



الجامعة الافتراضية السورية
SYRIAN VIRTUAL UNIVERSITY

تصميم وتطوير تطبيقات الويب

الدكتور باسل الخطيب

ISSN: 2617-989X



Books

تصميم وتطوير تطبيقات الويب

الدكتور باسل الخطيب

من منشورات الجامعة الافتراضية السورية

الجمهورية العربية السورية 2018

هذا الكتاب منشور تحت رخصة المشاع المبدع – النسب للمؤلف – حظر الاشتقاق (CC– BY– ND 4.0)

<https://creativecommons.org/licenses/by-nd/4.0/legalcode.ar>

يحق للمستخدم بموجب هذه الرخصة نسخ هذا الكتاب ومشاركته وإعادة نشره أو توزيعه بأية صيغة وبأية وسيلة للنشر ولأية غاية تجارية أو غير تجارية، وذلك شريطة عدم التعديل على الكتاب وعدم الاشتقاق منه وعلى أن ينسب للمؤلف الأصلي على الشكل الآتي حصراً:

باسل الخطيب، الإجازة في تقانة المعلومات، من منشورات الجامعة الافتراضية السورية، الجمهورية العربية السورية، 2018

متوفر للتحميل من موسوعة الجامعة <https://pedia.svuonline.org/>

Web application development

Bassel Al Khateeb

Publications of the Syrian Virtual University (SVU)

Syrian Arab Republic, 2018

Published under the license:

Creative Commons Attributions- NoDerivatives 4.0

International (CC-BY-ND 4.0)

<https://creativecommons.org/licenses/by-nd/4.0/legalcode>

Available for download at: <https://pedia.svuonline.org/>



الفهرس

1.....	الفصل الأول: أساسيات (XHTML)
2.....	أساسيات الانترنت والويب
3.....	الانترنت
3.....	البروتوكول TCP/IP
4.....	عناوين بروتوكول الانترنت IP Addresses
4.....	أسماء النطاق Domain Names
5.....	مخدمات الأسماء name servers
5.....	الويب World Wide Web
6.....	الويب أم الانترنت
6.....	متصفحات الويب Web Browsers
7.....	مخدمات الويب
8.....	المؤثرات الأساسية للغة التأشير XHTML
9.....	تطور HTML و XHTML
9.....	إصدارات HTML و XHTML
9.....	HTML و XHTML
10.....	الشكل الأساسي Syntax
10.....	البنية المعيارية لوثيقة XHTML
10.....	أساسيات تأشير النص
23.....	الجداول
24.....	الجداول
24.....	مؤثرات الجدول الأساسية
28.....	اقتراحات وتمارين
29.....	الفصل الثاني: مهارات متقدمة في XHTML
30.....	النماذج Forms

الفهرس

31.....	النماذج Forms
38.....	HTML5
39.....	المؤثرات الجديدة في HTML5
42.....	HTML5 - Web Forms 2.0
46.....	اقتراحات وتمارين
47.....	الفصل الثالث: أسلوب الصفحات المتتالي
48.....	أسلوب الصفحات المتتالي
49.....	مقدمة
49.....	مستويات أسلوب الصفحات
49.....	تنسيق تحديد الأسلوب
50.....	أشكال المحددات
53.....	أشكال خاصية - قيمة Property-Value
55.....	الخطوط
56.....	خصائص الخط Font Properties
59.....	خصائص القوائم
60.....	خصائص القوائم List Properties
63.....	أساليب أخرى
64.....	اللون Color
65.....	محاذاة النص
66.....	نموذج الصندوق The Box Model
72.....	المؤثرات < span > و < div >
74.....	اقتراحات وتمارين
78.....	الفصل الرابع: أساسيات لغة JavaScript
79.....	أساسيات JavaScript
80.....	مقدمة

الفهرس

80.....	البرمجة غرضية التوجه و JavaScript
80.....	الخصائص العامة للشكل
81.....	المُعَرِّفات
81.....	الكلمات المفتاحية
81.....	التعليقات
81.....	التعليمات
82.....	الأنماط والعمليات والتعبير
87.....	الإدخال والإظهار
90.....	تعليمات التحكم
91.....	تعليمات التحكم
91.....	التعبير المنطقية
91.....	تعليمات الاختيار Selection Statements
93.....	تعليمات التكرار
94.....	المصفوفات Arrays
95.....	المصفوفات Arrays
98.....	الوظائف Functions
99.....	الوظائف Functions
101.....	مطابقة النماذج
102.....	مطابقة النماذج Pattern Matching
107.....	الفصل الخامس: التفاعل بين JavaScript و HTML
108.....	التفاعل بين JavaScript و HTML
109.....	التعامل مع عناصر HTML من خلال JavaScript
112.....	الأحداث Events
113.....	الأحداث Events
114.....	معالجة أحداث جسم الوثيقة body

الفهرس

115	معالجة أحداث الزر
116	معالجة أحداث صندوق النص وصندوق كلمة السر
119	التحقق من الإدخال
120	التحقق من الإدخال
125	مثال تطبيقي
129	الفصل السادس: الوثائق الديناميكية مع JavaScript
130	الوثائق الديناميكية: توضع العناصر
131	مقدمة
131	توضع العناصر
134	تحريك العناصر
136	الوثائق الديناميكية: التعامل مع العناصر
137	إظهار وإخفاء العناصر
138	تغيير اللون والخط
140	المحتوى الديناميكي
141	المحتوى الديناميكي
142	تكديس العناصر
144	تحديد إحداثيات مؤشر الفارة
145	التفاعل مع نقر الفارة
147	التحكم بحركة العناصر
148	الوظيفة setTimeout
150	اقتراحات وتمارين
154	الفصل السابع: أساسيات JQuery
155	أساسيات JQuery
156	ماهي JQuery
156	تنزيل المكتبة

الفهرس

157	استخدام مكتبة JQuery
157	استدعاء وظائف JQuery
158	استخدام خطاطات مخصصة
160	محددات العناصر Selectors
161	المحددات Selectors
166	الفصل الثامن: الواصفات والأساليب في JQuery
167	التعامل مع الواصفات Attributes
168	التعامل مع الواصفات
170	بعض طرق الواصفات
174	أسلوب الصفحات المتتالي CSS
175	أسلوب الصفحات المتتالي
175	التعامل مع خصائص CSS
179	الفصل التاسع: مهارات JQuery
180	التعامل مع عناصر الوثيقة
181	التعامل مع نموذج كائن الوثيقة DOM
188	التعامل مع الأحداث
189	التعامل مع الأحداث
199	Ajax: Asynchronous JavaScript and XML
200	التعامل مع تقنية Ajax
200	تحميل البيانات
201	إرسال البيانات إلى المخدم
206	وظائف JQuery AJAX
207	أحداث JQuery AJAX
208	التأثيرات Effects
209	التأثيرات

الفهرس

212	طرق التأثيرات JQuery
213	مكتبة واجية المستخدم UI
216	اقتراحات وتمارين
217	الفصل العاشر: أساسيات ASP.Net 2013
218	أساسيات ASP.Net 2013
219	مقدمة
219	لغة التنفيذ المشتركة CLR
219	لغات .NET
220	أساسيات ASP.Net
221	وثائق ASP.Net
223	الكود الخفي
224	عناصر التحكم HTML Controls
225	عناصر التحكم ASP.NET
225	عناصر التحكم HTML Controls
229	دورة حياة وثيقة ASP.NET Life Cycle
233	أحداث الصفحة Page-Level Events
234	أحداث عناصر التحكم Control Events
237	عناصر التحكم Web Controls
238	عناصر تحكم الويب Web Controls
251	عناصر التحقق من الصحة
252	عناصر التحقق من الصحة
255	الفصل الحادي عشر: إدارة الحالة
256	متحولات الحالة View State
257	متحولات الحالة View State
258	الحفاظ على حالة الجلسة

الفهرس

259	ملاحقة الجلسات Session Tracking
259	متغيرات الجلسة Session Variables
265	الكعكات
266	الكعكات Cookies
271	الفصل الثاني عشر: ربط عناصر التحكم مع البيانات
272	ربط عناصر التحكم مع البيانات
273	مقدمة
273	ربط البيانات
273	الربط مع قيمة وحيدة
277	الربط مع قيم متعددة
278	الربط مع عدة قيم
284	الفصل الثالث عشر: التعامل مع قواعد البيانات
285	الاتصال مع قواعد البيانات
286	التعامل مع قواعد البيانات
286	إنشاء قاعدة بيانات
286	عناصر تحكم البيانات
288	عرض البيانات في عناصر التحكم
289	عرض البيانات في القوائم
293	استخدام عناصر البيانات
294	استخدام FormView
297	استخدام GridView و DetailsView
301	ربط معاملات الاستعلام مع عناصر التحكم
302	ربط معاملات الاستعلام مع عناصر التحكم
306	الفصل الرابع عشر: التعامل مع قواعد البيانات برمجياً
307	التعامل مع قواعد البيانات برمجياً

الفهرس

308.....	التعامل مع قواعد البيانات برمجياً
310.....	مثال تطبيقي
317.....	عنصر التحكم Repeater
318.....	استخدام عنصر التحكم Repeater
321.....	مجموعات البيانات Datasets
322.....	الصف SqlDataAdapter
322.....	الصف DataSet
326.....	مثال تطبيقي
334.....	الفصل الخامس عشر: خدمات الويب
335.....	خدمات الويب
336.....	خدمات الويب Web Services
337.....	آلية عمل خدمات الويب
339.....	إنشاء ونشر خدمات الويب
340.....	إنشاء خدمة ويب
344.....	نشر خدمة الويب
345.....	مثال تطبيقي لخدمة ويب
346.....	إنشاء خدمة ويب
348.....	إنشاء نموذج ويب يستهلك خدمة الويب
352.....	مثال : استخدام خدمة ويب
353.....	استخدام خدمة ويب
355.....	الفصل السادس عشر: مقدمة إلى XML
356.....	XML
357.....	مقدمة
358.....	لغة السلوب الموسعة XSL
360.....	مثال تطبيقي

الفهرس

361	مثال تطبيقي
364	عنصر التحكم XML Control
365	مجموعات البيانات و XML
366	مجموعات البيانات XML و Datasets
370	اقتراحات وتمارين

الفصل الأول: أساسيات (XHTML)

الكلمات المفتاحية:

الإنترنت، الويب، متصفحات الويب، الزيون، المخدم، مؤثرات XHTML.

الملخص:

يبدأ الفصل بمقدمة سريعة للتذكير بأساسيات عالم الويب. تُبين بعدها أساسيات لغة XHTML حيث نستعرض المؤثرات والواصفات الأساسية للغة.

الأهداف التعليمية:

يتعرف الطالب في هذا الفصل على:

- الإنترنت والويب.
- المخدم والزيون.
- مؤثرات التنسيق.
- التعامل مع الصور.
- التعامل مع الروابط التشعبية.
- التعامل مع الجداول.

المخطط:

يضم فصل أساسيات XHTML 3 وحدات (Learning Objects) هي:

- أساسيات الإنترنت والويب.
- المؤثرات الأساسية للغة التأشير XHTML.
- الجداول.

أساسيات الانترنت والوب

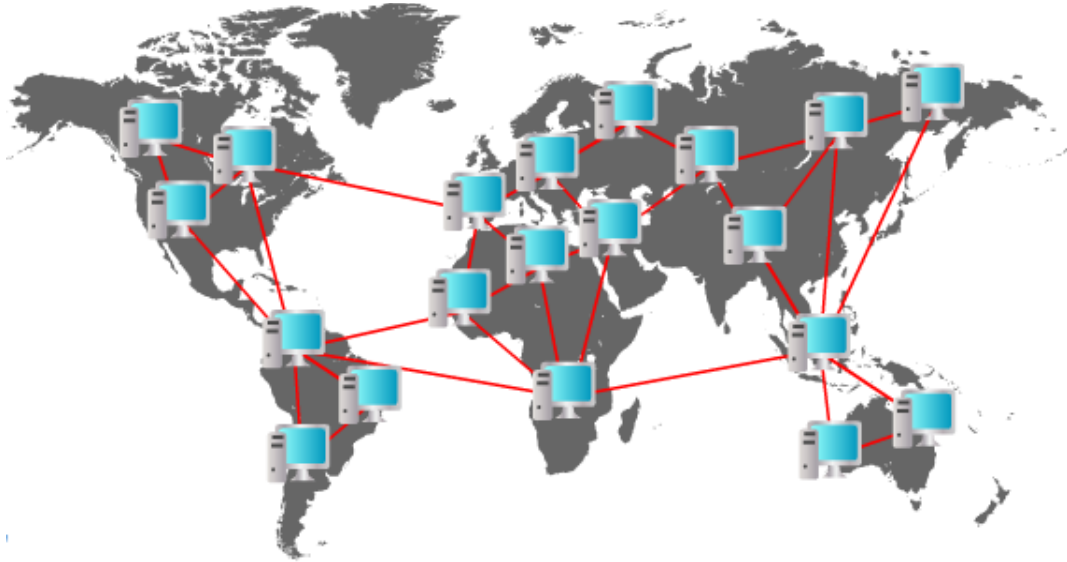
الأهداف التعليمية:

التعرف على:

- الإنترنت.
- الوب.
- الزبون.
- المخدم.
- البروتوكول TCP/IP.
- البروتوكول HTTP.
- متصفحات الوب.
- مخدمات الوب.

الإنترنت

تتألف الإنترنت من مجموعة ضخمة من الحواسيب المرتبطة مع بعضها عبر شبكات الاتصالات. تختلف هذه الحواسيب عن بعضها بحجمها ومصنعيها وأنظمتها.



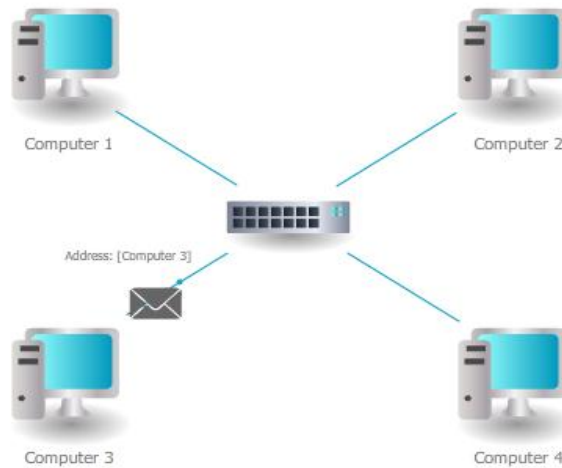
البروتوكول TCP/IP

وهو معيار يسمح بتواصل الأجهزة المختلفة مع بعضها البعض، وهو اختصار لـ:

Transmission Control Protocol/Internet Protocol (TCP/IP)

ويُمكن أن يُستخدم مباشرة ليرسل برنامج على حاسب من التواصل مع برنامج آخر على حاسب آخر عبر الإنترنت.

ومن الواضح أن كل جهاز متصل مع الإنترنت يجب أن يكون معرفاً بشكل فريد.



عناوين بروتوكول الانترنت IP Addresses

تُحدّد العقد على الانترنت بالنسبة للمستخدمين بأسماء. أما بالنسبة للحاسب، فهي تُعرّف بعناوين رقمية (IP Address).

يتكون العنوان الفريد Internet Protocol (IP) address لكل جهاز متصل بالانترنت من رقم فريد يتألف من 32 بت 32-bit. يُكتب هذا العنوان عادةً باستخدام أربعة أرقام يتألف كل منها من 8 بت يفصل بينها نقاط. (مثلاً: 191.57.126.121). يتم عادة إعطاء مجال متسلسل من هذه الأرقام لكل منظمة، والتي تقوم بدورها بتوزيع الأرقام ضمن هذا المجال على حواسيبها المتصلة مع الانترنت. مثلاً، لوزارة الدفاع الأمريكية المجال من 12.0.0.0 إلى 12.255.255.255 أي 16 مليون عنوان. ومع تزايد الطلب على هذه المجالات. ظهر في عام 1998 معيار IP جديد يُدعى IPv6 يوسع العنوان إلى 128 بت. إلا أنه لم ينتشر على نطاق واسع حتى الآن.



أسماء النطاق Domain Names

بما أن البشر يجدون صعوبة في التعامل مع الأرقام، تمّ التوافق على استخدام أسماء للحاسب عوضاً عن العناوين الرقمية.

يتألف الاسم عادة من اسم المضيف host متبوعاً بأسماء نطاقات domains (مجموعات من الحاسب) محتواه ضمن بعضها البعض. يُميز النطاق الأخير نوع المنظمة التي ينتمي الحاسب المضيف لها، فمثلاً، يُميز النطاق edu المؤسسات التعليمية و com المؤسسات التجارية.

يُدعى هذا الاسم الكامل باسم النطاق المحدد تماماً fully qualified domain name. مثلاً:

movies.comedy.marxbros.com

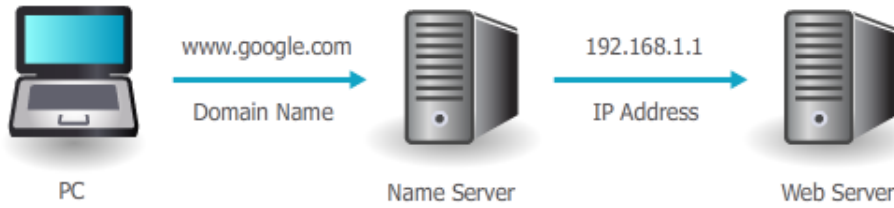
مخدمات الأسماء name servers

هي برمجيات خاصة تقوم بتحويل أسماء النطاقات التي يكتبها المستخدم في متصفح الويب إلى العناوين الرقمية الموافقة.

توضع هذه البرمجيات على مخدمات خاصة تُشغّل من قبل منظمات مسؤولة عن الحواسيب المرتبطة بها. عندما يقوم مستخدم بطلب وثيقة باستخدام اسم النطاق يتم توجيه هذا الطلب إلى أقرب مخدم أسماء للحصول على العنوان IP لمخدم الوثيقة المطلوبة.

وفي أواسط الثمانينات ظهر العديد من البروتوكولات التي تعمل فوق البروتوكول TCP/IP لتُخدّم مستخدمي الانترنت. ومن أشهر هذه البروتوكولات: **PC**

- البروتوكول telnet والذي يُستخدم للولوج لحاسب آخر على الانترنت.
- البروتوكول ftp والذي يُستخدم لنقل الملفات عبر الانترنت.
- البروتوكول http والذي يُستخدم لتبادل وثائق الويب.

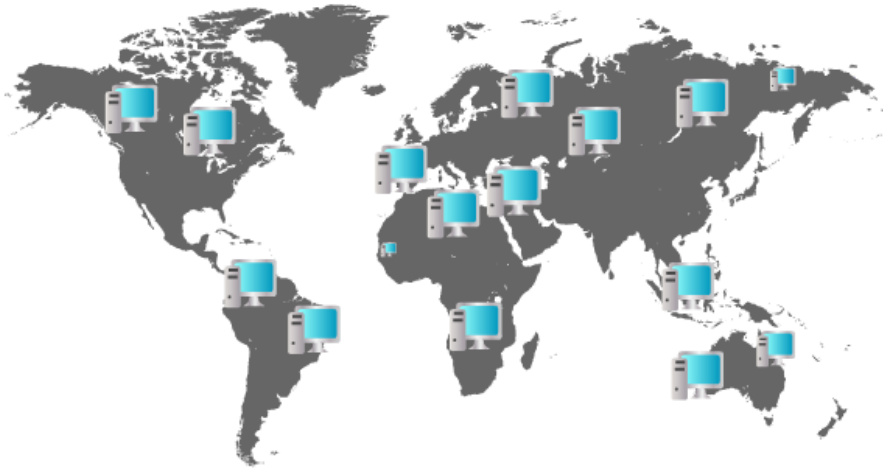


الويب World Wide Web

مجموعة من البرمجيات والبروتوكولات التي تم وضعها على حواسيب الانترنت بهدف السماح للأشخاص حول العالم باستخدام الانترنت للبحث والحصول على الوثائق أي حاسب آخر مخدم للوثائق.

تستخدم شكلاً معيناً للوثائق يُدعى النصوص الفائقة Hypertext والذي هو عبارة عن نص يحوي ارتباطات إلى نصوص أخرى في وثائق أخرى، مما يسمح باستعراض غير تسلسلي بالضرورة للنصوص.

يُمكن للوثائق أن تحوي صور أو صوت أو أنواع أخرى من الوسائط، فتُدعى فائقة الوسائط Hypermedia. يُمكن أن نرى الويب بأنها مجموعة ضخمة من الوثائق المترابطة فيما بينها.



الويب أم الانترنت

يجب أن نميز جيداً بين الانترنت وبين الويب. إذ أن الانترنت هي مجموعة من الحواسيب والتجهيزات المرتبطة مع بعضها بحيث يُمكن لكل منها التخاطب مع الآخر. أما الويب فهي مجموعة من البرمجيات والبروتوكولات التي تمّ وضعها على حواسيب الانترنت (تستخدم الويب البروتوكول HTTP). وبهذا فإن الانترنت كانت وماتزال مفيدة قبل ظهور الويب، مع ملاحظة أن معظم البشر اليوم يستخدمون الانترنت عبر الويب.

متصفحات الويب Web Browsers

عندما يتخاطب حاسبين في شبكة، يتصرف في معظم الأحيان أحدهما كزبون والآخر كمخدم. يبدأ الزبون Client بالتخاطب، فيطلب بشكل عام بيانات موجودة على المخدم Server، الذي يقوم بإرسال هذه البيانات للزبون.

تعمل الويب باستخدام هذا المبدأ والذي ندعوه زبون-مخدم Client/Server. يقوم المتصفح (برنامج على الزبون) بطلب وثائق من المخدم. يقوم المخدم بتحديد الوثائق المطلوبة وإرسالها للمتصفح (يُمكن أن تكون الوثائق ساكنة أو نتيجة طلب تنفيذ برنامج ما على المخدم، كما يُمكن للمخدم طلب بيانات معينة من مستخدم المتصفح كبيانات التسجيل في موقع مثلاً). تعتمد الويب لتحقيق التواصل بين المستعرض والمخدم البروتوكول HTTP¹، والذي يُشكل الشكل المعياري للتخاطب بين المستعرض والمخدم. من أشهر المتصفحات المستخدمة:



مخدمات الويب

برامج مهمتها تقديم الوثائق المطلوبة من قبل المتصفحات.

من أشهر هذه المخدمات:

- **المخدم Apache**

مخدم Apache هو مخدم مجاني مفتوح المصدر يعمل على جميع أنظمة التشغيل. ويتميز بوثوقيته ودعمه للعديد من لغات البرمجة.



- **المخدم IIS**

مخدم Internet Information Services هو مخدم Microsoft وبالتالي فهو يعمل على النظام Windows فقط. وهو المنافس الأقوى للمخدم Apache.



المؤثرات الأساسية للغة التأشير XHTML

الأهداف التعليمية:

- التعرف على المؤثرات الأساسية للغة التأشير XHTML.

تطور HTML وXHTML

اشتقت اللغة HTML من اللغة المعيارية SGML² والتي هي معيار لتوصيف تنسيق النصوص وفق المعايير الموضوعية من قبل المنظمة الدولية للمعايير ISO³. تختلف أهداف اللغة HTML عن باقي لغات تنسيق النصوص، والتي تهتم بكل تفاصيل النص مثل حجم الخط ولونه، في أنها مصممة لتحديد بنية الوثيقة بمستوى عالي ومجرد، كما أنها يجب أن تُستخدم على حواسيب مختلفة ومتصفحات متنوعة. ولقد أغنى أسلوب الصفحات المتتالي CSS⁴ والذي ظهر في أواخر التسعينات تفاصيل الإظهار التي يُمكن تحديدها من خلال هذه اللغة.

إصدارات HTML وXHTML

تعددت إصدارات HTML انتهاءً بالنسخة 5 عام 2012. أما المعيار XHTML 1.0 فقد تمت الموافقة عليه في عام 2000. تم إنشاء هذا المعيار بإعادة تشكيل اللغة HTML وبحيث تكون لغة تأشير موسعة eXtensible Hypertext Markup Language XML. وفي عام 2001 تمت الموافقة على النسخة XHTML 1.1 من قبل منظمة الويب W3C⁵ والتي دورها مراقبة تطور لغات الويب. ثم النسخة XHTML 2 عام 2010.

HTML وXHTML

يُمكن تلخيص الأسباب التي تدعونا لاستخدام XHTML بما يلي:

- تعتمد على معيار موثوق جيداً. بخلاف لغة HTML والتي يُمكن أن تختلف من منتج لآخر.
- تملك XHTML قواعد كتابة صارمة تفرض بنية متماسكة لجميع المستندات المكتوبة باستخدامها، بخلاف HTML والتي تكون قواعدها ضبابية ويُمكن تجاوزها في أغلب الأحيان.
- يُمكن التحقق من توافقية أي نص مع قواعد اللغة ومعياريها باستخدام أدوات التحقق التي توفرها المنظمة W3C.

Standard Generalized Markup Language 2

International Standard Organization 3

Cascading Style Sheet 4

World Wide Web Consortium 5

الشكل الأساسي Syntax

- تُعرف عناصر elements اللغة باستخدام مجموعة من المحددات Tags.
- شكل المحدد:
 - Opening tag: <name>
 - Closing tag: </name>
- يُشكّل محدد الفتح ومحدد الإغلاق حاوية container للمحتوى content.
- لا يكون لكل المؤثرات محتوى وفي هذه الحالة يكون شكلها <name />
- ندعو الحاوية والمحتوى بالعنصر element.
- يُمكن أن يكون للمحدد واصفات attributes توضع بعد اسم المؤثر:
 - <name attribute1="value1" attribute2="value2"...>
- يكون للتعليق الشكل: <!-- ... -->
- تتجاهل المتصفحات التعليقات والمؤثرات غير المفهومة والأسطر line breaks والفراغات المتتالية multiple spaces والمسافات .tabs.

البنية المعيارية لوثيقة XHTML

- يجب أن تبدأ كل وثيقة XHTML بـ:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
```

- تكون المؤثرات <html>, <head>, <title>, <body> مطلوبة في كل وثيقة.
- يكون المحدد <html> جذر كامل الوثيقة.
- يجب أن يكون للمؤثر <html> الواصفة: xmlns
- تتكون الوثيقة من رأس head وجسم body.
- يقوم المؤثر <title> بإظهار عنوان للوثيقة في شريط العنوان لنافذة المتصفح.

أساسيات تأشير النص

الفقرات Paragraph

يقوم المؤثر <p> بكسر السطر الحالي وحشر سطر فارغ جديد. يبدأ السطر الجديد محتوى الفقرة. يقوم المتصفح بوضع كلمات الفقرة بشكل يلائم عرض السطر. فثلاً يُمكن أن تظهر الفقرة التالية في المتصفح:

```
<!DOCTYPE html>
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> Paragraph Example </title> </head>
<body>
<p>
  Mary had
  a
  little lamb, its fleece was white as snow. And
every where that
  Mary went, the lamb
was sure to go.
</p>
</body>
</html>
```

كما يلي:

Mary had a little lamb, its fleece was white as snow. And every where that Mary went, the lamb was sure to go.

التحقق من الوثائق XHTML

تُقدم الهيئة W3C موقع ويب يسمح بالتحقق من موافقة نص XHTML إلى المعايير المعتمدة:

<http://validator.w3.org>

يُحدد الأمر DOCTYPE المعيار المطلوب.

كسر السطر

يقوم المؤثر `</ br>` بكسر السطر الحالي. لاحظ أن هذا المؤثر هو بنفس الوقت مؤثر فتح وإغلاق. وبالتالي تظهر الفقرة التالية:

```
<p>
  Mary had
  a
  little lamb, <br /> its fleece was white as snow. And
every where that
  Mary went, the lamb
was sure to go.
</p>
```

كما يلي:

Mary had a little lamb,
its fleece was white as snow. And everywhere that Mary went, the lamb was sure to
go.

الترويسات Headings

تتوفر ستة ترويسات مختلفة تُحددها المؤثرات من <h1> إلى <h6>. تستخدم الترويسات 1 و2 و3 أحجاماً للخط أكبر من الخط الافتراضي. أما الترويسة 4 فتستخدم الخط الافتراضي، وتستخدم الترويسات 5 و6 حجوماً أصغر من الحجم الافتراضي. إذا كان لدينا مثلاً:

```
<!DOCTYPE html>

<!-- headings example -->
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> Headings </title>
</head>
<body>

  <h1> Aidan's Airplanes (h1) </h1>
  <h2> The best in used airplanes (h2) </h2>
  <h3> "We've got them by the hangarful" (h3) </h3>
  <h4> We're the guys to see for a good used airplane (h4) </h4>
  <h5> We offer great prices on great planes (h5) </h5>
  <h6> No returns, no guarantees, no refunds,
    all sales are final! (h6) </h6>

</body>
</html>
```

يُظهر المتصفح:

Aidan's Airplanes (h1)

The best in used airplanes (h2)

"We've got them by the hangarful" (h3)

We're the guys to see for a good used airplane (h4)

We offer great prices on great planes (h5)

No returns, no guarantees, no refunds, all sales are final! (h6)

كتلة الاقتباس Block Quotations

يسمح المؤثر `<blockquote>` بإظهار نص محدد بشكل مختلف عن النص المحيط. فمن أجل المثال التالي:

```
<!DOCTYPE html>
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Blockquotes </title>
  </head>
  <body>
    <p>
      Lincoln is generally regarded as one of the greatest
      presidents of the U.S.
    </p>
    <blockquote>
      <p>
        "Fourscore and seven years ago our fathers brought forth on
        this continent, a new nation, conceived in Liberty, and
        dedicated to the proposition that all men are created equal."
      </p>
    </blockquote>
    <p>
      Whatever one's opinion of Lincoln, no one can deny the
      enormous and lasting effect he had on the U.S.
    </p>
  </body>
</html>
```

يُظهر المتصفح:

Lincoln is generally regarded as one of the greatest presidents of the U.S.

"Fourscore and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal."

Whatever one's opinion of Lincoln, no one can deny the enormous and lasting effect he had on the U.S.

المؤثر ``

يُستخدم لوضع كلمة أو جملة معينة بشكل مختلف.

مثال:

```
<p>"Did you say that you are a <em>chameleon</em>".</p>
```

يُظهر المتصفح:

"Did you say that you are a *chameleon*".

الخطوط Fonts

تتوفر المؤثرات التالية والتي تسمح بتنسيق الخط:

- **Boldface** خط غامق
- **Italics** خط مائل <i>
- **Larger** خط أكبر <big>
- **Smaller** خط أصغر <small>
- **Monospace** خط واحد <tt>
- **Superscript** خط أعلى <sup>
- **Subscript** خط أدنى <sub>

والتي يُبينها المثال التالي:

```
<p>
  The <big> sleet <big> in <big> <i> Crete
  </i><br /> lies </big> completely </big>
  in </big> the street
</p>
<p>
  <tt>
  Monospace <big> font </big>
  </tt>
</p>
x<sub>2</sub><sup>3</sup> + y<sub>1</sub><sup>2</sup>
</p>
```

والتي تُظهر:

The sleet in *Crete*
lies completely in the street

Monospace font

$$x_2^3 + y_1^2$$

المحارف الخاصة

يُبين الجدول التالي مجموعة المحارف الخاصة التي يُمكن استخدامها:

Character	Entity
&	&
<	<
>	>
“	"
‘	'
(space)	
¼	¼
½	½
¾	¾
°	°

فمثلاً تظهر الفقرة:

```
<p>  
&frac14;&nbsp; + &frac14;&nbsp; = &frac12;  
</p>
```

كما يلي:

```
¼ + ¼ = ½
```

المسطرة الأفقية

يقوم المؤثر `</hr>` بكسر السطر الحالي ورسم خط أفقي.

المؤثر meta

يُستخدم المؤثر `<meta>` لوضع معلومات مفيدة لمحركات البحث تساعد في تصنيف الوثيقة ووضعها في فهرسها.

يكون لهذا المؤثر واصفتين أساسيتين هما `<name>` و `<content>`.

يُحدد المثال التالي أن الوثيقة تحوي معلومات تتعلق بـ XHTML و JavaScript:

```
<meta name="keywords" content="XHTML, JavaScript" />
```

الصور Images

تُحسّن الصور مظهر صفحات الويب بشكل كبير، مع ملاحظة أنها تُبطئ استعراض الوثائق من قبل المستخدمين الذين لا يملكون سرعات وصول عالية للإنترنت. تُخزن الصور في ملفات مستقلة وتُستدعى من قبل .XHTML.

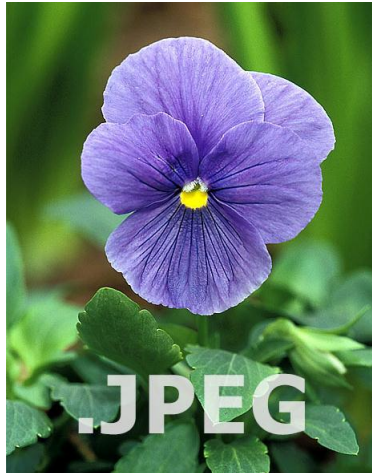
تنسيق الصور Image Formats

يوجد بشكل أساسي ثلاثة أشكال لتنسيق ملفات الصور:

- الملفات (GIF (Graphic Interchange Format تستخدم 8-bits لكل بيكسل pixel (أي 256 لون). يُمكن أن تكون هذه الصور شفافة transparent. يُفضل استخدامها للرسم المرسومة باستخدام برامج حاسوبية.



- الملفات (JPEG (Joint Photographic Experts Group تستخدم 24-bits لكل بيكسل pixel (أي 16 مليون لون). لا يُمكن أن تكون شفافة. يُفضل استخدامها للصور الطبيعية photo.



• الملفات (PNG (Portable Network Graphics)

ظهر هذا الشكل عام 1996 ليجمع بين ميزات الشكلين السابقين (شفافية JIF وعدد ألوان JPEG). ومن المتوقع أن يصبح الشكل الأوسع على الإنترنت، لاسيما وأنه مصدر حر free.



المؤثر

يُستخدم المؤثر لإدراج صورة في الوثيقة.

- تُحدد الوصفة src اسم ملف الصورة.
- تُحدد الوصفة alt نص الصورة أي النص الذي سيظهر في حال عدم تحميل الصورة أو عند وضع مؤشر الفأرة فوق الصورة. وهي واصفة مطلوبة required.
- تُحدد الوصفتان width و height أبعاد الصورة.

يُبين المثال التالي طريقة إدراج صورة:

```
<!DOCTYPE html>
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Images </title>
  </head>
  <body>
    <h1> Aidan's Airplanes </h1>
    <h2> The best in used airplanes </h2>
    <img src = "images/plane.png" alt = "Picture of a Cessna 210"
          width="160" height="120" />
    <br />
    Buy this fine airplane today at a remarkably low price
  </body>
</html>
```

Aidan's Airplanes

The best in used airplanes



Buy this fine airplane today at a remarkably low price

الروابط Hypertext Links

يتصرف الرابط كمؤشر إلى مورد آخر. يُمكن أن يكون هذا المورد أي وثيقة على الويب، أو مكان آخر في الوثيقة نفسها، أو مكان محدد في وثيقة أخرى. وبالطبع فإن الروابط هي التي تُعني مواقع الويب. يُستخدم المؤثر `<a>` لتحديد إرتباط، يملك هذا المؤثر الوصفة `href` التي تُحدد عنوان الهدف. أما محتوى المؤثر فهو الذي يُشكل الرابط (أي الذي يجب النقر عليه للانتقال إلى الهدف). يُمكن أن يكون المحتوى نص أو صورة أو عنوان.

يُبين المثال التالي استخدام رابط نصي لفتح وثيقة أخرى:

```
<body>
  <h1> Aidan's Airplanes </h1>
  <h2> The best in used airplanes </h2>
  <a href = "6.images.html">
    Information on the Cessna 210 </a>
</body>
```

حيث يظهر كما يلي:

Aidan's Airplanes
The best in used airplanes
[Information on the Cessna 210](#)

وعند النقر على الرابط سيتم فتح الملف 6.images.html:

Aidan's Airplanes
The best in used airplanes



Buy this fine airplane today at a remarkably low price

- عندما لا يكون الهدف بداية الوثيقة فيجب تحديد المكان المعين في الوثيقة الهدف.
- تُستخدم الوصفة `id` لإعطاء معرف هدف. فمثلاً، إذا أعطينا ترويسة معرف:

```
<h1 id = "baskets"> Baskets </h1>
```

فيمكن استخدام هذا المعرف في الرابط (يجب سبقة بالمحرف #):

```
<a href = "#baskets"> What about baskets? </a>
```

• إذا كان المعرف في وثيقة أخرى فيجب وضع اسم الوثيقة:

```
<a href = "myAd.html#baskets"> Baskets </a>
```

• يمكن أن يكون الرابط صورة، كما يُبين المثال التالي:

```
<a href = "6.images.html">
  <img src = "images/Plane.png"
    alt = "Small picture of an airplane " />
</a>
```

القوائم Lists

توفر XHTML إمكانية إدراج:

- قوائم غير مرتبة
- قوائم مرتبة
- قوائم تعريف

القوائم غير المرتبة

تكون القائمة محتوى المؤثر ، وحيث يُعرّف كل عنصر من عناصر القائمة باستخدام المؤثر . يمكن وضع أي مؤثر في عنصر قائمة. كما يُبين المثال التالي:

```
<body>
  <ul>
    <li> Syria </li>
    <ul>
      <li> Damascus </li>
      <li> Aleppo </li>
    </ul>
    <li> France </li>
    <ul>
      <li> Paris </li>
      <li> Boredaux </li>
    </ul>
    <li> Germany </li>
  </ul>
</body>
```

حيث يُظهر المتصفح:

- Syria
 - Damascus
 - Aleppo
- France
 - Paris
 - Boredaux
- Germany

القوائم المرتبة

تُعرّف في هذه الحالة القائمة باستخدام المؤثر ``. تُرقم عناصر القائمة تلقائياً بشكل تسلسلي. كما يُبين المثال التالي:

```
<body>
  <ol>
  <li> Syria </li>
    <ol>
    <li> Damascus </li>
    <li> Aleppo </li>
    </ol>
  <li> France </li>
    <ol>
    <li> Paris </li>
    <li> Boredaux </li>
    </ol>
  <li> Germany </li>
  </ol>
</body>
```

حيث يُظهر المتصفح:

1. Syria
 1. Damascus
 2. Aleppo
2. France
 1. Paris
 2. Boredaux
3. Germany

قوائم التعريف

يُستخدم هذا النوع من القوائم عادةً لسرد مجموعة من التعاريف. تكون القائمة محتوى المؤثر `<dl>`. يُستخدم المؤثر `<dt>` لعنوان كل تعريف، والمؤثر `<dd>` لمحتوى التعريف. كما يُبين المثال التالي:

```
<body>
  <dl>
    <dt>XML </dt>
    <dd> eXtensible Markup Language </dd>
    <dt>XHTML </dt>
    <dd> eXtensible Hyper Text Markup Language </dd>
  </dl>
</body>
```

حيث يُظهر المتصفح:

```
XML
  eXtensible Markup Language
XHTML
  eXtensible Hyper Text Markup Language
```

الجداول

الأهداف التعليمية:

- التعامل مع الجداول.

الجدول

- يتألف الجدول من مصفوفة من الخلايا.
- يُمكن أن يكون لكل منها محتوى.
- يُمكن للخلايا أن تحوي أي عنصر.

مؤثرات الجدول الأساسية

يكون الجدول محتوى المؤثر `<table>`. لهذا المؤثر الوصفة `border`. يُمكن وضع قيمة رقمية في هذه الوصفة لتحديد عرض حدود الجدول بالبيكسل، أو وضع قيمتها تساوي "border" فيكون عرض حدود الجدول العرض الافتراضي، وعند عدم وضع هذه الوصفة يبقى الجدول بلا حدود . يُسبق الجدول عادةً بعنوان وذلك باستخدام المؤثر `<caption>`. يُعرّف كل سطر في الجدول باستخدام المؤثر `<tr>`. أما كل خلية في السطر فتُعرّف باستخدام المؤثر `<td>`. يُمكن تعريف خلية عنوان باستخدام المؤثر `<th>`. يُبين المثال التالي استخدام المؤثرات السابقة:

```
<body>
<table border = "border">
  <caption> Fruit Juice Drinks </caption>
  <tr>
    <th> </th>
    <th> Apple </th>
    <th> Orange </th>
    <th> Pineapple </th>
  </tr>
  <tr>
    <th> Breakfast </th>
    <td> 0 </td>
    <td> 1 </td>
    <td> 0 </td>
  </tr>
  <tr>
    <th> Lunch </th>
    <td> 1 </td>
    <td> 0 </td>
    <td> 0 </td>
  </tr>
  <tr>
    <th> Dinner </th>
    <td> 0 </td>
    <td> 0 </td>
    <td> 1 </td>
  </tr>
</table>
</body>
```

حيث يُظهر المتصفح:

Fruit Juice Drinks			
	Apple	Orange	Pineapple
Breakfast	0	1	0
Lunch	1	0	0
Dinner	0	0	1

مؤثرات الجدول الأساسية

امتداد الخلايا

يُمكن استخدام الواصفة `colspan` لتحديد عدد الأعمدة التي تمتد خلية عليها. والواصفة `rowspan` لتحديد عدد الأسطر التي تمتد خلية عليها. كما يُبين المثال التالي:

```
<body>
<table border = "border">
  <tr>
    <td rowspan = "2"> </td>
    <th colspan = "3"> Fruit Juice Drinks </th>
  </tr>
  <tr>
    <th> Apple </th>
    <th> Orange </th>
    <th> Pineapple </th>
  </tr>
  <tr>
    <th> Breakfast </th>
    <td> 0 </td>
    <td> 1 </td>
    <td> 0 </td>
  </tr>
</table>
</body>
```

حيث يُظهر المتصفح:

Fruit Juice Drinks			
	Apple	Orange	Pineapple
Breakfast	0	1	0

المحاذاة

تُستخدم الواسفة align لتحديد المحاذاة الأفقية لخلية، وتأخذ القيم left و right و center وذلك مع المؤثرات <tr> و <td> و <th>.

أما المحاذاة العمودية فتُحدد باستخدام الواسفة valign والتي تأخذ القيم top و bottom و center وذلك مع المؤثرين <th> و <td>.

يُبين المثال التالي استخدام هذه الواسفات:

```
<body>
<table border = "border">
  <caption> The align and valign attributes </caption>
  <tr align = "center">
    <th> </th>
    <th> Column Label </th>
    <th> Another One </th>
    <th> Still Another One </th>
  </tr>
  <tr>
    <th> align </th>
    <td align = "left"> Left </td>
    <td align = "center"> Center </td>
    <td align = "right"> Right </td>
  </tr>
  <tr>
    <th> <br /> valign <br /> <br /> </th>
    <td> Default </td>
    <td valign = "top"> Top </td>
    <td valign = "bottom"> Bottom </td>
  </tr>
</table>
</body>
```

حيث يُظهر المتصفح:

The align and valign attributes			
	Column Label	Another One	Still Another One
align	Left	Center	Right
valign	Default	Top	Bottom

التباعد

تُحدد الواصفة cellspacing للمؤثر <table> تباعد خلايا الجدول عن بعضها. أما الواصفة cellpadding فتُحدد بعد محتوى الخلية عن حدودها. كما يُبين المثال التالي:

```
<body>
  <b>Table 1 (space = 10, pad = 30) </b><br />
  <table border = "5" cellspacing = "10" cellpadding = "30">
    <tr>
      <td> Small spacing, </td>
      <td> large padding </td>
    </tr>
  </table>
  <br /><br />
  <b>Table 2 (space = 30, pad = 10) </b><br />
  <table border = "5" cellspacing = "30" cellpadding = "10">
    <tr>
      <td> Large spacing, </td>
      <td> small padding </td>
    </tr>
  </table>
</body>
```

حيث يُظهر المتصفح:

Table 1 (space = 10, pad = 30)	
Small spacing,	large padding

Table 2 (space = 30, pad = 10)	
Large spacing,	small padding

اقتراحات وتمارين

قم بإنشاء الصفحة التالية والتي تحوي بعض المعلومات والروابط للجامعات السورية:

Syrian Universities

Syrian Universities

Type	Web Site	Address
Public	SVU	Damascus- Omayad Square
	Damascus University	Damascus- Mazza
Private	IUST	Damascus- Kiwan
	AIU	Damascus- Marjaa

© 2015 SVU. All rights reserved. For questions, send email to info@svuonline.org.

الفصل الثاني:

مهارات متقدمة في XHTML

الكلمات المفتاحية:

النماذج، الجديد في HTML5.

الملخص:

ندرس آلية إنشاء النماذج في XHTML والتعامل معها. ثم نستعرض أهم الإضافات الجديدة التي قدمتها HTML5.

الأهداف التعليمية:

يتعرف الطالب في هذا الفصل على:

- عناصر النماذج.
- المؤثرات الإضافية في HTML5.
- الواصفات الإضافية في HTML5.

المخطط:

يضم فصل مهارات متقدمة في XHTML وحدتين (Learning Objects) هي:

- النماذج Forms
- HTML5

النماذج Forms

الأهداف التعليمية:

- إنشاء النماذج.

النماذج Forms

تُستخدم النماذج عادةً لتحصيل مجموعة بيانات من المستخدم، ومن ثم إرسال هذه البيانات من المتصفح إلى المخدم.

توفر لغة XHTML مجموعة من المؤثرات التي تقوم بخلق أغراض تُحقق عملية جمع البيانات من المستخدم. تُدعى هذه الأغراض عناصر تحكم (Controls, Widgets) مثل صناديق التحقق check boxes وأزرار الخيار radio buttons.

يكون لكل عنصر تحكم قيمة value يُدخلها المستخدم عادةً. ندعو مجموعة القيم لعنصر التحكم في نموذج بيانات النموذج form data.

تُرسل هذه البيانات إلى المخدم عندما يقوم المستخدم بالنقر على زر الإرسال submit في النموذج.

المؤثر <form>

توضع جميع مكونات النموذج ضمن هذا المؤثر. لهذا المؤثر الوصفة المطلوبة action والتي تُحدد¹ عنوان التطبيق الذي سترسل بيانات النموذج إليه.

تُحدد الوصفة method طريقة إرسال البيانات. تأخذ هذه الوصفة إحدى القيمتين التاليتين:

- القيمة get (وهي القيمة الافتراضية): تُرسل البيانات في هذه الحالة في سلسلة محرفية تُضاف إلى المحدد URL.
- القيمة post تُرسل البيانات في هذه الحالة عبر أغراض خاصة إلى المخدم.

المؤثر <input>

يُستخدم هذا المؤثر لإنشاء 6 أنواع من الكائنات. تُحدد الوصفة المطلوبة type نوع الكائن المطلوب والتي تأخذ إحدى القيم التالية:

- القيمة text لإنشاء صندوق نص.
- القيمة password لإنشاء صندوق كلمة سر.
- القيمة checkbox لإنشاء صندوق تحقق.
- القيمة radio لإنشاء زر خيار.
- القيمة submit لإنشاء زر إرسال.
- القيمة reset لإنشاء زر إعادة.

¹ في حال عدم وجود عنوان يجب وضع السلسلة الفارغة "".

صندوق النص textbox

- يكون الحجم الافتراضي 20 حرف.
- يُمكن تحديد حجم معين باستخدام الوصفة size.
- في حال كتابة محارف أكثر من الحجم المحدد يظهر تلقائياً شريط إنزلاق.
- يُمكن تحديد عدد محارف أعظمي لا يُمكن تجاوزه باستخدام الوصفة maxlength.
- يجب إعطاء اسم لصندوق النص باستخدام الوصفة name.

Input Text:

صندوق كلمة السر password

يُمثل صندوق النص في واصفاته. إلا أنه يُظهر نجوم عوضاً عن المحارف المدخلة من قبل المستخدم.

Password:

صندوق التحقق checkbox

- يتم إعطاء اسم لصندوق التحقق باستخدام الوصفة name.
- يتم وضع الوصفة value لتحديد قيمة لصندوق التحقق.
- يكون صندوق التحقق إما محدداً أم لا.
- يُمكن استخدام الوصفة checked="checked" لجعل صندوق التحقق محدداً.

Checkbox checked

Checkbox unchecked

زر الخيار radio button

- لا يُمكن في مجموعة مترابطة من أزرار الخيار تحديد سوى زر واحد.
- يجب إعطاء نفس الاسم لكل أزرار الخيار في مجموعة مترابطة.
- تُحدد الوصفة value قيمة لزر الخيار.
- يُمكن استخدام الوصفة checked="checked" لجعل زر خيار محدداً.

- ◉ Radio button 1
- Radio button 2
- Radio button 3

زر الإرسال submit button

يقوم هذا الزر بحزم بيانات النموذج وإرسالها إلى الوجهة المحددة بالخاصية `action` للنموذج وبالطريقة المحددة بالخاصية `method`.

زر الإعادة reset button

يُعيد هذا الزر عناصر التحكم في النموذج إلى قيمها الابتدائية.

يُبين المثال التالي استخدام العناصر الستة السابقة:

```
<!DOCTYPE html >
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> Forms </title> </head>
<body>
<form action = "Server.html">
<p>Fiance<input type = "text" name ="Fiance" size="12" /></p>
<p>Salary <input type = "password" name ="Salary" size ="12" /> </p>
<p>
Own
<input type = "checkbox" name ="CO1" value="House"
checked = "checked"/> House
<input type = "checkbox" name ="CO2" value="Car"/> Car
<input type = "checkbox" name ="CO3" value="Gold"/> Gold
</p>
<p>
Age
<input type = "radio" name ="age" value = "under20"
checked = "checked"/> 0-19
<input type ="radio" name ="age" value = "20-35"/> 20-35
<input type ="radio" name ="age" value = "36-50"/> 36-50
<input type ="radio" name ="age" value = "over50"/> Over 50
</p>
<input type ="reset" value = "Reset Form"/>
<input type ="submit" value = "Submit Form"/>
</form>
</body>
</html>
```

حيث يُظهر المتصفح (بعد أن قمنا بإدخال بعض البيانات):

Fiance

Salary

Own House Car Gold

Age 0-19 20-35 36-50 Over 50

لاحظ أنه في هذا المثال تُرسل البيانات المدخلة إلى صفحة أخرى Server.html باستخدام الطريقة الافتراضية .get. وبهذا فإن البيانات ترسل في سلسلة محرفية في العنوان:

```
Server.html?Fiance=Bassel&Salary=15000&CO1=House&age=36-50
```

(لاحظ أن قيمة حقل كلمة السر المخفاة في النموذج تظهر في العنوان واضحة!).

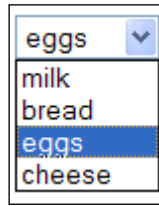
المؤثر <select>

يُمكن استخدام القوائم menu عوضاً عن صناديق التحقق أو أزرار الخيار لاسيما إذا كان عدد الخيارات كبيراً. يُستخدم المؤثر <select> لإنشاء قائمة. أما الخيارات فيستخدم المؤثر <option> لكل خيار. يُمكن السلوك الافتراضي للقائمة من اختيار عنصر وحيد (وبهذا يشابه سلوكها مجموعة أزرار خيار). أما إذا أردنا تحقيق إمكانية خيارات متعددة (وبهذا يصبح سلوك القائمة كمجموعة صناديق تحقق)، فيجب وضع الوصفة multiple="multiple" للمؤثر <select>. يُمكن استخدام الوصفة size لتحديد عدد العناصر الظاهرة. يُمكن جعل خيار محدد افتراضياً بوضع الوصفة selected="selected" للمؤثر <option>.

يُبين المثال التالي إنشاء قائمة:

```
<form action = "">
<p>
  <select name = "groceries" >
    <option> milk </option>
    <option> bread </option>
    <option selected="selected"> eggs </option>
    <option> cheese </option>
  </select>
</p>
```

حيث يُظهر المتصفح:



المؤثر <textarea>

يُستخدم المؤثر <textarea> لإنشاء صندوق نص متعدد الأسطر. تُحدد الوصفة rows عدد الأسطر والوصفة cols عدد الأعمدة. عند تجاوز قيم هذه الوصفات تظهر تلقائياً أشرطة إنزلاق موافقة. كما يُبين المثال التالي:

```
<form action = "">
  <p>
    Please provide your employment aspirations
  </p>
  <p>
    <textarea name = "aspirations" rows = "4" cols="40">
      (Be brief and concise)
    </textarea>
  </p>
</form>
```

حيث يُظهر المتصفح:

مثال:

يُبين المثال التالي استخدام العديد من العناصر المدروسة سابقاً لإنشاء واجهة تسوق بسيطة:

```
<!DOCTYPE html >

<!-- popcorn.html
  This describes popcorn sales form page
-->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Popcorn Sales Form </title>
  </head>
  <body>
    <form action = "Server.html" >
      <h2> Welcome to Millennium Gymnastics Booster Club Popcorn Sales
    </h2>

    <!-- A borderless table of text widgets for name and address -->

    <table>
```

```

    <tr>
      <td> Buyer's Name: </td>
      <td> <input type = "text" name = "name"
          size = "30" /> </td>
    </tr>
    <tr>
      <td> Street Address: </td>
      <td> <input type = "text" name = "street"
          size = "30" /> </td>
    </tr>
    <tr>
      <td> City, State, Zip: </td>
      <td> <input type = "text" name = "city"
          size = "30" /> </td>
    </tr>
  </table>
<p />

<!-- A bordered table for item orders -->

  <table border = "border">

<!-- First, the column headings -->

    <tr>
      <th> Product Name </th>
      <th> Price </th>
      <th> Quantity </th>
    </tr>

<!-- Now, the table data entries -->

    <tr>
      <th> Unpopped Popcorn (1 lb.) </th>
      <td> $3.00 </td>
      <td> <input type = "text" name = "unpop"
          size = "2" /> </td>
    </tr>
    <tr>
      <th> Caramel Popcorn (2 lb. canister) </th>
      <td> $3.50 </td>
      <td> <input type = "text" name = "caramel"
          size = "2" /> </td>
    </tr>
    <tr>
      <th> Caramel Nut Popcorn (2 lb. canister) </th>
      <td> $4.50 </td>
      <td> <input type = "text" name = "caramelnut"
          size = "2" /></td>
    </tr>
    <tr>
      <th> Toffey Nut Popcorn (2 lb. canister) </th>
      <td> $5.00 </td>
      <td> <input type = "text" name = "toffeynut"
          size = "2" /></td>
    </tr>
  </table>

<!-- The radio buttons for the payment method -->

  <h3> Payment Method: </h3>

```

```

<p>
  <input type = "radio" name = "payment"
    value = "visa"
    checked = "checked" /> Visa
  <input type = "radio" name = "payment"
    value = "mc" />
  Master Card
  <input type = "radio" name = "payment"
    value = "discover" /> Discover
  <input type = "radio" name = "payment"
    value = "check" /> Check <br/>
</p>

<!-- The submit and reset buttons -->
<p>
  <input type = "submit" value = "Submit Order" />
  <input type = "reset" value = "Clear Order Form" />
</p>
</form>
</body>
</html>

```

حيث يُظهر المتصفح:

**Welcome to Millennium
Gymnastics Booster Club Popcorn
Sales**

Buyer's Name:

Street Address:

City, State, Zip:

Product Name	Price	Quantity
Unpopped Popcorn (1 lb.)	\$3.00	<input type="text"/>
Caramel Popcorn (2 lb. canister)	\$3.50	<input type="text"/>
Caramel Nut Popcorn (2 lb. canister)	\$4.50	<input type="text"/>
Toffee Nut Popcorn (2 lb. canister)	\$5.00	<input type="text"/>

Payment Method:

Visa Master Card Discover Check

HTML5

الأهداف التعليمية:

- أهم المؤثرات والواصفات في HTML5.

المؤثرات الجديدة في HTML5

قدّمت HTML5 مجموعة من المؤثرات والواصفات الجديدة التي سهلت الكثير من القضايا التي كانت تتطلب من المبرمج كتابة العديد من الأسطر والتعليمات منها:

- المؤثر <video>
- المؤثر <audio>
- المؤثر <svg>
- النوع date للمؤثر <input>
- النوع number للمؤثر <input>
- النوع email للمؤثر <input>
- النوع url للمؤثر <input>
- المؤثر <output>
- الواصفة <placeholder>
- الواصفة <autofocus>
- الواصفة <required>

المؤثر <video>

يسمح بتضمين فيديو في الصفحة.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Video</title>
</head>
<body>
  <video src="media/HB.mp4" width="300" height="200" controls autoplay>
    Your browser does not support the video element.
  </video>
</body>
</html>
```

يقوم المتصفح بتشغيل ملف الفيديو الموافق:



المؤثر <video>

واصفات المؤثر <video>:

الخاصية	الوصف
autoplay	في حال وضع هذه الواصفة، سوف يتم تشغيل الفيديو تلقائياً.
controls	في حال وضع هذه الواصفة، سوف يتمكن المستخدم من التحكم بالفيديو (التشغيل، الإيقاف، الصوت، ...).
height	التحكم بارتفاع العنصر.
loop	في حال وضع هذه الواصفة، سوف يتم تشغيل الفيديو بشكل حلقة متواصلة.
poster	تُستخدم هذه الواصفة لتحديد الملف مصدر الصورة التي يُمكن إظهارها لحين قيام المستخدم بتشغيل الفيديو.
src	تُستخدم هذه الواصفة لتحديد الملف مصدر الفيديو.

المؤثر <audio>

يسمح هذا المؤثر بتضمين ملف صوتي في الصفحة بشكل بسيط.

يُبين المثال التالي استخدام هذا المؤثر:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Audio</title>
</head>
<body>
  <audio src="media/m.wav" controls autoplay>
    Your browser does not support the audio element.
  </audio>
</body>
</html>
```

حيث يقوم المتصفح بتشغيل ملف الصوت الموافق.

واصفات المؤثر <audio>:

الوصف	الخاصية
في حال وضع هذه الواصفة، سوف يتم تشغيل الملف الصوتي تلقائياً.	autoplay
في حال وضع هذه الواصفة، سوف يتمكن المستخدم من التحكم بالصوت (التشغيل، الإيقاف، الصوت، ...).	controls
في حال وضع هذه الواصفة ، سوف يتم تشغيل الصوت بشكل حلقة متواصلة.	loop
تُستخدم هذه الواصفة لتحديد الملف المصدر.	src

المؤثر <svg>

يُستخدم المؤثر <svg> لإنشاء الرسوم المتجهة.

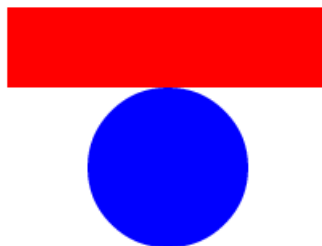
حيث SVG هي اختصار لـ Scalable Vector Graphics والتي هي لغة لتوصيف الرسومات ثنائية البعد في XML.

يوضح المثال التالي كيفية رسم مستطيل ودائرة:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>SVG</title>
</head>
<body>
  <h2>HTML5 SVG </h2>
  <svg id="svgelem" height="200" >
    <rect id="redrect" width="200" height="50" fill="red" />
    <circle id="redcircle" cx="100" cy="100" r="50" fill="blue" />
  </svg>
</body>
</html>
```

حيث يُظهر المتصفح:

HTML5 SVG



HTML5 – Web Forms 2.0

تم أيضاً إضافة العديد من الواصفات إلى مؤثرات النماذج.

- المؤثر `<input>`
 - النوع `date` للمؤثر `<input>`
 - النوع `number` للمؤثر `<input>`
 - النوع `email` للمؤثر `<input>`
 - النوع `url` للمؤثر `<input>`
- المؤثر `<output>`
 - الواصفة `<placeholder>`
 - الواصفة `<autofocus>`
 - الواصفة `<required>`

المؤثر `<input>`

إضافة للأأنواع المعروفة السابقة (`text`, `password`, `radio`, ...)، تمت إضافة مجموعة من الأنواع الجديدة. من أهمها:

Type	الوصف
<code>date</code>	يُظهر تقويم للمساعدة في إدخال التاريخ. لا يقبل إدخال تاريخ غير صحيح.
<code>number</code>	يقبل قيم رقمية فقط.
<code>email</code>	يُستخدم لإدخال بريد إلكتروني صالح.
<code>url</code>	يُستخدم لإدخال عنوان ويب صالح.

النوع `date` للمؤثر `<input>`

يُبين المثال التالي استخدام النوع `date` لإدخال قيمة تاريخ صحيح في نموذج:

```
<!DOCTYPE html>
<html>
<body>
  <form method="get">
    Date : <input type="date" name="newinput" />
    <input type="submit" value="submit" />
  </form>
</body></html>
```

حيث يظهر المتصفح تقويم لمساعدة المستخدم في إدخال التاريخ:

Date :

January, 2015

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

النوع number للمؤثر <input>

يُبين المثال التالي استخدام النوع number لإدخال قيمة رقمية محصورة ضمن مجال محدد في نموذج:

```
<!DOCTYPE html>
<html>
<body>
  <form method="get">
    Select Number : <input type="number" min="0" max="10" step "1" value="5"
name="newinput" />
    <input type="submit" value="submit" />
  </form>
</body></html>
```

حيث يُظهر المتصفح:

Select Number :

النوع email للمؤثر <input>

يُبين المثال التالي استخدام النوع email لإدخال قيمة بريد الكتروني صحيح في نموذج:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<form method="get">
  Enter email : <input type="email" name="newinput" />
  <input type="submit" value="submit" />
</form>
</body>
</html>
```

النوع url للمؤثر <input>

يُبين المثال التالي استخدام النوع url لإدخال قيمة عنوان صحيح في نموذج:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
</head>
<body>
  <form method="get">
    Enter URL : <input type="url" name="newinput" />
    <input type="submit" value="submit" />
  </form>
</body></html>
```

المؤثر <output>

يُستخدم هذا المؤثر الجديد لإظهار قيمة نتيجة ما على الصفحة.

يُبين المثال التالي استخدام المؤثر <output>. الذي يظهر النتيجة على الصفحة بالاعتماد على الاجرائية

المكتوبة بلغة² Java Script

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title> output </title>
  <script type="text/javascript">
    function showResult() {
      x = document.forms["myform"]["newinput"].value;
      document.forms["myform"]["result"].value = x;
    }
  </script>
</head>
<body>
  <form method="get" name="myform">
    Enter a value : <input type="text" name="newinput" />
    <input type="button" value="Result" onclick="showResult();" />
    <output name="result" />
  </form>
</body></html>
```

حيث سيتم كتابة القيمة المدخلة في صندوق الإدخال على العنصر :output

Enter a value : test

بعض الواصفات الجديدة

تم إضافة بعض الواصفات، من أهمها:

- الواصفة <placeholder>
- الواصفة <autofocus>
- الواصفة <required>

الواصفة <placeholder>

تسمح هذه الواصفة بإظهار محتوى باهت لإرشاد المستخدم ويختفي حالما يبدأ المستخدم بالكتابة.

```
<!DOCTYPE html>
<html>
<body>
  <form action="/cgi-bin/html5.cgi" method="get">
    Enter email : <input type="email" name="newinput"
placeholder="email@example.com" />
    <input type="submit" value="submit" />
  </form>
</body></html>
```

حيث يُظهر المتصفح:

Enter email :

الواصفة <autofocus>

تقوم هذه الواصفة بوضع التركيز على العنصر.

الواصفة <required>

تجعل هذه الواصفة قيمة العنصر واجبة الإدخال.

```
<!DOCTYPE html>
<html>
<body>
  <form action="/cgi-bin/html5.cgi" method="get">
    Enter email : <input type="email" name="newinput" required autofocus />
    <p>Try to submit using Submit button</p>
    <input type="submit" value="submit" />
  </form>
</body></html>
```


Registration Form

First Name:

Middle Initial:

Last Name:

City:

State/Province:

Country: United States
 Canada
 Other

Zip:

Email:

Username:

Password:

Repeat Password:

Comments:

I agree to the site terms.

Submit

Reset Form

الفصل الثالث:

أسلوب الصفحات المتتالي

الكلمات المفتاحية :

أسلوب الصفحات، التنسيق، الخطوط، الألوان، القوائم، الصور، نموذج الصندوق.

الملخص:

نستعرض في هذا الفصل أسلوب الصفحات المتتالي والذي يُمكن من إجراء التنسيق المطلوب لصفحات الويب.

الأهداف التعليمية:

يتعرف الطالب في هذا الفصل على:

- مستويات الأساليب.
- تنسيق الأسلوب.
- أشكال المحددات.
- أشكال خاصة-قيمة.
- أساليب الخطوط.
- أساليب القوائم.
- اللون.
- المحاذاة.
- نموذج الصندوق.
- صور الخلفية.

المخطط:

يضم فصل أسلوب الصفحات المتتالي 4 وحدات (Learning Objects) هي:

- أسلوب الصفحات المتتالي
- الخطوط
- خصائص القوائم
- أساليب أخرى

أسلوب الصفحات المتتالي

الأهداف التعليمية:

- مستويات الأساليب
- تنسيق الأسلوب
- أشكال المحددات
- أشكال خاصة-قيمة

مقدمة

يوفر أسلوب الصفحات المتتالي (CSS) Cascade Style Sheet طرق التحكم بمظهر الوثائق. كما يسمح بإعطاء مظهر متجانس لمجموعة من الوثائق. إذ يُمكن أسلوب الصفحات المتتالي بشكل رئيسي من تعريف مظهر معين لكل ظهور لمؤثر في وثيقة. سُمي بالأسلوب المتتالي لأنه يُمكن تعريف أسلوب الصفحات وفق ثلاثة مستويات متدرجة. يُهيمن المستوى الأدنى على المستوى الأعلى، وبهذا فإن الأسلوب لمحتوى مؤثر يُحدّد عبر تتالي تطبيق أسلوب الصفحات.

مستويات أسلوب الصفحات

يوجد ثلاثة مستويات من أسلوب الصفحات.

- الأسلوب الفوري Inline: يُحدّد الأسلوب لظهور معين لمؤثر. يظهر هذا الأسلوب في المؤثر نفسه.
- أسلوب الوثيقة Document-level style sheets: يُحدّد الأسلوب لكامل الوثيقة. يظهر هذا الأسلوب في رأس head الوثيقة.
- الأسلوب الخارجي External style sheets: يُمكن تطبيقه على مجموعة من الوثائق. يُكتب هذا الأسلوب في ملف نصي له اللاحقة CSS. يتم استخدام المؤثر <link> في رأس الوثيقة لتحديد ملف أسلوب خارجي:

```
<link rel = "stylesheet" type = "text/css"
href = "http://www.wherever.org/termpaper.css">
</link>
```

تنسيق تحديد الأسلوب

يختلف تنسيق الأسلوب وفق مستوى الأسلوب:

- الأسلوب الفوري Inline: يكون الأسلوب قيمة للواصفة style، ويأخذ الشكل العام التالي:

```
style = "property_1: value_1;
property_2: value_2;
...
property_n: value_n"
```

- أسلوب الوثيقة Document-level style sheets: يُحدّد الأسلوب كمجموعة من القواعد ضمن المؤثر <style>. يجب وضع قيمة الوصفة type=text/css. تُستخدم /* ... */ لوضع تعليقات ضمن القواعد.

```
<style type = "text/css">
rule list
</style>
```

1 عندما يُطبق أكثر من أسلوب على مؤثر في وثيقة يكون للمستوى الأدنى الأولوية

حيث تأخذ القاعدة الشكل:

```
selector {property_1: value_1;
          property_2: value_2;
          ...
          property_n: value_n}
```

- الأسلوب الخارجي External style sheets: يُكتب هذا الأسلوب في ملف نصي يحوي مجموعة من القواعد من الشكل السابق.

أشكال المحددات

يُمكن أن يأخذ المحدد selector عدة أشكال:

- شكل المحدد البسيط Simple Selectors
- التحديد السياقي Contextual Selectors
- محددات الصف Class Selectors
- المحددات العامة Generic Selectors
- محددات المعرفات id Selectors

شكل المحدد البسيط Simple Selectors

- يُمكن أن يكون المحدد اسم مؤثر وحيد مثل h1. وبالتالي، فإن قيم الخصائص في القاعدة ستُطبق على كل ظهور لهذا المؤثر في الوثيقة.
 - يُمكن أيضاً أن يكون المحدد مجموعة من المؤثرات المفصولة بفواصل. وفي هذه الحالة، ستُطبق قيم الخصائص على كل ظهور لهذه المؤثرات في الوثيقة.
 - إذا وضعنا * في المحدد، فإن قيم الخصائص ستُطبق على كامل الوثيقة.
- يُبين المثال التالي تحديد قيمة خاصية حجم الخط لبعض المؤثرات:

```
h1 {font-size: 24pt;}
h2, h3 {font-size: 20pt;}
```

التحديد السياقي Contextual Selectors

يُمكن أن يُخصص المحدد أن الأسلوب المطبق على عنصر يكون لوضع معين في الوثيقة. كما يُبين المثال التالي:

```
body b i {font-size: 30pt; }
```

والذي يُخصص أن حجم الخط لمحتوى غامق ومائل هو 30.

يُبين المثال التالي ذلك:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> Selector </title>
<style type="text/css" >
h1 {font-size: 24pt;}
h2, h3 {font-size: 20pt;}
body b i {font-size: 30pt; }
</style>
</head>
<body>
<h1> This is heading 1</h1>
<h2> This is heading 2</h2>
<h3> This is heading 3</h3>
<p>
This is to test <b> <i> contextual </i></b> selector
</p>
</body>
</html>
```

حيث يُظهر المتصفح:

This is heading 1

This is heading 2

This is heading 3

This is to test ***contextual*** selector

محددات الصف Class Selectors

- يسمح هذا النوع بتحديد مجموعة من الأساليب لنفس المؤثر.
- يكون للصف اسم معين يرتبط مع المؤثر.

مثلاً:

```
p.narrow {font-size:14;}  
p.wide {font-size:20;}
```

تُستخدم الوصفة class مع المؤثر لتحديد الصف المطلوب. كما يُبين المثال التالي:

```
<html xmlns = "http://www.w3.org/1999/xhtml">  
<head> <title> Selector </title>  
<style type="text/css" >  
p.narrow {font-size:14;}  
p.wide {font-size:20;}  
</style>  
</head>  
<body>  
<p class="narrow">  
This is a narrow paragraph !  
</p>  
<p class="wide">  
This is a wide paragraph !  
</p>  
</body>  
</html>
```

حيث يُظهر المتصفح:

```
This is a narrow paragraph !  
  
This is a wide paragraph !
```

المحددات العامة Generic Selectors

- يُمكن تعريف صفوف عامة بهدف تطبيق نفس الأسلوب على عدة مؤثرات.
- يجب إعطاء اسم للصف العام مسبقاً بنقطة.

مثلاً:

```
.really-big {font-size:30; }
```

تُستخدم الوصفة class مع مؤثرات مختلفة:

```
<h1 class = "really-big"> ...</h1>  
...  
<p class = "really-big"> ...</p>
```

محددات المعرفات id Selectors

يسمح المحدد المعرف بتطبيق أسلوب على عنصر واحد معين (له نفس المعرف)، ويأخذ الشكل:

```
#specific-id {property-value list }
```

فمثلاً إذا تم تعريف:

```
#section14 {font-size: 20}
```

فسيتم تطبيق هذا الأسلوب على:

```
<h2 id="section14"> Special Heading </h2>
```

أشكال خاصة - قيمة Property-Value

يوجد حوالي 60 خاصية تنتزع على 7 فئات:

- الخطوط Fonts
- القوائم Lists
- محاذاة النص Text Alignment
- الهوامش Margins
- الألوان Colors
- الخلفيات Backgrounds
- الحدود Borders

يُمكن أن تكون قيم الخصائص:

- كلمات مفتاحية:

large ■

medium ■

small ■

مثلاً: font-size:large

- قيم رقمية مع الوحدات التالية:

px – pixels ■

in – inches ■

cm – centimeters ■

mm – millimeters ■

pt – points ■

pc – picas (12 points) ■

em – height of the letter 'm' ■

ex-height – height of the letter 'x' ■

مثلاً: font-size:14pt

● يُمكن استخدام النسبة المئوية لتحديد أن القيمة الحالية هي نسبة مئوية من القيمة السابقة. فمثلاً إذا وضعنا: font-size:75%، يُصبح حجم الخط 75% من القيمة السابقة لحجم الخط.

● تأخذ قيم محددات المصدر الشكل: url (...)

مثلاً: body {background-image: url(photo.png);}

● يُمكن تحديد قيمة لون:

■ باستخدام اسم اللون: Color:white

■ بوضع قيمة اللون بالنظام السداسي عشر: Color:#FFFFFF

■ باستخدام الوظيفة rgb^2 : Color:rgb(255, 255, 255)

2 للوظيفة rgb ثلاثة معاملات تأخذ قيمها بين 0 و 255 وتُحدد بالترتيب درجة الأحمر، الأخضر، الأزرق

الخطوط

الأهداف التعليمية:

التعرف على أساليب الخطوط.

خصائص الخط Font Properties

- **اسم الخط font-family**

يستخدم المتصفح أول خط يدعمه من القائمة: font-family: Arial, Helvetica, Courier
إذا كان اسم الخط أكثر من كلمة فيجب استخدام المحرف ' :font-family: 'Times New Roman

- **حجم الخط font-size**

رقمية: حيث تكون القيمة إما رقم يُحدد حجم الخط أو نسبة مئوية من الخط السابق:

font-size:10pt

font-size:75%

كلمة مفتاحية: xx-small, x-small, small, medium, large, x-large, xxlarge

font-size:medium

- **نمط الخط font-style**

كلمة مفتاحية: bolder, lighter, bold, normal

font-weight:bolder

رقمية حيث تكون القيمة من مضاعفات 100 و محصورة بين 100 و 900. حيث القيمة 400 تكافئ

normal والقيمة 700 تكافئ bold:

font-weight:800

- **وزن الخط font-weight**

تكون قيمها normal, italic.

- **الخاصية font**

يُمكن استخدام الخاصية font لوضع قيم للخصائص السابقة وبالترتيب التالي: النمط، الوزن، الحجم، أسماء الخطوط:

font: italic bolder 14pt Arial Helvetica

- **زخرفة الخط font-decoration**

تكون قيمها: line-through, overline, underline

يُبين المثال التالي استخدام خصائص الخط:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Font properties </title>
  <style type = "text/css">
    p.big {font-size: 14pt;
           font-style: italic;
           font-family: 'Times New Roman';}
    p.small {font: 10pt bold 'Courier New';}
    h2 {font-family: 'Times New Roman';
        font-size: 24pt; font-weight: bold}
    h3 {font-family: 'Courier New'; font-size: 18pt}
  </style>
</head>
<body>
  <p class = "big">
    If a job is worth doing, it's worth doing right.
  </p>
  <p class = "small">
    Two wrongs don't make a right, but they certainly
    can get you in a lot of trouble.
  </p>
  <h2> Chapter 1 Introduction </h2>
  <h3> 1.1 The Basics of Computer Networks </h3>
</body> </html>
```

حيث يُظهر المتصفح:

If a job is worth doing, it's worth doing right.

Two wrongs don't make a right, but they certainly can get you in a lot of trouble.

Chapter 1 Introduction

1.1 The Basics of Computer Networks

يُبين المثال التالي استخدام ملف خارجي CSS:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> External style sheets </title>
  <link rel = "stylesheet" type = "text/css"
    href = "styles.css" />
</head>
<body>
  <p class = "big">
    If a job is worth doing, it's worth doing right.
  </p>
  <p class = "small">
    Two wrongs don't make a right, but they certainly
    can get you in a lot of trouble.
  </p>
  <h2> Chapter 1 Introduction </h2>
  <h3> 1.1 The Basics of Computer Networks </h3>
</body>
</html>
```

حيث يكون محتوى الملف styles.css:

```
p.big {font-size: 14pt;
      font-style: italic;
      font-family: 'Times New Roman';
      }
p.small {font: 10pt bold 'Courier New';}
h2 {font-family: 'Times New Roman';
    font-size: 24pt; font-weight: bold}
h3 {font-family: 'Courier New';
    font-size: 18pt}
```

ويُظهر المتصفح:

If a job is worth doing, it's worth doing right.

Two wrongs don't make a right, but they certainly can get you in a lot of trouble.

Chapter 1 Introduction

1.1 The Basics of Computer Networks

خصائص القوائم

الأهداف التعليمية:

- التعرف على أساليب القوائم.

خصائص القوائم List Properties

- يُمكن في القوائم تحديد:
 - شكل الرسمة التي تسبق العنصر في القوائم غير المرتبة
 - طريقة ترقيم القوائم المرتبة
- تُحدد الخاصية `list-style-type` شكل الرسمة التي تسبق العنصر وتكون قيمها: `disc`, `circle`, `square`, `none`
- يُمكن وضع هذه الخاصية للمؤثر `` وبالتالي يكون تأثيرها على جميع عناصر القائمة. أو وضعها للمؤثر `` فيكون تأثيرها على العنصر المعني فقط.
كما يُبين المثال التالي:

```
<h3> Some Common Single-Engine Aircraft </h3>
<ul style = "list-style-type: square">
  <li> Cessna Skyhawk </li>
  <li> Beechcraft Bonanza </li>
  <li style = "list-style-type: circle">
    Piper Cherokee </li>
</ul>
```

حيث يُظهر المتصفح:

Some Common Single-Engine Aircraft

- Cessna Skyhawk
- Beechcraft Bonanza
- Piper Cherokee

- يُمكن وضع صورة محددة باستخدام الخاصية `:list-style-image`
`<li style = "list-style-image: url(bird.jpg)">`
- يُمكن استخدام الخاصية `list-style-type` مع القوائم المرتبة حيث تأخذ قيمها:

قيمة الخاصية	نوع الترقيم	مثال
decimal	Arabic numerals	1, 2, 3, 4
upper-alpha	Uc letters	A, B, C, D
lower-alpha	Lc letters	a, b, c, d
upper-roman	Uc Roman	I, II, III, IV
lower-roman	Lc Roman	i, ii, iii, iv

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Sequence types </title>
  <style type="text/css">
    ol {list-style-type: upper-roman;}
    ol ol {list-style-type: upper-alpha;}
    ol ol ol {list-style-type: decimal;}
  </style>
</head>
<body>
  <h3>
    Aircraft Types
  </h3>
  <ol>
    <li>General Aviation (piston-driven engines)
      <ol>
        <li>Single-Engine Aircraft
          <ol>
            <li>Tail wheel </li>
            <li>Tricycle </li>
          </ol>
        </li>
        <li>Dual-Engine Aircraft
          <ol>
            <li>Wing-mounted engines </li>
            <li>Push-pull fuselage-mounted engines </li>
          </ol>
        </li>
      </ol>
    </li>
    <li>Commercial Aviation (jet engines)
      <ol>
        <li>Dual-Engine
          <ol>
            <li>Wing-mounted engines </li>
            <li>Fuselage-mounted engines </li>
          </ol>
        </li>
        <li>Tri-Engine
          <ol>
            <li>Third engine in vertical stabilizer </li>
            <li>Third engine in fuselage </li>
          </ol>
        </li>
      </ol>
    </li>
  </ol>
</body> </html>

```


Aircraft Types

I. General Aviation (piston-driven engines)

A. Single-Engine Aircraft

1. Tail wheel
2. Tricycle

B. Dual-Engine Aircraft

1. Wing-mounted engines
2. Push-pull fuselage-mounted engines

II. Commercial Aviation (jet engines)

A. Dual-Engine

1. Wing-mounted engines
2. Fuselage-mounted engines

B. Tri-Engine

1. Third engine in vertical stabilizer
2. Third engine in fuselage

أساليب أخرى

الأهداف التعليمية

- اللون
- المحاذاة
- نموذج الصندوق
- صور الخلفية

اللون Color

• الألوان الـ 16 الأساسية

هي الألوان التي تظهر على أي متصفح:

black	000000	green	008000
silver	C0C0C0	lime	00FF00
gray	808080	olive	808000
white	FFFFFF	yellow	FFFF00
maroon	800000	navy	000080
red	FF0000	blue	0000FF
purple	800080	teal	008080
fuchsia	FF00FF	aqua	00FFFF

• لوحة ألوان الويب المولفة من 216 لون

هي الألوان التي تُظهرها معظم المتصفحات. يُمكن توليدها من تركيب الأرقام: 00,33,66,99,CC,FF لتحديد الأحمر والأخضر والأزرق. (الأحمر مثلاً FF0000).

• جميع الألوان الـ 16 مليون (bits24).

في حال عدم تمكن المتصفح من إظهار نفس اللون فإنه يُظهر أقرب لون له. تُستخدم الخاصية color لتحديد اللون الأمامي والخاصية background-color لتحديد لون الخلفية. كما يُبين المثال التالي:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Colors </title>
  <style type = "text/css">
    p.redback {font-size: 24pt; color:blue;
               background-color: red;}
  </style>
</head>
<body>
  <p class = "redback">
    To really make it stand out, use a red background!
  </p>
</body>
</html>
```

حيث يُظهر المتصفح:

To really make it stand out,
use a red background!

محاذاة النص

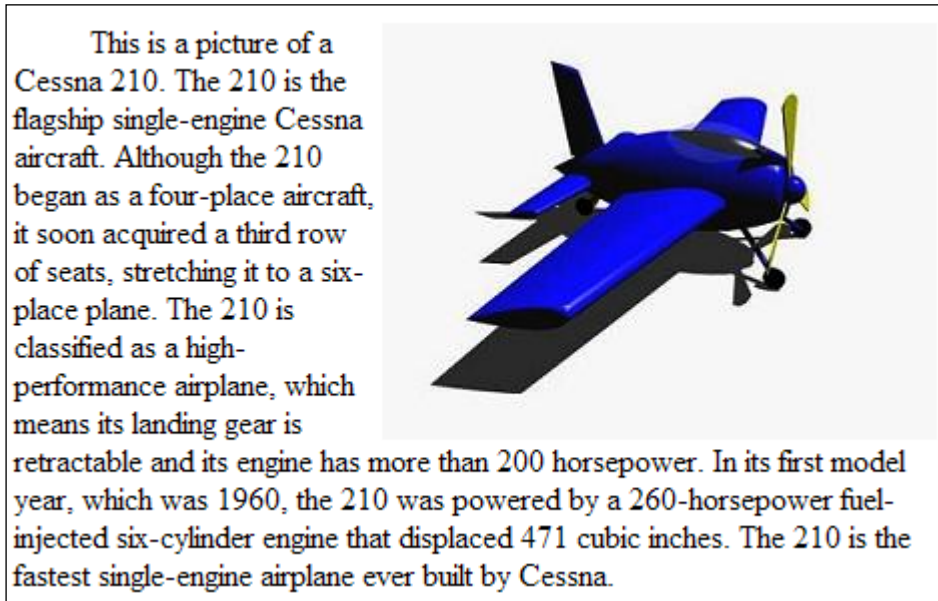
يُمكن استخدام الخاصية `text-indent` لتحديد المسافات البادئة للسطر الأول لفقرة. تأخذ هذه الخاصية قيمة رقمية أو نسبة مئوية.

أما الخاصية `text-align` فنُحدد محاذاة النص وتأخذ إحدى القيم التالية: `left`, `center`, `right`, `justify`. تُستخدم الخاصية `float` غالباً مع الصور والجداول لتحديد كيفية تدفق النص حول الصورة أو الجدول، وتكون قيمها `left`, `right`, `none`.

يُبين المثال التالي استخدام جميع الخصائص السابقة:

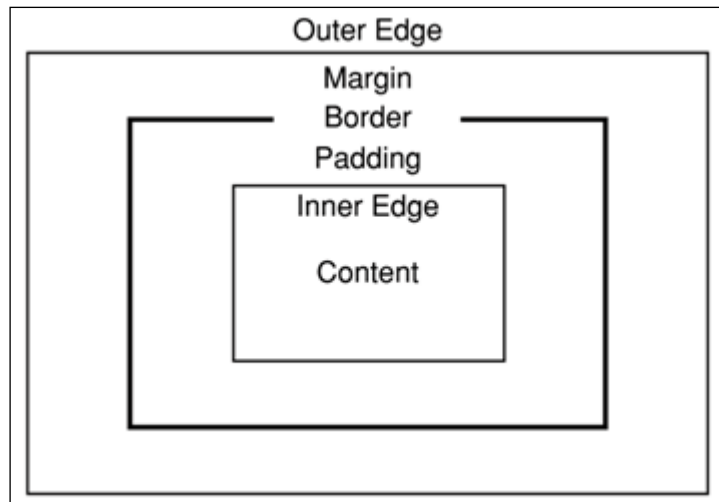
```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> The float property </title>
  <style type = "text/css">
    p.indent {text-indent: 1cm; text-align:left}
    img {float: right}
  </style>
</head>
<body>
  <p>
  
  </p>
  <p class="indent">
    This is a picture of a Cessna 210. The 210 is the flagship
    single-engine Cessna aircraft. Although the 210 began as a
    four-place aircraft, it soon acquired a third row of seats,
    stretching it to a six-place plane. The 210 is classified
    as a high-performance airplane, which means its landing
    gear is retractable and its engine has more than 200
    horsepower. In its first model year, which was 1960,
    the 210 was powered by a 260-horsepower fuel-injected
    six-cylinder engine that displaced 471 cubic inches.
    The 210 is the fastest single-engine airplane ever
    built by Cessna.
  </p>
</body>
</html>
```

حيث يُظهر المتصفح:



نموذج الصندوق The Box Model

يُمكن لكل عنصر في وثيقة أن يكون له حدود تأخذ أنماطاً مختلفة كاللون والعرض. تُدعى المسافة بين العنصر وحدوده بالبطانة padding أما المسافة بين العنصر والعناصر الأخرى المجاورة له فتُدعى الهامش margin. كما يُبين الشكل التالي:



نموذج الصندوق The Box Model

الحدود

- تُستخدم الخاصية `border-style` لتحديد نمط الحدود وتأخذ القيم: `solid`, `dotted`, `dashed`, `double`, `none`.
- تُحدد الخاصية `border-width` عرض الحد ويُمكن أن تأخذ القيم: `thin`, `medium` (default), `thick`, أو قيمة رقمية بالكسل. كما يُمكن تحديد عرض مختلف لكل حد باستخدام الخصائص التالية: `border-top-width`, `border-bottom-width`, `border-left-width`, `border-right-width`.
- يُحدد لون الحدود باستخدام الخاصية `border-color`. كما يُمكن تحديد لون لكل حد باستخدام الخصائص: `border-top-color`, `border-bottom-color`, `border-left-color`, `border-right-color`.

يُبين المثال التالي استخدام هذه الخصائص:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Table borders </title>
  <style type = "text/css">
    table {border-top-width: medium;
           border-bottom-width: thick;
           border-top-color: red;
           border-bottom-color: blue;
           border-top-style: dotted;
           border-bottom-style: dashed;
          }
    p {border-style: dashed; border-width: thin;
       border-color: green
      }
  </style>
</head>
<body>
  <table border = "5">
    <caption> Fruit Juice Drinks </caption>
    <tr>
      <th> </th>
      <th> Apple </th>
      <th> Orange </th>
      <th> Screwdriver </th>
    </tr>
    <tr>
      <th> Breakfast </th>
      <td> 0 </td>
      <td> 0 </td>
    </tr>
    <tr>
      <th> Lunch </th>
      <td> 1 </td>
      <td> 0 </td>
      <td> 0 </td>
    </tr>
  </table>
</body>
</html>
```

```

    <th> Dinner </th>
    <td> 0 </td>
    <td> 0 </td>
    <td> 1 </td>
  </tr>
</table>
<p>
  Now is the time for all good Web programmers to
  learn to use style sheets.
</p>
</body>
</html>

```

حيث يُظهر المتصفح:

Fruit Juice Drinks			
	Apple	Orange	Screwdriver
Breakfast	0	0	
Lunch	1	0	0
Dinner	0	0	1

Now is the time for all good Web programmers to learn to use style sheets.

البطانة والهوامش

- يُمكن استخدام الخاصية margin لتحديد هوامش العنصر الأربعة. أو استخدام الخصائص -margin: top, margin-bottom, margin-left, margin-right.
- يُمكن استخدام الخاصية padding لتحديد بطانة العنصر، أو الخصائص -padding: top, padding-bottom, padding-left, padding-right.

يُبين المثال التالي مختلف الحالات الممكنة:

```

<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Table borders </title>
  <style type = "text/css">
    p.one   {margin: 0.2in;
             padding: 0.2in;
             background-color: #C0C0C0;
             border-style: solid;
            }
    p.two   {margin: 0.1in;
             padding: 0.3in;
             background-color: #C0C0C0;
             border-style: solid;
            }
    p.three {margin: 0.3in;
             padding: 0.1in;
             background-color: #C0C0C0;
             border-style: solid;
            }
    p.four  {margin:0.4in;
             background-color: #C0C0C0;}
    p.five  {padding: 0.4in;
             background-color: #C0C0C0;
            }
  </style>
</head>
<body>
  <p>
    Here is the first line.
  </p>
  <p class = "one">
    Now is the time for all good Web programmers to
    learn to use style sheets. <br /> [margin = 0.2in,
    padding = 0.2in]
  </p>
  <p class = "two">
    Now is the time for all good Web programmers to
    learn to use style sheets. <br /> [margin = 0.1in,
    padding = 0.3in]
  </p>
  <p class = "three">
    Now is the time for all good Web programmers to
    learn to use style sheets. <br /> [margin = 0.3in,
    padding = 0.1in]
  </p>
  <p class = "four">
    Now is the time for all good Web programmers to
    learn to use style sheets. <br /> [margin = 0.4in,
    no padding, no border]
  </p>
  <p class = "five">
    Now is the time for all good Web programmers to
    learn to use style sheets. <br /> [padding = 0.4in,
    no margin, no border]
  </p>
  <p>
    Here is the last line.
  </p>
</body>

```


Here is the first line.

Now is the time for all good Web programmers to learn to use style sheets.
[margin = 0.2in, padding = 0.2in]

Now is the time for all good Web programmers to learn to use style sheets.
[margin = 0.1in, padding = 0.3in]

Now is the time for all good Web programmers to learn to use style sheets.
[margin = 0.3in, padding = 0.1in]

Now is the time for all good Web programmers to learn to use style sheets.
[margin = 0.4in, no padding, no border]

Now is the time for all good Web programmers to learn to use style sheets.
[padding = 0.4in, no margin, no border]

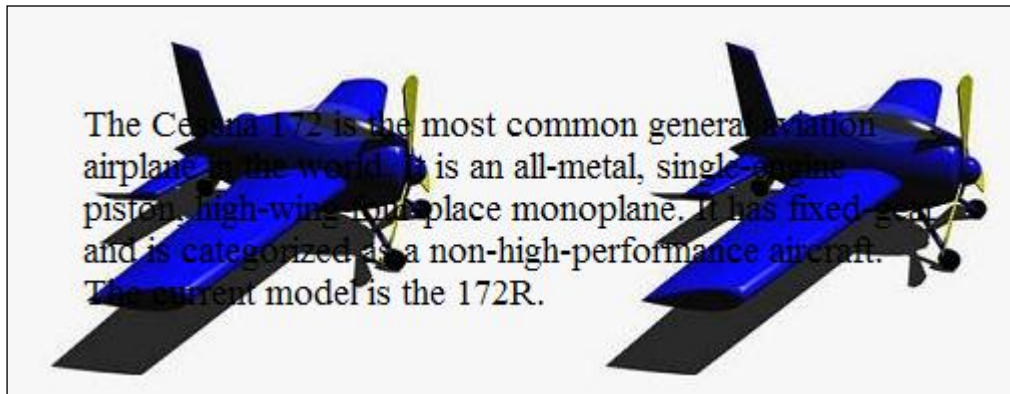
Here is the last line.

صور الخلفية

تُستخدم الخاصية background-image لوضع صورة في الخلفية. كما يُبين المثال التالي:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Background images </title>
  <style type = "text/css">
    body {background-image: url(plane.png);}
    p {margin-left: 30px; margin-right: 30px;
      margin-top: 50px; font-size: 14pt;}
  </style>
</head>
<body>
  <p >
    The Cessna 172 is the most common general aviation
    airplane
    in the world. It is an all-metal, single-engine piston,
    high-wing four-place monoplane. It has fixed-gear and is
    categorized as a non-high-performance aircraft. The current
    model is the 172R.
  </p>
</body>
</html>
```

حيث يُظهر المتصفح:



يقوم المتصفح بتكرار الصورة بحيث تملئ مساحة العنصر.

يُمكن التحكم بعملية التكرار باستخدام الخاصية background-repeat والتي تأخذ القيم: repeat (default), .no-repeat, repeat-x, repeat-y.

تُحدد القيمة repeat-x أن التكرار يكون أفقياً فقط. أما القيمة repeat-y فتُحدد أن التكرار يكون عمودياً فقط.

يُمكن استخدام الخاصية background-position والتي يُمكن أن تأخذ عدد من القيم: top, center, .bottom, right, center.

لتحديد مكان توضع الصورة.

المؤثرات و <div>

يوفر المؤثر إمكانية تخصيص نمط لجزء من عنصر. ففي المثال التالي نقوم بتخصيص نمط كلمة ضمن فقرة:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Span </title>
  <style type = "text/css">
    .spanred {font-size: 24pt;
              font-family: Arial; color: red}
  </style>
</head>
<body>
  <p >
    It sure is fun to be in <span class="spanred"> total </span>
    control of text
  </p>
</body>
</html>
```

يُظهر المتصفح:

It sure is fun to be in **total** control of text

يُمكن أيضاً استخدام المؤثر <div> لتحديد نفس الأسلوب لمجموعة من العناصر:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> Div </title>
  <style type = "text/css">
    .divred {font-size: 16pt;
              font-family: Ariel; color: red}
  </style>
</head>
<body>
<p >
  It sure is fun to be in total
  control of text
</p>
<div class="divred">
  <p >
    It sure is fun to be in total
    control of text
  </p>
  <p >
    It sure is fun to be in total
    control of text
  </p>
</div>
</body>
</html>
```

حيث يُظهر المتصفح:

It sure is fun to be in total control of text

It sure is fun to be in total control of text

It sure is fun to be in total control of text

اقتراحات وتمارين

1. قم بتعريف ترويساتك المخصصة في ملف (cssExercise.css):

h1, h2, h3, h4, h5, h6

2. قم بتعريف ثلاثة صفوف للفقرات p1,p2,p3 وفق ما يلي (في الملف cssExercise.css):

- p1:
 - left and right margins of 20 pixels
 - background color of Pink (#FF00FF)
 - foreground color of blue.
 - font-family of Arial,Helvetica,sans-serif
 - font-size 10pt
 - font-weight normal
 - left aligned text
- p2:
 - left and right margins of 30 pixels
 - background color of black
 - foreground color of #FFFF00
 - font-family of Arial,Helvetica,sans-serif
 - font-size 10pt
 - font-weight normal
- p3:
 - text indent of 1 centimeter
 - background color of rgb(204, 51, 51)
 - foreground color of white
 - font-family of Arial,Helvetica,sans-serif
 - font-size 10pt
 - font-weight normal

3. قم بإنشاء ست فقرات لكل فقرة ترويسة (استخدم من h1 إلى h6). تستخدم الفقرة الأولى والرابعة p1.

تستخدم الفقرة الثانية والخامسة p2. تستخدم الفقرة الثالثة والسادسة p3.

يجب أن تظهر الصفحة كما يلي:

Heading 1

This is paragraph 1

Heading 2

This is paragraph 2

Heading 3

This is paragraph 3

Heading 4

This is paragraph 4

Heading 5

This is paragraph 5

Heading 6

This is paragraph 3

4. قم بتعريف الأساليب اللازمة لكتابة القوائم وفق المثال التالي:


- I. ant farming
- II. hang gliding
 - A. falling down
 - 1. leaking blood
 - B. breaking things
- III. drooling

5. قم بتعريف الأساليب اللازمة لإظهار الفقرتين التاليتين مع الصور الموافقة كما يلي:

Raymond Feist is a remarkable author who has written over 30 books. His first series was The Riftwar Saga, which includes Magician: Apprentice, Magician: Master, Silverthorn, and A Darkness at Sethanon. Other series include The Empire Trilogy, Kronadors Sons, The Serpentwar Saga, The Riftwar Legacy, Legends of the Riftwar, Conclave of Shadows, The Darkwar Saga, Demonwar Saga, and the Chaoswar Saga. Feist is a very prolific author whose stories sprang from his involvement in the Dungeons and Dragons role-playing game.



Terry Brooks is another of my favorite authors. His Shannara books are among the best of the swords and sorcery genre. Beginning with the Shannara Trilogy, the stories continue with the Heritage of Shannara tetralogy, The Voyage of the Jerle Shannara trilogy, the High Druid of Shannara trilogy, and the Genesis of Shannara trilogy. He has also written the Word and the Void series and the Magic Kingdom of Landover series. Brooks was inspired by The Lord of the Rings by J.R.R. Tolkien.



6. قم بتعريف الأساليب اللازمة لإنشاء قائمة غير مرتبة من الكتب وبحيث يكون لكل عنصر في القائمة صورة الكتاب الموافق:

Some of Feist's books include



Magician



Silverthorn



A Darkness at Sethanon

الفصل الرابع:

أساسيات لغة JavaScript

الكلمات المفتاحية

المتغيرات، أنماط البيانات، التعبيرات والعمليات، الإدخال والإخراج، الشرط، التكرار، الوظائف، المصفوفات، مطابقة النماذج.

المُلخَص

نستعرض في هذا الفصل أساسيات لغة البرمجة JavaScript.

الأهداف التعليمية

يتعرف الطالب في هذا الفصل على

- المتغيرات.
- الأنماط.
- العمليات.
- التعبيرات.
- الإدخال والإظهار.
- تعليمات التحكم.
- المصفوفات.
- الوظائف.
- مطابقة النماذج.

المخطط:

يضم فصل أساسيات لغة JavaScript 5 وحدات (Learning Objects) هي:

- أساسيات JavaScript
- تعليمات التحكم
- المصفوفات
- الوظائف Functions
- مطابقة النماذج

أساسيات JavaScript

الأهداف التعليمية:

- أساسيات البرمجة غرضية التوجه و JavaScript
- الخصائص العامة للشكل
- التعليمات
- الأنماط والعمليات والتعبير
- التصريح عن المتغيرات
- العمليات الرقمية
- تحويل الأنماط الضمني
- تحويل الأنماط الصريح
- الغرض Date
- الإدخال والإظهار
- الغرض window

مقدمة

ظهرت اللغة الخطاطية JavaScript عام 1996 نتيجة تعاون مئثر بين الشركتين Netscape و Sun. وأصبحت اليوم معيار عالمي معتمد ISO-16262. تتألف لغة JavaScript من ثلاثة أجزاء رئيسية:

- **قلب اللغة Core**

تحتوي التعليمات الأساسية للغة

- **جهة الزبون Client Side**

تحتوي مجموعة الأغراض التي تدعم التحكم بالمستعرض والتفاعل مع المستخدم (وهو الجزء الأكثر استخداماً من قبل مطوري الويب)

- **جهة المخدم Server Side**

تحتوي مجموعة الأغراض التي يُمكن أن تتعامل مع مخدم الويب، كعمليات الوصول إلى قواعد البيانات مثلاً.

البرمجة غرضية التوجه و JavaScript

لا تدعم JavaScript البرمجة غرضية التوجه. أي أنها لا تحتوي صفوف classes. بل تعمل أغراضها objects كأغراض وكنموذج للأغراض في آن واحد models of objects. كما لا تدعم JavaScript الوراثة Inheritance وتعددية الأشكال Polymorphism. يكون للأغراض في JavaScript مجموعة من الخصائص التي يُمكن أن تكون إما خاصية معطيات property أو طريقة method.

الخصائص العامة للشكل

- يُمكن تضمين الخططات مباشرة في ملف XHTML:

```
<script type="text/javascript">
  -- JavaScript script -
</script>
```

- أو (وهو الأفضل) وضع الخططات في ملف نصي مستقل وتضمينها باستخدام:

```
<script type="text/javascript"
  src="myScript.js">
</script>
```

- يُمكن وضع تعليمات اللغة ضمن تعليقات خاصة من الشكل التالي وذلك بهدف إخفاء التعليمات عن المتصفحات التي لاتدعم JavaScript:

```
<!--
-- JavaScript script -
//-->
```

المُعَرِّفَات

- يجب أن تبدأ المعرفات Identifiers في JavaScript بحرف أو تحت السطر (_) أو إشارة الدولار \$.
بعدها يُمكن أن يحوي المعرف على حرف أو رقم أو تحت السطر أو دولار.
- لا يوجد حد لطول المعرف.
- تكون JavaScript حساسة لحالة الأحرف case sensitive.

الكلمات المفتاحية

تحتوي اللغة على مجموعة من الكلمات المفتاحية:

break, case, catch, continue, default, delete, do, else, finally, for, function, if, in, .instanceof, new, return, switch, this, throw, try, typeof, var, void, while, with

التعليقات

يُمكن وضع التعليقات على سطر باستخدام // أو على أكثر من سطر باستخدام /* */ .

التعليمات

يُستحسن وضع كل تعليمة على سطر وإنهائها بوضع فاصلة منقوطة (;) . إذ أن مفسر اللغة interpreter يضع فاصلة منقوطة عند كل نهاية سطر لما يعتبره تعليمة. مما قد يقود لخطأ كما يُبين المثال التالي:

```
return x;
```

(حيث سيقوم المفسر بوضع فاصلة منقوطة بعد return، مما سيجعل وضع x غير قانوني).

الأنماط والعمليات والتعابير

الأنماط الأساسية

يكون للقيم أحد الأنماط الأساسية التالية:

Number, String, Boolean, Undefined, Null.

الأرقام والسلاسل النصية

تُخزن الأرقام باستخدام الفاصلة العائمة مع دقة مضاعفة. تُبين الأمثلة التالية أشكال صحيحة للأرقام:

72, 7.2, .72, 72., 7E2, 7e2, .7e2, 7.e2, 7.2E-2

تُحاط السلاسل النصية إما بإشارة تنصيص واحدة (') أو بإشارتي تنصيص ("). يُمكن أن تحوي السلاسل النصية على محارف خاصة مثل \n و \t.

يُمكن استخدام المحرف (\) لإلغاء: (')

'You're the most freckly person I've ever met'

كما يجب وضع (\) قبل كل (\) إذا كان من محارف السلسلة: "d:\\bookfiles"

لايوجد فرق بين السلسلة المحاطة بـ (') والمحاطة بـ ("). كما يُمكن التعبير عن السلسلة الفارغة بـ "" أو "".

الأنماط الأساسية الأخرى

- يكون في النمط Boolean القيمتين true و false فقط. تنتج هذه القيم عادةً عن حساب تعبير منطقي.
- يكون في النمط Null قيمة وحيدة هي الكلمة المفتاحية null. وتُعامل كـ 0 عند معاملتها كرقم، وكـ false عند معاملتها كمتغير منطقي.
- يكون في النمط Undefined قيمة وحيدة هي undefined. وتُعامل كـ NaN عند معاملتها كرقم، وكـ false عند معاملتها كمتحول منطقي.
- يكون متغير undefined عندما يكون معرفاً ولم تُسند له قيمة.

فمثلاً إذا كتبنا التعليمات التالية:

```
<script type="text/javascript">
  var a;
  var b = 10;
  b = b + a;
  document.write("a: ", a, "<br />");
  document.write("b: ", b, "<br />");
</script>
```

تكون النتيجة:

a: undefined
b: NaN

التصريح عن المتغيرات

تتميز JavaScript بأنها تقوم بتحديد نمط المتغير بشكل ديناميكي حسب القيمة المسندة له. كما يُمكن إسناد قيمة من أي نمط لنفس المتغير.

يُمكن التصريح عن متغير بشكل ضمني وذلك بإسناد قيمة له: `a= 10;`
أو بشكل صريح باستخدام الكلمة المفتاحية `var`:

```
var a,  
sum = 0,  
today = "Monday",  
flag = false;
```

العمليات الرقمية

توفر JavaScript العمليات الرقمية:

`++`, `--`, `+`, `-`, `*`, `/`, `%`

تتبع JavaScript أفضلية العمليات حيث تكون أفضلية `*/%` أعلى من `+,-`. وفي حال تساوي الأفضليات في تعبير يتم تطبيق العمليات من اليسار لليمين.

مثال:

يُبين المثال التالي كيفية تطبيق الأفضليات والتجميع:

```
<script type="text/javascript">  
  var a = 2,  
      b = 4,  
      c,  
      d;  
  c = 3 + a * b;  
  // * is first, so c is now 11 (not 24)  
  d = b / a / 2;  
  // / association left, so d is now 1 (not 4)  
  document.write("c: ", c, "<br />");  
  document.write("d: ", d, "<br />");  
</script>
```

يُمكن استخدام الأقواس لتحديد الأفضلية المطلوبة.

الغرض Math

يوفر الغرض Math مجموعة من الطرق على الأرقام مثل:

`floor`, `round`, `max`, `min`, `cos`, `sin`. . . ,

مثلاً:

`Math.sin(x)`

الغرض Number

- يوفر الغرض Number مجموعة من الخصائص ذات القيم الثابتة الرقمية: MAX_VALUE, MIN_VALUE, NaN, POSITIVE_INFINITY, NEGATIVE_INFINITY, PI.
(مثلاً، تُعطي Number.MIN_VALUE أصغر قيمة ممكنة).
- تُعيد عملية جبرية مع فيضان overflow القيمة NaN.
- تُستخدم الدالة isNaN() لاختبار أن متغير له القيمة NaN.
- للغرض Number الطريقة toString() لإرجاع الرقم كسلسلة نصية:

```
Var price= 477,  
Str_price;  
...  
Str_price = price.toString();
```

جمع السلاسل النصية

تُستخدم إشارة الجمع + لجمع السلاسل النصية:

```
<script type="text/javascript">  
var x = "Hello";  
x = x + " World";  
// x now is Hello World  
document.write( x, "<br />");  
</script>
```

تحويل الأنماط الضمني

تقوم JavaScript بمجموعة من التحويلات الضمنية بين الأنماط وفق ما يلي:

- إذا كانت العملية عملية جمع بين رقم وسلسلة نصية يتم تحويل الرقم إلى سلسلة نصية.
- إذا كانت العملية عملية حسابية (غير الجمع) يتم تحويل السلسلة النصية إلى رقم.
- إذا فشلت عملية تحويل السلسلة النصية إلى رقم تُعاد القيمة NaN.

مثال:

يُبين المثال التالي مختلف حالات التحويل الضمني:

```
<script type="text/javascript">
var x, y, z, t;
x = "August " + 2007;
// x now is August 2007
document.write("x now is ", x, "<br />");
y = 2007 + " August";
// y now is 2007 August
document.write("y now is ", y, "<br />");
z = 7 * "3";
// z now is 21
document.write("z now is ", z, "<br />");
t = "lo" * 3;
// t now is NaN
document.write("t now is ", t, "<br />");
</script>
```

تحويل الأنماط الصريح

يُمكن طلب التحويل بين الأنماط بشكل صريح كما يلي:

- يُستخدم الباني String للحصول على سلسلة نصية.
- يُستخدم الباني Number للحصول على رقم.
- تُستخدم الطريقة toString() على رقم لتحويله إلى سلسلة نصية.
- يُمكن استخدام الدالة parseInt لتحويل سلسلة نصية إلى رقم صحيح.
- يُمكن استخدام الدالة parseFloat لتحويل سلسلة نصية إلى رقم عشري.

مثال:

يُبين المثال التالي مختلف حالات التحويل الصريح:

```
<script type="text/javascript">
var str1 = String(33.33);
// str1 now is "33.33"
document.write("str1 now is ", str1, "<br />");
var num1 = 6.6;
var str2 = num1.toString();
//str2 now is "6.6"
document.write("str2 now is ", str2, "<br />");
var num2 = Number(str1);
// num2 now is 33.33
document.write("num2 now is ", num2, "<br />");
var num3 = str1 - 0;
// num3 now is 33.33
document.write("num3 now is ", num3, "<br />");
var num4 = parseInt(str1);
document.write("num4 now is ", num4, "<br />");
// num4 now is 33
var num5 = parseFloat(str1);
// num5 now is 33.33
document.write("num5 now is ", num5, "<br />");
</script>
```


خصائص وطرق السلاسل String

- للغرض String خاصية واحدة هي length وتعطي عدد الأحرف في سلسلة نصية.
- للغرض String مجموعة من الطرق أهمها:

- charAt(number)
- indexOf(One-character string)
- substring(number1, number2)
- toLowerCase()
- toUpperCase()

كما تُبين الأمثلة:

```
var str="George";  
str.length is 6  
str.charAt(2) is 'o'  
str.indexOf('r') is 3  
str.substring(2, 4) is 'or'  
str.toLowerCase() is 'george'
```

الطريقة typeof

تُعيد typeof نمط متغير كما تُبين الأمثلة التالية:

```
typeof("George") is string  
typeof(33) is number  
typeof(true) is Boolean  
var a; typeof(a) is undefined  
typeof(b) is undefined (b is a not defined var)
```

الإسناد

تُستخدم إشارة المساواة للإسناد:

```
a = a + 7;  
a += 7;
```

الغرض Date

يُمكن التصريح عن متغير تاريخ يأخذ التاريخ والوقت الحالي:

```
var today = new Date();
```

يوفر الغرض Date مجموعة من الطرق:

toLocaleString, getDate, getMonth, getDay, getFullYear, getTime, getHours, getMinutes, getSeconds, getMilliseconds.

الإدخال والإظهار

يقوم مفسر JavaScript بتنفيذ التعليمات حيثما يجدها ضمن المستند XHTML. وبالتالي فإن شاشة الإظهار الطبيعية لـ JavaScript هي نفسها شاشة إظهار مخرجات XHTML. تُستخدم الطريقة write للغرض document بشكل أساسي لخلق خطأ خرج. كما يُبين المثال التالي:

```
var a = 25;  
document.write("The result is : <b> ", a, "</b> <br />");
```

حيث يكون الخرج:

The result is : 25

لاحظ أن معامل الطريقة write يُمكن أن يحوي أي مؤثر XHTML.

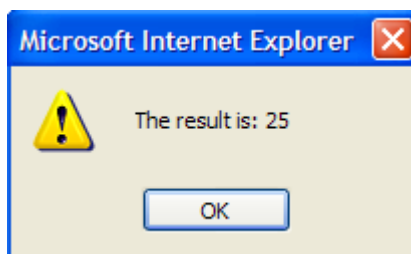
الغرض window

يوفر الغرض window ثلاثة طرق لإنشاء صناديق حوار بهدف التفاعل مع المستخدم.

• الطريقة alert

تقوم الطريقة alert بفتح نافذة حوارية وإظهار محتوى معاملها فيها.

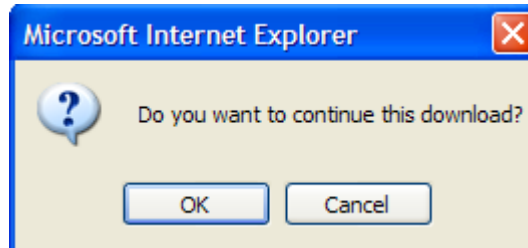
```
var a = 25;  
alert("The result is: " + a + "\n");
```



الطريقة confirm

تقوم الطريقة confirm بفتح نافذة حوارية مع زرّي OK و Cancel. تكون القيمة المرجعة true في حال نقر المستخدم على الزر OK، و false في حال نقره على الزر Cancel.

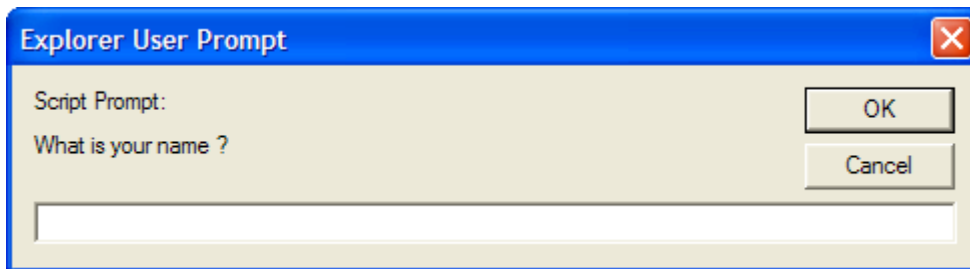
```
var question =  
confirm("Do you want to continue this download?");
```



الطريقة prompt

تقوم الطريقة prompt بإظهار نافذة حوارية تحوي صندوق نص للكتابة فيه. وتكون القيمة المرجعة هي محتوى هذا النص إذا نقر المستخدم على الزر OK، و null إذا نقر المستخدم على الزر Cancel.

```
a = prompt("What is your name ?", "");
```



يقوم المثال التالي بحل معادلة من الدرجة الثانية بعد طلب حدودها من المستخدم:

```
<!DOCTYPE html >

<!-- roots.html
  Compute the real roots of a given quadratic
  equation. If the roots are imaginary, this script
  displays NaN, because that is what results from
  taking the square root of a negative number
-->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Real roots of a quadratic equation </title>
  </head>
  <body>
    <script type = "text/javascript">
      <!--
        var question=true;
        var a, b, c, root_part, denom, root1, root2;
        while (question)
        {
          // Get the coefficients of the equation from the user
          a = prompt("What is the value of 'a'? \n", "1");
          b = prompt("What is the value of 'b'? \n", "");
          c = prompt("What is the value of 'c'? \n", "");
          // Compute the square root and denominator of the result
          root_part = Math.sqrt(b * b - 4.0 * a * c);
          denom = 2.0 * a;
          // Compute and display the two roots
          root1 = (-b + root_part) / denom;
          root2 = (-b - root_part) / denom;
          if (isNaN(root1))
          {
            alert("No real roots !");
            question=confirm("Try another equation?");
          }
          else
          {
            document.write("The first root is: ", root1, "<br />");
            document.write("The second root is: ", root2, "<br />");
            question=false;
          }
        }
      // -->
    </script>
  </body>
</html>
```

تعليمات التحكم

الأهداف التعليمية

- التعرف على تعليمات التحكم

تعليمات التحكم

- تُشابه تعليمات JavaScript تعليمات لغة C++ و Java.
- تكون التعليمات المركبة تسلسل من التعليمات المحاطة ب { }.

التعبير المنطقية

تكون نتيجة تقييم تعبير منطقي القيمة true أو القيمة false. يوجد ثلاثة أنواع من التعبيرات المنطقية:

- القيم الأساسية Primitive values
 - إذا كانت القيمة رقمية فهي تُعتبر true ما لم تكن مساوية للصفر.
 - إذا كانت القيمة نصية تُعتبر true ما لم تكن فارغة "" أو "0".
- التعبيرات العلائقية Relational Expressions
 - تستخدم العلاقات المعروفة ==, !=, >, <, >=, <=.
 - في حال كون المعاملات من أنماط مختلفة يتم إجراء التحويل الضمني.
- التعبيرات المركبة Compound Expressions
 - يُمكن إنشاء تعابير مركبة من التعبيرات السابقة باستخدام العمليات المنطقية: &&(And), ||(Or), !(Not).

تعليمات الاختيار Selection Statements

التعليمة الشرطية if

يكون للتعليمة الشرطية if في JavaScript نفس الشكل في C. كما يُبين المثال التالي:

```
if (a > b)
  document.write("a is greater than b <br />");
else {
  a = b;
  document.write(" a was not greater than b <br />",
    "Now they are equal <br /> ");}
```

التعليمة switch

تأخذ التعليمة switch الشكل:

```
switch (expression) {
  case value_1:
    // value_1 statements
  case value_2:
    // value_2 statements
  ...
  [default:
    // default statements]
}
```

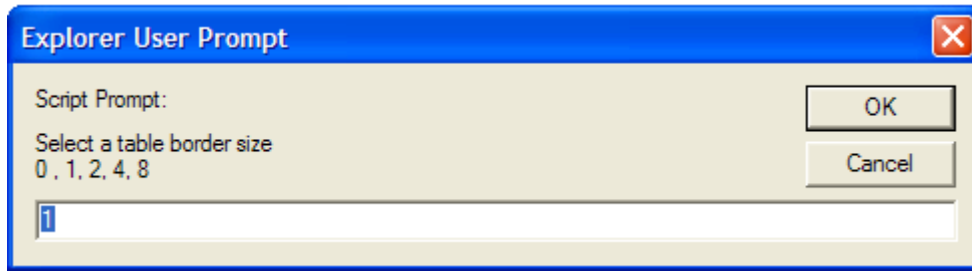
يُبين المثال التالي استخدام switch حيث يتم الطلب من المستخدم إدخال عرض الإطار المطلوب لجدول:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> A switch statement </title>
</head>
<body>
<script type = "text/javascript">
  <!--
  var bordersize;
  bordersize = prompt("Select a table border size \n" +
    "0 , 1, 2, 4, 8 \n","1" );
  switch (bordersize) {
    case "0": document.write("<table>");
      break;
    case "1": document.write("<table border = '1'>");
      break;
    case "4": document.write("<table border = '4'>");
      break;
    case "8": document.write("<table border = '8'>");
      break;
    default: document.write("Error - invalid choice: ",
      bordersize, "<br />");
  }

  document.write("<caption> 2003 NFL Divisional",
    " Winners </caption>");
  document.write("<tr>",
    "<th />",
    "<th> American Conference </th>",
    "<th> National Conference </th>",
    "</tr>",
    "<tr>",
    "<th> East </th>",
    "<td> New England Patriots </td>",
    "<td> Philadelphia Eagles </td>",
    "</tr>",
    "<tr>",
    "<th> North </th>",
    "<td> Baltimore Ravens </td>",
    "<td> Green Bay Packers </td>",
    "</tr>",
    "<tr>",
    "<th> West </th>",
    "<td> Kansas City Chiefs </td>",
    "<td> St. Louis Rams </td>",
    "</tr>",
    "<tr>",
    "<th> South </th>",
    "<td> Indianapolis Colts </td>",
    "<td> Carolina Panthers </td>",
    "</tr>",
    "</table>");

  // -->
</script>
</body>
</html>
```

يبدأ التنفيذ بالطلب من المستخدم إدخال عرض الحدود المطلوب:



ومن ثم يتم إظهار الجدول بالعرض المحدد:

2003 NFL Divisional Winners		
	American Conference	National Conference
East	New England Patriots	Philadelphia Eagles
North	Baltimore Ravens	Green Bay Packers
West	Kansas City Chiefs	St. Louis Rams
South	Indianapolis Colts	Carolina Panthers

تعليمات التكرار

توفر لغة JavaScript التعليمتين الأساسيتين للتكرار while و for. يكون للتعليمية while الشكل:

```
while ( control expression)
{
    Statements
}
```

أو الشكل:

```
do
{
    statements
}
while ( control expression)
```

يكون للتعليمية for الشكل:

```
for (initial expression; control expression; increment expression)
{
    statements
}
```


المصفوفات Arrays

الأهداف التعليمية

- المصفوفات

المصفوفات Arrays

يتم تعريف المصفوفات في JavaScript إما باستخدام التعليمة `new` أو بإسناد عناصر المصفوفة مباشرة. كما تُبين الأمثلة التالية:

```
var myList1 = new Array(24);  
var myList2 = ["bread", 99, true];
```

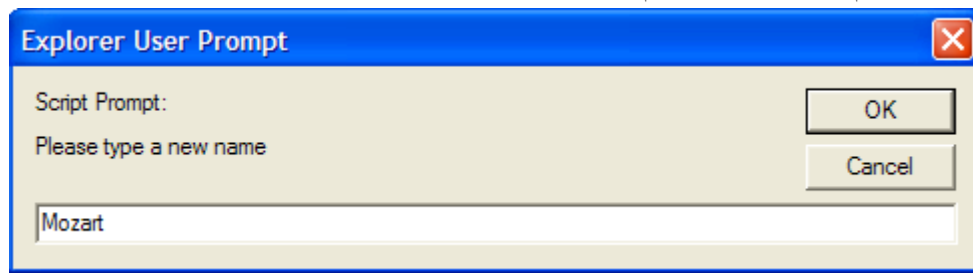
- يكون طول المصفوفة `length` هو فهرس آخر عنصر فيها زائد 1.
- يُمكن تغيير طول المصفوفة بإسناد قيمة إلى الخاصية `length`: `myList.length = 150;`

مثال:

يوضح المثال التالي استخدام المصفوفات. نصح في البداية عن مصفوفة تحوي بعض الأسماء مرتبة أبجدياً. نطلب من المستخدم إدخال اسم جديد ونقوم بإدراجه في المصفوفة وبحيث نحافظ على الترتيب الأبجدي.

```
<!DOCTYPE html >  
<html xmlns = "http://www.w3.org/1999/xhtml">  
  <head> <title> Name list </title>  
  </head>  
  <body>  
    <script type = "text/javascript">  
      <!--  
      // The original list of names  
      var name_list= new Array("Al", "Betty", "Kasper",  
                               "Michael", "Roberto", "Zimbo");  
      var new_name, index, last;  
      // Loop to get a new name and insert it  
      while (new_name = prompt("Please type a new name", "")) {  
        // Loop to find the place for the new name  
        last = name_list.length - 1;  
        while (last >= 0 && name_list[last] > new_name) {  
          name_list[last + 1] = name_list[last];  
          last--;  
        }  
        // Insert the new name into its spot in the array  
        name_list[last + 1] = new_name;  
        // Display the new array  
        document.write("<p><b>The new name list is:</b> ",  
                       "<br />");  
        for (index = 0; index < name_list.length; index++)  
          document.write(name_list[index], "<br />");  
        document.write("</p>");  
      } /** end of the outer while loop  
      // -->  
    </script>  
  </body>  
</html>
```

يبدأ البرنامج بطلب اسم جديد من المستخدم:



ثم يظهر المصفوفة الجديدة بعد إضافة الاسم الجديد في مكانه الصحيح:

```
The new name list is:  
Al  
Betty  
Kasper  
Michael  
Mozart  
Roberto  
Zimbo
```

يوجد مجموعة من الطرق للتعامل مع المصفوفات:

- concat لوصل مصفوفة مع أخرى.
- join لتشكيل سلسلة نصية من عناصر المصفوفة مع فصلها بفاصل محدد.
- reverse لعكس عناصر المصفوفة.
- slice للحصول على جزء من المصفوفة.

مثال:

يُبين المثال التالي بعض طرق التعامل مع المصفوفات:

```
<script type="text/javascript">  
a = new Array("a", "b", "c", "d");  
n = new Array(1, 2, 3);  
an = a.concat(n);  
document.write("a concat n= ", an, "<br />");  
document.write("a.join(',') = ", a.join(", "), "<br />");  
document.write("a.slice(1,3) =", a.slice(1, 3), "<br />");  
document.write("a.reverse() =", a.reverse(), "<br />");  
</script>
```

حيث يكون الخرج:

```
a concat n= a,b,c,d,1,2,3  
a.join(',') = a,b,c,d  
a.slice(1,3) =b,c  
a.reverse() =d,c,b,a
```

الوظائف Functions

الأهداف التعليمية

- الوظائف

الوظائف Functions

- يُشبه شكل الوظائف في JavaScript شكل الوظائف في لغة C.
- يتمّ التصريح عن الوظائف ضمن المؤثر <head>.
- لا يتمّ التحقق عند استدعاء الوظيفة من نمط المعاملات الممرّة ولا من عددها. حيث يتمّ تجاهل المعاملات الزائدة. أما المعاملات الناقصة فتُعتبر undefined.
- يتمّ إرسال المعاملات عبر مصفوفة arguments يُمكن الحصول على طولها من الخاصية length.

مثال:

يُبين المثال التالي استخدام المصفوفة arguments:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> Parameters </title>
  <script type = "text/javascript">
    <!--
    // Function params
    // Parameters: two named parameters and one unnamed
    // parameter, all numbers
    // Returns: nothing
    function params(a, b) {
      document.write("Function params was passed ",
        arguments.length, " parameter(s) <br />");
      document.write("Parameter values are: <br />");
      for (var arg = 0; arg < arguments.length; arg++)
        document.write(arguments[arg], "<br />");
      document.write("<br />");
    }
    // -->
  </script>
</head>
<body>
<script type = "text/javascript">
  params("Mozart");
  params("Mozart", "Beethoven");
  params("Mozart", "Beethoven", "Tchaikowsky");
</script>
</body>
</html>
```

حيث يكون الخرج:

```
Function params was passed 1 parameter(s)
Parameter values are:
Mozart

Function params was passed 2 parameter(s)
Parameter values are:
Mozart
Beethoven

Function params was passed 3 parameter(s)
Parameter values are:
Mozart
Beethoven
Tchaikowsky
```

مطابقة النماذج

الأهداف التعليمية

- مطابقة النماذج

مطابقة النماذج Pattern Matching

توفر لغة JavaScript طرق قوية لمطابقة النماذج اعتماداً على التعبيرات المنتظمة Regular Expressions.

المحارف والمحارف المترفعة

يوجد نوعين من المحارف في النموذج:

- المحارف العادية وهي المحارف التي تتطابق مع نفسها.
 - المحارف المترفعة وهي المحارف التي لها معنى خاص ولا تتطابق مع نفسها:
?. * \$ ^ { } [] () \ | (يُمكن جعل محرف مترفع يُعامل كمحرف عادي بسببه ب \).
- تُعتبر الطريقة search على الغرض String من أبسط الطرق لمطابقة النماذج. يكون النموذج معامل الدخول لهذه الطريقة، وتُعيد موقع بداية النموذج في السلسلة في حال وجوده (تُفهرس الأحرف اعتباراً من الصفر)، أو - 1 في حال عدم وجوده. كما يُبين المثال التالي:

```
var str = "Rabbits are furry";
var position = str.search(/bits/);
if (position >= 0)
    document.write("'bits' appears in position",
                    position, " <br />");
else
    document.write("'bits' does not appear in str <br />");
```

الجواب: 'bits' appears in position 3

المحارف والمحارف المترفعة

- يُطابق المحرف المترفع (.) أي محرف عدا السطر الجديد. فمثلاً، يُطابق النموذج /snow./ كل من snowy, snowe, snowd. (لمطابقة نقطة في سلسلة نصية يجب سبق النقطة في النموذج ب \، فمثلاً، يُطابق النموذج \3\4\ السلسلة النصية 3.4 ولا يطابق 374.
- عند وضع مجموعة من الأحرف ضمن []. فهذا يعني أن المطابقة يجب أن تتم مع أحد هذه الحروف. فمثلاً يُطابق النموذج /oi]n/ كل من on و in.
- يُمكن استخدام المحرف (-) لتعيين مجال من القيم. فمثلاً، /[a-h]/ تعني أي محرف بين a و h.
- يُمكن استخدام المحرف (^) لعكس المحارف المعينة. فمثلاً /[^abc]/ تعني أي محرف ما عدا الأحرف a, b, c.

المحارف والمحارف المترفعة

يوجد بعض الصفوف المعرفة مسبقاً لبعض النماذج الأكثر استخداماً:

Name	Equivalent Pattern	Matches
\d	[0-9]	a digit
\D	[^0-9]	not a digit
\w	[A-Za-z_0-9]	a word character
\W	[^A-Za-z_0-9]	not a word character
\s	[\r\t\n\f]	a whitespace character
\S	[^\r\t\n\f]	not a whitespace character

فمثلاً:

- يُطابق النموذج $/d\d\d/$ أي رقم تليه نقطة يليها رقمين.
- يُطابق النموذج $/D\dD/$ رقم واحد.

يُمكن في العديد من الحالات تحديد تكرار معين:

Quantifier	Meaning
{n}	exactly n repetitions
{m,}	at least m repetitions
{m, n}	at least m but not more than n repetitions

فمثلاً يُطابق النموذج $/xy\{4\}z/$ السلسلة $xyyyyyz$.

- يُستخدم المحرف (*) لتحديد صفر أو أي عدد من التكرارات.
- يُستخدم المحرف (+) لتحديد تكرار واحد أو أكثر.
- يُستخدم المحرف (?) لتحديد صفر أو تكرار واحد.

فمثلاً، يُطابق النموذج $/?x*y+z/$ سلسلة محرفية تبدأ بعدد من x (أو ولا x) يليها تكرار أو أكثر لـ y ، يليها z واحدة (أو ولا z).

يُطابق النموذج $/d+\d*/$ رقم أو أكثر يليه نقطة يليها عدد من الأرقام (أو ولا رقم).

يُبين المثال التالي استخدام مطابقة النماذج لاختبار شكل سلسلة نصية:

```
<!DOCTYPE html >
<!-- forms_check.html
      A function tst_phone_num is defined and tested.
      This function checks the validity of phone
      number input from a form
-->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Phone number tester </title>
  <script type = "text/javascript">
    <!--
  /* Function tst_phone_num
     Parameter: A string
     Result: Returns true if the parameter has the form of a legal
           seven-digit phone number (3 digits, a dash, 4 digits)
  */
function tst_phone_num(num) {
    var ok = num.search(/\d{3}-\d{4}/);
    if (ok == 0)
        return true;
    else
        return false;
  } // end of function tst_phone_num
  // -->
</script>
</head>
<body>
  <script type = "text/javascript">
    <!--
var tst = tst_phone_num("444-5432");
if (tst)
  document.write("444-5432 is a legal phone number <br />");
else
  document.write("Error in tst_phone_num <br />");
tst = tst_phone_num("444-r432");
if (tst)
  document.write("Error in tst_phone_num <br />");
else
  document.write("444-r432 is not a legal phone number <br />");
tst = tst_phone_num("44-1234");
if (tst)
  document.write("Error in tst_phone_num <br />");
else
  document.write("44-1234 is not a legal phone number <br />");
// -->
</script>
</body>
</html>
```

حيث يكون الخرج:

```
444-5432 is a legal phone number
444-r432 is not a legal phone number
44-1234 is not a legal phone number
```

تحديد الموقع

يُمكن استخدام المحرف (^) لتحديد أن موقع النموذج يجب أن يكون بداية السلسلة. أو المحرف (\$) لتحديد أن النموذج يجب أن يكون نهاية السلسلة. فمثلاً، يُطابق النموذج /gold\$/ السلسلة "I like gold" بينما لا يُطابق "golden". كذلك يُطابق النموذج /^pearl/ السلسلة "pearls are pretty" ولا يُطابق "My pearls are pretty".

تعديل النماذج

يُمكن استخدام المحرف (i) لطلب تجاهل حالة الأحرف. فمثلاً، يُطابق النموذج /ok/i كل من OK, Ok, ok. .oK

الطريقة replace

يُمكن استخدام الطريقة replace لاستبدال سلسلة جزئية بأخرى. كما يُمكن استخدام المحرف (g) لطلب الاستبدال لكل ظهور للسلسلة الجزئية.

مثال:

```
var str = "Some rabbits are rabid";  
str = str.replace(/rab/g, "tim");  
document.write("str is ",str ,"<br />");
```

تصبح str مساوية إلى "Some timbits are timid".

الطريقة match

تُستخدم الطريقة match لإرجاع مصفوفة من السلاسل الجزئية المطابقة للنموذج.

مثال:

```
var str = "My 3 kings beat your 2 aces";  
var matches = str.match(/[ab]/g);  
document.write("matches is ", matches ,"<br />");
```

تصبح قيمة matches مساوية إلى b,a,a.

الطريقة split

تقوم الطريقة split بتجزئة السلسلة إلى سلاسل جزئية.

مثال:

```
var str = "red,green,blue";  
var colors = str.split(",");  
document.write("colors is ", colors,"<br />");
```

تصبح قيمة colors مساوية إلى [red, green , blue].

الفصل الخامس:

التفاعل بين JavaScript وHTML

الكلمات المفتاحية

نموذج كائن الوثيقة، الأحداث.

المُلخَص

نستعرض في هذا الفصل آليات التعامل مع عناصر XHTML من خلال تعليمات JavaScript. كما نعرض لبعض آليات التحقق من القيم المدخلة من قبل المستخدم.

الأهداف التعليمية

يتعرف الطالب في هذا الفصل على:

- نموذج كائن الوثيقة.
- الأحداث.
- أحداث جسم الوثيقة.
- أحداث الأزرار.
- أحداث مربعات النص.
- التركيز.

المخطط

يضم فصل التفاعل بين JavaScript وHTML 4 وحدات (Learning Objects) هي:

- التفاعل بين JavaScript وHTML
- الأحداث Events
- التحقق من الإدخال
- مثال تطبيقي

التفاعل بين JavaScript و HTML

الأهداف التعليمية

- الوصول لعناصر HTML من خلال Java Script

التعامل مع عناصر HTML من خلال JavaScript

يُمكن الوصول لعناصر HTML من خلال تعليمات JavaScript بعدة طرق:

- نموذج كائن الوثيقة (DOM (Document Object Model
- استخدام أسماء العناصر
- استخدام الطريقة getElementById
- استخدام الطريقة getElementsByTagName

نموذج كائن الوثيقة (DOM (Document Object Model

- تُكون العناصر elements في هذا النموذج أغراض objects، وتكون واصفاتها attributes خصائص properties.
- يحوي الكائن document المصفوفة forms[] لكل النماذج المحتواة في الوثيقة والمحددة باستخدام المؤثر <form>.
- يحوي كل عنصر من المصفوفة السابقة على مصفوفة elements[] للوصول لجميع عناصر النموذج.
- بالتالي يُمكن الوصول لأي عنصر في الوثيقة باستخدام: document.forms[i].elements[j]

مثال:

يُبين المثال التالي استخدام هذه الطريقة في العنونة:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Access to form elements </title>
  </head>
  <body>
    <form action = "">
      <input type="button" name="turnItOn" />
    </form>
    <script type="text/javascript" >
      document.forms[0].elements[0].value="on";
    </script>
  </body>
</html>
```

من الواضح أن مشكلة هذه الطريقة في العنونة تكون عند إضافة أو حذف عناصر جديدة. فمثلاً، لو أضفنا زر جديد قبل الزر السابق سيُصبح العنوان السابق عنوان الزر الجديد.

استخدام أسماء العناصر

مثال:

في المثال السابق يُمكن كتابة:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Access to form elements </title>
  <body>
    <form name="myForm" action = "">
      <input type="button" name="turnItOn" />
    </form>
    <script "type=text/javascript" >
      myForm.turnItOn.value="on";
    </script>
  </body>
</html>
```

تُمكن المشكلة في هذه الطريقة أن المعيار XHTML لايسمح بإعطاء الوصفة name للنموذج (إلا أنه يسمح بإعطائها لعناصر النموذج).

استخدام الطريقة getElementById

مثال:

يُمكن الوصول إلى عنصر عن طريق معرفه (يتم إعطاء معرف فريد للعنصر باستخدام الوصفة id). ويصبح المثال السابق في هذه الحالة:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Access to form elements </title>
  <body>
    <form action = "">
      <input type="button" id="turnItOn" />
    </form>
    <script "type=text/javascript" >
      document.getElementById("turnItOn").value="on";
    </script>
  </body>
</html>
```

نضع في غالب الأحيان لعناصر النموذج اسم name ومُعَرَّف id¹ (نضع نفس القيمة عادةً).

نقوم عادةً بإعطاء نفس الاسم لمجموعة من أزرار الخيار أو مجموعة من صناديق التحقق، وعندما يكون هذا الاسم مصفوفة يُمكن من خلالها الوصول لكل عنصر في المجموعة².

1 سنرى لاحقاً أن البرمجة من جهة المخدم تتعامل مع المعرف

2 بالطبع، يُمكن إعطاء معرف لكل عنصر في المجموعة إلا أن ذلك لن يكون عملياً للوصول لعناصر المجموعة

استخدام الطريقة getElementById

مثلاً لو كان لدينا مجموعة من صناديق التحكم:

```
<div id = "VehicleGroup">
<input type="checkbox" name="vehicles" value="car" /> Car
<input type="checkbox" name="vehicles" value="truck"/> Truck
<input type="checkbox" name="vehicles" value="bike"/> Bike
</div>
```

فيمكن الوصول لعناصرها كما يلي (لنحسب مثلاً عدد الصناديق المختارة):

```
var numChecked = 0;
var dom = document.getElementById("VehicleGroup");
for (index = 0; index < dom.vehicles.length; index++)
    if (dom.vehicles[index].checked)
        numChecked++;
```

استخدام الطريقة getElementsByName

يُمكن الوصول إلى مجموعة من العناصر التي لها نفس الاسم باستخدام الطريقة getElementsByName كما يُبين المثال التالي:

```
var numChecked = 0;
var dom = document.getElementsByName("vehicles");
for (index = 0; index < dom.length; index++)
    if (dom[index].checked)
        numChecked++;
```

الأحداث Events

الأهداف التعليمية

- التعرف على الأحداث

الأحداث Events

- يُخبر الحدث بأن شيء محدد قد حصل بفعل المتصفح أو بفعل المستخدم.
 - ندعو معالج الحدث Event handler الخطاطة التي تُنفذ عند وقوع الحدث.
 - ندعو عملية ربط حدث بمعالج حدث بالتسجيل Registration.
- يُبين الجدول التالي الأحداث والواصفات الموافقة والمؤثرات التي تتحسس هذه الأحداث:

الحدث	Event	Tag Attribute	Tag
ترك العنصر	blur	onblur	<a> <button>
			<input> <textarea>
			<select>
تغيير القيمة	change	onchange	<input> <textarea> <select>
النقر	click	onclick	<a> <input>
انتقال الفارة إلى العنصر	focus	onfocus	<a> <input> <textarea> <select>
التحميل	load	onload	<body>
ضغط زر الفارة	mousedown	onmousedown	Most elements
تحريك الفارة	mousemove	onmousemove	Most elements
مغادرة الفارة العنصر	mouseout	onmouseout	Most elements
توضع الفارة فوق العنصر	mouseover	onmouseover	Most elements
تحرير زر الفارة	mouseup	onmouseup	Most elements
تحديد العنصر	select	onselect	<input> <textarea>
الإرسال	submit	onsubmit	<form>
إنهاء التحميل	unload	onunload	<body>

يتم تسجيل معالج الحدث:

- إما بإسناد التعليمات لوصفة المؤثر:

```
<input type="button" name="myButton" id="myButton"
      onclick="alert('You clicked my button!');" />
```

- أو بكتابة تعليمات الحدث في وظيفة ومن ثم استدعاء الوظيفة في واصفة المؤثر:

```
<input type="button" name="myButton" id="myButton"
      onclick="myHandler( );" />
```

- أو بإسناد اسم الوظيفة في JavaScript إلى خاصية الحدث:

```
document.getElementById("myButton").onclick = myHandler;
```

معالجة أحداث جسم الوثيقة body

يُعتبر حدثي التحميل load وإنهاء التحميل unload من أهم أحداث المؤثر <body>. يُبين المثال التالي استخدام حدث التحميل لإظهار رسالة ترحيبية:

```
<!DOCTYPE html >
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
<title> onLoad event handler </title>
    <script type = "text/javascript">
      <!--
// The onload event handler
      function load_greeting () {
        alert("You are visiting the home page of \n" +
              "SVU \n" +
              "WELCOME!!!");
      }
      // -->
    </script>
  </head>
  <body onload="load_greeting();" >
    <p />
  </body>
</html>
```

حيث يُظهر المتصفح حال فتح الوثيقة:



معالجة أحداث الزر

يُعتبر الحدث onclick الأكثر استخداماً مع زر الأمر. ويتم تسجيل الحدث عادةً:

- إما بإسناد اسم الوظيفة إلى الوصفة onclick:

```
<input type="button" name="freeButton"
      id="freeButton" onclick="freeButtonHandler();">
```

- أو باستخدام:

```
document.getElementById("freeButton").onclick = freeButtonHandler();
```

يُبين المثال التالي استخدام الحدث onclick مع مجموعة من أزرار الخيار لإظهار رسالة موافقة عندما يقوم المستخدم بالنقر على زر خيار من المجموعة:

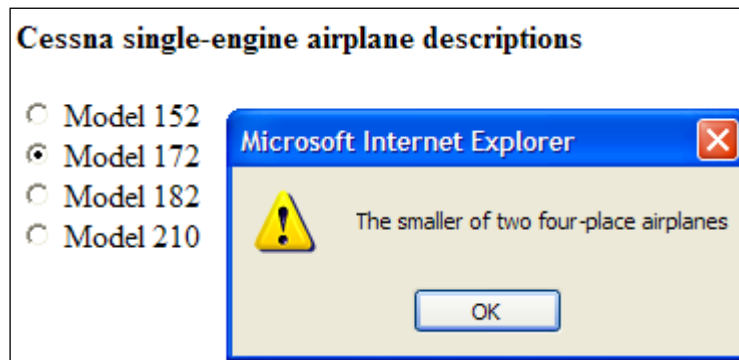
```
<!DOCTYPE html >
< An example of the use of the click event with radio buttons,
registering the event handler by assignment to the button attributes
-->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Illustrate messages for radio buttons </title>
    <script type = "text/javascript">
      <!--
      // The event handler for a radio button collection
      function planeChoice (plane) {
      // Produce an alert message about the chosen airplane
      switch (plane) {
        case 152:
          alert("A small two-place airplane for flight training");
          break;
        case 172:
          alert("The smaller of two four-place airplanes");
          break;
        case 182:
          alert("The larger of two four-place airplanes");
          break;
        case 210:
          alert("A six-place high-performance airplane");
          break;
        default:
          alert("Error in JavaScript function planeChoice");
          break;
      }
      }
      // -->
    </script>
  </head>
  <body>
    <h4> Cessna single-engine airplane descriptions </h4>
    <form id = "myForm" action = "handler">
      <p>
        <input type = "radio" name = "planeButton"
          value = "152" onclick = "planeChoice(152)" /> Model 152
        <br />
        <input type = "radio" name = "planeButton"
```

```

value = "172" onclick = "planeChoice(172)" />
Model 172 <br />
<input type = "radio" name = "planeButton"
value = "182" onclick = "planeChoice(182)" /> Model 182
<br />
<input type = "radio" name = "planeButton"
value = "210" onclick = "planeChoice(210)" /> Model 210
</p>
</form>
</body>
</html>

```

عند النقر على أحد عناصر مجموعة أزرار الخيار تظهر الرسالة الموافقة:



معالجة أحداث صندوق النص وصندوق كلمة السر

يُمكن لصندوق النص إنشاء أربعة أحداث مختلفة:

- blur •
- focus •
- change •
- select •

حدث التركيز Focus

يُبين المثال التالي استخدام الحدث onfocus وذلك لمنع المستخدم من الكتابة في صندوق نص وذلك بإخراج التركيز من صندوق النص فور حدوثه (وذلك باستخدام الطريقة blur). يسمح هذا المثال للمستخدم بإدخال الكمية المطلوبة من كل منتج، وعند النقر على الزر "Total Cost" يظهر الإجمالي في صندوق النص الموافق. لن يتمكن المستخدم من الكتابة في صندوق النص المخصص للإجمالي إذ سنقوم بإخراج التركيز عن هذا الصندوق كلما قام المستخدم بالانتقال إليه.

```

<!DOCTYPE html >
<!-- This document illustrates using the focus event
      to prevent the user from changing a text field
-->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> The focus event </title>
    <script type = "text/javascript">
      <!--
// The event handler function to compute the cost
function computeCost() {
var french = document.getElementById("french").value;
var hazlenut = document.getElementById("hazlenut").value;
var columbian = document.getElementById("columbian").value;
// Compute the cost
document.getElementById("cost").value =
totalCost = french * 3.49 + hazlenut * 3.95 +
              columbian * 4.59;
} /* end of computeCost
-->
</script>
</head>
  <body>
    <form action = "">
      <h3> Coffee Order Form </h3>
<!-- A bordered table for item orders -->
<table border = "border">
<!-- First, the column headings -->
  <tr>
    <th> Product Name </th>
    <th> Price </th>
    <th> Quantity </th>
  </tr>
<!-- Now, the table data entries -->
  <tr>
    <th> French Vanilla (1 lb.) </th>
    <td> $3.49 </td>
    <td> <input type = "text" id = "french"
              size = "2" /> </td>
  </tr>
  <tr>
    <th> Hazlenut Cream (1 lb.) </th>
    <td> $3.95 </td>
    <td> <input type = "text" id = "hazlenut"
              size = "2" /> </td>
  </tr>
  <tr>
    <th> Columbian (1 lb.) </th>
    <td> $4.59 </td>
    <td> <input type = "text" id = "columbian"
              size = "2" /></td>
  </tr>
</table>
<!-- Button for precomputation of the total cost -->
  <p>
    <input type = "button" value = "Total Cost"
            onclick = "computeCost();" />
    <input type = "text" size = "5" id = "cost"
            onfocus = "this.blur();" />
  </p>

```



```
<!-- The submit and reset buttons -->
<p>
  <input type = "submit" value = "Submit Order" />
  <input type = "reset" value = "Clear Order Form" />
</p>
</form>
</body>
</html>
```

Coffee Order Form

Product Name	Price	Quantity
French Vanilla (1 lb.)	\$3.49	<input type="text" value="2"/>
Hazlenut Cream (1 lb.)	\$3.95	<input type="text" value="1"/>
Columbian (1 lb.)	\$4.59	<input type="text" value="1"/>

التحقق من الإدخال

الأهداف التعليمية

- التحقق من القيم المدخلة من قبل المستخدم.

التحقق من الإدخال

تُعتبر عمليات التحقق من الإدخالات التي يقوم بها المستخدم من العمليات الأساسية التي يُحذ القيام بها على الزبون باستخدام JavaScript وذلك تلافياً للقيام بهذه التحققات على المخدم مما يوفر زمن معالجة على المخدم وزمن تراسل على الشبكة. يجب عادة القيام بالخطوات الثلاثة التالية:

- اكتشاف الخطأ وإظهار رسالة alert موافقة.
- وضع التركيز على العنصر منشأ الخطأ باستخدام الطريقة ().focus()
- تحديد محتوى العنصر باستخدام الطريقة ().select() (ليقوم المستخدم بتعديله مباشرة).

مثال 1:

يقوم المثال التالي بإظهار صندوق نص لإدخال كلمة السر وتأكيد كلمة السر. يقوم الزر submit باستدعاء معالج الحدث. يقوم معالج الحدث أولاً بالتأكد من أن صندوق كلمة السر غير فارغ، وفي هذه الحالة يقوم بإظهار رسالة خطأ موافقة ومن ثم يضع التركيز على الصندوق. أما إذا كان صندوق كلمة السر غير فارغ، فيقوم بالتحقق من تطابق كلمة السر وتأكيد كلمة السر، وفي حال عدم تطابقهما، يقوم بإظهار رسالة خطأ موافقة ومن ثم يقوم بوضع التركيز على كلمة السر وتحديد القيمة المدخلة.

```
<!DOCTYPE html >
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Illustrate password checking </title>
    <script type = "text/javascript">
      <!--
        function chkPasswords() {
var init=document.getElementById("initial");
var sec=document.getElementById("second");
if (init.value==""){
  alert("You did not enter a password");
  init.focus();
  return false;
}
if (init.value != sec.value) {
  alert("The two passwords are not the same");
  init.focus();
  init.select();
  return false;
} else
return true;
}
      // -->
    </script>
  </head>
  <body>
    <h3> Password Input </h3>
    <form id = "myForm" action = "">
      <p>
        Your password
        <input type = "password" id = "initial" size = "10" />
        <br /><br />
        Verify password
        <input type = "password" id = "second" size = "10" />
      </p>
    </form>
  </body>
</html>
```

```

<br /><br />
<input type = "reset" name = "reset" />
<input type = "submit" name = "submit" />
</p>
</form>
<script type = "text/javascript">
  <!--
// Set submit button onsubmit property to the event handler
document.getElementById("second").onblur = chkPasswords;
document.getElementById("myForm").onsubmit=chkPasswords;
  // -->
</script>
</body> </html>

```

Password Input

Your password

Verify password

مثال 2:

يقوم المثال التالي من التحقق من القيمة المدخلة لصندوق الاسم ومن القيمة المدخلة لصندوق الهاتف.

- تقوم الوظيفة chkName من التحقق من أن الاسم المدخل له الشكل: Last Name, First Name, Middle initial.

وذلك عن طريق التحقق من مطابقة القيمة المدخلة للنموذج Pattern:

$^[A-Z][a-z]^+, ?[A-Z][a-z]^+, ?[A-Z]\.?$$

- كذلك تقوم الوظيفة chkPhone من التحقق من مطابقة القيمة المدخلة في صندوق الهاتف مع النموذج: $d\{3}-\d\{3}-\d\{4}$.
- يتم تسجيل الوظيفة مع حدث onchange لصندوق النص الموافق. عند إدخال قيمة في صندوق لا تتطابق مع النموذج، يتم إظهار رسالة تنبيه موافقة ومن ثم وضع التركيز على الصندوق وتحديد القيمة المدخلة.

```

<!DOCTYPE html >
<!-- An example of input validation using the change and submit
events
-->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>

```

```

<title> Illustrate form input validation </title>

<script type = "text/javascript">

    <!--

// The event handler function for the name text box

    function chkName() {

        var myName = document.getElementById("custName");

// Test the format of the input name
// Allow the spaces after the commas to be optional
// Allow the period after the initial to be optional

        var pos = myName.value.search(

            /^[A-Z][a-z]+, ?[A-Z][a-z]+, ?[A-Z]\.?$/);

        if (pos != 0) {

            alert("The name you entered (" + myName.value +

                ") is not in the correct form. \n" +

                "The correct form is: " +

                "last-name, first-name, middle-initial \n" +

                "Please go back and fix your name");

            myName.focus();

            myName.select();

            return false;

        } else

            return true;

    }

// The event handler function for the phone number text box

    function chkPhone() {

        var myPhone = document.getElementById("phone");

// Test the format of the input phone number

        var pos = myPhone.value.search(/^\d{3}-\d{3}-\d{4}$/);

        if (pos != 0) {

            alert("The phone number you entered (" + myPhone.value +

                ") is not in the correct form. \n" +

                "The correct form is: ddd-ddd-dddd \n" +

                "Please go back and fix your phone number");

            myPhone.focus();

            myPhone.select();


```

```

        return false;
    } else
        return true;
    }
    // -->
</script>
</head>
<body>
    <h3> Customer Information </h3>
    <form action = "">
        <p>
            <input type = "text" id = "custName"
                onchange = "chkName();" />
            Name (last name, first name, middle initial)
            <br /><br />
            <input type = "text" id = "phone" />
            Phone number (ddd-ddd-dddd)
            <br /><br />
            <input type = "reset" id = "reset" />
<input type = "submit" id = "submit" />
        </p>
    </form>
    <script type = "text/javascript">
        <!--
// Set form element object properties to their
// corresponding event handler functions
document.getElementById("custName").onchange = chkName;
document.getElementById("phone").onchange = chkPhone;
        // -->
    </script>
</body>
</html>

```

Customer Information

Name (last name, first name, middle initial)

Phone number (ddd-ddd-dddd)

مثال تطبيقي

الأهداف التعليمية

- مراجعة مفاهيم الفصل.

تهدف الصفحة التالية إلى حساب سعر الإقامة في فندق وذلك وفق مجموعة من الخيارات التي يُمكن أن يقوم بها المستخدم:

- تصنيف الفندق: *، **، **، ****، *****
- إطلالة الفندق: البحر، الغابة، الجبل.
- الوجبات المطلوبة: فطور، غداء، عشاء.
- خدمات إضافية: تلفزيون، مسبح، مطبخ.

ولتكن مثلاً الأسعار التالية المعتمدة:

Stars		Where		Food		Extra	
Stars	Price	Sea	+50%	Breakfast	50	TV	50
*	100	Forest	+30%	Lunch	100	Pool	100
**	200	Mountain	+20%	Dinner	200	Kitchen	150
***	300						
****	400						
*****	500						

يُمكن في هذه المسألة البسيطة تثبيت الأسعار في قيم عناصر HTML للرجوع إليها في خطأ Java المستخدمة لحساب سعر الليلة:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head> <title>Hotel Club HTML</title>
<script language="javascript" type="text/javascript">
function ChangeFlag()
{
document.getElementById("imgFlag").src=
document.getElementById("SelectFlag").value
}

function getPrice()
{
var x;
x=document.getElementById("SelectStars").value;

if (document.getElementById("RadWhere1").checked)
x= x* document.getElementById("RadWhere1").value;
else if (document.getElementById("RadWhere2").checked)
x= x* document.getElementById("RadWhere2").value;
else if (document.getElementById("RadWhere3").checked)
x= x* document.getElementById("RadWhere3").value;

if (document.getElementById("chkBreak").checked)
x = x + parseInt( document.getElementById("chkBreak").value);

if (document.getElementById("chkLunch").checked)
x = x + parseInt( document.getElementById("chkLunch").value);
}
</script>
</head>
<body>
<div style="display: flex; justify-content: space-around; border: 1px solid black; padding: 5px;>
<div style="border: 1px solid black; padding: 5px; width: 20%; text-align: center;>
<table border="1" style="width: 100%; border-collapse: collapse;>
<thead>
<tr><th>Stars</th><th>Price</th></tr>
<tbody>
<tr><td>*</td><td>100</td></tr>
<tr><td>**</td><td>200</td></tr>
<tr><td>***</td><td>300</td></tr>
<tr><td>****</td><td>400</td></tr>
<tr><td>*****</td><td>500</td></tr>
</tbody>
</table>
</div>
<div style="border: 1px solid black; padding: 5px; width: 20%; text-align: center;>
<table border="1" style="width: 100%; border-collapse: collapse;>
<thead>
<tr><th colspan="2">Where</th></tr>
<tbody>
<tr><td>Sea</td><td>+50%</td></tr>
<tr><td>Forest</td><td>+30%</td></tr>
<tr><td>Mountain</td><td>+20%</td></tr>
</tbody>
</table>
</div>
<div style="border: 1px solid black; padding: 5px; width: 20%; text-align: center;>
<table border="1" style="width: 100%; border-collapse: collapse;>
<thead>
<tr><th colspan="2">Food</th></tr>
<tbody>
<tr><td>Breakfast</td><td>50</td></tr>
<tr><td>Lunch</td><td>100</td></tr>
<tr><td>Dinner</td><td>200</td></tr>
</tbody>
</table>
</div>
<div style="border: 1px solid black; padding: 5px; width: 20%; text-align: center;>
<table border="1" style="width: 100%; border-collapse: collapse;>
<thead>
<tr><th colspan="2">Extra</th></tr>
<tbody>
<tr><td>TV</td><td>50</td></tr>
<tr><td>Pool</td><td>100</td></tr>
<tr><td>Kitchen</td><td>150</td></tr>
</tbody>
</table>
</div>
</div>
</body>
</html>
```

```

if (document.getElementById("chkDinner").checked)
    x = x + parseInt( document.getElementById("chkDinner").value);

var i;
for (i=0; i<document.getElementById("SelectExtra").length; i++)
    if (document.getElementById("SelectExtra")[i].selected)
        x= x + parseInt(document.getElementById("SelectExtra")[i].value);

document.getElementById("txtOut").value=x;
}
</script>

</head>
<body bgcolor="#ffffcc">
<form id="frmHotel" name="frmHotel" action="">
<table border="1">
<caption>Hotel Club (HTML Controls/ Client)</caption>
<tr>
    <td> Country:</td>
    <td>
        <select name="SelectFlag" id="SelectFlag" onchange="ChangeFlag()" >
            <option value="images\USA.PNG">USA</option>
            <option value="images\FRANCE.PNG">France</option>
            <option value="images\ITALY.PNG">Italy</option>
            <option value="images\GERMANY.PNG">Germany</option></select></td>
    <td> </td></tr>
<tr>
    <td> Stars:</td>
    <td>
        <select id="SelectStars" name="SelectStars">
            <option value="100">*</option>
            <option value="200">**</option>
            <option value="300">***</option>
            <option value="400">****</option>
            <option value="500">*****</option>
        </select></td><td></td></tr>
<tr>
    <td>Where:</td>
    <td>
        <input id="RadWhere1" name="RadWhere" checked type="radio" value="1.5"
/> Sea<br />
        <input id="RadWhere2" type="radio" name="RadWhere" title="Forest"
value="1.3" /> Forest   <br />
        <input id="RadWhere3" type="radio" name="RadWhere" title="Mountain"
value="1.2" /> Mountain   <br />
    </td><td></td></tr>
<tr>
    <td>Meals:</td>
    <td>
        <input id="chkBreak" name="chkBreak" type="checkbox" title="Breakfast"
value="50" />Breakfast<br />
        <input id="chkLunch" name="chkLunch" type="checkbox" title="Lunch"
value="100" />Lunch<br />
        <input id="chkDinner" name="chkDinner" type="checkbox" title="Dinner"
checked="CHECKED" value="200" />Dinner</td>
    <td></td></tr>
<tr>
    <td>Extra:</td>
    <td>

```

```

<select id="SelectExtra" multiple name="SelectExtra" dir="ltr"
size="3">
<option selected="selected" value="50">TV</option>
<option value="75">Pool</option>
<option value="100">Kitchen</option>
</select><br /></td><td></td></tr>
<tr><td>
<input id="Submit1" type="button" value="How Much" onclick="getPrice()"
/></td>
<td>
<input id="txtOut" name="txtOut" type="text" />
</td><td></td></tr>
</table></form>
</body> </html>

```

Hotel Club (HTML Controls/ Client)		
Country:	France ▾	
Stars:	*** ▾	
Where:	<input checked="" type="radio"/> Sea <input type="radio"/> Forest <input type="radio"/> Mountain	
Meals:	<input type="checkbox"/> Breakfast <input type="checkbox"/> Lunch <input checked="" type="checkbox"/> Dinner	
Extra:	TV Pool Kitchen	
How Much	800	

- توجد أعلام الدول في المجلد .images.
- تكون قيمة كل خيار من قائمة البلدان مسار ملف صورة علم البلد.
- تقوم وظيفة Java المسماة ChangeFlag() بإسناد قيمة الخيار المحدد من قائمة البلاد (أي اسم ملف صورة علم البلد المحدد) إلى الخاصية src للصورة التي يظهر فيها علم البلد.
- يتم استدعاء هذه الوظيفة على الحدث onchange لقائمة البلاد.
- تقوم الوظيفة getPrice() بحساب سعر الإقامة استناداً لخيارات المستخدم. تعود هذه الوظيفة إلى قيمة كل عنصر لمعرفة سعر الخيار المحدد.

الفصل السادس:

الوثائق الديناميكية مع Java Script

الكلمات المفتاحية

توضع العناصر، تحريك العناصر، إخفاء وإظهار العناصر، تغيير الخصائص، تكديس العناصر، تحريك العناصر.

الملخص

نستعرض في هذا الفصل آليات إنشاء وثائق ديناميكية باستخدام خطاطات Java Scripts.

الأهداف التعليمية

يتعرف الطالب في هذا الفصل على:

- توضع العناصر.
- تحريك العناصر.
- إخفاء العناصر.
- إظهار العناصر.
- تغيير اللون والخط.
- تكديس العناصر.
- التعامل مع الفأرة.
- تحريك العناصر

المخطط

يضم فصل الوثائق الديناميكية مع Java Scripts 4 وحدات (Learning Objects) هي:

- الوثائق الديناميكية: توضع العناصر
- الوثائق الديناميكية: التعامل مع العناصر
- المحتوى الديناميكي
- التحكم بحركة العناصر

الوثائق الديناميكية: توضع العناصر

الأهداف التعليمية

- طرق توضع العناصر
- تحريك العناصر

مقدمة

نقول عن وثيقة إنها ديناميكية إذا كانت واصفات المؤثرات أو محتواها أو نمطها يتغير بعد فتح الوثيقة من قبل المتصفح. يُمكن أن تكون هذه التغييرات نتيجة التفاعل مع المستخدم، أو نتيجة مرور فترة من الزمن. تُعتبر JavaScript من أكثر الأدوات ملائمةً للقيام بذلك من جهة الزبون.

توضع العناصر

يُحدّد توضع العنصر في الوثيقة بالخصائص الثلاث التالية للواصفة `:style position, left, top`.

ويوجد ثلاثة حالات لتوضع العناصر:

- التوضع المطلق Absolute
- التوضع النسبي Relative
- التوضع الساكن Static

التوضع المطلق Absolute

يُستخدم التوضع المطلق لوضع عنصر في الوثيقة في مكان محدد وبغض النظر عن توضع العناصر الأخرى.

```
<p style="position: absolute; left: 50px; top: 100px;">
```

يُمكن استخدام التوضع المطلق لترتيب نص معين فوق نص آخر مما يخلق ما يُشبه العلامة المائية، وذلك بتوضيح النص الخاص في موقع معين:

```
<!DOCTYPE html
<!-- Illustrates absolute positioning of elements -->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Absolute positioning </title>
    <style type = "text/css">
/* A style for a paragraph of text */
    .regtext {font-family: Times; font-size: 14pt;
              color: rgb(102,102,102); width: 600px}
/* A style for the text to be absolutely positioned */
    .abstext {position: absolute; top: 25px; left: 50px;
              font-family: Times; font-size: 24pt;
              font-style: italic; letter-spacing: lem;
              color: rgb(255,0,150); width: 500px}
    </style>
  </head>
  <body>
    <p class = "regtext">
      Apple is the common name for any tree of the genus Malus, of
      the family Rosaceae. Apple trees grow in any of the temperate
      areas of the world. Some apple blossoms are white, but most
      have stripes or tints of rose. Some apple blossoms are bright
      red. Apples have a firm and fleshy structure that grows from
      the blossom. The colors of apples range from green to very
      dark red. The wood of apple trees is fine-grained and hard.
```

```

</p>
<p class = "abstext">
  APPLES ARE GOOD FOR YOU
</p>
</body>
</html>

```

Apple is the common name for any tree of the genus Malus, of the family Rosaceae. Apple trees grow in any of the temperate areas of the world. Some apple blossoms are white, but most have stripes or tints of rose. Some apple blossoms are bright red. Apples have a firm and fleshy structure that grows from the blossom. The colors of apples range from green to very dark red. The wood of apple trees is fine-grained and hard.

التوضيح المطلق Absolute

عندما يكون عنصر متوضع بشكل مطلق ضمن عنصر آخر، فإن إحداثياته top, left تكون بالنسبة للعنصر المحتوي له.

مثال:

في المثال التالي سوف نزيح النص العادي بـ 100, 100 عن زاوية المتصفح، والنص الخاص بـ 25, 50 عن النص العادي.

```

<!DOCTYPE html >
<!-- Illustrates nested absolute positioning of elements -->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Nested absolute positioning </title>
    <style type = "text/css">
/* A style for a paragraph of text */
    .regtext {font-family: Times; font-size: 14pt; width: 500px;
              color: rgb(102,102,102);
              position: absolute; top: 100px; left: 100px;}
/* A style for the text to be absolutely positioned */

    .abstext {position: absolute; top: 25px; left: 50px;
              font-family: Times; font-size: 24pt;
              font-style: italic; letter-spacing: 1em;
              color: rgb(255,0,150); width: 400px;}

    </style>
  </head>
  <body>
    <div class = "regtext">
Apple is the common name for any tree of the genus Malus, of
the family Rosaceae. Apple trees grow in any of the temperate
areas of the world. Some apple blossoms are white, but most
have stripes or tints of rose. Some apple blossoms are bright
red. Apples have a firm and fleshy structure that grows from
the blossom. The colors of apples range from green to very
dark red. The wood of apple trees is fine-grained and hard.
It is, therefore, good for furniture construction.
    <span class = "abstext">
      APPLES ARE GOOD FOR YOU

```

```
</span>
</div>
</body>
</html>
```

Apple is the common name for any tree of the genus Malus, of the family Rosaceae. Apple trees grow in any of the temperate areas of the world. Some apple blossoms are white, but most have stripes or tints of rose. Some apple blossoms are bright red. Apples have a firm and fleshy structure that grows from the blossom. The colors of apples range from green to very dark red. The wood of apple trees is fine-grained and hard. It is, therefore, good for furniture construction.

التوضع النسبي Relative

عندما يتم وضع `position:relative` دون تحديد للإحداثيات `top,left` يُعامل العنصر كأن لم تُسند أي قيمة للخاصية `position` (إلا أنه سيُمكن تحريكه فيما بعد). أما إذ حُددت قيم الإحداثيات `left,top` فيتمّ إزاحة العنصر بمقدار هذه الإحداثيات عن الموضع الذي كان سيأخذه فيما لو لم يتمّ تحديد هذه القيم.

مثال 1:

لو وضعنا زرّين في وثيقة (القيمة الافتراضية للخاصية `position` تكون `static`) سيتوضعان بجانب بعضهما في سطر واحد (في حال وجود مساحة كافية على نفس السطر).

```
<input type="button" value="1">
```

```
<input type="button" value="2">
```



مثال 2:

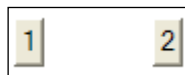
إذا قمنا بوضع الخصائص:

```
<input type="button" value="1">
```

```
<input type="button" value="2"
```

```
style="position:relative;left:50px">
```

فسيبعد الزر الثاني عن الزر الأول بمقدار 50 بكسل.



يُمكن استخدام التوضع النسبي لرفع أو خفض جزء من نص وذلك بوضع هذا الجزء من النص في span وضبط التوضع والإحداثيات.

مثال:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
  <title> Relative positioning </title>
</head>
<body style = "font-family: Times; font-size: 24pt;">
  <p>
    Apples are <span style =
      "position: relative; top: -10px;
      font-family: Times; font-size: 48pt;
      font-style: italic; color: red;">
      GOOD </span> for you.
  </p>
</body></html>
```



التوضع الساكن Static

لايُمكن إسناد قيم للإحداثيات top, left لعنصر متوضع بشكل ساكن، كما لا يُمكن تحريكه إذاً. (إذا لم يتم تحديد قيمة للخاصية position فتكون القيمة الافتراضية هي .static)

تحريك العناصر

يُمكن تحريك عنصر وذلك بتغيير إحداثياته left, top. بشرط أن يكون توضع مطلق absolute أو نسبي relative.

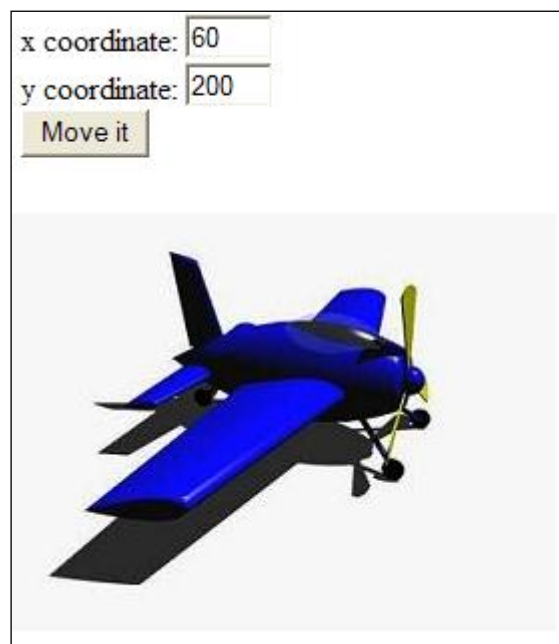
مثال:

```
<!DOCTYPE html >
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Moving elements </title>
    <script type = "text/javascript">
<!--
// *****
// The event handler function to move an element
function moveIt(movee, newTop, newLeft) {
  dom = document.getElementById(movee).style;
// Change the top and left properties to perform the move
// Note the addition of units to the input values
  dom.top = newTop + "px";
  dom.left = newLeft + "px";
  }
// *****
// -->
```

```

</script>
</head>
<body>
  <form action = "">
    <p>
      x coordinate: <input type = "text" id = "leftCoord"
                    size = "3" />
      <br />
      y coordinate: <input type = "text" id = "topCoord"
                    size = "3" />
      <br />
      <input type = "button" value = "Move it"
            onclick = "moveIt('plane',
                          document.getElementById('leftCoord').value,
                          document.getElementById('topCoord').value)"
    />
    </p>
  </form>
  <div id = "plane" style = "position: absolute;
    top: 115px; left: 0;">
    <img src = "plane.png"
        alt = "(Picture of a plane)" />
  </div>
</body>
</html>

```



الوثائق الديناميكية: التعامل مع العناصر

الأهداف التعليمية

- التعامل مع عناصر الوثائق

إظهار وإخفاء العناصر

يتم إخفاء أو إظهار عنصر عن طريق الخاصية `visibility` والتي تأخذ إحدى القيمتين: `visible` و `hidden`.

مثال:

```
<!DOCTYPE html >
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Visibility control </title>
    <script type = "text/javascript">
<!--
      function flipImag() {
        dom = document.getElementById("plane").style;
// Flip the visibility adjective to whatever it is not now

        if (dom.visibility == "visible")
        {
          dom.visibility = "hidden";
          document.getElementById("MyButton").value="Show";
        }
        else {
          dom.visibility = "visible";
          document.getElementById("MyButton").value="Hide";
        }
      }
// -->
    </script>
  </head>
  <body>
    <form action = "">
      <div id = "plane" style = "position: relative;
        visibility: visible;">
        <img src = "plane.png"
          alt = "(Picture of plane)" />
      </div>
      <p>
        <br />
        <input type = "button" id="MyButton" value = "Hide"
          onclick = "flipImag()" />
      </p>
    </form>
  </body>
</html>
```



تغيير اللون والخط

يُمكن تغيير لون الخلفية `backgroundcolor` أو اللون `color` باستخدام الخصائص الموافقة.

مثال:

يُمكن للمستخدم في المثال التالي ادخال اسم اللون المطلوب لخلفية ولون جسم الوثيقة:

```
<!DOCTYPE html >
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Dynamic colors </title>
    <script type = "text/javascript">
<!--
function setColor(where, newColor) {
  if (where == "background")
    document.body.style.backgroundColor = newColor;
  else
    document.body.style.color = newColor;
  }
-->
    </script>
  </head>
  <body>
<p style = "font-family: Times; font-style: italic; ">

<font size="4">This small page illustrates dynamic setting of the
  foreground and background colors for a document
  </font>
  </p>
  <form action = "">
    <p>
Background color: <input type = "text" name = "background"
                      size = "10"
                      onchange = "setColor('background',
                                           this.value)" />

      <br />
Foreground color: <input type = "text" name = "foreground"
                       size = "10"
                       onchange = "setColor('foreground',
                                             this.value)" />

      <br />
    </p>
  </form>
  </body>
</html>
```

This small page illustrates dynamic setting of the foreground and background colors for a document

Background color:

Foreground color:

يُمكن تغيير الخط عن طريق الخاصية `font`.

مثال:

يتم في المثال التالي تغيير لون ونوع الخط للكلمة Washington عند تحريك الفأرة فوق الكلمة `onmouseover`، وتعود الكلمة إلى تنسيقها الأول عند ابتعاد الفأرة عنها `onmouseout`.

```
<!DOCTYPE html >
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Dynamic fonts for links </title>
    <style type = "text/css">
      .regText {font: Times; font-size: 16pt;}
    </style>
  </head>
  <body>
    <p class = "regText">
      The state of
      <a style = "color: blue;"
        onmouseover = "this.style.color = 'red';
                       this.style.font = 'italic 16pt Times';"
        onmouseout = "this.style.color = 'blue';
                     this.style.font = 'normal 16pt Times';">
        Washington
      </a>
      produces many of our nation's apples.
    </p>
  </body>
</html>
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Dynamic fonts for links </title>
    <style type = "text/css">
      .regText {font: Times; font-size: 16pt;}
    </style>
  </head>
  <body>
    <p class = "regText">
      The state of
      <a style = "color: blue;"
        onmouseover = "this.style.color = 'red';
                       this.style.font = 'italic 16pt Times';"
        onmouseout = "this.style.color = 'blue';
                     this.style.font = 'normal 16pt Times';">
        Washington
      </a>
      produces many of our nation's apples.
    </p>
  </body>
</html>
```

The state of **Washington**
produces many of our nation's
apples.

The state of *Washington*
produces many of our nation's
apples.

المحتوى الديناميكي

الأهداف التعليمية

- استخدام خطاطات Java Script للحصول على محتوى ديناميكي

المحتوى الديناميكي

يُمكن تغيير محتوى أي عنصر بإسناد قيمة إلى الخاصية `value`. يستخدم هذه الخاصية المثال التالي لإظهار

نص مساعد للمستخدم عند مرور الفأرة فوق كل صندوق إدخال:

تُخزن قيم نصوص المساعدة في مصفوفة `helpers` تتألف من خمسة عناصر، وتقوم الوظيفة `messages(adviceNumber)` بوضع العنصر ذو الفهرس `adviceNumber` في صندوق المساعدة. يتم استدعاء هذه الوظيفة مع تمرير الفهرس الموافق عند مرور الفأرة `onmouseover` فوق كل عنصر.

```
<!DOCTYPE html >
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Dynamic values </title>
    <script type = "text/javascript">
<!--
var helpers = ["Your name must be in the form: \n \
  first name, middle initial., last name",
  "Your email address must have the form: \
  user@domain",
  "Your user ID must have at least six characters",
  "Your password must have at least six \
  characters and it must include one digit",
  "This box provides advice on filling out\
  the form on this page. Put the mouse cursor over any \
  input field to get advice"]

function messages(adviceNumber) {
  document.getElementById("adviceBox").value =
    helpers[adviceNumber];}
// -->
</script>
</head>
<body>
  <form action = "">
    <p style = "font-weight: bold">
      <span style = "font-style: italic">
        Customer information
      </span>
      <br /><br />
      Name: <input type = "text"
        onmouseover = "messages(0)"
        onmouseout = "messages(4)" />

      <br />
      Email: <input type = "text"
        onmouseover = "messages(1)"
        onmouseout = "messages(4)" />

      <br /> <br />
      <span style = "font-style: italic">
        To create an account, provide the following:
      </span>
      <br /> <br />
      User ID: <input type = "text"
        onmouseover = "messages(2)"
        onmouseout = "messages(4)" />

      <br />
      Password: <input type = "password"
```



```

        onmouseover = "messages(3)"
        onmouseout = "messages(4)" />
    <br />
    <textarea id = "adviceBox" rows = "8" cols = "32"
        name="adviceBox" style="position: absolute;
            left: 220px; top: 5px">

    </textarea>
    <input type = "submit" value = "Submit" />
    <input type = "reset" value = "Reset" />
</p>
</form>
</body>
</html>

```

تكديس العناصر

تُحدد الخاصية z-index لمجموعة من العناصر المتراكبة من الأمام في الأمام¹. يُمكن تغيير هذه الخاصية² في JavaScript. يُمكن تغيير هذه الخاصية ديناميكياً لتغيير تراكب العناصر. عند النقر على أحد الصور الثلاث في المثال التالي، يتم استدعاء الوظيفة toTop التي تعطي القيمة 10 للخاصية للصورة التي تمّ النقر عليها.

```

<!DOCTYPE html >
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Dynamic stacking of images </title>
    <script type = "text/javascript">
<!--
    var top = "p1anel1";
    // The event handler function to move the given element
    // to the top of the display stack

    function toTop(newTop) {
    // Set the two dom addresses, one for the old top
    // element and one for the new top element

```

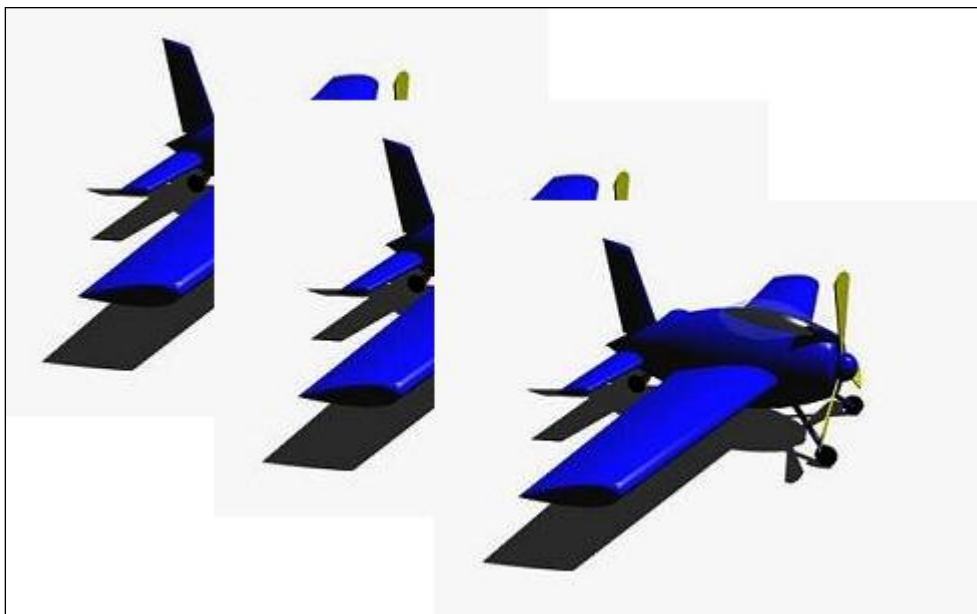
1 صاحب القيمة الأكبر

2 تُدعى zIndex

```

        domTop = document.getElementById(top).style;
        domNew = document.getElementById(newTop).style;
// Set the zIndex properties of the two elements, and
// reset top to the new top
        domTop.zIndex = "0";
        domNew.zIndex = "10";
        top = newTop;
    }
// -->
</script>
<style type = "text/css">
    .plane1 {position: absolute; top: 0; left: 0;
            z-index: 0;}
    .plane2 {position: absolute; top: 50px; left: 110px;
            z-index: 0;}
    .plane3 {position: absolute; top: 100px; left: 220px;
            z-index: 0;}
</style>
</head>
<body>
    <p>
        <img class = "plane1" id = "plane1"
            src = "plane.png"
            alt = "(Picture 1)"
            onclick = "toTop('plane1')" />
        <img class = "plane2" id = "plane2"
            src = "plane.png"
            alt = "(Picture 2)"
            onclick = "toTop('plane2')" />
        <img class = "plane3" id = "plane3"
            src = "plane.png"
            alt = "(Picture 3)"
            onclick = "toTop('plane3')" />
    </p>
</body></html>

```



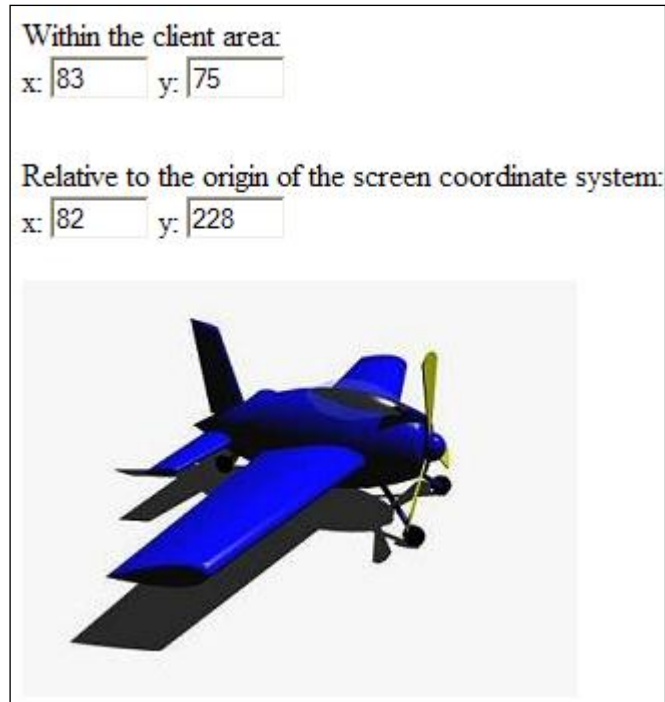
تحديد إحداثيات مؤشر الفأرة

- تُعطي الخاصيتين clientX و clientY للغرض event معامل دخل الحدث إحداثيات العنصر الذي استدعى الحدث بالنسبة للزاوية اليسرى العليا لنافذة العرض للمتصفح.
- تُعطي الخاصيتين screenX و screenY إحداثيات العنصر بالنسبة للزاوية اليسرى العليا لنافذة المتصفح.

مثال:

عند النقر في المثال التالي في جسم الوثيقة يتم إظهار القيم الأربع السابقة.

```
<!DOCTYPE html >
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Where is the cursor? </title>
    <script type = "text/javascript">
<!--
// The event handler function to get and display the
// coordinates of the cursor, both in an element and
// on the screen
function findIt(evt) {
  document.getElementById("xcoor1").value = evt.clientX;
  document.getElementById("ycoor1").value = evt.clientY;
  document.getElementById("xcoor2").value = evt.screenX;
  document.getElementById("ycoor2").value = evt.screenY;
}
// -->
</script>
</head>
<body onclick = "findIt(event)">
  <form action = "">
    <p>
      Within the client area: <br />
      x:
      <input type = "text" id = "xcoor1" size = "4" />
      y:
      <input type = "text" id = "ycoor1" size = "4" />
      <br /><br />
      Relative to the origin of the screen coordinate system:
      <br />
      x:
      <input type = "text" id = "xcoor2" size = "4" />
      y:
      <input type = "text" id = "ycoor2" size = "4" />
    </p>
  </form>
  <p>
    <img src = "plane.png" alt = "(Picture of C172)" />
  </p>
</body> </ html>
```



التفاعل مع نقر الفأرة

مثال:

يُبين المثال التالي استخدام حدثي `onmousedown` و `onmouseup` للفأرة. حيث يتم إظهار نص معين عند إحدائيات الفأرة.

```

<!DOCTYPE html >
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Sense events anywhere </title>
    <script type = "text/javascript">
<!--
// The event handler function to display the message

function displayIt(evt) {
  var dom = document.getElementById("message");
  dom.style.left = (evt.clientX - 130) + "px";
  dom.style.top = (evt.clientY - 25) + "px";
  dom.style.visibility = "visible";
}
// The event handler function to hide the message

function hideIt() {
  document.getElementById("message").style.visibility =
    "hidden";
}
// -->
</script>
</head>
<body onmousedown="displayIt(event);"
      onmouseup = "hideIt();" >
  <p>
    <span id= "message"

```

```
        style = "color: red; visibility: hidden;
                position: relative;
                font-size: 20pt; font-style: italic;
                font-weight: bold; ">
    Please don't click here!
</span>
<br /><br /><br /><br /><br /><br /><br /><br /><br />
<br /><br /><br /><br /><br /><br /><br /><br /><br />
</p>
</body>
</html>
```

Please don't click here!

التحكم بحركة العناصر

الأهداف التعليمية

- تحريك العناصر

التحكم بحركة العناصر

الوظيفة `setTimeout`

- تُستخدم لتحديد فترة زمنية قبل تنفيذ الوظيفة المحددة في المعامل الأول.
- يُحدّد المعامل الثاني الفترة الزمنية بالميلي ثانية.

```
setTimeout("mover()", 20);
```

- يتم تنفيذ الوظيفة `mover()` بعد 20 ميلي ثانية.

مثال:

يقوم المثال التالي بتحريك نص بشكل قطري وذلك عن طريق تغيير إحداثيات النص كل فترة زمنية (10 ميلي ثانية). حيث يتضمن المثال ملف `javascript`

```
<!DOCTYPE html >
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
  <title> Moving text </title>
  <script type = "text/javascript" src= "moveTextfuns.js" >
  </script>
</head>
<!-- Call the initializing function on load, giving the
      destination coordinates for the text to be moved
-->
<body onload="initText()">
<!-- The text to be moved, including its initial position -->
<p>
  <span id = 'theText' style =
    "position: absolute; left: 100px; top: 100px;
    font: bold 20pt 'Times Roman';
    color: blue;"> Jump in the lake!
  </span>
</p>
</body>
</html>
```

وحيث محتوى الملف `moveTextfuns.js`

```
var dom, x, y, finalx = 300, finaly = 300;

// ***** //
// A function to initialize the x and y coordinates
// of the current position of the text to be moved,
// and then call the mover function

function initText() {
  dom = document.getElementById('theText').style;

  /* Get the current position of the text */

  var x = dom.left;
  var y = dom.top;

  /* Convert the string values of left and top to
  numbers by stripping off the units */
```

```

    x = x.match(/\d+/);
    y = y.match(/\d+/);

    /* Call the function that moves it */
    moveText(x, y);
} /*** end of function initText */

// A function to move the text from its original
// position to (finalx, finaly)
function moveText(x, y) {
    /* If the x coordinates are not equal, move
    x toward finalx */
    if (x != finalx)
        if (x > finalx) x--;
        else if (x < finalx) x++;
    /* If the y coordinates are not equal, move
    y toward finaly */
    if (y != finaly)
        if (y > finaly) y--;
        else if (y < finaly) y++;
    /* As long as the text is not at the destination,
    call the mover with the current position */
    if ((x != finalx) || (y != finaly)) {
    /* Put the units back on the coordinates before
    assigning them to the properties to cause the
    move */
        dom.left = x + "px";
        dom.top = y + "px";
    /* Recursive call, after a 1-millisecond delay */
        setTimeout("moveText(" + x + "," + y + ")", 10);
    }
} /*** end of function moveText */

```

Jump in the lake!

1. قم بإنشاء استمارة التسجيل التالية:

Registration Form

First Name:

Middle Initial:

Last Name:

City:

State/Province:

Country: United States
 Canada
 Other

Zip:

Email:

Username:

Password:

Repeat Password:

Comments:

I agree to the site terms.

Submit

Reset Form

واكتب خطاطات جافا للتحقق مما يلي:

- الحقول واجبة الإدخال: الاسم الأول First Name، الحرف البادئ Middle Initial، الاسم الأخير Last Name، اسم الدخول Username، المدينة City، البلد Country، كلمة السر Password.
- الحرف البادئ حرف واحد فقط Middle Initial.
- يكون الرمز Zip محصور بين 5 و 10 محارف.
- يجب أن يحقق البريد الإلكتروني Email الشروط التالية:
 - يحوي 6 محارف على الأقل
 - يحوي المحرف @.
 - يحوي المحرف . وذلك بعد المحرف @.
- يجب أن يكون تأكيد كلمة السر Repeat Password مطابق لكلمة السر Password.
- في حال إدخال أكثر من 100 محرف في صندوق الملاحظات Comments سيتم حذف كل المحارف الزائدة وإعلام المستخدم بالنص المقبول.

- يجب تحديد صندوق التحقق I agree.

2. المطلوب بناء نموذج يقوم المستخدم من خلاله بإدخال رقم شهر للحصول على اسم الشهر:

Month Check

Month Number:
Month Name:

Month Check

Month Number:
Month Name:

3. المطلوب إظهار التاريخ والوقت الحالي على صفحة كما يلي:

Today is : Monday.
4:14:45

4. المطلوب صفحة يستطيع المستخدم من خلالها التحويل من درجات مئوية إلى فهرنهايت وبالعكس.

يجب أن يتم التحويل حال قيام المستخدم بالكتابة في أي صندوق نص موافق.

degrees Celsius
equals
 degrees Fahrenheit

5. المطلوب كتابة وظيفة تقوم بتحويل الحرف الأول من كل كلمة إلى حرف كبير:

6. عربة تسوق بسيطة:

المطلوب صفحة أولى تطلب اسم المستخدم وكلمة السر:

Log In Page

User Name	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Log In"/>	

في حال كون القيم المدخلة غير صحيحة تظهر رسالة إعلام موافقة.
في حال كون القيم المدخلة صحيحة (admin, 1234) يتم الانتقال إلى صفحة عربة التسوق التالية:

	Product name	Price	
1	Plasma	100,000	Add to cart
2	PC Dell	70,000	Add to cart
3	HP Printer	30,000	Add to cart
4	Sony Camera	25,000	Add to cart
<input type="button" value="Show Cart"/> <input type="button" value="Delete Cart"/>			

في كل مرة يقوم المستخدم بتحديد منتج والنقر على الرابط Add to cart يتم تلوينه بالأحمر.

	Product name	Price	
1	Plasma	100,000	Add to cart
2	PC Dell	70,000	Delete from cart
3	HP Printer	30,000	Add to cart
4	Sony Camera	25,000	Delete from cart
<input type="button" value="Show Cart"/> <input type="button" value="Delete Cart"/>			

عند النقر على الزر Show Cart تظهر جميع المنتجات المختارة في صفحة:

	Product name	Price	
1	Plasma	100,000	Add to cart
2	PC Dell	70,000	Delete from cart
3	HP Printer	30,000	Add to cart
4	Sony Camera	25,000	Delete from cart

Show Cart

D:\Java\Products.html - Internet Explorer

The Products in your cart is:

1- PC Dell
2- Sony Camera

The total cost of the cart is: 95000

عند النقر على الزر Delete Cart يتم إلغاء تلوين جميع العناصر وحذفها من عربة التسوق.

الفصل السابع:

أساسيات JQuery

الكلمات المفتاحية

تنزيل المكتبة، استدعاء وظائف المكتبة، آليات تحديد العناصر.

المُلخَص

نستعرض في هذا الفصل أساسيات التعامل مع المكتبة JQuery. نُبين أولاً كيفية استدعاء الوظائف التي توفرها المكتبة، ثم نعرض لآليات تحديد العناصر للتعامل معها.

الأهداف التعليمية

يتعرف الطالب في هذا الفصل على:

- كيفية استدعاء وظائف المكتبة JQuery.
- الآليات المختلفة لتحديد العناصر.

المخطط

يضم فصل أساسيات JQuery وحدتين (Learning Objects) هي:

- أساسيات JQuery
- محددات العناصر Selectors

أساسيات JQuery

الأهداف التعليمية

- أساسيات JQuery.

ماهي JQuery

- هي مكتبة تحوي مجموعة من الوظائف المكتوبة بلغة JavaScript. قام ببنائها John Resig عام 2006 بهدف: الكتابة أقل، الفعل أكثر Write less, do more.
- تُسهّل هذه المكتبة التعامل مع مختلف مكونات صفحة HTML. من أهم الميزات التي تقدمها هذه المكتبة:

■ التعامل مع العناصر DOM manipulation

تُسهّل الوصول إلى مختلف العناصر وتعديل المحتوى باستخدام محرك تحديد مفتوح المصدر يُدعى Sizzle.

■ معالجة الأحداث Event handling

تُقدّم طرق أنيقة لإلتقاط أحداث المستخدم مثل النقر على رابط مثلاً، دون الحاجة لخلط كود HTML نفسه مع معالجات الأحداث.

■ دعم Ajax support

تُساعد في تطوير صفحات غنية وتفاعلية باستخدام تكنولوجيا Ajax.

■ الإحياء Animations

توفر مجموعة من إمكانيات الإحياء الجاهزة.

■ الخفة Lightweight

لايتجاوز حجم المكتبة 200K.

■ دعم المتصفحات Cross Browser Support

تعمل على جميع المتصفحات المشهورة.

■ أحدث التكنولوجيا Latest Technology

تدعم التكنولوجيا الحديثة مثل CSS3 و XPath.

تنزيل المكتبة

يُمكن تنزيل إصدار المكتبة الأخير مجاناً من الرابط jquery.com.
(jquery-1.10.2.min.js).

استخدام مكتبة JQuery

يُمكنك تضمين المكتبة في صفحة HTML كما يلي:

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>The jQuery Example</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript">
    // you can add our javascript code here
  </script>
</head>
<body>
  .....
</body>
</html>
```

استدعاء وظائف JQuery

بما أن معظم ما نفعله عند استخدام JQuery هو التعامل مع نموذج كائن الوثيقة (DOM)، فإننا بحاجة للتأكد من أننا نُضيف الأحداث عند جهوزية هذا النموذج. إذا أردت أن يعمل حدث في الصفحة، فيجب استدعائه داخل `$(document).ready()`. كل شيء داخلها سوف يُحمّل حال تحميل عناصر الوثيقة. نقوم باستخدام حدث جهوزية الوثيقة كما يلي:

```
$(document).ready(function () {
  // do stuff when DOM is ready
});
```

مثال:

في المثال التالي نقوم بتسجيل حدث النقر على أي مقطع div في الصفحة. سيؤدي النقر إلى ظهور رسالة ترحيب.

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>The jQuery Example</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function () {
      $("div").click(function () {
        alert("Hello world!");
      });
    });
  </script>
</head>
```



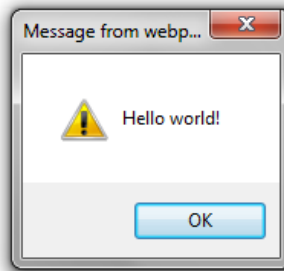
```
<body>
  <div >
    Click on this to see a dialogue box.
  </div>

  <div >
    Click also on this to see a dialogue box.
  </div>

</body>
</html>
```

يُظهر المتصفح عند النقر الرسالة الموافقة:

Click on this to see a dialogue box.
Click also on this to see a dialogue box.



استخدام خطاطات مخصصة

من الأفضل كتابة تعليماتك المخصصة في ملف جافا. مثلاً:

```
/* Filename: custom.js */
$(document).ready(function() {
  $("div").click(function() {
    alert("Hello world!");
  });
});
```

ومن ثم تضمين الملف في الصفحة:

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>The jQuery Example</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript"
    src="jquery/custom.js"></script>
</head>
<body>
  <div>
    Click on this to see a dialogue box.
  </div>

  <div>
    Click also on this to see a dialogue box.
  </div>

</body>
</html>
```

محددات العناصر Selectors

الأهداف التعليمية

- التعرف على مختلف طرق تحديد العناصر.

المحددات Selectors

- توفر مكتبة JQuery إمكانية الوصول للعناصر في نموذج كائن الوثيقة (DOM) بسرعة وسهولة.
- يبدأ المحدد ب (\$) ويمكن أن يستخدم:
 - اسم المؤثر Tag Name: مثلاً \$('p') يُحدّد جميع الفقرات في المستند.
 - معرف مؤثر Tag ID: مثلاً \$('#some-id') يُحدّد العنصر الوحيد الذي له المعرف some-id.
 - صف مؤثر Tag Class: مثلاً \$('.some-class') يُحدّد جميع المؤثرات من الصف some-class.

أمثلة:

\$('*')	تحديد جميع عناصر الوثيقة
\$('p')	تحديد جميع الفقرات
\$('div, p')	تحديد العناصر من النوع p أو div
\$("p > *")	تحديد جميع العناصر أبناء فقرة
\$("#specialID")	تحديد عنصر له المعرف المحدد specialID
\$('div#yourid')	تحديد عنصر من النوع div له المعرف yourid
\$(".specialClass")	تحديد العناصر من الصف المحدد specialClass
\$("li:not(.myclass)")	تحديد جميع العناصر من النوع li والتي ليست من الصف myclass
\$("a#specialID.specialClass")	تحديد الروابط لها المعرف specialID والصف specialClass
\$("p a.specialClass")	تحديد الروابط من الصف specialClass ضمن فقرة
\$("ul li:first")	تحديد العنصر الأول فقط من قائمة ul
\$("#container p")	تحديد الفقرات الموجودة تحت عنصر له المعرف container
\$("li > ul")	تحديد جميع العناصر ul الموجودة تحت li
\$("li:eq(2)")	تحديد العنصر الثالث في قائمة li
\$("li:lt(2)")	تحديد العناصر قبل العنصر الثالث في قائمة li
\$("li:gt(1)")	تحديد العناصر بعد العنصر الثاني في قائمة li
\$("li:even")	تحديد العناصر ذات الفهرس الزوجي في قائمة li
\$("li:odd")	تحديد العناصر ذات الفهرس الفردي في قائمة li
\$("li:visible")	تحديد العناصر الظاهرة li
\$("li:hidden")	تحديد العناصر المخفية li

\$("#:radio")	تحديد جميع أزرار الراديو في نموذج
\$("#:checked")	تحديد جميع صناديق التحقق التي لها القيمة checked
\$("#:text")	تحديد صناديق النص
\$('#div, p')	تحديد جميع العناصر من النوع div أو من النوع p
\$('#p.small')	تحديد الفقرات من الصف small
\$("#a[@href*=domain.com]")	تحديد جميع الروابط التي تحوي domain.com
\$("#input[@name^=myname]")	تحديد العناصر من النوع input والتي اسمها يبدأ بـ myname
\$("#input[@name=myname]")	تحديد العناصر من النوع input والتي اسمها يساوي myname

مثال: استخدام محدد المؤثر

عند تنفيذ المثال التالي، ستظهر 3 رسائل على التوالي تُظهر محتوى الفقرات.

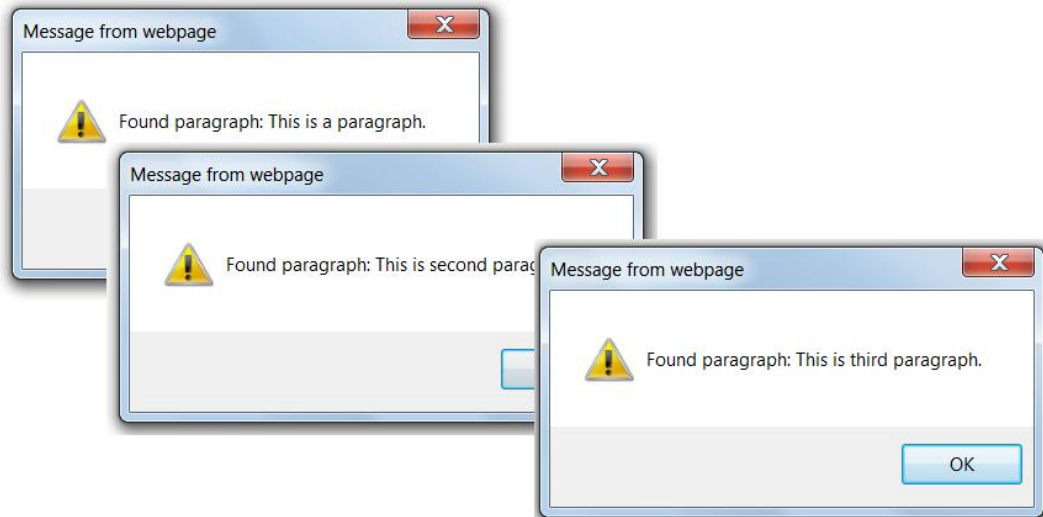
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>The jQuery Example</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function () {
      var pars = $("p");
      for (i = 0; i < pars.length; i++) {
        alert("Found paragraph: " + pars[i].innerHTML);
      }
    });
  </script>
</head>
<body>
  <div>
    <p class="myclass">This is a paragraph.</p>
    <p id="myid">This is second paragraph.</p>
    <p>This is third paragraph.</p>
  </div>
</body> </html>
```

حيث يُظهر المتصفح:

This is a paragraph.

This is second paragraph.

This is third paragraph.



مثال: استخدام محدد الصف

نقوم في المثال التالي بتعديل جميع العناصر من الصف small لتصبح من الصف big.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>The jQuery Example</title>
  <style type="text/css">
    .small {
      font-size:small;
      color:green ;
    }
    .medium {
      font-size: medium;
      color: blue;
    }
    .big {
      font-size: larger;
      color: red;
    }
  </style>

  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function () {
      $('small').toggleClass('big');
    })
  </script>
</head>
<body>

  <div class="small" id="divid1">
```

```

    <p class="para1" id="pid1">This is first paragraph.</p>
    <p class="para2" id="pid2">This is second paragraph.</p>
    <p class="para3" id="pid3">This is third paragraph.</p>
</div>
<br />

<div class="medium" id="divid2">
    <p>This is second division of the DOM.</p>
    <p>This is second para inside second division.</p>
</div>
<br />

<div class="big" id="divid3">
    <p>This is a para inside third division</p>
</div>

</body></html>

```

ويُظهر المتصفح:

This is first paragraph.

This is second paragraph.

This is third paragraph.

This is second division of the DOM.

This is second para inside second division.

This is a para inside third division

مثال: تحديد جميع العناصر *

نقوم في المثال التالي بإضافة الصف big إلى جميع عناصر الوثيقة:

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>The jQuery Example</title>

    <script type="text/javascript"
        src="jquery/jquery-1.10.2.min.js"></script>
    <script type="text/javascript" language="javascript">
        $(document).ready(function () {
            $("*").addClass("big");
        })
    </script>
<style type="text/css">
    .big {

```

```
        font-size: larger;
        color: red;
    }
</style>
</head>
<body>
    <p >This is first paragraph.</p>
    <p >This is second paragraph.</p>
    <p >This is third paragraph.</p>
</body></html>
```

حيث يُظهر المتصفح:

This is first paragraph.

This is second paragraph.

This is third paragraph.

الفصل الثامن:

الواصفات والأساليب في JQuery

الكلمات المفتاحية

التعامل مع الواصفات، التعامل مع الأساليب.

المُلخَص

نستعرض في هذا الفصل التعامل مع الواصفات والأساليب في JQuery.

الأهداف التعليمية

يتعرف الطالب في هذا الفصل على:

- أشكال التعامل مع الواصفات.
- أشكال التعامل مع الأسلوب.

المخطط

يضم فصل الواصفات والأساليب في JQuery وحدتين (Learning Objects) هي:

- التعامل مع الواصفات Attributes
- أسلوب الصفحات المتتالي CSS

Attributes التعامل مع الواصفات

الأهداف التعليمية

- التعرف على طرق التعامل مع الواصفات.

التعامل مع الواصفات

تُمكن مكتبة JQuery من التعامل مع الواصفات بسهولة.

لنتذكر أولاً أنه في المثال البسيط التالي:

```

```

يكون اسم المؤثر `img` وتكون الواصفات هي `id`, `src`, `alt`, `class`. كل واصفة هي زوج مفتاح/قيمة.

قراءة/إسناد قيمة واصفة

تسمح الطريقة `attr()` بإيجاد قيمة واصفة العنصر الأول في مجموعة من العناصر المحددة. أو بإسناد قيمة إلى جميع العناصر المحددة.

مثال:

نقوم في المثال التالي بإيجاد قيمة المؤثر `em` وإسناد هذه القيمة إلى نص `div` له المعرف `divid`.

```
<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">

  $(document).ready(function() {
    var title = $("em").attr("title");
    $("#divid").text(title);
  });

  </script>
</head>
<body>
  <div>
    <em title="Bold and Brave">This is first paragraph.</em>
    <p id="myid">This is second paragraph.</p>
    <div id="divid"></div>
  </div>
</body>
</html>
```

يُظهر المتصفح:

This is first paragraph.

This is second paragraph.

Bold and Brave

تطبيق الأنماط

يُمكن استخدام الطريقة (addClass(classes) لتطبيق نمط معرف على العناصر المحددة.

مثال:

```
<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
  src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
  $(document).ready(function() {
    $("em").addClass("selected");
    $("#myid").addClass("highlight");
  });
  </script>
  <style>
    .selected { color:red; }
    .highlight { background:yellow; }
  </style>
</head>
<body>
  <em title="Bold and Brave">This is first paragraph.</em>
  <p id="myid">This is second paragraph.</p>
</body>
</html>
```

يُظهر المتصفح

This is first paragraph.

This is second paragraph.

بعض طرق الواصفات

يُبين الجدول التالي بعض الطرق المفيدة التي تسمح بالتعامل مع الواصفات.

الطريقة	الوصف
<code>attr(properties)</code>	تسمح بإسناد زوج مفتاح/قيمة للواصفات للعناصر المحددة.
<code>attr(key, fn)</code>	تسمح بإسناد قيمة لوصفة.
<code>removeAttr(name)</code>	حذف واصفة معينة من كل العناصر المحددة.
<code>hasClass(class)</code>	تُعيد true إذا كان لأحد العناصر المحددة الصف class.
<code>removeClass(class)</code>	حذف الصف من العناصر المحددة.
<code>toggleClass(class)</code>	إضافة الصف إذا كان غير موجود وحذفه إن كان موجوداً.
<code>html()</code>	إعادة المحتوى html لأول عنصر محدد.
<code>html(val)</code>	إسناد المحتوى html لجميع العناصر المحددة.
<code>text()</code>	إعادة نص جميع العناصر المحددة.
<code>text(val)</code>	إسناد نص إلى جميع العناصر المحددة.
<code>val()</code>	إعادة قيمة أول عنصر محدد.
<code>val(val)</code>	إسناد قيمة الوصفة value لجميع العناصر المحددة إذا كانوا من النوع input. أما مع عنصر select فيتمّ تحديد الخيار option الموافق للقيمة. ومع صندوق تحقق أو زر راديو، فسيتمّ تحديد العنصر الموافق للقيمة.

<code>\$("#myID").attr("custom")</code>	إعادة قيمة الوصفة custom لأول عنصر له المعرف myID.
<code>\$("#img").attr("alt", "Sample Image")</code>	إسناد قيمة الوصفة alt بجميع العناصر img إلى القيمة Sample Image.
<code>\$("#input").attr({ value: "", title: "Please enter a value" });</code>	إسناد القيمة الفارغة إلى جميع العناصر input والعنوان title إلى القيمة Please enter a value.
<code>\$("#a[href^=http://]").attr("target", "_blank")</code>	إسناد الوصفة target إلى القيمة _blank لجميع الروابط التي تبدأ فيها الخاصية href بـ http://.
<code>\$("#a").removeAttr("target")</code>	حذف الوصفة target لجميع الروابط.
<code>\$("#form").submit(function() {\$("#:submit",this).attr("disabled", "disabled");});</code>	إسناد القيمة disabled إلى الوصفة disabled عند النقر على زر submit.
<code>\$("#p").text()</code>	تُعيد النص لكل الفقرات
<code>\$("#p").text("<i>Hello World</i>")</code>	تقوم بإسناد النص <i>Hello World</i> إلى محتوى جميع الفقرات
<code>\$("#p").html()</code>	تُعيد محتوى HTML لكل الفقرات.
<code>\$("#div").html("Hello World")</code>	تُسند Hello World إلى محتوى html لكل div.
<code>\$("#input:checkbox:checked").val()</code>	تُعيد قيمة أول صندوق تحقق مُحدّد.
<code>\$("#input:radio[name=bar]:checked").val()</code>	تُعيد أول قيمة لمجموعة من أزرار الراديو.
<code>\$("#button").val("Hello")</code>	إسناد القيمة Hello لكل button.

مثال: تغيير قيمة واصفة

```
<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function() {
      $("img").attr({
        src: "images/jquery.jpg",
        title: "jQuery",
        alt: "jQuery Logo"
      });
    });
  </script>
</head>
<body>
<div class="div1" id="divid">
  <p>Following is the logo of jQuery</p>
  
</div>
</body>
</html>
```

يُظهر المتصفح:

Following is the logo of jQuery



مثال HTML و Text

نسخ محتوى HTML الموجود في فقرة معرفها pid1 وإسناده إلى نص فقرة أخرى معرفها pid2.

```
<!doctype html>
<html>
<head>
<title>the title</title>
<script type="text/javascript"
src="jquery/jquery-1.10.2.min.js"></script>
<script type="text/javascript" language="javascript">
$(document).ready(function() {
    var content = $("#pid1").html();
    $("#pid2").text(content);
});
</script>
<style>
    .red { color:red; }
    .green { color:green; }
</style>
</head>
<body>
<p class="green" id="pid1">This is <i>first paragraph</i>.</p>
<p class="red" id="pid2">This is second paragraph.</p>
</body>
</html>
```

يُظهر المتصفح

This is *first paragraph*.

This is <i>first paragraph</i>.

أسلوب الصفحات المتتالي CSS

الأهداف التعليمية

- التعامل مع أسلوب الصفحات المتتالي.

أسلوب الصفحات المتتالي

تدعم المكتبة JQuery معظم محددات CSS بإصداراتها 1 إلى 3.

التعامل مع خصائص CSS

تُستخدم الطريقة (css(PropertyName, PropertyValue) للتعامل مع الخصائص.

مثال:

يقوم المثال التالي بتلوين العنصر الثالث في قائمة:

```
<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">

    $(document).ready(function() {
      $("li").eq(2).css("color", "red");
    });
  </script>
</head>
<body>
  <div>
    <ul>
      <li>list item 1</li>
      <li>list item 2</li>
      <li>list item 3</li>
      <li>list item 4</li>
      <li>list item 5</li>
      <li>list item 6</li>
    </ul>
  </div>
</body>
</html>
```

يُظهر المتصفح:

- list item 1
- list item 2
- list item 3
- list item 4
- list item 5
- list item 6

تطبيق أكثر من خاصية CSS

مثال:

يُبين المثال التالي تطبيق أكثر من خاصية:

```
<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function() {
      $("li").eq(2).css({"color":"red",
        "background-color":"green"});
    });
  </script>
</head>
<body>
  <div>
    <ul>
      <li>list item 1</li>
      <li>list item 2</li>
      <li>list item 3</li>
      <li>list item 4</li>
      <li>list item 5</li>
      <li>list item 6</li>
    </ul>
  </div>
</body>
</html>
```

يُظهر المتصفح:

- list item 1
- list item 2
- **list item 3**
- list item 4
- list item 5
- list item 6

تغيير عرض وارتفاع العناصر

مثال:

يُبين المثال التالي كيفية تغيير عرض وارتفاع العناصر:

```
<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">

    $(document).ready(function() {
      $("div:first").width(100);
      $("div:first").css("background-color", "blue");
    });

  </script>
  <style>
  div{ width:70px; height:50px; float:left; margin:5px;
    background:red; cursor:pointer; }
  </style>
</head>
<body>
  <div></div>
  <div>d</div>
  <div>d</div>
  <div>d</div>
  <div>d</div>
</body>
</html>
```

يُظهر المتصفح:



وظائف JQuery CSS

يُبين الجدول التالي

Method	الوصف
<code>css(name)</code>	يُعيد خاصية الأسلوب لأول عنصر محدد.
<code>css(name, value)</code>	يُسند قيمة لخاصية لجميع العناصر المحددة.
<code>css(properties)</code>	يُسند قيمة/خاصية لجميع العناصر المحددة.
<code>height(val)</code>	يُسند الإرتفاع لجميع العناصر المحددة.
<code>height()</code>	يُعيد قيمة الإرتفاع لأول عنصر محدد.
<code>width(val)</code>	يُسند العرض لجميع العناصر المحددة.
<code>width()</code>	يُعيد قيمة العرض لأول عنصر محدد.

الفصل التاسع: مهارات JQuery

الكلمات المفتاحية

عناصر الوثيقة، الأحداث، Ajax، التأثيرات.

الملخص

نستعرض في هذا الفصل مجموعة من المهارات التي توفرها JQuery.

الأهداف التعليمية:

يتعرف الطالب في هذا الفصل على:

- التعامل مع عناصر الوثيقة.
- التعامل مع الأحداث.
- استخدام تقانات Ajax.
- التأثيرات.

المخطط

يضم فصل مهارات JQuery 4 وحدات (Learning Objects) هي:

- التعامل مع عناصر الوثيقة
- التعامل مع الأحداث
- Ajax: Asynchronous JavaScript and XML
- التأثيرات Effects

التعامل مع عناصر الوثيقة

الأهداف التعليمية

- التعرف على مختلف طرق التعامل مع عناصر الوثيقة.

التعامل مع نموذج كائن الوثيقة DOM

تُمكن المكتبة JQuery من التعامل مع عناصر نموذج كائن الوثيقة DOM بسهولة.

التعامل مع المحتوى

تسمح الوظيفة `html()` بالحصول على المحتوى `.html`.

مثال:

يُبين المثال التالي استخدام الطريقة `html()` للحصول على المحتوى، واستخدام الطريقة `text(val)` لإسناد قيمة للنص.

```
<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function() {
      $("div").click(function () {
        var content = $(this).html();
        $("#result").text( content );
      });
    });
  </script>
  <style>
    #division{ margin:10px;padding:12px;
      border:2px solid #666;
      width:60px;
    }
  </style>
</head>
<body>
  <p>Click on the square below:</p>
  <span id="result"> </span>
  <div id="division" style="background-color:blue;">
    This is Blue Square!!
  </div>
</body>
</html>
```

يُظهر المتصفح أولاً:

Click on the square below:



ويعد النقر على العنصر div :

Click on the square below:

This is Blue Square!!



استبدال عناصر DOM

تسمح الوظيفة `replaceWith(content)` باستبدال عنصر بـ HTML أو بعناصر أخرى.

مثال:

يُبين المثال التالي استخدام الطريقة `replaceWith(content)`:

```
<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function() {
      $("div").click(function () {
        $(this).replaceWith("<h1>jQuery is Great</h1>");
      });
    });
  </script>
  <style>
    #division{ margin:10px;padding:12px;
      border:2px solid #666;
      width:60px;
    }
  </style>
</head>
<body>
  <p>Click on the square below:</p>
  <span id="result"> </span>
  <div id="division" style="background-color:blue;">
    This is Blue Square!!
  </div>
</body>
</html>
```

يُظهر المتصفح أولاً:

Click on the square below:



ويعد النقر على العنصر `:div`

Click on the square below:

JQuery is Great

حذف عناصر DOM

تسمح الوظيفة `remove(expr)` بحذف جميع العناصر المحددة.

مثال:

يُبين المثال التالي استخدام الطريقة `:remove(expr)`

```
<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function() {

      $("div").click(function () {
        $(this).remove( );
      });

    });
  </script>
  <style>
    .div{ margin:10px;padding:12px;
          border:2px solid #666;
          width:60px;
        }
  </style>
</head>
<body>
  <p>Click on any square below:</p>
  <span id="result"> </span>
  <div class="div" style="background-color:blue;"></div>
  <div class="div" style="background-color:green;"></div>
  <div class="div" style="background-color:red;"></div>
</body>
</html>
```

يُظهر المتصفح أولاً:

Click on any square below:



النقر على أي div سيؤدي إلى إخفائه:

Click on any square below:



إدراج عناصر DOM

تسمح الوظيفة `after (content)` بإدراج محتوى بعد العناصر المحددة، والوظيفة `before (content)` بإدراج محتوى قبل العناصر المحددة.

مثال:

يُبين المثال التالي استخدام الطريقة `:before (content)`:

```

<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function() {

      $("div").click(function () {
        $(this).before('<div class="div"></div>' );
      });

    });
  </script>
  <style>
    .div{ margin:10px;padding:12px;
          border:2px solid #666;
          width:60px;
        }
  </style>
</head>
<body>
  <p>Click on any square below:</p>
  <span id="result"> </span>
  <div class="div" style="background-color:blue;"></div>
  <div class="div" style="background-color:green;"></div>
  <div class="div" style="background-color:red;"></div>
</body>
</html>

```

يُظهر المتصفح أولاً:

Click on any square below:



النقر على أي div سيؤدي إلى ظهور div جديد قبله:

Click on any square below:



طرق التعامل مع العناصر

يُبين الجدول التالي **طرق التعامل** مع عناصر الوثيقة:

للحصول على أمثلة إضافية يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-dom.htm>

Method	Description
after(content)	Insert content after each of the matched elements.
append(content)	Append content to the inside of every matched element.
appendTo(selector)	Append all of the matched elements to another, specified, set of elements.
before(content)	Insert content before each of the matched elements.
clone(bool)	Clone matched DOM Elements, and all their event handlers, and select the clones.
clone()	Clone matched DOM Elements and select the clones.
empty()	Remove all child nodes from the set of matched elements.
html(val)	Set the html contents of every matched element.
html()	Get the html contents (innerHTML) of the first matched element.
insertAfter(selector)	Insert all of the matched elements after another, specified, set of elements.
insertBefore(selector)	Insert all of the matched elements before another, specified, set of elements.

prepend(content)	Prepend content to the inside of every matched element.
prependTo(selector)	Prepend all of the matched elements to another, specified, set of elements.
remove(expr)	Removes all matched elements from the DOM.
replaceAll(selector)	Replaces the elements matched by the specified selector with the matched elements.
replaceWith(content)	Replaces all matched elements with the specified HTML or DOM elements.
text(val)	Set the text contents of all matched elements.
text()	Get the combined text contents of all matched elements.
wrap(elem)	Wrap each matched element with the specified element.
wrap(html)	Wrap each matched element with the specified HTML content.
wrapAll(elem)	Wrap all the elements in the matched set into a single wrapper element.
wrapAll(html)	Wrap all the elements in the matched set into a single wrapper element.
wrapInner(elem)	Wrap the inner child contents of each matched element (including text nodes) with a DOM element.
wrapInner(html)	Wrap the inner child contents of each matched element (including text nodes) with an HTML structure.

التعامل مع الأحداث

الأهداف التعليمية

- التعامل مع الأحداث المختلفة.

التعامل مع الأحداث

تُمكن المكتبة JQuery من إنشاء صفحات ويب ديناميكية عن طريق استخدام الأحداث المختلفة مثل النقر وتحريك الفأرة فوق العناصر أو ضغط مفاتيح لوحة المفاتيح وغيرها. يُمكنك ربط وظيفة لتُنفذ عند وقوع الحدث الموافق. تُدعى هذه الوظيفة بمعالج الحدث Event Handler.

ربط العناصر بمعالجات الأحداث

تسمح الوظيفة bind() بربط عنصر/حدث/وظيفة.

كما يُبين المثال التالي والذي يربط كل عنصر من النوع div بمعالج لحدث النقر click:

```
$('#div').bind('click', function (event) {  
    alert('Hi there!');  
});
```

ولها ثلاث معاملات:

- نوع الحدث eventType: اسم الحدث مثل click و submit.
 - بيانات الحدث eventData: معامل اختياري يحوي البيانات المراد تمريرها لمعالج الحدث.
 - معالج الحدث handler: وظيفة سيتم تنفيذها في كل مرة يقع الحدث.
- كما تسمح الوظيفة unbind() بإزالة معالج الحدث.

مثال:

يُبين المثال التالي استخدام الطريقة bind() لربط كل العناصر من النوع div مع معالج لحدث النقر click.

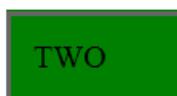
```
<!doctype html>  
<html>  
<head>  
  <title>the title</title>  
  <script type="text/javascript"  
    src="jquery/jquery-1.10.2.min.js"></script>  
  <script type="text/javascript" language="javascript">  
    $(document).ready(function () {  
      $('#div').bind('click', function (event) {  
        alert('Hi there!');  
      });  
    });  
  </script>  
  <style>  
    .div {  
      margin: 10px;  
      padding: 12px;  
      border: 2px solid #666;  
      width: 60px;  
    }  
  </style>  
</head>
```



```
<body>
  <p>Click on any square below to see the result:</p>
  <div class="div" style="background-color:blue;">ONE</div>
  <div class="div" style="background-color:green;">TWO</div>
  <div class="div" style="background-color:red;">THREE</div>
</body>
</html>
```

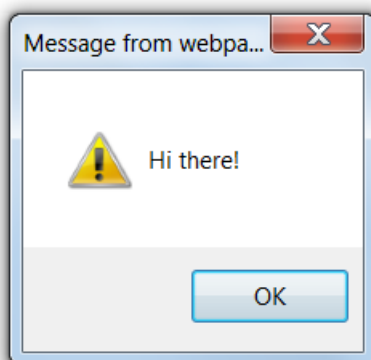
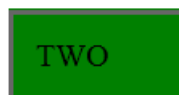
حيث يُظهر المتصفح:

Click on any square below to see the result:



وعند النقر على أي div تظهر الرسالة الموافقة.

Click on any square below to see the result:



أنواع الأحداث

يُبين الجدول التالي أنواع الأحداث:

Event Type	الوصف
blur	يحدث عندما يفقد العنصر التركيز.
change	يحدث عندما يتغير العنصر.
click	يحدث عند النقر بالفأرة على العنصر.
dblclick	يحدث عند النقر المزدوج بالفأرة على العنصر.
error	يحدث عند حصول خطأ ما (في التحميل مثلاً).
focus	يحدث عندما يصبح التركيز على العنصر.
keydown	يحدث عند ضغط مفتاح من لوحة المفاتيح.
keypress	يحدث عند ضغط مفتاح ورفع مفتاح من لوحة المفاتيح.
keyup	يحدث عند رفع مفتاح من لوحة المفاتيح.
load	يحدث عند تحميل العنصر.
mousedown	يحدث عند ضغط زر من أزرار الفأرة.
mouseenter	يحدث عند دخول الفأرة في منطقة العنصر.
mouseleave	يحدث عند خروج الفأرة من منطقة العنصر.
mousemove	يحدث عند تحريك الفأرة فوق منطقة العنصر.
mouseout	يحدث عند حركة الفأرة خروجاً من العنصر.
mouseover	يحدث عند حركة الفأرة فوق العنصر.
mouseup	يحدث عند تحرير زر الفأرة.
resize	يحدث عند تغيير حجم النافذة.
scroll	يحدث عند التمرير في النافذة.
select	يحدث عند تحديد نص.
submit	يحدث عند إرسال بيانات النموذج.
unload	يحدث عند تفريغ النموذج.

الكائن Event

يكون لمعالج الحدث معامل وحيد¹ يُدعى event.

يوفر هذا الكائن مجموعة من الخصائص والطرق وبعض طرق معالجة الأحداث.

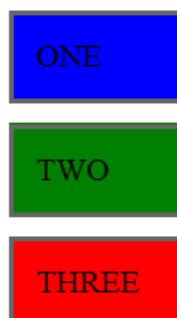
مثال:

يُبين المثال التالي استخدام بعض الخصائص للكائن event.

```
<!doctype html>
<html>
<head>
  <title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
$(document).ready(function() {
  $('div').bind('click', function( event ){
    alert('Event type is ' + event.type);
    alert('pageX : ' + event.pageX);
    alert('pageY : ' + event.pageY);
    alert('Target : ' + event.target.innerHTML);
  });
});
</script>
<style>
  .div{ margin:10px;padding:12px;
    border:2px solid #666;
    width:60px;
  }
</style>
</head>
<body>
  <p>Click on any square below to see the result:</p>
  <div class="div" style="background-color:blue;">ONE</div>
  <div class="div" style="background-color:green;">TWO</div>
  <div class="div" style="background-color:red;">THREE</div>
</body>
</html>
```

حيث يُظهر المتصفح:

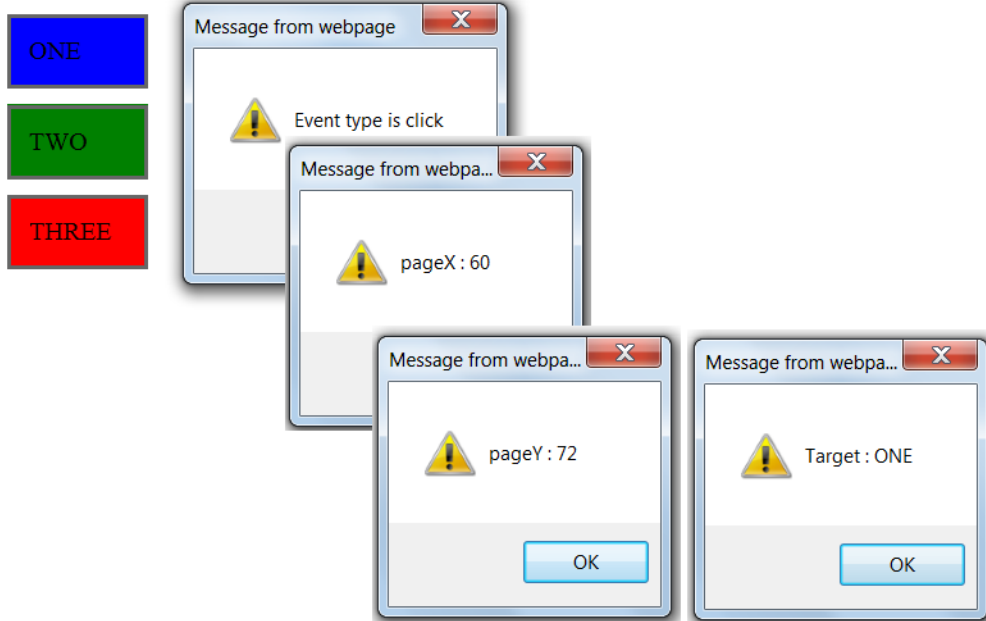
Click on any square below to see the result:



1 في الكثير من الأحيان لا يُستخدم

وعند النقر على أي div تظهر الرسائل الموافقة.

Click on any square below to see the result:



الطرق المساعدة للأحداث Event Helper Methods

توفر المكتبة مجموعة من الطرق المساعدة التي يُمكن أن تُستخدم لإطلاق حدث أو للربط مع أي نوع حدث من الأنواع السابقة.

طرق إطلاق الأحداث Trigger Methods

يُبين المثال التالي كيفية إطلاق حدث الخروج blur لجميع الفقرات.

```
$("#p").blur();
```

طرق الربط Binding Methods

يُبين المثال التالي ربط حدث النقر click مع مقطع div:

```
$("#div").click(function () {  
    // do something here  
});
```

```

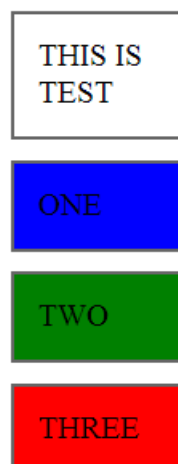
<!doctype html>
<html>
<head>
  <title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function () {

      $("div").click(function () {
        var content = "<b></b>";
        $(this).wrapInner(content);
      });
    });
  </script>
  <style>
    .div {
      margin: 10px;
      padding: 12px;
      border: 2px solid #666;
      width: 60px;
    }
  </style>
</head>
<body>
  <p>Click on any square below to see the result:</p>
  <div class="div" id="destination">THIS IS TEST</div>
  <div class="div" style="background-color:blue;">ONE</div>
  <div class="div" style="background-color:green;">TWO</div>
  <div class="div" style="background-color:red;">THREE</div>
</body>
</html>

```

يُظهر المتصفح

Click on any square below to see the result:



وعند النقر على أي مقطع سنلاحظ احتواء النص الداخلي:

Click on any square below to see the result:

THIS IS
TEST

ONE

TWO

THREE

خصائص الحدث

يُبين الجدول التالي **خصائص الحدث**:

للحصول على أمثلة إضافة يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-events.htm>

Property	Description
altKey	Set to true if the Alt key was pressed when the event was triggered, false if not. The Alt key is labeled Option on most Mac keyboards.
ctrlKey	Set to true if the Ctrl key was pressed when the event was triggered, false if not.
data	The value, if any, passed as the second parameter to the bind() command when the handler was established.
keyCode	For keyup and keydown events, this returns the key that was pressed.
metaKey	Set to true if the Meta key was pressed when the event was triggered, false if not. The Meta key is the Ctrl key on PCs and the Command key on Macs.

pageX	For mouse events, specifies the horizontal coordinate of the event relative from the page origin.
pageY	For mouse events, specifies the vertical coordinate of the event relative from the page origin.
relatedTarget	For some mouse events, identifies the element that the cursor left or entered when the event was triggered.
screenX	For mouse events, specifies the horizontal coordinate of the event relative from the screen origin.
screenY	For mouse events, specifies the vertical coordinate of the event relative from the screen origin.
shiftKey	Set to true if the Shift key was pressed when the event was triggered, false if not.
target	Identifies the element for which the event was triggered.
timeStamp	The timestamp (in milliseconds) when the event was created.
type	For all events, specifies the type of event that was triggered (for example, click).
which	For keyboard events, specifies the numeric code for the key that caused the event, and for mouse events, specifies which button was pressed (1 for left, 2 for middle, 3 for right)

طرق الحدث

يُبين الجدول التالي **طرق الحدث**:

للحصول على أمثلة إضافة يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-events.htm>

Method	Description
preventDefault()	Prevents the browser from executing the default action.
isDefaultPrevented()	Returns whether event.preventDefault() was ever called on this event object.
stopPropagation()	Stops the bubbling of an event to parent elements, preventing any parent handlers from being notified of the event.
isPropagationStopped()	Returns whether event.stopPropagation() was ever called on this event object.
stopImmediatePropagation()	Stops the rest of the handlers from being executed.
isImmediatePropagationStopped()	Returns whether event.stopImmediatePropagation() was ever called on this event object.

طرق التعامل مع الأحداث

يُبين الجدول التالي طرق **التعامل مع الأحداث**:

للحصول على أمثلة إضافة يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-events.htm>

Method	Description
bind(type, [data], fn)	Binds a handler to one or more events (like click) for each matched element. Can also bind custom events.
die(type, fn)	This does the opposite of live, it removes a bound live event.
hover(over, out)	Simulates hovering for example moving the mouse on, and off, an object.
live(type, fn)	Binds a handler to an event (like click) for all current – and future – matched element. Can also bind custom events.
one(type, [data], fn)	Binds a handler to one or more events to be executed once for each matched element.
ready(fn)	Binds a function to be executed whenever the DOM is ready to be traversed and manipulated.
Toggle (fn, fn2, fn3,...)	Toggle among two or more function calls every other click.
trigger(event, [data])	Trigger an event on every matched element.
triggerHandler(event, [data])	Triggers all bound event handlers on an element.
unbind([type], [fn])	This does the opposite of bind, it removes bound events from each of the matched elements.

Ajax: Asynchronous JavaScript and XML

الأهداف التعليمية

- استخدام تقانات Ajax.

التعامل مع تقنية Ajax

تحتوي المكتبة JQuery الكثير من الأدوات للتعامل مع تقنية Ajax² والتي تسمح بتحميل بيانات من المخدم بدون عملية إنعاش refresh للصفحة.

تحميل البيانات

تسمح الطريقة load بتحميل البيانات ولها الشكل العام:

```
[selector].load( URL, [data], [callback] );
```

حيث:

- URL عنوان المخدم الذي سيُرسل الطلب له (صفحة مثلاً).
- data معامل اختياري. البيانات المررة للطلب.
- callback معامل اختياري. وظيفة سيتم تنفيذها بعد تحميل النتائج.

مثال:

```
<!doctype html>
<html>
<head>
  <title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
$(document).ready(function() {
  $("#driver").click(function(event){
    $('#stage').load('jquery/result.html');
  });
});
</script>
</head>
<body>
  <p>Click on the button to load result.html file:</p>
  <div id="stage" style="background-color:blue;">
    STAGE
  </div>
  <input type="button" id="driver" value="Load Data" />
</body></html>
```

يحتوي الملف result.html

```
<h1>THIS IS RESULT...</h1>
```

يُظهر المتصفح أولاً:

Click on the button to load result.html file:



وبعد النقر على الزر:

Click on the button to load result.html file:



إرسال البيانات إلى المخدم

يُبين المثال التالي استخدام الطريقة load() مع إرسال بيانات إلى المخدم:

```
<!DOCTYPE html>
<html>
<head>
  <title>the title</title>
  <script src="Scripts/jquery-1.10.2.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function () {
      $("#driver").click(function (event) {
        var name = $("#name").val();
        $("#stage").load('result.aspx', { "name": name });
      });
    });
  </script>
</head>
<body>

  <p>Enter your name and click on the button:</p>
  <input type="input" id="name" size="40" /><br />
  <div id="stage" style="background-color:green;">
    STAGE
  </div>
  <input type="button" id="driver" value="Show Result" />
</body></html>
```

يجوي ملف المخدم result.aspx التعليمات التالية:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class result : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string v;
        v= Request.Params["name"];
        if (v != null)
        {
            Response.Write("Welcome ");
            Response.Write(v);
        }
    }
}
```

يُظهر المتصفح أولاً:

Enter your name and click on the button:

STAGE

وبعد النقر على الزر:

Enter your name and click on the button:

Welcome Bassel

الحصول على بيانات JSON

تُستخدم الوظيفة `getJSON()` عندما يقوم المخدم بإرجاع البيانات كسلسلة `JSON3`.

تأخذ هذه الوظيفة الشكل العام التالي:

```
[selector].getJSON( URL, [data], [callback] );
```

حيث:

- عنوان المخدم الذي سيُرسل الطلب له (صفحة مثلاً).
- data معامل اختياري. البيانات المُررة للطلب.
- callback معامل اختياري. وظيفة سيتم تنفيذها بعد تحميل النتائج وحيث يكون المعامل الأول لها البيانات المعادة من المخدم JSON.

مثال:

ليكن لدينا صفحة المخدم json.aspx والتي تُعيد بيانات بشكل json:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Script.Serialization;
public partial class json : System.Web.UI.Page
{
    class Data
    {
        public string Name { set; get; }
        public string Age { set; get; }
        public string Sex { set; get; }
    }
    protected void Page_Load(object sender, EventArgs e)
    {
        List<Data> data = new List<Data>();
        data.Add(new Data()
        {
            Name = "Bassel",
            Age = "45",
            Sex = "Male"
        });
        data.Add(new Data()
        {
            Name = "Sami",
            Age = "34",
            Sex = "Male"
        });
        var json = new JavaScriptSerializer().Serialize(data);
        Response.Clear();
        Response.ContentType = "application/json; charset=utf-8";
        Response.Write(json);
        Response.End();
    }
}
```

يُمكن استخدام الطريقة getJSON() لإظهار البيانات:

```
<!DOCTYPE html>
<html>
<head>
  <title>JSON</title>
  <script src="Scripts/jquery-1.10.2.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function () {
      $("#driver").click(function (event) {
        $.getJSON('json.aspx', function (jd) {
          $('#stage').html('<p>' + jd.length + '</p>');
          for (var i = 0; i < jd.length; i++) {
            $('#stage').append('<p> Name: ' + jd[i].Name + '</p>');
            $('#stage').append('<p> Age : ' + jd[i].Age + '</p>');
            $('#stage').append('<p> Sex: ' + jd[i].Sex + '</p>');
          }
        });
      });
    });
  </script>
</head>
<body>
  <p>Click on the button to load json data:</p>
  <div id="stage" style="background-color:blue;">
    STAGE
  </div>
  <input type="button" id="driver" value="Load Data" />
</body></html>
```

يُظهر المتصفح أولاً:

Click on the button to load json data:



وبعد النقر على الزر:

Click on the button to load json data:



JQuery AJAX وظائف

يُبين الجدول التالي **طرق التعامل** مع الأحداث:
للحصول على أمثلة إضافة يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-ajax.htm>

Methods and Description
<code>jQuery.ajax(options)</code> Load a remote page using an HTTP request.
<code>jQuery.ajaxSetup(options)</code> Setup global settings for AJAX requests.
<code>jQuery.get(url, [data], [callback], [type])</code> Load a remote page using an HTTP GET request.
<code>jQuerygetJSON(url, [data], [callback])</code> Load JSON data using an HTTP GET request.
<code>jQuery.getScript(url, [callback])</code> Loads and executes a JavaScript file using an HTTP GET request.
<code>jQuery.post(url, [data], [callback], [type])</code> Load a remote page using an HTTP POST request.
<code>load(url, [data], [callback])</code> Load HTML from a remote file and inject it into the DOM.
<code>serialize()</code> Serializes a set of input elements into a string of data.
<code>serializeArray()</code> Serializes all forms and form elements like the <code>.serialize()</code> method but returns a JSON data structure for you to work with.

أحداث JQuery AJAX

يُبين الجدول التالي **طرق التعامل** مع الأحداث:
للحصول على أمثلة إضافة يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-ajax.htm>

Methods and Description
ajaxComplete (callback) Attach a function to be executed whenever an AJAX request completes.
ajaxStart (callback) Attach a function to be executed whenever an AJAX request begins and there is none already active.
ajaxError (callback) Attach a function to be executed whenever an AJAX request fails.
ajaxSend (callback) Attach a function to be executed before an AJAX request is sent.
ajaxStop (callback) Attach a function to be executed whenever all AJAX requests have ended.
ajaxSuccess (callback) Attach a function to be executed whenever an AJAX request completes successfully.

التأثيرات Effects

الأهداف التعليمية

- استخدام التأثيرات المختلفة.

التأثيرات

تُمكن المكتبة JQuery من القيام بعمليات تحريك وإحياء ممتعة.

إظهار وإخفاء العناصر

تسمح الوظيفة `show()` بإظهار العناصر كما تسمح الوظيفة `hide()` بإخفائها.

لكلا الوظيفتين المعاملين التاليين:

- `Speed` ويأخذ قيمة رقمية بالملي ثانية أو أحد القيم النصية: ("slow", "normal", "fast")
- `callback` وهو معامل اختياري يُحدد وظيفة للتنفيذ بعد كل عملية انتهاء للتأثير.

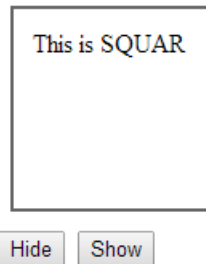
مثال:

```
<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">

    $(document).ready(function() {

      $("#show").click(function () {
        $(".mydiv").show( 1000 );
      });
      $("#hide").click(function () {
        $(".mydiv").hide( 1000 );
      });
    });
  </script>
  <style>
  .mydiv{ margin:10px;padding:12px;
    border:2px solid #666;
    width:100px;
    height:100px;
  }
  </style>
</head>
<body>
  <div class="mydiv">
    This is SQUAR
  </div>
  <input id="hide" type="button" value="Hide" />
  <input id="show" type="button" value="Show" />
</body>
</html>
```

يُظهر المتصفح أولاً:



وبعد النقر على الزر hide يختفي العنصر div:



قلب العناصر

تسمح الوظيفة toggle() بقلب حالة ظهور العنصر .

للوظيفة المعاملين التاليين:

- Speed وهو معامل اختياري ويأخذ قيمة رقمية بالملي ثانية أو أحد القيم النصية ("normal", "slow", "fast" or "fast").
- callback وهو معامل اختياري يُحدد وظيفة للتنفيذ بعد كل عملية انتهاء للتحريك.

مثال:

```
<!doctype html>
<html>
<head>
<title>the title</title>
<script type="text/javascript"
src="jquery/jquery-1.10.2.min.js"></script>
<script type="text/javascript" language="javascript">
$(document).ready(function() {
    $(".clickme").click(function(event){
        $(".target").toggle('slow', function(){
            $(".log").text('Transition Complete');
        });
    });
});
</script>
<style>
.clickme{ margin:10px;padding:12px;
border:2px solid #666;
width:100px;
height:50px;
}
</style>
</head>
<body>
<div class="content">
<div class="clickme">Click Me</div>
```

```
<div class="target">
  
</div>
<div class="log"></div>
</body>
</html>
```

يُظهر المتصفح أولاً:

Click Me



وبعد النقر على المقطع div الأول يختفي المقطع الثاني:

Click Me

Transition Complete

عند إعادة النقر على المقطع div الأول يظهر المقطع الثاني:

Click Me



Transition Complete

وهكذا دواليك.

طرق التأثيرات JQuery

يُبين الجدول التالي **طرق التأثيرات** في المكتبة JQuery. للحصول على أمثلة إضافة يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-effects.htm>

Methods and Description
<code>animate(params, [duration, easing, callback])</code> A function for making custom animations.
<code>fadeIn(speed, [callback])</code> Fade in all matched elements by adjusting their opacity and firing an optional callback after completion.
<code>fadeOut(speed, [callback])</code> Fade out all matched elements by adjusting their opacity to 0, then setting display to "none" and firing an optional callback after completion.
<code>fadeTo(speed, opacity, callback)</code> Fade the opacity of all matched elements to a specified opacity and firing an optional callback after completion.
<code>hide()</code> Hides each of the set of matched elements if they are shown.
<code>hide(speed, [callback])</code> Hide all matched elements using a graceful animation and firing an optional callback after completion.
<code>show()</code> Displays each of the set of matched elements if they are hidden.
<code>show(speed, [callback])</code> Show all matched elements using a graceful animation and firing an optional callback after completion.
<code>slideDown(speed, [callback])</code> Reveal all matched elements by adjusting their height and firing an optional

callback after completion.
<p><code>slideToggle(speed, [callback])</code></p> <p>Toggle the visibility of all matched elements by adjusting their height and firing an optional callback after completion.</p>
<p><code>slideUp(speed, [callback])</code></p> <p>Hide all matched elements by adjusting their height and firing an optional callback after completion.</p>
<p><code>stop([clearQueue, gotoEnd])</code></p> <p>Stops all the currently running animations on all the specified elements.</p>
<p><code>toggle()</code></p> <p>Toggle displaying each of the set of matched elements.</p>
<p><code>toggle(speed, [callback])</code></p> <p>Toggle displaying each of the set of matched elements using a graceful animation and firing an optional callback after completion.</p>
<p><code>toggle(switch)</code></p> <p>Toggle displaying each of the set of matched elements based upon the switch (true shows all elements, false hides all elements).</p>
<p><code>jQuery.fx.off</code></p> <p>Globally disable all animations.</p>

مكتبة واجهة المستخدم UI

توفر هذه **المكتبة** مجموعة إضافية من التأثيرات المخصصة والتي نعرضها في الجدول المرجع التالي:
للحصول على أمثلة إضافية يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-effects.htm>

يجب تنزيل المكتبة من الرابط:

<http://jqueryui.com/download/>

Methods and Description
<p>Blind</p> <p>Blinds the element away or shows it by blinding it in.</p>
<p>Bounce</p> <p>Bounces the element vertically or horizontally n-times.</p>
<p>Clip</p> <p>Clips the element on or off, vertically or horizontally.</p>
<p>Drop</p> <p>Drops the element away or shows it by dropping it in.</p>
<p>Explode</p> <p>Explodes the element into multiple pieces.</p>
<p>Fold</p> <p>Folds the element like a piece of paper.</p>
<p>Highlight</p> <p>Highlights the background with a defined color.</p>
<p>Puff</p> <p>Scale and fade out animations create the puff effect.</p>
<p>Pulsate</p> <p>Pulsates the opacity of the element multiple times.</p>
<p>Scale</p> <p>Shrink or grow an element by a percentage factor.</p>
<p>Shake</p> <p>Shakes the element vertically or horizontally n-times.</p>
<p>Size</p> <p>Resize an element to a specified width and height.</p>
<p>Slide</p>

Slides the element out of the viewport.

Transfer

Transfers the outline of an element to another.

اقتراحات وتمارين

1. قم بالدخول إلى الرابط التالي وحل جميع التمارين التفاعلية الموجودة:

<http://jqexercise.droppages.com/>

2. قم بالدخول إلى الرابط التالي وحل جميع التمارين:

<http://courses.cs.washington.edu/courses/cse190m/12su/sections/section07.shtml#>

الفصل العاشر:

أساسيات ASP .Net 2013

الكلمات المفتاحية

ASPX، لغة التنفيذ المشتركة CLR، وثيقة ASPX، الكود الخلفي، عناصر التحكم HTML، عناصر التحكم .Web Controls

المُلخَص

نستعرض في هذا الفصل أساسيات صفحات ASPX. ثم نُبين استخدام عناصر التحكم من النوع HTML ومن ثم من النوع .Web Controls.

الأهداف التعليمية

يتعرف الطالب في هذا الفصل على:

- مفهوم ASPX.
- وثيقة ASPX.
- عناصر التحكم HTML.
- أحداث عناصر التحكم HTML.
- عناصر التحكم .Web Controls.
- عناصر التحكم من الصحة.

المخطط

يضم فصل أساسيات ASP .Net 2013 4 وحدات (Learning Objects) هي:

- أساسيات ASP.NET 2013
- عناصر التحكم HTML Controls
- عناصر التحكم Web Controls
- عناصر التحقق من الصحة

أساسيات ASP.NET 2013

الأهداف التعليمية

- ASPX
- لغة التنفيذ المشتركة CLR
- وثيقة ASPX
- الكود الخلفي

مقدمة

تُعتبر .NET. المظلة التي تجمع مجموعة من التقانات التي أطلقتها Microsoft في عام 2002 انطلاقةً من حقيقة توزع التطبيقات على مكونات برمجية components تعمل على حواسيب مختلفة في أمكنة متعددة من العالم.

يكون المكون البرمجي component عبارة عن كبسلة لبرمجيات يُمكن أن تعمل بمفردها، أو أن تُستخدم من قبل مكونات أخرى وبدون أن تعرف هذه المكونات كيفية التحقيق البرمجي لوظائف المكونة المستخدمة. يُشكّل الإطار .NET. الإطار الذي يسمح بتطوير مكونات برمجية باستخدام لغات برمجية مختلفة، أما أدوات التطوير والنشر فتكون مستقلة عن لغة البرمجة.

لغة التنفيذ المشتركة CLR

تُعتبر لغة التنفيذ المشتركة Common Language Runtime التقانة الأساسية لـ .NET، والتي تؤمن خدمات لغة حيادية language-neutral لمعالجة وتنفيذ برمجيات .NET. ومن أهم هذه الخدمات:

- التحقق من أنماط البيانات type checking
- تتبع التنفيذ debugging
- التنظيف garbage collection
- معالجة الاستثناءات exception handling

يتوفر للغة التنفيذ المشتركة، ومن أجل أي لغة برمجة، مترجم يقوم بترجمة البرنامج المصدر إلى لغة مشتركة وسيطة تُدعى (IL) Intermediate Language. يكون لجميع البرامج IL نفس الشكل مهما كانت اللغة المصدر. يقوم المترجم Just-In Time (JIT) compiler (والذي هو جزء من CLR) بترجمة طريقة method إلى لغة الآلة عند استدعاء هذه الطريقة.

لغات .NET

حوت .NET. في البداية على خمسة لغات:

- VB .NET
- C++ .NET
- JScript .NET
- J# .NET
- C#

أما اليوم فيوجد حوالي أكثر من عشرين لغة مثل: COBOL, Fortran, Perl, Python (تحوي كل لغة على مترجم إلى IL).

توفر ميزة وجود هذا العدد من اللغات سهولة تهجير البرامج المكتوبة سابقاً إلى الإطار .NET. كما أن المبرمجين الخبراء في لغة معينة يمكن أن يستمروا باستخدام لغتهم المفضلة. وبالطبع فلا يُفضل كتابة منظومة باستخدام لغات مختلفة.

أساسيات ASP.NET

تُعتبر ASP.NET (Active Server Pages) تقانة لبناء وثائق الويب الديناميكية. تدعم هذه الوثائق تنفيذ الكود على مخدّم الويب. كما يُمكن أن تحوي بالطبع على خطوات تُنفذ من جهة الزبون. يُمكن كتابة الكود بأي لغة من لغات .NET. (ستتم ترجمة هذا الكود). يُمكن وضع الكود ضمن الوثيقة نفسها أو في ملف مستقل يُدعى عادةً ملف الكود الخلفي code-behind file. تُتيح هذه الميزة الفصل بين مصمّم الصفحات وبين المبرمج.

تتم ترجمة كل وثيقة إلى صف class يُخزن في مجمع Assembly. يكون هذا الصف مشتق من الصف System.Web.UI.Page، وبالتالي فهو يرث مجموعة من الأعضاء members التي يوفرها هذا الصف. من أكثر هذه الأعضاء استخداماً الأغراض Request و Response، والصفوف HTMLControls و WebControls، والخاصية IsPostBack.

- تُستخدم مثلاً الطريقة Write للغرض Response (من الصف Page) لكتابة خرج في وثيقة.
- يُعرّف الصفان HTMLControls و WebControls مجموعة من عناصر التحكم التي يُمكن استخدامها في الوثائق.
- أما الخاصية IsPostBack فتُستخدم لمعرفة فيما إذا كان الطلب الحالي للوثيقة هو نتيجة تفاعل المستخدم مع نموذج أو (على عكس ذلك) الحالة الابتدائية لوثيقة.

تتم ترجمة الوثائق ASP.NET التي لا تحوي ملفات كود خلفي مباشرةً إلى صف مشتق من الصف System.Web.UI.Page.

بينما يكون صف الوثيقة التي تستخدم ملفات كود خلفي صف مشتق من الصف الموجود في الكود الخلفي والذي هو بدوره صف مشتق من الصف System.Web.UI.Page² (إشارة تولد) (إشارة تولد)

1 ندعو عادةً الصف المولد من ترجمة وثيقة بصف الوثيقة document class

2 وبهذا يكون البرنامج في هذه الحالة مشتق من صف الكود الخلفي ومن الصف System.Web.UI.Page

وثائق ASP.NET

يُمكن لوثيقة ASP.NET أن تحوي عدد من العناصر المختلفة:

1. مؤثرات XHTML

تُعرّف هذه المؤثرات عناصر الوثيقة. يُمكن جعل هذه العناصر ديناميكية باستخدام خطاطات من جهة الزبون أو كود من جهة المخدم.

2. تعليقات من جهة الزبون

حيث يتم استخدام <!--

3. تعليمات Directives

من الواصفات attributes. وهي تعليمة مطلوبة في كل وثيقة ASP.NET تحوي على كود برمجي. كما أن الواصفة language لهذه التعليمة مطلوبة لتحديد لغة البرمجة المستخدمة.

4. كتل تعريف Declaration blocks

يوضع كود هذه الكتل ضمن المؤثر script مع إسناد قيمة الواصفة runat إلى server:

```
<script runat = "server">  
.....  
</script>
```

لا يتم تنفيذ كود هذه الكتل بشكل مباشر. إذ تحوي هذه الكتل عادةً على تعاريف البرامج الفرعية (بما فيها معالجات الأحداث). يتم وضع هذه الكتل ضمن صف الوثيقة مباشرة عند الترجمة.

5. كتل الإعادة البرمجية Render blocks

يُكتب كود هذه الكتل ضمن <% ... %>. لا يُمكن أن يحوي هذا الكود على تعريف برامج فرعية. يتم وضع هذا الكود ضمن وظيفة في صف الوثيقة عند ترجمة الوثيقة إلى صف الوثيقة. تحوي هذه الكتل عادةً على استدعاءات لإجرائيات فرعية وعلى تعليمات إظهار. يتم تنفيذ هذه الوظيفة عندما يتم تنفيذ صف الوثيقة (والذي يولد وثيقة XHTML التي يتم إرجاعها إلى المتصفح الطالب).

6. تعليقات من جهة المخدم

تُستخدم <!-- ... --> لوضع التعليقات من جهة المخدم.

مثال:

يُبين المثال التالي استخدام مختلف العناصر السابقة. حيث تُعرّف في كتلة التعريف الغرض randomGen من الصف Random، والمتحول msg من النمط string، والمصفوفة myArray من 10 عناصر من النمط int كما تُعرّف الوظيفة fillArray() والتي تقوم بملء المصفوفة myArray بعشرة أرقام عشوائية. نقوم في كتلة الإعادة البرمجية باستدعاء الوظيفة fillArray() وإظهار قيم المصفوفة myArray.

```
<!-- ex1.aspx
  A simple example of an ASP.NET document
  It uses a function to fill an array with pseudorandom numbers,
  which are then displayed
-->
<%@ Page language="c#" %>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head> <title> Ex1 </title>
  <script runat = "server">
// Build a pseudorandom number method
  Random randomGen = new Random();
  string msg;
  private int [ ] myArray = new int[10];
// A method to fill the array with pseudorandom numbers
  public void fillArray() {
    for (int index = 0; index < 10; index++)
// Generate a pseudorandom number in the range of 0 to 100
      myArray[index] = randomGen.Next(0, 100);
  }
  </script>
</head>
<body>
<%-- Code to call the fillArray method and display the array--%>
<% fillArray();
  Response.Write(
    "<br /> <b>The array's contents are: </b><br /><br />");
  for (int index = 0; index < 10; index++) {
    msg = string.Format("The element at {0} is: {1} <br />",
      index, myArray[index]);
    Response.Write(msg);
  }
%>
</body> </html>
```

حيث يكون الخرج:

The array's contents are:

```
The element at 0 is: 69
The element at 1 is: 31
The element at 2 is: 91
The element at 3 is: 34
The element at 4 is: 71
The element at 5 is: 99
The element at 6 is: 21
The element at 7 is: 28
The element at 8 is: 9
The element at 9 is: 65
```

الكود الخلفي

سيكون من الأفضل فصل التصريحات البرمجية عن الوثيقة، مما يسمح بفصل منطق البرنامج عن مسائل الإظهار. يُمكن مثلاً إعادة المثال السابق كما يلي:

يتم كتابة كتلة التعريف ضمن صف (ندعوه ex2) مشتق من الصف Page:

```
// ex2.aspx.cs
// The code behind file for ex2.aspx
// Includes a function to build an array of ten pseudorandom
// numbers

using System;
using System.Web;
using System.Web.UI;

public class ex2 : Page {
    // Build a pseudorandom number method
    Random randomGen = new Random();
    protected int [] myArray = new int[10];
    // A method to fill the array with pseudorandom numbers
    public void fillArray() {
        for (int index = 0; index < 10; index++)
    // Generate a pseudorandom number in the range of 0 to 100
        myArray[index] = randomGen.Next(0, 100);
    }
}
```

أما في الملف aspx فيتم في التعليمة @Page التصريح عن الوراثة من الصف السابق باستخدام الوصفة Inherits وتحديد اسم الملف الذي يحوي الصف باستخدام الوصفة Src أو CodeFile:

```
<!-- ex2.aspx
A simple example of an ASP.NET document with a code-behind file
It has the same functionality as ex1.aspx
-->
<%@ Page language="C#" Inherits = "ex2" Src = "ex2.aspx.cs" %>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head> <title> Ex2 </title>
  </head>
  <body>

  <!-- Code to call the fillArray method and display the array-->

  <% string msg;
fillArray();
Response.Write(
    "<br /> <b>The array's contents are: </b><br /><br />");
for (int index = 0; index < 10; index++) {
    msg = string.Format("The element at {0} is: {1} <br />",
        index, myArray[index]);
    Response.Write(msg);
}
%>
  </body>
</html>
```

عناصر التحكم HTML Controls

الأهداف التعليمية

- التعامل مع عناصر التحكم HTML في صفحات ASPX.

عناصر التحكم ASP.NET

تُدعى عناصر النموذج widgets في الوثائق ASP.NET بعناصر التحكم controls. كما ندعو الكود المرتبط مع هذه العناصر والذي يُنفذ على المخدم بكود تحكم المخدم server controls. يُعالج هذا الكود الأحداث المرتبطة مع عناصر التحكم. يوجد نوعين من عناصر التحكم:

- عناصر التحكم HTML Controls
- عناصر الويب Web Controls

عناصر التحكم HTML Controls

يتضمن الفضاء (namespace) System.Web.UI.HTML.Controls عناصر التحكم HTML. تُشابه هذه العناصر العناصر XHTML إلا أنها تتميز بأنها تتفاعل مع الكود من جهة المخدم والذي يُمكن أن يُغير ديناميكياً من محتوى وسلوك هذه العناصر، حيث يُمكن الوصول للواصفات التي تتحكم بهذه العناصر عن طريق أغراض في صف الوثيقة. يُمكن لعناصر التحكم في أغلب الأحيان أن ترفع أحد الحدثين على المخدم ServerClick أو ServerChange. يُبين الجدول التالي أنماط أكثر عناصر التحكم HTML استخداماً وعناصر XHTML الموافقة والأحداث التي ترفعها على المخدم:

Control Type	XHTML Element	Event
HtmlHidden	<input type="hidden">	ServerChange
HtmlInputButton	<input type="button">	ServerClick
	<input type="submit">	ServerClick
	<input type="reset">	ServerClick
HtmlInputCheckBox	<input type="checkbox">	ServerChange
HtmlInputRadioButton	<input type="radio">	ServerChange
HtmlInputText	<input type="text">	ServerChange
	<input type="password">	ServerChange
HtmlAnchor	<a>	ServerClick
HtmlButton	<button>	ServerClick
HtmlSelect	<select>	ServerChange
HtmlTextArea	<textarea>	ServerChange
HtmlForm	<form>	None
HtmlTable	<table>	None
HtmlTableCell	<td> and <th>	None
HtmlTableRow	<tr>	None

لاحظ أن بعض أنماط عناصر التحكم HTML ليست عناصر نموذج، أي لا يستطيع المستخدم التفاعل معها (مثل HTMLTableCell)، إلا أن وجود الأنماط يسمح بالوصول إلى واصفاتها من خلال التعليمات البرمجية مما يُمكن من تغييرها ديناميكياً.

يتم ترجمة عناصر التحكم من جهة المخدم إلى حقول fields في صف الوثيقة. ويكون معرف العنصر id اسم الحقل. ولهذا فيجب وضع معرف لعناصر التحكم حين استخدامها من جهة المخدم. كما يتم تحديد العنصر بأنه من جهة المخدم عن طريق إسناد القيمة server إلى الوصفة runat (بدون هذا يصبح العنصر عبارة عن عنصر XHTML ساكن لا يصل له المخدم). لنفرض أن لدينا مثلاً:

```
<form runat="server">
  <input type="text" id="address"
        runat="server" />
  . . .
</form>
```

يُمكن في هذه الحالة أن يحوي صف الوثيقة:

```
protected HtmlInputText address;
```

لاحظ أن المؤثر <form> لا يحوي الوصفة action وذلك لأن جميع الأحداث البرمجية التي تنجم عن تفاعل المستخدم مع عناصر التحكم تُعرّف في الوثائق نفسها أو في ملف الكود الخلفي. يكون صف أي عنصر تحكم HTML مشتق من الصف HtmlControl وبالتالي فهو يرث خصائص وطرق هذا الصف. أما الخصائص والطرق الخاصة بعنصر التحكم فتكون معرفة في صف العنصر الموافق. فمثلاً، يحوي الصف HtmlAnchor الخاصية Href.

تختلف عناصر التحكم HTML عن عناصر XHTML العادية في أنها تُحول إلى أغراض في صف الوثيقة. مما يعني أن هذه العناصر وواصفاتها يُمكن الوصول لها وتغييرها من خلال التعليمات البرمجية.

مثال: استخدام عناصر HTML Controls من جهة المخدم

نقوم في المثال التالي ببناء صفحة تستخدم تستخدم عناصر HTML. إنما ستكون البرمجة من جهة المخدم أي باستخدام C#.

تهدف الصفحة إلى حساب سعر الإقامة في فندق وذلك وفق مجموعة من الخيارات التي يُمكن أن يقوم بها المستخدم:

- تصنيف الفندق: *, **, ***, ****, *****.
- إطلالة الفندق: البحر، الغابة، الجبل.
- الوجبات المطلوبة: فطور، غداء، عشاء.
- خدمات إضافية: تلفزيون، مسبح، مطبخ.

Hotel Club (HTML Controls/ Server)		
Country:	Italy ▾	
Stars:	*** ▾	
Where:	<input checked="" type="radio"/> Sea <input type="radio"/> Forest <input type="radio"/> Mountain	
Meals:	<input checked="" type="checkbox"/> Breakfast <input checked="" type="checkbox"/> Lunch <input checked="" type="checkbox"/> Dinner	
Extra:	TV ▲ Pool Kitchen ▾	
How Much	925	

ولتكن مثلاً الأسعار التالية المعتمدة:

Stars		Where		Food		Extra	
Stars	Price	Sea	+50%	Breakfast	50	TV	50
*	100	Forest	+30%	Lunch	100	Pool	100
**	200	Mountain	+20%	Dinner	200	Kitchen	150
***	300						
****	400						
*****	500						

يُمكن في هذه المسألة البسيطة تثبيت الأسعار في قيم عناصر HTML للرجوع إليها في إجرائية #C المستخدمة لحساب سعر الليلة:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>HTML Server</title></head>
<body runat="server" bgcolor="papayawhip">
<form id="form1" runat="server">
<table border="1">
<caption>Hotel Club (HTML Controls/ Server)</caption>
<tr>
<td>Country:</td>
<td>
<select name="SelectFlag" id="SelectFlag" runat="server"
onserverchange="SelectFlag_ServerChange">
<option value="images\usa.png">USA</option>
<option value="images\france.png">France</option>
<option value="images\italy.png">Italy</option>
```

```

<option value="images\germany.png">Germany</option></select></td>
<td>
</td></tr>
<tr><td>Stars:</td>
<td>
<select id="SelectStars" name="SelectStars" runat="server">
<option value="100">*</option>
<option value="200">**</option>
<option value="300">***</option>
<option value="400">****</option>
<option value="500">*****</option></select></td><td></td></tr>
<tr><td>Where:</td>
<td>
<input id="RadWhere1" name="RadWhere" checked type="radio" runat="server"
value="1.5" />Sea<br />
<input id="RadWhere2" type="radio" name="RadWhere" runat="server"
value="1.3" />Forest &nbsp;&nbsp;<br />
<input id="RadWhere3" type="radio" name="RadWhere" runat="server"
value="1.2" />Mountain &nbsp;&nbsp;<br /></td>
<td></td></tr>
<tr><td>Meals:</td>
<td>
<input id="chkBreak" name="chkBreak" type="checkbox" runat="server"
value="50" />Breakfast<br />
<input id="chkLunch" name="chkLunch" type="checkbox" runat="server"
value="100" />Lunch<br />
<input id="chkDinner" name="chkDinner" type="checkbox" runat="server"
checked="CHECKED" value="200" />Dinner</td><td></td></tr>
<tr><td>Extra:</td><td>
<select id="SelectExtra" multiple name="SelectExtra" runat="server"
dir="ltr" size="3">
<option selected="selected" value="50">TV</option>
<option value="75">Pool</option>
<option value="100">Kitchen</option></select><br /></td>
<td></td></tr>
<tr>
<td>
<input id="Submit1" type="button" runat="server" value="How Much"
onserverclick="Submit1_ServerClick" /></td>
<td>
<input id="txtOut" name="txtOut" runat="server" type="text" />
</td><td></td></tr>
</table>
</form>
</body>
</html>

```

لاحظ إسناد الخاصية `runat=server` إلى عناصر التحكم HTML. كي تستطيع إجراءات #C التالية الوصول إلى هذه العناصر.

توجد أعلام الدول في المجلد `images`. تكون قيمة كل خيار من قائمة البلدان مسار ملف صورة علم البلد. تقوم إجراءات المخدم التالية والمكتوبة على حدث تغيير العنصر المحدد من قائمة البلاد بإسناد قيمة عنصر القائمة (والذي يحوي اسم ملف العلم) إلى الخاصية `src` لعنصر الصورة.

```

protected void SelectFlag_ServerChange(object sender, EventArgs e)
{
    this.imgFlag.Src = this.SelectFlag.Value;
}

```

تقوم إجرائية المخدم التالية بحساب سعر الإقامة استناداً لخيارات المستخدم. تعود هذه الوظيفة إلى قيمة كل عنصر لمعرفة سعر الخيار المحدد.

```
protected void Submit1_ServerClick(object sender, EventArgs e)
{
    double x = 0;
    x = double.Parse(this.SelectStars.Value);

    if (this.RadWhere1.Checked)
        x = x * double.Parse(this.RadWhere1.Value);
    else if (this.RadWhere2.Checked)
        x = x * double.Parse(this.RadWhere2.Value);
    else if (this.RadWhere3.Checked)
        x = x * double.Parse(this.RadWhere3.Value);

    if (this.chkBreak.Checked)
        x = x + double.Parse(this.chkBreak.Value);
    if (this.chkLunch.Checked)
        x = x + double.Parse(this.chkLunch.Value);
    if (this.chkDinner.Checked)
        x = x + double.Parse(this.chkDinner.Value);

    for (int i = 0; i < this.SelectExtra.Items.Count; i++)
        if (this.SelectExtra.Items[i].Selected)
            x = x +
                double.Parse(this.SelectExtra.Items[i].Value);

    this.txtOut.Value = x.ToString();
}
```

دورة حياة وثيقة ASP.NET Life Cycle

يكون لكل وثيقة ASP.NET نوعين من الطلب request:

- الطلب الأول initial request والذي ينتج عن طلب الوثيقة وإظهار نماذجها للمستخدم.
- الطلب اللاحق postback request والذي يكون بعد تغيير قيم عناصر النموذج من قبل المستخدم. يُدعى هذا الطلب بالطلب اللاحق postback لأن قيم عناصر النموذج تُعاد إلى الوثيقة على المخدم.

لتوضيح تسلسل الأحداث لوثيقة ASP.NET تحوي نموذج لنأخذ المثال التالي:

```
<!-- ex3.aspx
A simple example of an ASP.NET document with HTML controls.
It uses textboxes to get the name and age of the client,
which are then displayed.
-->
<%@ Page language="c#" %>

<html xmlns="http://www.w3.org/1999/xhtml" >
  <head> <title> Ex3 </title>
  </head>
  <body>
    <form runat = "server">
      <p>
        Your Name:
        <input type = "text" id = "name" runat = "server" />
      </p>
    </form>
  </body>
</html>
```



```

<br />
Your age:
<input type = "text" id = "age" runat = "server" />
<br />
<input type = "submit" value = "submit" />
<br />
<% if (IsPostBack) { %>
    Hello <%= name.Value %> <br />
    You are <%= age.Value %> years old <br />
<% } %>
</p>
</form>
</body>
</html>

```

لاحظ أن كل من النموذج وعناصر التحكم يجب أن تحوي القيمة server للوصفة .runat.

يقوم ASP.NET ضمناً بتخزين حالة كل عنصر تحكم لمنتسخ صف الوثيقة document class instance قبل أن يُعيد المخدم خرج المنتسخ إلى الزبون. تُخزن هذه المعلومات في عنصر تحكم مخفي على الصفحة يُدعى ViewState:

```

<input type="hidden"
name="__VIEWSTATE" id="__VIEWSTATE"

```

عندما تُعاد الوثيقة إلى المخدم تُستخدم بيانات ViewState ضمناً لإعطاء قيم ابتدائية للمنتسخ الجديد عند إعادة توليده. تسمح ViewState إذاً بالمحافظة على الحالة (قيم عناصر النموذج) بين الطلبات المتتالية لنفس الصفحة.

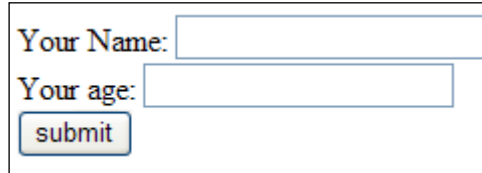
تُبين القائمة التالية الأمور التي تحدث إذا طلبت الوثيقة السابقة، فُقدت للمتصفح، ثم قام المستخدم بتعبئة صناديق النص فيها، ثم أعادها للمخدم، ثم أُعيدت مرة أخرى للمتصفح:

1. يطلب المستخدم الوثيقة ex3.aspx.
2. تتم ترجمة الوثيقة المطلوبة في المخدم فيُنشئ صف الوثيقة ويُستدعى باني الصف.
3. يتم إعطاء قيم ابتدائية لحالة عناصر التحكم باستخدام ViewState (في الطلب الأول لا تحوي ViewState أية بيانات).
4. تُستخدم بيانات النموذج للطلب لإسناد حالة عناصر التحكم لمنتسخ صف الوثيقة (لا يوجد بيانات نموذج في الطلب الأول).
5. تُسجل الحالة الحالية للمنتسخ في الحقل المخفي ViewState.
6. يُنفذ المنتسخ وتُعاد النتيجة إلى الزبون.
7. يُمسح كل من صف الوثيقة والمنتسخ من المخدم.
8. يقوم المستخدم بالتفاعل مع النموذج في الوثيقة (تغيير قيم بعض عناصر النموذج).
9. يقوم المستخدم بطلب إعادة postback إلى المخدم (عند النقر على زر الإرسال submit مثلاً).
10. تتم ترجمة الوثيقة المطلوبة في المخدم فيُنشئ صف الوثيقة ويُستدعى باني الصف.
11. يتم إسناد حالة عناصر التحكم في المنتسخ باستخدام ViewState.
12. تُستخدم بيانات نموذج الطالب لإسناد قيم لحالة عناصر التحكم لمنتسخ صف الوثيقة.

13. تُسجل الحالة الحالية للصف في ViewState.

14. يتم تنفيذ المنتسخ وتُعاد النتيجة إلى الزبون. ويُمسح الصف والمنتسخ من المخدم.

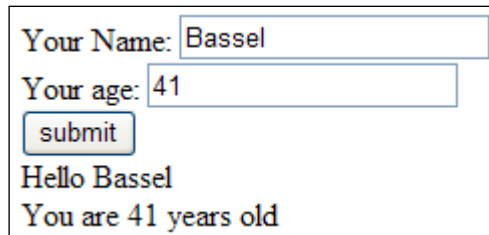
تكون الوثيقة المنشأة من صف الوثيقة المترجم من الوثيقة ex3.aspx كما يلي:



Your Name:
Your age:

```
<html xmlns="http://www.w3.org/1999/xhtml" >
  <head <title> Ex3 </title>
  </head>
  <body>
    <form name="ctl100" method="post"
      action="ex3.aspx" id="ctl100">
    <div>
    <input type="hidden"
name="__VIEWSTATE"
id="__VIEWSTATE"
value="/wEPDwULLTEzNTg2NzExNzZkZPMc7/+Qlyt5as8R30c8Ya5ME5fW" />
    </div>
    <p>
      Your Name:
      <input name="name" type="text" id="name" />
      <br />
      Your age:
      <input name="age" type="text" id="age" />
      <br />
      <input type = "submit" value = "submit" />
    <br />
    </p>
    <div>
      <input type="hidden"
name="__EVENTVALIDATION"
id="__EVENTVALIDATION"
value="/wEWAwKL1uvQCgL7uPQdAtCCr6oGMAo8C59NSjolaHXJlcGerJ0RLFk=" />
    </div></form>
    </body>
  </html>
```

وبعد أن تُعبأ من قبل المستخدم وترسل للمخدم (بالنقر على submit) كما يلي:



Your Name:
Your age:

Hello Bassel
You are 41 years old

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Ex3 </title>
</head>
<body>
  <form name="ctl100" method="post"
    action="ex3.aspx" id="ctl100">
    <div>
      <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
        value="/wEPDwULLTEzNTg2NzExNzZkZPMc7/+Qlyt5as8R30c8Ya5ME5fW"
      />
    </div>
    <p>
      Your Name:
      <input name="name" type="text" id="name" value="Bassel" />
      <br />
      Your age:
      <input name="age" type="text" id="age" value="40" />
      <br />
      <input type="submit" value="submit" />
      <br />
      Hello Bassel
      <br />
      You are 40 years old
      <br />
    </p>
    <div>
      <input type="hidden" name="__EVENTVALIDATION"
        id="__EVENTVALIDATION"
        value="/wEWAwKL1uvQCgL7uPQdAtCCr6oGMa08C59NSjo1aHXJ1cGerJ0RLFk=" />
    </div>
  </form>
</body></html>

```

تختلف هذه الوثيقة عن النسخة الأصلية من ex3.aspx في ثلاثة مواضع:

1. تحوي عنصر التحكم المخفي ViewState والذي يحوي بيانات النموذج بشكل مشفر.
2. يكون للنموذج اسم ومعرف (ctl100).
3. تم استبدال كتلة الإعادة البرمجية بخرجها.

يُمكن طلب الإرجاع (أي ارسال قيم عناصر النموذج إلى المخدم) postback من قبل المستخدم بعدة طرق. فبالطبع، يحدث الإرجاع عند نقر المستخدم على زر الإرسال submit. كما أنه يحدث عند النقر على أي زر أمر. كذلك يُمكن للمبرمج تحديد طلب الإرجاع لمختلف عناصر التحكم كصناديق الخيار مثلاً وذلك بإسناد القيمة true إلى الخاصية AutoPostBack، وعندها يتم الإرجاع عند تغيير قيمة العنصر.

أحداث الصفحة Page-Level Events

يُنشئ الصف Page أربعة أحداث للصفحة:

- الحدث Init والذي يُرفع مباشرةً بعد إنشاء المنتسخ صف الوثيقة.
- الحدث Load والذي يُرفع بعد إسناد حالة المنتسخ من بيانات النموذج و ViewState.
- الحدث PreRender والذي يُرفع قبل تنفيذ المنتسخ لبناء الوثيقة الجواب للزبون.
- الحدث Unload والذي يُرفع قبل مسح المنتسخ.

يكفي لكتابة معالجات هذه الأحداث التصريح عن الإجراءات باستخدام أسمائها المعرفة مسبقاً والمسجلة ضمناً عند إنشاء صف الوثيقة.

مثال:

يُبين المثال التالي التصريح عن الأحداث الأربع السابقة وتسلسل وقوعها:

```
<%@ Page language="c#" %>

<script runat="server">

    protected void Page_Load(object sender, EventArgs e)
    {
        T.Value = T.Value + " Load ";
    }
    protected void Page_Unload(object sender, EventArgs e)
    {
        T.Value = T.Value + " Unload ";
    }
    protected void Page_PreRender(object sender, EventArgs e)
    {
        T.Value = T.Value + " PreRender ";
    }
    protected void Page_Init(object sender, EventArgs e)
    {
        T.Value = T.Value + " Init ";
    }
</script>

<html xmlns="http://www.w3.org/1999/xhtml" >
  <head> <title> Page Events </title>
  </head>
  <body>
    <form runat = "server" id="MyForm" >
      Page Events Sequence:
      <input id="T" runat="server" />
    </form>
  </body>
</html>
```

حيث يكون الخرج:

Page Events Sequence: Init Load PreRender

يوجد طريقتين لإنشاء وتسجيل معالج حدث لعنصر HTML. تُسند في الطريقة الأولى اسم إجرائية معالج الحدث إلى الوصفة OnServerClick أو الوصفة OnServerChange. يكون لإجرائية معالج الحدث البروتوكول **البروتوكول** وهو البروتوكول للتفويض (delegate) EventHandler والذي يُعرّف المنهج القياسي لمعالجة الأحداث في لغة التنفيذ الموحدة CLR التالي:

- تُعيد void.
- يوجد لها معاملي دخل الأول من النوع object، والثاني من النوع System.EventArgs.

أحداث عناصر التحكم Control Events

مثال:

يُبين المثال التالي الطريقة الأولى لتسجيل معالج الحدث.

```
protected void TextHandler (object sender, EventArgs e)
```

```
{
...
}
```

```
<input type="text" id="Name"
    OnServerChange="TextHandler"
    Runat="server" />
```

تعتمد الطريقة الثانية على كتابة وتسجيل الحدث باستخدام المنهج القياسي في CLR، والذي يستخدم التفويض لتسجيل معالج الحدث. حيث تتبع الخطوات الثلاث التالية:

- كتابة معالج الحدث والذي هو عبارة عن وظيفة تُعيد void، ولها معاملي دخل من النوع object و EventArgs.
- إنشاء منتسخ جديد من النمط EventHandler باستخدام new وتمرير اسم الوظيفة السابقة للبانى.
- تسجيل منتسخ التفويض السابق إلى خاصية حدث عنصر التحكم بإضافته إلى المنتسخات المسجلة سابقاً.

يتم عادة دمج الخطوتين السابقتين في تعليمة واحدة توضع في معالج الحدث Page_Init كما يُبين المثال التالي:

```
protected void TextboxHandler(object sender, EventArgs e)
```

```
{
...
}
```

```

protected void Page_Init(object sender, EventArgs e)
{
    Name.ServerChange +=
        New EventHandler(TextBoxHandler);
}
...
<input type="text" id="Name" runat="server" />

```

مثال:

يُبين المثال التالي استخدام الطريقتين السابقتين وتسلسل وقوع الأحداث:

```

<%@ Page language="c#" %>
<script runat="server">
    protected void Page_Load(object sender, EventArgs e)
    {
        T.Value = T.Value + " Load ";
    }
    protected void Page_Unload(object sender, EventArgs e)
    {
        T.Value = T.Value + " Unload ";
    }
    protected void Page_PreRender(object sender, EventArgs e)
    {
        T.Value = T.Value + " PreRender ";
    }
    protected void Page_Init(object sender, EventArgs e)
    {
        T.Value = T.Value + " Init ";
        T2.ServerChange += new EventHandler(T2Handler);
    }

    protected void T1Handler(object sender, EventArgs e)
    {
        T.Value = T.Value + " T1 Handler ";
        T1.Style.Add(HtmlTextWriterStyle.BackgroundColor, "red");
    }

    protected void T2Handler(object sender, EventArgs e)
    {
        T.Value = T.Value + " T2 Handler ";
        T2.Style.Add(HtmlTextWriterStyle.BackgroundColor, "blue");
    }
}
</script>
<html xmlns="http://www.w3.org/1999/xhtml" >
    <head> <title> Control Events </title>
    </head>
    <body>
        <form runat = "server" id="MyForm" >

```

```

        <br/>
        <input type="text" id="T1" onserverchange="T1Handler"
runat="server" />
        <br/>

        <input type="text" id="T2" runat="server" />
        <br/>
        <input type = "submit" value = "submit" />
        <br/>
        Events Sequence:
        <input id="T" runat="server" style="width: 424px" />
    </form>
</body>
</html>

```

حيث يظهر بعد النقر على الزر :submit

The screenshot shows a web form with two text input fields, T1 (red) and T2 (blue), and a submit button. Below the form, an 'Events Sequence' box displays the sequence: Init Load PreRender Load T1 Handler T2 Handler PreRender.

عناصر التحكم Web Controls

الأهداف التعليمية

- التعرف على عناصر التحكم Web Controls.

عناصر تحكم الويب Web Controls

يتضمن الفضاء System.Web.UI.WebControls عناصر تحكم الويب وهي مجموعة أكبر وأغنى من عناصر التحكم HTML. فبالإضافة إلى عناصر التحكم التي توافقت العناصر العادية XHTML، يوجد عناصر عديدة أخرى.

فمثلاً، يوجد عناصر تحكم لقوائم صناديق الاختيار checkbox list، قوائم أزرار الخيار radio button lists، القوائم المنسدلة drop-down lists، القوائم list boxes. كذلك يوجد عناصر تحكم خاصة للربط مع قواعد البيانات وللتحقق من صحة القيم المدخلة.

يُبين الجدول التالي أهم عناصر تحكم الويب والتقابل، إن وجد، مع عناصر XHTML:

Web Control	XHTML Element
<ASP:HyperLink>	<a>...
<ASP:LinkButton>	<a>
<ASP:Image>	
<ASP:Panel>	<div>...</div>
<ASP:Label>	...
<ASP:Button>	<input type="submit"/> or <input type="button"/> or <input type="reset"/>
<ASP:TextBox>	<input type="text"/> or <input type="password"/> or <textarea>...</textarea>
<ASP:CheckBox>	<input type="checkbox"/>
<ASP:RadioButton>	<input type="radio"/>
<ASP:ImageButton>	<input type="image"/>
<ASP:Table>	<table>...</table>
<ASP:TableRow>	<tr>...</tr>
<ASP:TableCell>	<td>...</td>
AdRotator	None
Calender	None
CheckListBox	None
RadioButtonList	None

توجد جميع عناصر تحكم الويب ضمن الفضاء asp، وبالتالي فيجب أن تبدأ جميع الأسماء المؤثرات بـ asp.:
 فمثلاً، لتعريف مربع نص نكتب:

```
<asp:textbox id="phone" runat="server" />
```

توفر عناصر تحكم الويب أسلوب برمجي أكثر تجانساً من عناصر HTML، وهي تُعاد render للزبون كتركيب من عناصر التحكم XHTML.

تملك عناصر تحكم الويب مجموعة من الخصائص والطرق والأحداث. يُبين الجدول التالي أهم الخصائص المشتركة والموروثة من الصف WebControls:

Property	Description
Attributes	إعادة جميع الثنائيات (خاصية، قيمة الخاصية)
AccessKey	إعادة مفتاح الاختصار الذي يساعد على اختيار العنصر
BackColor	لون الخلفية
Border Color	لون الإطار
BorderStyle	نمط الإطار
BorderWidth	عرض الإطار
ClientID	معرف العنصر المعطى
Controls	عناصر التحكم الأبناء التابعة للعنصر الحالي
Enabled	مفعّل أم لا
EnableViewState	قيمة منطقية تُحدد فيما إذا العنصر سيُحافظ على قيمة ViewState لهذا العنصر ولجميع أبنائه
Font	الخط
ForeColor	اللون
Height	الارتفاع

Id	معرف العنصر
Page	مؤشر إلى الصفحة التي تحوي العنصر
Parent	مؤشر لأب العنصر
Style	نمط العنصر
TabIndex	تسلسل ترتيب العنصر
ToolTip	نص المساعدة
Visible	مرئي
Width	العرض

يُبين الجدول التالي أهم الطرق المشتركة والموروثة من الصف WebControls:

Method	Description
DataBind	تفعيل الربط بمصدر البيانات
FindControl	البحث ضمن الكائن الحالي عن عنصر تحكم معين
HasControls	قيمة منطقية فيما إذا كان لعنصر التحكم أبناء

كذلك تملك عناصر تحكم الويب خصائص وطرق مميزة لكل عنصر. يُبين الجدول التالي أهمها:

Control	Properties	Events
HyperLink	ImageUrl, NavigateUrl, Target, Text	-
LinkButton	CommandArgument, CommandName, Text, CausesValidation	OnClick, OnCommand
Image	AlternateText, ImageAlign, ImageUrl	-
Panel	BackImageUrl, HorizontalAlign, Wrap	-
Label	Text	-
Button	CommandArgument, CommandName, Text, CausesValidation	OnClick, OnCommand
TextBox	AutoPostBack, Columns, MaxLength, ReadOnly, Rows, Text, TextMode, Wrap	OnTextChanged
CheckBox	AutoPostBack, Checked, Text, TextAlign	OnCheckChanged
RadioButton	AutoPostBack, Checked, GroupName, Text, TextAlign	OnCheckChanged
ImageButton	CommandArgument, CommandName, CausesValidation	OnClick, OnCommand
Table	BackImageUrl, CellPadding, CellSpacing, GridLines, HorizontalAlign, Rows	-
TableRow	Cells, HorizontalAlign, VerticalAlign	-
TableCell	ColumnSpan, HorizontalAlign, RowSpan, Text, VerticalAlign, Wrap	-
Literal	Text	-
Placeholder	-	-

إنشاء عناصر التحكم ضمن الكود

يُمكن إنشاء عناصر تحكم الويب إما بتعريفها مباشرة:

```
<asp:button id="helpButton" Text="help"
    OnClick="OnClickHandler"
    Runat="server" />
```

أو بإنشائها ضمن كود C#:

```
protected Button helpButton= new Button();
helpButton.Text="help";
helpButton.id=" helpButton";
helpButton.OnClick="OnClickHandler ";
helpButton.runat=" server";
```

يُمكن عند إنشاء عناصر الويب برمجياً استخدام عنصر التحكم placeholder وذلك لتحديد مكان توضع عناصر التحكم. فمثلاً، يُمكن أولاً إنشاء عنصر تحكم placeholder:

```
<asp:placeholder id="buttonPlace" runat="server" />
```

ووضع الزر المنشئ داخله باستخدام التعليمة:

```
buttonPlace.Controls.Add(helpButton);
```

خرج عناصر التحكم

يُمكن استخدام الطريقة Response.Write لإظهار خرج. إلا أنه من الأفضل، وللتحكم بموضع الإخراج، استخدام عنصر التحكم Label للإظهار ضمنه وذلك بإسناد الخرج إلى الخاصية Text. لنفرض مثلاً أن الوثيقة تحوي العنصر Label التالي في الموضع الذي نريد إظهار الخرج فيه:

```
<asp:label id="output" runat="server" />
```

تقوم التعليمات التالية بوضع نص معين في العنصر السابق:

```
<% string msg= string.Format(
    "The result is {0} <br />", result);
output.Text=msg ; %>
```

مثال 1:

يقوم المثال التالي بإنشاء صندوق نص TextBox وقائمة منسدلة Drop-Down List وزر أمر Button في وثيقة ASP.NET. كما يستخدم الكود الخلفي لملء عناصر القائمة المنسدلة. تحوي الوثيقة أيضاً عنصر تحكم Label لتأمين مكان وضع الرسالة الصادرة عن الكود الخلفي. يحوي الكود الخلفي معالج لحدث النقر على زر الأمر لكتابة نص معين حسب العنصر المختار من القائمة.

تتم إضافة عنصر إلى القائمة المنسدلة باستخدام الطريقة Add على الخاصية Items للقائمة. كما يتم إنشاء كل عنصر باستدعاء باني الصف ListItem مع تمرير قيمة العنصر. فمثلاً، لإضافة عنصر إلى القائمة mySelect مع القيمة "red":

```
mySelect.Items.Add(new ListItem("red"))
```

```
<!-- ex4.aspx
An example of an ASP.NET document that creates a textbox,
a drop-down list, a submit button, and a label.
A code-behind file is used to populate the drop-down list and
handle the button clicks. The label is used for the return
message
-->
<%@ Page language="c#" Inherits = "ex4" Src = "ex4.aspx.cs" %>

<html xmlns="http://www.w3.org/1999/xhtml">
  <head> <title> Ex4 </title>
  </head>
  <body>
    <form id="Form1" runat = "server">
      Name: <asp:TextBox runat = "server" id = "name" />
      <br /><br />
      Favorite Color:<asp:DropDownList runat = "server"
                          id = "color" />

      <br /><br />
      <asp:button runat = "server" id = "submit"
                  text = "Submit" OnClick = "OnClickHandler" />

      <br /><br />
      <asp:label id = "message" runat = "server" />
    </form>
  </body>
</html>
```

وملف الكود الخلفي. :aspx.cs

```
// The code behind file for ex4.aspx.
// In an OnLoad handler, it populates the drop-down
// list created in the associated ASP.NET document.
// It also includes a handler for the button, which
// produces a message to the client, including the
// client's name and the chosen item from the drop
// down list

using System;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```

public class ex4 : System.Web.UI.Page
{
    protected DropDownList color;
    protected TextBox name;
    protected Button submit;
    protected Label message;

    // OnLoad handler to populate the dropdownlist

    override protected void OnLoad(EventArgs e)
    {
        if (!IsPostBack)
        {
            color.Items.Add(new ListItem("blue"));
            color.Items.Add(new ListItem("red"));
            color.Items.Add(new ListItem("green"));
            color.Items.Add(new ListItem("yellow"));
        }
    }

    // Handler for the button

    protected void OnClickHandler(object src, EventArgs e)
    {
        string newMsg = string.Format(
            "Hi {0}, your favorite color is {1}",
            name.Text, color.SelectedItem);
        message.Text = newMsg;
    }
}

```

حيث يُظهر المتصفح:

وبعد اختيار عنصر من القائمة المنسدلة والنقر على زر الأمر:

يستخدم المثال التالي مجموعة من عناصر التحكم الأساسية لتحقيق نموذج تسجيل بيانات.

```
<%@ Page Language="C#" %>
<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml" >
  <head id="Head1" runat="server">
    <title>Web Controls Demonstration</title>
  </head>
  <body>
    <form id="form1" runat="server">
      <div>
        <h3>This is a sample registration form.</h3>
        <p><em>Please fill in all fields and click Register.</em></p>
        <p>
          <asp:Image ID="UserInformationImage" runat="server"
            ImageUrl="~/Images/user.png"
            EnableViewState="False" />
          &nbsp;
          <span style="color: teal">
            Please fill out the fields below.</span>
        </p>
        <table>
          <tr>
            <td style="width: 230px; height: 21px"
              valign="top">
              <asp:Image ID="FirstNameImage" runat="server"
                ImageUrl="~/Images/fname.png"
                EnableViewState="False" />
              <asp:TextBox ID="FirstNameTextBox"
                runat="server"
                EnableViewState="False"></asp:TextBox>
            </td>
            <td style="width: 231px; height: 21px"
              valign="top">
              <asp:Image ID="LastNameImage" runat="server"
                ImageUrl="~/Images/lname.png"
                EnableViewState="False" />
              <asp:TextBox ID="LastNameTextBox" runat="server"
                EnableViewState="False"></asp:TextBox>
            </td>
          </tr>
          <tr>
            <td style="width: 230px" valign="top">
              <asp:Image ID="EmailImage" runat="server"
                ImageUrl="~/Images/email.png"
                EnableViewState="False" />
              <asp:TextBox ID="EmailTextBox" runat="server"
                EnableViewState="False"></asp:TextBox>
            </td>
            <td style="width: 231px" valign="top">
              <asp:Image ID="PhoneImage" runat="server"
                ImageUrl="~/Images/phone.png"
                EnableViewState="False" />
              <asp:TextBox ID="PhoneTextBox" runat="server"
                EnableViewState="False"></asp:TextBox>
              Must be in the form (555) 555-5555.
            </td>
          </tr>
        </table>
      </div>
    </form>
  </body>
</html>
```



```

<p>
  <asp:Image ID="PublicationsImage" runat="server"
    ImageUrl="~/Images/publications.png"
    EnableViewState="False" />
  &nbsp;
  <span style="color: teal">
    Which book would you like information about?</span>
</p>
<p>
  <asp:DropDownList ID="BooksDropDownList"
    runat="server"
    EnableViewState="False">
    <asp:ListItem>Visual Basic 2013 How to Program 3e
  </asp:ListItem>
    <asp:ListItem>Visual C# 2013 How to Program 2e
  </asp:ListItem>
    <asp:ListItem>Java How to Program 6e</asp:ListItem>
    <asp:ListItem>C++ How to Program 5e</asp:ListItem>
    <asp:ListItem>XML How to Program 1e</asp:ListItem>
  </asp:DropDownList>
</p>
<p>
  <asp:HyperLink ID="BooksHyperLink" runat="server"
    NavigateUrl="http://www.deitel.com" Target="_blank"
    EnableViewState="False">
    Click here to view more information about our books
  </asp:HyperLink>
</p>
<p>
  <asp:Image ID="OSImage" runat="server"
    ImageUrl="~/Images/os.png"
    EnableViewState="False"/>
  &nbsp;
  <span style="color: teal">
    Which operating system are you using?</span>
</p>
<p>
  <asp:RadioButtonList
    ID="OperatingSystemRadioButtonList"
    runat="server" EnableViewState="False">

    <asp:ListItem>Windows 8</asp:ListItem>
    <asp:ListItem>Windows 7</asp:ListItem>
    <asp:ListItem >Windows XP</asp:ListItem>
    <asp:ListItem>Linux</asp:ListItem>
    <asp:ListItem>Other</asp:ListItem>
  </asp:RadioButtonList>
</p>
<p>
  <asp:Button ID="RegisterButton" runat="server"
    Text="Register" EnableViewState="False" />
</p>
</div>
</form>
</body>
</html>

```

This is a sample registration form.

Please fill in all fields and click Register.

User Information Please fill out the fields below.

First Name Bassel **Last Name** ALKHATIB

Email t_balkhatib@svuonline.org **Phone**

Must be in the form (555) 555-5555.

Publications Which book would you like information about?

Visual Basic 2013 How to Program 3e

[Click here to view more information about our books](#)

Operating System Which operating system are you using?

Windows 8

Windows 7

Windows XP

Linux

Other

مثال 3:

نقوم في المثال التالي ببناء صفحة تستخدم تستخدم عناصر Web Controls. ستكون البرمجة طبعاً من جهة المخدم أي باستخدام C#. تهدف الصفحة التالية إلى حساب سعر الإقامة في فندق وذلك وفق مجموعة من الخيارات التي يُمكن أن يقوم بها المستخدم:

- تصنيف الفندق: *، **، ***، ****، *****.
- إطلالة الفندق: البحر، الغابة، الجبل.
- الوجبات المطلوبة: فطور، غداء، عشاء.
- خدمات إضافية: تلفزيون، مسبح، مطبخ.
- خدمات إضافية: تلفزيون، مسبح، مطبخ.

نتكن مثلاً الأسعار التالية المعتمدة:

Stars		Where		Food		Extra	
Stars	Price	Sea	+50%	Breakfast	50	TV	50
*	100	Forest	+30%	Lunch	100	Pool	100
**	200	Mountain	+20%	Dinner	200	Kitchen	150
***	300						
****	400						
*****	500						

نقوم في هذه المسألة بتهيئة الأسعار في قيم عناصر Web Controls للرجوع إليها في إجرائية C# المستخدمة لحساب سعر الليلة.

```

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server"><title>Web Controls</title></head>
<body bgcolor="#ccffcc">
<form id="form1" runat="server">
<div>
<table border="1">
<caption>Hotel Club (Web Controls)</caption>
<tr>
<td style="width: 100px">Country:</td>
<td style="width: 100px">
<asp:DropDownList ID="lstCty" runat="server" AutoPostBack="True"
OnSelectedIndexChanged="lstCty_SelectedIndexChanged"
Style="position: relative">
<asp:ListItem Value="images\usa.png">USA</asp:ListItem>
<asp:ListItem Value="images\france.png">France</asp:ListItem>
<asp:ListItem Value="images\germany.png">Germany</asp:ListItem>
<asp:ListItem Value="images\italy.png">Italy</asp:ListItem>
</asp:DropDownList></td><td style="width: 100px">
<asp:Image ID="imgFlag" runat="server" ImageUrl="~/images/usa.png"
Style="position: relative" /></td></tr><tr>
<td style="width: 100px">Stars:</td>
<td style="width: 100px">
<asp:DropDownList ID="lstStars" runat="server" Style="position: relative">
<asp:ListItem Value="100">*</asp:ListItem>
<asp:ListItem Value="200">**</asp:ListItem>
<asp:ListItem Value="300">***</asp:ListItem>
<asp:ListItem Value="400">****</asp:ListItem>
<asp:ListItem Value="500">*****</asp:ListItem>
</asp:DropDownList></td></tr>
<tr>
<td style="width: 100px">Where:</td>
<td style="width: 100px">
<asp:RadioButtonList ID="lstWhere" runat="server" Style="position:
relative">
<asp:ListItem Value="1.5" Selected="True">Sea</asp:ListItem>
<asp:ListItem Value="1.3">Mountain</asp:ListItem>
<asp:ListItem Value="1.2">Forest</asp:ListItem>
</asp:RadioButtonList></td><td style="width: 100px"></td></tr>
<tr><td style="width: 100px">Meals:</td>
<td style="width: 100px">

```

```

<asp:CheckBoxList ID="lstFood" runat="server" Style="position: relative">
<asp:ListItem Value="50" Selected="True">Breakfast</asp:ListItem>
<asp:ListItem Value="100" Selected="True">Lunch</asp:ListItem>
<asp:ListItem Value="200">Dinner</asp:ListItem>
</asp:CheckBoxList></td><td style="width: 100px"></td></tr>
<tr>
<td style="width: 100px">Extra:</td>
<td style="width: 100px">
<asp:ListBox ID="lstExtra" runat="server" Style="position: relative"
SelectionMode="Multiple">
<asp:ListItem Value="50">TV</asp:ListItem>
<asp:ListItem Value="100">Pool</asp:ListItem>
<asp:ListItem Value="150">Kitchen</asp:ListItem></asp:ListBox></td>
<td style="width: 100px"></td></tr>
<tr>
<td style="width: 100px">
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click"
Style="position: relative" Text="How Much" /></td>
<td style="width: 100px">
<asp:TextBox ID="txtOut" runat="server" Style="position:
relative">0</asp:TextBox></td>
<td style="width: 100px"></td></tr></table></div></form>
</body>
</html>

```

توجد أعلام الدول في المجلد images. تكون قيمة كل خيار من قائمة البلدان مسار ملف صورة علم البلد. تقوم إجرائية المخدم التالية والمكتوبة على حدث تغيير العنصر المحدد من قائمة البلاد بإسناد قيمة عنصر القائمة (والذي يحوي اسم ملف العلم) إلى الخاصية imageUrl لعنصر الصورة.

```

protected void lstCty_SelectedIndexChanged(object sender, EventArgs e)
{
    this.imgFlag.imageUrl = this.lstCty.SelectedValue;
}

```

تقوم إجرائية المخدم التالية بحساب سعر الإقامة استناداً لخيارات المستخدم. تعود هذه الوظيفة إلى قيمة كل عنصر لمعرفة سعر الخيار المحدد.

```

protected void Button1_Click(object sender, EventArgs e)
{
    double x = 0;

    x = double.Parse(this.lstStars.SelectedValue);

    x = double.Parse(this.lstWhere.SelectedValue) * x;

    for (int i = 0; i < this.lstFood.Items.Count; i++)
        if (this.lstFood.Items[i].Selected)
            x = x + double.Parse(this.lstFood.Items[i].Value);
    for (int i = 0; i < this.lstExtra.Items.Count; i++)
        if (this.lstExtra.Items[i].Selected)
            x = x + double.Parse(this.lstExtra.Items[i].Value);

    this.txtOut.Text = x.ToString();
}

```

Hotel Club (Web Controls)

Country:	Italy ▾	
Stars:	*** ▾	
Where:	<input checked="" type="radio"/> Sea <input type="radio"/> Mountain <input type="radio"/> Forest	
Meals:	<input checked="" type="checkbox"/> Breakfast <input checked="" type="checkbox"/> Lunch <input checked="" type="checkbox"/> Dinner	
Extra:	TV Pool Kitchen	
How Much	950	

عناصر التحقق من الصحة

الأهداف التعليمية

- استخدام عناصر التحقق من الصحة.

عناصر التحقق من الصحة

استعرضنا سابقاً كيفية كتابة تعليمات JavaScript للتحقق من صحة القيم المدخلة على الزبون. يوجد العديد من الأسباب التي تستدعي إعادة التحقق من صحة القيم المدخلة على المخدم لاسيما أنه يُمكن لمستخدم مراوغ تجنب عمليات التحقق من جهة الزبون. يوجد ستة عناصر تحكم ويب للتحقق من صحة القيم المدخلة. يُبين الجدول التالي العناصر الأربعة الأكثر استخداماً:

Control	Properties	Values
RequiredFieldValidator	None	None
CompareValidator	Operator	Equal, NotEqual, GreaterThan, GreaterThanEqual, LessThan, LessThanEqual, DataTypeCheck
	Type	String, Currency, Date, Double, Integer
	ValueToCompare	Constant
	ControlToCompare	Another Control
RangeValidator	MaximumValue	Constant
	MinimumValue	Constant
	Type	String, Currency, Date, Double, Integer
RegularExpressionValidator	ValidationExpression	Regular Expression

يتمّ وضع عناصر التحقق من الصحة مباشرةً بعد العناصر التي نريد التحقق من قيمها، وذلك كي تظهر رسائل الخطأ الموافقة جانبها.

يتمّ كتابة رسالة الخطأ في الخاصية ErrorMessage لعنصر التحقق من الصحة. كما يتمّ الربط بين عنصر التحقق من الصحة وعنصر التحكم المطلوب التحقق من قيمه بوضع معرفه في الخاصية ControlToValidate لعنصر التحقق.

مثال:

يُبين المثال التالي استخدام بعض عناصر التحقق من الصحة حيث يكون حقل الاسم في النموذج واجب الإدخال، ورقم الهاتف من الشكل ddd-ddd-dddd، والعمر محصور بين 10 و100.

```
<!-- ex5.aspx
An example of an ASP.NET document to illustrate server-side
validation Web controls.
It uses Web control textboxes to get the name, phone number,
and age of the client. These three are validated on the
server
1. The name must be present
2. The phone number must be in the form ddd-ddd-dddd
3. The range of the age must be 10 to 110
-->
<%@ Page language="c#" %>
<html>
  <head> <title> Ex5 </title>
</head>
  <body>
    <form id="Form1" runat = "server">
      <p>
        Your name:
        <asp:TextBox id = "name" runat = "server" />
        <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
          ControlToValidate = "name"
          Display = "Static"
          runat = "server"
          ErrorMessage = "Please enter your name">
        </asp:RequiredFieldValidator>
        <br />
        Your phone number:
        <asp:TextBox id = "phone" runat = "server" />
        <asp:RegularExpressionValidator
          ID="RegularExpressionValidator1"
          ControlToValidate = "phone"
          Display = "Static"
          runat = "server"
          ErrorMessage = "Phone number form must be ddd-ddd-dddd"
          ValidationExpression = "\d{3}-\d{3}-\d{4}">
        </asp:RegularExpressionValidator>
        <br />
        Your age:
        <asp:TextBox id = "age" runat = "server" />
        <asp:RangeValidator ID="RangeValidator1"
          ControlToValidate = "age"
          Display = "Static"
          runat = "server"
          MaximumValue = "110"
          MinimumValue = "10"
          Type = "Integer"
          ErrorMessage = "Age must be in the range of 10 to 110">
        </asp:RangeValidator>
        <br />
        <input type = "submit" value = "Submit" />
      </p>
    </form>
  </body>
</html>
```


عند إدخال قيم غير محققة لأحد الشروط تظهر رسالة خطأ موافقة بجانب عنصر التحكم المعني:

Your name:	<input type="text"/>	Please enter your name
Your phone number:	<input type="text" value="2222"/>	Phone number form must be ddd-ddd-dddd
Your age:	<input type="text" value="140"/>	Age must be in the range of 10 to 110
<input type="button" value="Submit"/>		

ولا تظهر رسائل خطأ عند إدخال قيم موافقة للشروط:

Your name:	<input type="text" value="Bassel"/>
Your phone number:	<input type="text" value="111-222-3333"/>
Your age:	<input type="text" value="41"/>
<input type="button" value="Submit"/>	

الفصل الحادي عشر: إدارة الحالة

الكلمات المفتاحية

متغيرات الحالة View state variables، متغيرات الحالة Session variables، الكعكات Cookies.

الملخص

نستعرض في هذا الفصل أساسيات إدارة الحالة.

الأهداف التعليمية

يتعرف الطالب في هذا الفصل على:

- التعامل مع متغيرات الحالة View state variables.
- التعامل مع متغيرات الحالة Session variables.
- التعامل مع الكعكات Cookies.

المخطط

يضم فصل إدارة الحالة 3 وحدات (Learning Objects) هي:

- متحولات الحالة View State
- الحفاظ على حالة الجلسة
- الكعكات

متحولات الحالة View State

الأهداف التعليمية

- استخدام متحولات الحالة من النوع View State.

متحولات الحالة View State

- تسمح هذه المتحولات بتخزين قيم والحفاظ عليها من استدعاء لاستدعاء آخر لنفس الصفحة.
- تسمح المجموعة ViewState (collection) والتي هي عبارة عن أزواج مفتاح/قيمة بتخزين قيم واسترجاعها في الاستدعاءات التالية للصفحة نفسها.
- لا يُمكن الوصول لهذه القيم من صفحات أخرى.

مثال:

يُبين المثال التالي استخدام تخزين قيم صناديق النص في نموذج واسترجاعها حين الحاجة.

Description	Value
Name:	<input type="text" value="Bassel"/>
ID:	<input type="text" value="1965"/>
Age:	<input type="text" value="45"/>
E-mail:	<input type="text" value="Bassel@gmail.com"/>

نقوم بحفظ قيم كل صناديق النص في الإجراءية التالية (زر Save):

```
protected void cmdSave_Click(object sender, EventArgs e)
{
    // Save the current text.
    ViewState["Name"] = Name.Text;
    ViewState["EmpID"] = EmpID.Text;
    ViewState["Age"] = Age.Text;
    ViewState["Email"] = Email.Text;
}
```

كما نقوم باسترجاع القيم في الإجراءية التالية (زر Restore):

```
protected void cmdRestore_Click(object sender, EventArgs e)
{
    // Retrieve the last saved text.
    Name.Text=ViewState["Name"].ToString ();
    EmpID.Text = ViewState["EmpID"].ToString();
    Age.Text = ViewState["Age"].ToString();
    Email.Text = ViewState["Email"].ToString();
}
```

الحفاظ على حالة الجلسة

الأهداف التعليمية

- التعامل مع متغيرات الجلسة Session variables.

ملاحظة الجلسات Session Tracking

نحتاج في العديد من الحالات لتخزين بعض البيانات المدخلة من قبل المستخدم. يوجد بشكل أساسي أسلوبين لتخزين هذه البيانات:

- إما في ملفات خاصة على حاسب المستخدم (ندعو هذه الملفات بالكعكات Cookies) ولفترة زمنية معينة.
- أو في متغيرات على المخدم لمدة معينة ولطالما الجلسة مفتوحة (ندعو هذه المتغيرات بمتغيرات الجلسة (Session Variables)).

متغيرات الجلسة Session Variables

تُستخدم متغيرات الجلسة لتخزين بيانات معينة لمستخدم في جلسة مفتوحة و**لمدة معينة** عادةً القيمة الافتراضية 20 دقيقة

مثال:

نقوم في المثال التالي وفي كل مرة يختار فيها المستخدم عنوان كتاب بتخزين هذا الخيار في متغير جلسة وذلك في إجرائية الزر submit.

Select a programming language:

- Visual Basic 2013
- Visual C# 2013
- C
- C++
- Java

Submit

يكون الملف Options.aspx في هذا المثال:

```
<%-- Options.aspx --%>
<%-- Allows client to select programming languages and access --%>
<%-- book recommendations. --%>
<%@ Page Language="C#" AutoEventWireup="true"
    CodeFile="Options.aspx.cs" Inherits="Options" %>

<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml" >
  <head id="Head1" runat="server">
    <title>Sessions</title>
  </head>
  <body>
    <form id="form1" runat="server">
      <div>
```

```

<asp:Label ID="promptLabel" runat="server" Font-Bold="True"
    Font-Size="Large" Text="Select a programming language:">
</asp:Label>

<asp:RadioButtonList ID="languageList" runat="server">
    <asp:ListItem>Visual Basic 2013</asp:ListItem>
    <asp:ListItem>Visual C# 2013</asp:ListItem>
    <asp:ListItem>C</asp:ListItem>
    <asp:ListItem>C++</asp:ListItem>
    <asp:ListItem>Java</asp:ListItem>
</asp:RadioButtonList>

<asp:Button ID="submitButton" runat="server" Text="Submit"
OnClick="submitButton_Click" />

<asp:Label ID="responseLabel" runat="server" Font-Bold="True"
    Font-Size="Large" Text="Welcome to cookies!"
    Visible="False"></asp:Label><br /><br />

<asp:Label ID="idLabel" runat="server" Visible="False">
</asp:Label><br /><br />

<asp:Label ID="timeoutLabel" runat="server" Visible="False">
</asp:Label><br /><br />

<asp:HyperLink ID="languageLink" runat="server"
    Visible="False" NavigateUrl="~/Sessions/Options.aspx">
    Click here to choose another language
</asp:HyperLink><br /><br />

<asp:HyperLink ID="recommendationsLink" runat="server"
    Visible="False"
NavigateUrl="~/Sessions/Recommendations.aspx">
    Click here to get book recommendations</asp:HyperLink>
</div>
</form>
</body>
</html>

```

: Options.aspx.cs وملف الكود الخلفي المرتبط به

```

using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
public partial class Options : System.Web.UI.Page
{
    // stores values to represent books as cookies
    private System.Collections.Hashtable books =
        new System.Collections.Hashtable();

    //initializes the Hashtable of values to be stored as cookies
    protected void Page_Init( object sender, EventArgs e )

```

```

{
    books.Add( "Visual Basic 2013", "0-13-186900-0" );
    books.Add( "Visual C# 2013", "0-13-152523-9" );
    books.Add( "C", "0-13-142644-3" );
    books.Add( "C++", "0-13-185757-6" );
    books.Add( "Java", "0-13-148398-6" );
} // end method Page_Init

// if postback, hide form and display links to make additional
// selections or view recommendations
protected void Page_Load( object sender, EventArgs e )
{
    if ( IsPostBack )
    {
        // user has submitted information,
        //     so display appropriate labels
        // and hyperlinks
        responseLabel.Visible = true;
        idLabel.Visible = true;
        timeoutLabel.Visible = true;
        languageLink.Visible = true;
        recommendationsLink.Visible = true;

        // hide other controls used to make language selection
        promptLabel.Visible = false;
        languageList.Visible = false;
        submitButton.Visible = false;

        // if the user made a selection, display it in responseLabel
        if ( languageList.SelectedItem != null )
            responseLabel.Text += " You selected " +
                languageList.SelectedItem.Text.ToString();
        else
            responseLabel.Text += " You did not select a language.";

        // display session ID
        idLabel.Text = "Your unique session ID is: " +
            Session.SessionID;

        // display the timeout
        timeoutLabel.Text = "Timeout: " +
            Session.Timeout + " minutes.";
    } // end if
} // end method Page_Load

// write a cookie to record the user's selection
protected void submitButton_Click( object sender, EventArgs e )
{
    // if the user made a selection
    if ( languageList.SelectedItem != null )
    {
        string language = languageList.SelectedItem.ToString();

        // get ISBN number of book for the given language
        string ISBN = books[ language ].ToString();
        // add name/value pair to Session
        Session.Add( language, ISBN );    } // end if
    } // end method submitButton_Click
} // end class Options

```


بعد النقر على زر الإرسال يتم إظهار الرابطين التاليين:
الأول للعودة إلى إظهار قائمة الخيارات واختيار كتاب آخر.
الثاني للانتقال إلى صفحة إظهار بعض المعلومات حول جميع الكتب المختارة سابقاً من قبل المستخدم
(والمخزنة في متغيرات الجلسة). كما يتم إظهار معرف الجلسة ومدتها.

Welcome to sessions! You selected Visual C# 2013

Your unique session ID is: n2po5qeyxhifo4maowmeilo1

Timeout: 20 minutes.

[Click here to choose another language](#)

[Click here to get book recommendations](#)

عند النقر على الرابط الثاني يتم الانتقال إلى الصفحة التالية والتي تقوم بإظهار معلومات عن الكتب المخزنة في
متغيرات الجلسة:

Recommendations

```
Java How to Program. ISBN#: 0-13-148398-6
Visual C# 2013 How to Program. ISBN#: 0-13-152523-9
```

[Click here to choose another language](#)

حيث يكون الملف Recommendations.aspx:

```
<%-- Recommendations.aspx --%>
<%-- Displays book recommendations using cookies. --%>
<%@ Page Language="C#" AutoEventWireup="true"
    CodeFile="Recommendations.aspx.cs" Inherits="Recommendations" %>

<!DOCTYPE html >

<html xmlns="http://www.w3.org/1999/xhtml" >
  <head id="Head1" runat="server">
    <title>Book Recommendations</title>
  </head>
```

```

<body>
  <form id="form1" runat="server">
    <div>
      <asp:Label ID="recommendationsLabel"
        runat="server" Text="Recommendations"
        Font-Bold="True" Font-Size="X-Large">
      </asp:Label><br /><br />

      <asp:ListBox ID="booksListBox" runat="server"
        Height="125px"
        Width="450px"></asp:ListBox><br /><br />

      <asp:HyperLink ID="languageLink" runat="server"
        NavigateUrl="~/Sessions/Options.aspx">
        Click here to choose another language
      </asp:HyperLink>
    </div>
  </form>
</body></html>
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class Recommendations : System.Web.UI.Page
{
  // read cookies and populate ListBox with any book recommendations
  protected void Page_Init( object sender, EventArgs e )
  {
    // stores a key name found in the Session object
    string keyName;

    // determine whether Session contains any information
    if ( Session.Count != 0 )
    {
      for ( int i = 0; i < Session.Count; i++ )
      {
        keyName = Session.Keys[ i ]; // store current key name

        // use current key to display one
        // of session's name-value pairs
        booksListBox.Items.Add( keyName +
          " How to Program. ISBN#: " +
          Session[ keyName ].ToString() );
      } // end for
    } // end if
    else
    {
      // if there are no session items, no language was chosen, so
      // display appropriate message and clear and hide booksListBox
      recommendationsLabel.Text = "No Recommendations";
      booksListBox.Items.Clear();
      booksListBox.Visible = false;

      // modify languageLink because no language was selected

```

```
        languageLink.Text = "Click here to choose a language";
    } // end else
} // end method Page_Init
} // end class Recommendations
```

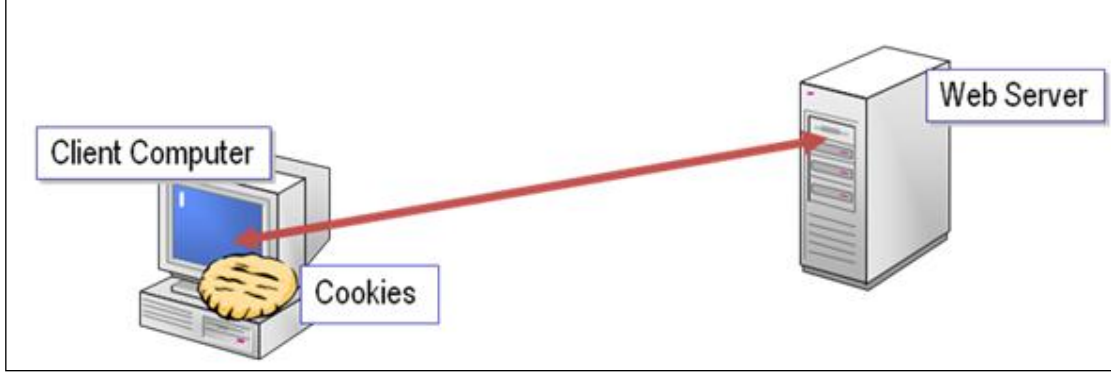
الكعكات

الأهداف التعليمية

- التعامل مع الكعكات Cookies.

الكعكات Cookies

تُستخدم الكعكات لتخزين بيانات معينة في ملفات خاصة على حاسب المستخدم وذلك للمحافظة على هذه البيانات خلال الجلسة وبين الجلسات المختلفة للمستخدم. يُمكن أن تُستخدم الكعكات مثلاً لتخزين تفضيلات مستخدم عند زيارته لموقع تسوق للمرة الأولى. حيث سيتم استخدام هذه التفضيلات في الزيارات التالية للموقع لتغيير ترتيب إظهار السلع مثلاً.



مثال:

نقوم في المثال التالي وفي كل مرة يختار فيها المستخدم عنوان كتاب بتخزين هذا الخيار في كعكة وذلك في إجرائية الزر submit.

Select a programming language:

- Visual Basic 2013
- Visual C# 2013
- C
- C++
- Java

Submit

يكون الملف Options.aspx في هذا المثال:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Options.aspx.cs" Inherits="Options" %>

<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml" >
  <head id="Head1" runat="server">
    <title>Cookies</title>
  </head>
  <body>
    <form id="form1" runat="server">
      <div>
        <asp:Label ID="promptLabel" runat="server" Font-Bold="True"
Font-Size="Large" Text="Select a programming language:">
```

```

        </asp:Label>

        <asp:RadioButtonList ID="languageList" runat="server">
            <asp:ListItem>Visual Basic 2013</asp:ListItem>
            <asp:ListItem>Visual C# 2013</asp:ListItem>
            <asp:ListItem>C</asp:ListItem>
            <asp:ListItem>C++</asp:ListItem>
            <asp:ListItem>Java</asp:ListItem>
        </asp:RadioButtonList>

        <asp:Button ID="submitButton" runat="server" Text="Submit"
OnClick="submitButton_Click" />

        <asp:Label ID="responseLabel" runat="server" Font-Bold="True"
Font-Size="Large" Text="Welcome to cookies!"
Visible="False"></asp:Label><br /><br />

        <asp:HyperLink ID="languageLink" runat="server"
Visible="False" NavigateUrl="~/Cookies/Options.aspx">
Click here to choose another language
</asp:HyperLink><br /><br />

        <asp:HyperLink ID="recommendationsLink" runat="server"
Visible="False"
NavigateUrl="~/Cookies/Recommendations.aspx">
Click here to get book recommendations</asp:HyperLink>
    </div>
</form>
</body>
</html>

```

: Options.aspx.cs وملف الكود الخلفي المرتبط به

```

// Options.aspx.cs
// Processes user's selection of a programming language
// by displaying links and writing a cookie to the user's machine.
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
public partial class Options : System.Web.UI.Page
{
    // stores values to represent books as cookies
    private System.Collections.Hashtable books =
        new System.Collections.Hashtable();

    // initializes the Hashtable of values to be stored as cookies
    protected void Page_Init( object sender, EventArgs e )
    {
        books.Add( "Visual Basic 2013", "0-13-186900-0" );
        books.Add( "Visual C# 2013", "0-13-152523-9" );
        books.Add( "C", "0-13-142644-3" );
        books.Add( "C++", "0-13-185757-6" );
    }
}

```

```

        books.Add( "Java", "0-13-148398-6" );
    } // end method Page_Init

    // if postback, hide form and display links to make additional
    // selections or view recommendations
    protected void Page_Load( object sender, EventArgs e )
    {
        if ( IsPostBack )
        {
            // user has submitted information, so display message
            // and appropriate hyperlinks
            responseLabel.Visible = true;
            languageLink.Visible = true;
            recommendationsLink.Visible = true;

            // hide other controls used to make language selection
            promptLabel.Visible = false;
            languageList.Visible = false;
            submitButton.Visible = false;

            // if the user made a selection, display it in responseLabel
            if ( languageList.SelectedItem != null )
                responseLabel.Text += " You selected " +
                    languageList.SelectedItem.Text.ToString();
            else
                responseLabel.Text += " You did not select a language.";
        } // end if
    } // end method Page_Load

    // write a cookie to record the user's selection
    protected void submitButton_Click( object sender, EventArgs e )
    {
        // if the user made a selection
        if ( languageList.SelectedItem != null )
        {
            string language = languageList.SelectedItem.ToString();

            // get ISBN number of book for the given language
            string ISBN = books[ language ].ToString();

            // create cookie using language-ISBN name-value pair
            HttpCookie cookie = new HttpCookie( language, ISBN );
            cookie.Expires = DateTime.Now.AddHours( 2 );
            // add cookie to response to place it on the user's machine
            Response.Cookies.Add( cookie );
        } // end if
    } // end method submitButton_Click
} // end class Options

```

بعد النقر على زر الإرسال يتم إظهار الرابطين التاليين:
الأول للعودة إلى إظهار قائمة الخيارات واختيار كتاب آخر.
الثاني للانتقال إلى صفحة إظهار بعض المعلومات حول جميع الكتب المختارة سابقاً من قبل المستخدم
(والمخزنة في الكعكة).

Welcome to cookies! You selected Visual C# 2005

[Click here to choose another language](#)

[Click here to get book recommendations](#)

عند النقر على الرابط الثاني يتم الانتقال إلى الصفحة التالية والتي تقوم بإظهار معلومات عن الكتب المخزنة في الكعكة:

Recommendations

Visual C# 2013 How to Program. ISBN#: 0-13-152523-9

[Click here to choose another language](#)

حيث يكون الملف Recommendations.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Recommendations.aspx.cs" Inherits="Recommendations" %>

<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml" >
  <head id="Head1" runat="server">
    <title>Book Recommendations</title>
  </head>
  <body>
    <form id="form1" runat="server">
      <div>
        <asp:Label ID="recommendationsLabel"
          runat="server" Text="Recommendations"
          Font-Bold="True" Font-Size="X-Large">
        </asp:Label><br /><br />

        <asp:ListBox ID="booksListBox" runat="server"
          Height="125px"
          Width="450px"></asp:ListBox><br /><br />

        <asp:HyperLink ID="languageLink" runat="server"
          NavigateUrl="~/Cookies/Options.aspx">
          Click here to choose another language
        </asp:HyperLink>
      </div>
    </form>
  </body>
</html>
```



```
// Recommendations.aspx.cs
// Creates book recommendations based on cookies.
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class Recommendations : System.Web.UI.Page
{
    // read cookies and populate ListBox with any book recommendations
    protected void Page_Init( object sender, EventArgs e )
    {
        // retrieve client's cookies
        HttpCookieCollection cookies = Request.Cookies;

        // if there are cookies,
        // list the appropriate books and ISBN numbers

        if ( cookies.Count > 0 )
        {
            for ( int i = 0; i < cookies.Count; i++ )
                booksListBox.Items.Add( cookies[ i ].Name +
                    " How to Program. ISBN#: " + cookies[ i ].Value );
        } // end if
        else
        {
            // if there are no cookies, then no language was chosen, so
            // display appropriate message and
            // clear and hide booksListBox
            recommendationsLabel.Text = "No Recommendations";
            booksListBox.Items.Clear();
            booksListBox.Visible = false;

            // modify languageLink because no language was selected
            languageLink.Text = "Click here to choose a language";
        } // end else
    } // end method Page_Init
} // end class Recommendations
```

الفصل الثاني عشر:

ربط عناصر التحكم مع البيانات

الكلمات المفتاحية

ربط عناصر التحكم مع البيانات، مصدر بيانات قيمة واحدة، مصدر بيانات قيم متعددة.

الملخص

نستعرض في هذا الفصل آليات ربط البيانات مع عناصر التحكم المختلفة.

الأهداف التعليمية:

يتعرف الطالب في هذا الفصل على:

- ربط البيانات.
- الربط مع قيمة وحيدة.
- الربط مع قيم متعددة.

المخطط

يضم فصل ربط عناصر التحكم مع البيانات وحدتين (Learning Objects) هي:

- ربط عناصر التحكم مع البيانات
- الربط مع قيم متعددة

ربط عناصر التحكم مع البيانات

الأهداف التعليمية

- ربط عناصر التحكم مع مصدر بيانات قيمة واحدة.

مقدمة

تحتاج معظم مواقع الويب إلى التعامل مع البيانات. تؤمن ASP.NET مجموعة من الطرق للتعامل مع البيانات من جميع الأنواع (قواعد البيانات العلائقية، الملفات النصية، ملفات XML، مخدمات البريد الإلكتروني). كما تؤمن مجموعة من عناصر التحكم من جهة المخدم التي يُمكن استخدامها لإظهار البيانات وتحريرها.

ربط البيانات

يعتمد مبدأ ربط البيانات على إدراج قيمة أو عدة قيم من مصدر بيانات على الوثيقة أو ضمن عنصر تحكم على الوثيقة. أي أننا نُميز بين حالتين من ربط البيانات:

• ربط بيانات قيمة وحيدة

وهي حالة ربط قيمة وحيدة مع عنصر تحكم. كإسناد قيمة إلى خاصية عنصر تحكم مثلاً (مثل `Text` أو `Value`). تُستخدم هذه الحالة عادةً مع العناصر التي لها قيمة وحيدة مثل: `asp:TextBox`, `asp:HyperLink`, `<Input>`.

• ربط بيانات قيم متعددة

وهي حالة ربط مجموعة من القيم من مصدر بيانات مع عنصر تحكم يُمكنه إظهار أكثر من قيمة. كالقوائم مثلاً. تُستخدم هذه الطريقة عادةً مع العناصر مثل: `asp:ListBox`, `asp:CheckBoxListBox`, `<SELECT>`.

وفي كلا الحالتين سنستخدم تقانات متشابهة.

يُستخدم الشكل التالي للدلالة على صيغة ربط بيانات:

```
<%# name-of-data-source %>
```

الربط مع قيمة وحيدة

يُمكن ربط عنصر تحكم مع قيمة وحيدة. يُمكن أن يكون مصدر هذه القيمة أحد الحالات التالية:

- قيمة خاصة مصرح عنها في الصفحة أو من عنصر تحكم أو من غرض آخر. ونستخدم الصيغة التالية:

```
<%# property-name %>
```

- القيمة المعادة من طريقة مصرح عنها في الصفحة أو من عنصر تحكم أو من غرض آخر. ونستخدم الصيغة التالية:

```
<%# method-name (parameter1, parameter2,...) %>
```

- القيمة المعادة نتيجة حساب تعبير. ونستخدم الصيغة التالية:

```
<%# expression %>
```

يتم تفعيل الربط باستخدام الطريقة DataBind والمُعرفة على مستوى الصفحة Page وعلى مستوى عناصر التحكم. يُمكن استدعاء هذه الطريقة مثلاً على حدث تحميل الصفحة Page_Load لتفعيل الربط أو على زر أمر مثلاً.

مثال:

نقوم بربط:

- الخاصية Text لعنصر Label بالخاصية Test المعرفة في كود صف الصفحة.
- الخاصية ImageUrl لعنصر Image بالخاصية SelectedValue لعنصر DropDownList.
- الخاصية Text لعنصر Label باستدعاء للطريقة GCD والمعرفة في كود الصفحة (والتي تحسب القاسم المشترك الأعظم لعددين).
- الخاصية Value لعنصر <INPUT> مع تعبير حسابي.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="SimpleBind.aspx.cs"
Inherits="SimpleBind" %>
<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Simple Data Binding</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<table style="position: relative" border="1">
<caption>
    <span style="color: #0000cc">Single-Value Data Binding</span> </caption>
<tr>
    <td style="width: 100px">Property</td>
    <td style="width: 100px">
        <asp:Label ID="Label1" runat="server"
            Text="<%# Test %>" Style="position: relative">
</asp:Label></td>
    <td style="width: 55px"></td></tr>
<tr>
    <td style="width: 100px">Property</td>
    <td style="width: 100px">
        <asp:DropDownList ID="lstCty" runat="server" Style="position: relative">
            <asp:ListItem Value="images\usa.png">USA</asp:ListItem>
            <asp:ListItem Value="images\france.png">France</asp:ListItem>
            <asp:ListItem Value="images\germany.png">Germany</asp:ListItem>
            <asp:ListItem Value="images\italy.png">Italy</asp:ListItem>
        </asp:DropDownList></td>
    <td style="width: 55px">
        <asp:Image ID="imgCty" runat="server"
            ImageUrl="<%# lstCty.SelectedValue %>"
            Style="position: relative" /></td></tr>
<tr>
    <td style="width: 100px">GCD Method</td>
    <td style="width: 100px">
        <asp:Label ID="lblGCD" runat="server"
            Style="position: relative">

```

```

        Text="<%# GCD(int.Parse(T1.Text), int.Parse(T2.Text)) %>">
</asp:Label></td>
    <td style="width: 55px">
    <asp:TextBox ID="T1" runat="server"
        Style="position: relative">1</asp:TextBox>
    <asp:TextBox ID="T2" runat="server"
        Style="position: relative">1</asp:TextBox></td></tr><tr>
    <td style="width: 100px">Expression</td>
    <td style="width: 100px">
    <input type="text" id="htmlinput"
        runat="server" style="position: relative"
        value="<%# 3+4*5 %>" /></td>
    <td style="width: 55px"></td></tr>
<tr>
    <td style="width: 100px">
    <asp:Button ID="Button1" runat="server"
        OnClick="Button1_Click" Style="position: relative"
        Text="Bind" /></td>
    <td style="width: 100px"></td>
    <td style="width: 55px"></td></tr></table></div>
</form>
</body>
</html>

```

نقوم باستدعاء الطريقة DataBind على مستوى الصفحة Page في الحدث Page_Load. أما على الزر Bind فنقوم باستدعاء هذه الطريقة فقط على العنصرين الذين قد تتغير قيمهما من قبل المستخدم (نتيجة تغيير البلد المختار من قائمة البلاد أو إدخال عددين جديدين في صندوق النص المخصصين لحساب القاسم المشترك الأعظم لهما).

يكون كود صف الصفحة الموافق:


```

public partial class SimpleBind : System.Web.UI.Page
{
    public string Test
    {
        get { return "Test"; }
    }
    public static int GCD(int n, int m)
    {
        while (n != m)
            if (n > m)
                n = n - m;
            else
                m = m - n;
        return n;
    }
    protected void Page_Load(object sender, EventArgs e)
    {
        Page.DataBind();
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        lblGCD.DataBind();
        imgCty.DataBind();
    }
}

```

يُبين الشكل التالي مظهر الصفحة بعد اختيار بلد وإدخال رقمين ومن ثم النقر على الزر Bind:

Single-Value Data Binding

Property	Test	
Property	France ▼	
GCD Method	12	36 24
Expression	23	
<input type="button" value="Bind"/>		

الربط مع قيم متعددة

الأهداف التعليمية

ربط عناصر التحكم مع بيانات قيم متعددة.

الربط مع عدة قيم

توفر .NET ثمانية عناصر تحكم للتعامل مع القيم المتعددة حيث تملك هذه العناصر مجموعة من الخصائص والطرق للاتصال بمصادر البيانات المتعددة:

- <SELECT>
- ListBox
- DropDown
- CheckBoxList
- RadioButtonList
- Repeater
- DataList
- DataGrid

تختلف طريقة الربط مع البيانات وفق طبيعة عنصر التحكم:

- إذا كان العنصر لا يدعم القوالب Templates، فيتم إسناد الحقول في مصدر البيانات إلى خصائص العنصر.
 - أما إذا كان العنصر يدعم القوالب (Repeater, DataList)، فيتم التصريح عن قالب لكل صف سيُظهره عنصر التحكم.
- يُبين الجدول التالي أهم خصائص الربط مع مصدر البيانات:

الوصف	الخاصية
تُحدد حقل أو عمود مصدر البيانات الذي سيتم إظهاره في العنصر.	DataTextField
تُحدد حقل أو عمود مصدر البيانات الذي سيتم إسناده إلى قيمة Value العنصر.	DataValueField
تُحدد تنسيق القيم الظاهرة في العنصر. مثلاً: {0: dddd MMMM dd yyyy} for Date {0: C} for Currency	DataTextFormatString
تُحدد مجموعة الصفوف التي سيتم الارتباط بها عندما يحوي مصدر البيانات على أكثر من مجموعة صفوف. مثلاً: اسم الجدول عند الربط مع غرض DataSet.	DataMember

أما أهم طريقة فهي الطريقة `DataBind()` والتي تقوم بتفعيل الربط بين عنصر التحكم ومصدر البيانات.
أما أهم الأحداث فهي:

الوصف	الحدث
يظهر هذا الحدث من أجل كل صف في مصدر البيانات. يتم تمرير محتوى الصف إلى الحدث ضمن معاملات الحدث حيث يُمكن فحص وتعديل بيانات الصف أثناء ملء عنصر التحكم بالبيانات.	DataBinding
يظهر هذا الحدث بعد تغيير الصف المختار في العنصر ومن ثم إرسال بيانات الصفحة إلى المخدم.	SelectedIndexChanged

يُمكن ربط عناصر تحكم القوائم إلى:

- **Collection**: مثل مجموعة الجداول في الغرض `DataSet` مثلاً.
- **ArrayList**: والتي تحوي عادةً مجموعة من القيم.
- **Hashtable**: يكون لكل عنصر مفتاح `key` وقيمة `value` (يتم الوصول عادة للعنصر عن طريق المفتاح) مما يجعلها مناسبة في حالة القوائم حيث يختلف عادة النص المراد إظهاره عن القيمة.
- **DataGridView**: يحوي هذا الغرض على صفوف من الغرض `DataTable`. يُمكن تعبئتها من قاعدة بيانات، أو إنشائها وتعبئتها برمجياً.
- **DataReader**: يوفر هذا الغرض اتصال سريع للقراءة وباتجاه واحد مع قاعدة بيانات. يحوي عادةً مجموعة صفوف.

مثال: الربط مع `ArrayList`

```
protected void Page_Load(object sender, EventArgs e)
{
    //create an ArrayList of values to bind to
    ArrayList arrValues = new ArrayList();
    arrValues.Add("Microsoft");
    arrValues.Add("Sun");
    arrValues.Add("IBM");
    arrValues.Add("Compaq");
    arrValues.Add("Oracle");

    //set the DataSource property of the controls to the array
    MySelectList.DataSource = arrValues;
    MyDropDown.DataSource = arrValues;
    MyASPList.DataSource = arrValues;
    MyDataGrid.DataSource = arrValues;
    MyRepeater.DataSource = arrValues;
    MyDataList.DataSource = arrValues;
    MyCheckList.DataSource = arrValues;
    MyRadioList.DataSource = arrValues;

    //bind all the controls on the page
}
```

```
Page.DataBind();
}
```

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="RepeatBind1.aspx.cs" Inherits="RepeatBind1" %>
<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Repeated-Value Data Binding</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<table style="position: relative" cellpadding="5" cellspacing="5">
<caption >
    <span style="color: #0000cc">Repeated-Value Data Binding 1 </span>
</caption>
    <tr><td >
        HTML <b>&lt;select&gt;</b> element:<br />
        <b>&lt;select id="MySelectList" runat="server" /><p />
        <b>&lt;ASP:DropDownList&gt;</b> control:<br />
        <b>&lt;asp:DropDownList ID="MyDropDown" runat="server" /><p />
        <b>&lt;ASP:ListBox&gt;</b> control:<br />
        <b>&lt;asp:ListBox ID="MyASPList" runat="server" /><p />
        <b>&lt;ASP:DataGrid&gt;</b> control:<br />
        <b>&lt;asp:DataGrid ID="MyDataGrid" runat="server" /><p />
        <b>&lt;ASP:Repeater&gt;</b> control:<br />
        <b>&lt;asp:Repeater ID="MyRepeater" runat="server">
            <ItemTemplate>
                <%# Container.DataItem %>
            </ItemTemplate>
        </asp:Repeater>
        <p /></td>
        <td valign="top" >
            <b>&lt;ASP:DataList&gt;</b> control:<br />
            <b>&lt;asp:DataList ID="MyDataList" runat="server">
                <ItemTemplate>
                    <%# Container.DataItem %>
                </ItemTemplate>
            </asp:DataList><p />
            <b>&lt;ASP:CheckBoxList&gt;</b> control:<br />
            <b>&lt;asp:CheckBoxList ID="MyCheckList" runat="server" /><p />
            <b>&lt;ASP:RadioButtonList&gt;</b> control:<br />
            <b>&lt;asp:RadioButtonList ID="MyRadioList" runat="server" />
            <p /> </td></tr></table></div>
    </form>
</body>
</html>
```

Repeated-Value Data Binding 1

HTML <select> element:

Microsoft ▼

<ASP:DropDownList> control:

Microsoft ▼

<ASP:ListBox> control:

Microsoft ▲
Sun
IBM
Compaq ▼

<ASP:DataGrid> control:

Item
Microsoft
Sun
IBM
Compaq
Oracle

<ASP:Repeater> control:

Microsoft Sun IBM Compaq Oracle

<ASP:DataList> control:

Microsoft
Sun
IBM
Compaq
Oracle

<ASP:CheckBoxList> control:

Microsoft
 Sun
 IBM
 Compaq
 Oracle

<ASP:RadioButtonList> control:

Microsoft
 Sun
 IBM
 Compaq
 Oracle

مثال: الربط مع Hashtable

```
protected void Page_Load(object sender, EventArgs e)
{
    //create a HashTable of values to bind to
    Hashtable tabValues = new Hashtable();
    tabValues.Add("Microsoft", 49.56);
    tabValues.Add("Sun", 28.33);
    tabValues.Add("IBM", 55);
    tabValues.Add("Compaq", 20.74);
    tabValues.Add("Oracle", 41.1);
    //first <select> displays the Keys in the HashTable
    MySelectList1.DataSource = tabValues;
    MySelectList1.DataTextField = "key";
    //second one displays the Values in the HashTable
    //and uses the Keys as the <option> values
    MySelectList2.DataSource = tabValues;
    MySelectList2.DataValueField = "key";
    MySelectList2.DataTextField = "value";
    //same applies to ASP: controls, except here
    //we can also specify the format of the Key
    MyDropDown1.DataSource = tabValues;
    MyDropDown1.DataTextField = "key";
    MyDropDown2.DataSource = tabValues;
    MyDropDown2.DataValueField = "key";
}
```

```

MyDropDown2.DataTextField = "value";
MyDropDown2.DataTextFormatString = "{0:F}";
MyASPList1.DataSource = tabValues;
MyASPList1.DataTextField = "key";
MyASPList2.DataSource = tabValues;
MyASPList2.DataValueField = "key";
MyASPList2.DataTextField = "value";
MyASPList2.DataTextFormatString = "{0:C}";
MyDataGrid.DataSource = tabValues;
MyRepeater.DataSource = tabValues;
MyDataList.DataSource = tabValues;
//in the CheckBoxList we'll display the Title and
//use the Value as the control value
MyCheckBoxList.DataSource = tabValues;
MyCheckBoxList.DataValueField = "value";
MyCheckBoxList.DataTextField = "key";
//in the RadioList we'll display and format the
//Value and use the Key as the control value
MyRadioList.DataSource = tabValues;
MyRadioList.DataValueField = "key";
MyRadioList.DataTextField = "value";
MyRadioList.DataTextFormatString = "Percentage rate {0:F}%";
Page.DataBind(); //bind all the controls on the page
}

```

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="RepeatBind2.aspx.cs" Inherits="RepeatBind2" %>
<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Repeat Binding 2</title>
</head>
<body>
<form id="form1" runat="server">
<div >
<table style="position: relative; left: 0px; top: 0px;" cellpadding="5"
cellspacing="5" title="Repeated-Value Data Binding 2"><caption >
<span style="color: #0000cc">Repeated-Value Data Binding 2
</span></caption><tr><td >
HTML <b>&lt;select&gt;</b> elements:<br />
<select id="MySelectList1" runat="server" />&nbsp;
<select id="MySelectList2" runat="server" /><p />
<b>&lt;ASP:DropDownList&gt;</b> controls:<br />
<asp:DropDownList ID="MyDropDown1" runat="server" />&nbsp;
<asp:DropDownList ID="MyDropDown2" runat="server" /><p />
<b>&lt;ASP:ListBox&gt;</b> controls:<br />
<asp:ListBox ID="MyASPList1" runat="server" />&nbsp;
<asp:ListBox ID="MyASPList2" runat="server" /><p />
<b>&lt;ASP:DataGrid&gt;</b> control:<br />
<asp:DataGrid ID="MyDataGrid"
runat="server"AutoGenerateColumns="false">
<Columns>
    <asp:BoundColumn HeaderText="key" DataField="key" />
    <asp:BoundColumn HeaderText="value" DataField="value"
DataFormatString="{0:C}" />
</Columns>
</asp:DataGrid><p />
<b>&lt;ASP:Repeater&gt;</b> control:<br />
<asp:Repeater ID="MyRepeater" runat="server">

```

```

<ItemTemplate>
<%# DataBinder.Eval(Container.DataItem, "key") %>
=
<%# DataBinder.Eval(Container.DataItem, "value") %><br />
</ItemTemplate>
</asp:Repeater><p />
</td><td valign="top" >
<b>&lt;ASP:DataList&gt;</b> control:<br />
<asp:DataList ID="MyDataList" runat="server">
<ItemTemplate>
'<%# DataBinder.Eval(Container.DataItem, "key") %>' - value:
<%# DataBinder.Eval(Container.DataItem, "Value", "{0:E}") %>
</ItemTemplate>
</asp:DataList><p />
<b>&lt;ASP:CheckBoxList&gt;</b> control:<br />
<asp:CheckBoxList ID="MyCheckList" runat="server" /><p />
<b>&lt;ASP:RadioButtonList&gt;</b> control:<br />
<asp:RadioButtonList ID="MyRadioList" runat="server" /><p />
</td></tr></table></div></form>
</body>
</html>

```

Repeated-Value Data Binding 2

<p>HTML <select> elements:</p> <div style="border: 1px solid #ccc; padding: 5px; display: inline-block;"> Compaq 20.74 </div> <p><ASP:DropDownList> controls:</p> <div style="border: 1px solid #ccc; padding: 5px; display: inline-block;"> Compaq 20.74 </div> <p><ASP:ListBox> controls:</p> <table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>Compaq</td><td>€20.74</td></tr> <tr><td>Microsoft</td><td>€49.56</td></tr> <tr><td>IBM</td><td>€55.00</td></tr> <tr><td>Oracle</td><td>€41.10</td></tr> </table> <p><ASP:DataGrid> control:</p> <table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr><th>key</th><th>value</th></tr> </thead> <tbody> <tr><td>Compaq</td><td>€20.74</td></tr> <tr><td>Microsoft</td><td>€49.56</td></tr> <tr><td>IBM</td><td>€55.00</td></tr> <tr><td>Oracle</td><td>€41.10</td></tr> <tr><td>Sun</td><td>€28.33</td></tr> </tbody> </table> <p><ASP:Repeater> control:</p> <p>Compaq = 20.74 Microsoft = 49.56 IBM = 55 Oracle = 41.1 Sun = 28.33</p>	Compaq	€20.74	Microsoft	€49.56	IBM	€55.00	Oracle	€41.10	key	value	Compaq	€20.74	Microsoft	€49.56	IBM	€55.00	Oracle	€41.10	Sun	€28.33	<p><ASP:DataList> control:</p> <p>'Compaq'- value: 2.074000E+001 'Microsoft'- value: 4.956000E+001 'IBM'- value: 5.500000E+001 'Oracle'- value: 4.110000E+001 'Sun'- value: 2.833000E+001</p> <p><ASP:CheckBoxList> control:</p> <p><input type="checkbox"/> Compaq <input type="checkbox"/> Microsoft <input type="checkbox"/> IBM <input type="checkbox"/> Oracle <input type="checkbox"/> Sun</p> <p><ASP:RadioButtonList> control:</p> <p><input type="radio"/> Percentage rate 20.74% <input type="radio"/> Percentage rate 49.56% <input type="radio"/> Percentage rate 55.00% <input type="radio"/> Percentage rate 41.10% <input type="radio"/> Percentage rate 28.33%</p>
Compaq	€20.74																				
Microsoft	€49.56																				
IBM	€55.00																				
Oracle	€41.10																				
key	value																				
Compaq	€20.74																				
Microsoft	€49.56																				
IBM	€55.00																				
Oracle	€41.10																				
Sun	€28.33																				

الفصل الثالث عشر: التعامل مع قواعد البيانات

الكلمات المفتاحية

الاتصال مع قاعدة بيانات، إنشاء قاعدة بيانات، عناصر تحكم البيانات.

الملخص

نستعرض في هذا آليات التعامل مع قواعد البيانات من النمط SQL Server.

الأهداف التعليمية:

يتعرف الطالب في هذا الفصل على:

- كيفية الاتصال مع قاعدة بيانات.
- إنشاء قاعدة بيانات.
- عناصر تحكم البيانات.
- عرض البيانات في عناصر التحكم.
- استخدام عناصر البيانات.

المخطط

يضم فصل التعامل مع قواعد البيانات 4 وحدات (Learning Objects) هي:

- الاتصال مع قواعد البيانات
- عرض البيانات في عناصر التحكم
- استخدام عناصر البيانات
- ربط معاملات الاستعلام مع عناصر التحكم

الاتصال مع قواعد البيانات

الأهداف التعليمية

- الاتصال مع قاعدة بيانات.

التعامل مع قواعد البيانات

يُمكن في .NET التعامل مع قواعد البيانات باستخدام مجموعة من عناصر التحكم الجاهزة. كما يُمكن التعامل مع قواعد البيانات وتنفيذ العمليات الأساسية عليها برمجياً.

إنشاء قاعدة بيانات

يُمكن في .NET إنشاء قاعدة بيانات من النوع SQL Server Database File. (Add New Item → SQL Database). ينتهي ملف قاعدة البيانات باللاحقة `mdf`، ونقوم بوضعه عادةً في المجلد `App_Data` كي يكون مسار الوصول للقاعدة نسبي.

مثال:

نقوم بإنشاء قاعدة بيانات `Hotel.mdf` تحوي خمسة جداول:

- تُخزن في الجدول `Countries` أسماء البلاد `Country` مع اسم ملف صورة علم كل بلد `Flag` (نضع الصور في المجلد `Images`).
- تُخزن في الجدول `Stars` تصنيف الفندق `Star` وسعر الليلة `Price`.
- تُخزن في الجدول `Places` مكان الفندق `Place` ومعامل زيادة السعر `factor`.
- تُخزن في الجدول `Meals` اسم الوجبة `Meal` وسعرها `Price`.
- تُخزن في الجدول `Extras` الخدمات الإضافية: الخدمة `Extra`، السعر `Price`، متاحة أم لا `Available`.

عناصر تحكم البيانات

توفر .NET مجموعة من عناصر التحكم للاتصال مع قواعد البيانات، وتنفيذ العمليات الأساسية عليها هي:

- الاستعلام `Select`
- الإضافة `Insert`
- الحذف `Delete`
- التعديل `Update`

عنصر مصدر البيانات SqlDataReader

يسمح هذا العنصر من تعريف اتصال مع مصدر بيانات من النوع SQL. عند استخدام عنصر SqlDataReader لأول مرة في المشروع، نختار القاعدة Hotel.mdf. سيُنشئ المعالج تلقائياً سلسلة اتصال Connection String مع قاعدة البيانات. نقوم بإعطاء اسم لهذه السلسلة. تُخزن **سلسلة الاتصال** يُمكن الآن استخدام سلسلة الاتصال هذه في أي عملية وصول لاحق لقاعدة البيانات سواء عبر عناصر التحكم أو برمجياً. مما يعني عملياً أن كل اتصالات المشروع تستخدم هذه السلسلة وبالتالي في حال تعديل مسار قاعدة البيانات (كأن توضع على مخدم آخر مثلاً) يكفي تعديل هذه السلسلة في الملف **Web.Config**. في الملف **Web.Config**.

```
<connectionStrings>
  <add name="ConnectionStringHotel" connectionString="Data
Source=(LocalDB)\v11.0;AttachDbFilename=|DataDirectory|\Hotel.mdf;Integrated
Security=True"
  providerName="System.Data.SqlClient" /></connectionStrings>
```

بعد تحديد سلسلة الاتصال للعنصر SqlDataReader، يتم تحديد الصفوف التي نريد الاستعلام عنها من قاعدة البيانات.

وفي حال أننا سنقوم بإجراء عمليات إضافة أو حذف أو تعديل، فيجب اختيار الزر Advanced للطلب بتوليد عبارات SQL اللازمة.

يُمكن معاينة عبارات SQL المولدة في خصائص العنصر SqlDataReader: SelectQuery, InsertQuery, DeleteQuery, UpdateQuery.

1 الموجود في القائمة Data من صندوق الأدوات Toolbox
2 ليكن مثلاً ConnectionStringHotel

عرض البيانات في عناصر التحكم

الأهداف التعليمية

- عرض البيانات في عناصر التحكم.

عرض البيانات في القوائم

- يُمكن استخدام جميع أنواع القوائم لعرض البيانات.
- نحتاج من أجل كل قائمة إلى عنصر مصدر بيانات يتم من خلاله تحديد حقول الجدول الذي نريد ربط القائمة معه.
- يتمّ الربط بين القائمة ومصدر البيانات باستخدام الخاصية DataSourceID للقائمة.
- يجب تحديد قيمة كل من الخاصيتين التاليتين للقائمة:

DataTextField	حقل الجدول الذي ستملأ قيمه القائمة
DataValueField	حقل الجدول الذي سيُعطي قيمة القائمة

مثال:

ترتبط قائمة البلاد lstCty مع مصدر البيانات (DataSourceID=SqlDataSourceCountry) المرتبط مع الجدول Countries. تُحدد الحقول المرتبطة:

DataTextField	Country
DataValueField	Flag

عند اختيار بلد من القائمة سيكون اسم ملف علم البلد في الخاصية SelectedValue للقائمة. لاحظ أننا كتبنا الإجرائية التالية على القائمة وذلك لإظهار العلم في الصورة imgFlag عند كل تغيير للبلد في القائمة.

```
protected void lstCty_SelectedIndexChanged(object sender, EventArgs e)
{
    this.imgFlag.ImageUrl = "Images\\" + this.lstCty.SelectedValue;
}
```

ترتبط القائمة lstStars مع الجدول Stars عبر العنصر SqlDataSourceStars مع إظهار الحقل Star وإسناد الحقل Price إلى قيمة القائمة.

ترتبط القائمة lstWhere مع الجدول Places عبر العنصر SqlDataSourcePlaces مع إظهار الحقل Place وإسناد الحقل Price إلى قيمة القائمة.


ترتبط القائمة lstFood مع الجدول Meals عبر العنصر SqlDataSourceMeals مع إظهار الحقل Meal وإسناد الحقل Price إلى قيمة القائمة.

ترتبط القائمة lstExtra مع الجدول Extras عبر العنصر SqlDataSourceExtras مع إظهار الحقل Extra وإسناد الحقل Price إلى قيمة القائمة.

مثال:

يكون الملف HotelDB.aspx في هذا المثال:

Hotel Example (DB)

Country:	<input type="text" value="Databound"/> SqlDataSource - SqlDataSourceCountry 	
Stars:	<input type="text" value="Databound"/> SqlDataSource - SqlDataSourceStars	
Where:	<input type="radio"/> Databound SqlDataSource - SqlDataSourcePlaces	
Meals:	<input type="checkbox"/> Databound SqlDataSource - SqlDataSourceMeals	
Extra:	<input type="text" value="Databound"/> SqlDataSource - SqlDataSourceExtras	
How Much	<input type="text" value="0"/>	

[Home](#)

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="HotelDB.aspx.cs"
Inherits="WebServer" %>
<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
  <title>Hotel Example (DB)</title>
</head>
<body >
  <form id="form1" runat="server">
    <div> <table border="1">
      <caption> Hotel Example (DB) &nbsp;  </caption>
<tr>
<td style="width: 100px">Country:</td>
<td style="width: 100px">
<asp:DropDownList ID="lstCty" runat="server" AutoPostBack="True"
OnSelectedIndexChanged="lstCty_SelectedIndexChanged"
Style="position: relative"
DataSourceID="SqlDataSourceCountry" DataTextField="Country"
DataValueField="Flag" OnPreRender="lstCty_PreRender" >
</asp:DropDownList>
<asp:SqlDataSource ID="SqlDataSourceCountry" runat="server"
ConnectionString="<%"$ ConnectionStrings:ConnectionStringHotel %>"
SelectCommand="SELECT * FROM [Countries]"></asp:SqlDataSource></td>
<td style="width: 100px">
<asp:Image ID="imgFlag" runat="server" Style="position: relative"
/></td></tr>
```

```

<tr>
<td style="width: 100px">
Stars:</td>
<td style="width: 100px">
<asp:DropDownList ID="lstStars" runat="server" Style="position: relative"
DataSourceID="SqlDataSourceStars" DataTextField="Star"
DataValueField="Price"></asp:DropDownList>
<asp:SqlDataSource ID="SqlDataSourceStars" runat="server"
ConnectionString="<%"$ ConnectionStrings:ConnectionStringHotel %>"
SelectCommand="SELECT * FROM [Stars]"></asp:SqlDataSource></td>
<td style="width: 100px"></td></tr>
<tr>
<td style="width: 100px">Where:</td>
<td style="width: 100px">
<asp:RadioButtonList ID="lstWhere" runat="server" Style="position:
relative" DataSourceID="SqlDataSourcePlaces" DataTextField="Place"
DataValueField="Factor">
</asp:RadioButtonList>
<asp:SqlDataSource ID="SqlDataSourcePlaces" runat="server"
ConnectionString="<%"$ ConnectionStrings:ConnectionStringHotel %>"
SelectCommand="SELECT * FROM [Places]">
</asp:SqlDataSource></td>
<td style="width: 100px"></td></tr>
<tr>
<td style="width: 100px">Meals:</td>
<td style="width: 100px">
<asp:CheckBoxList ID="lstFood" runat="server" Style="position: relative"
DataSourceID="SqlDataSourceMeals" DataTextField="Meal"
DataValueField="Price"></asp:CheckBoxList>
<asp:SqlDataSource ID="SqlDataSourceMeals" runat="server"
ConnectionString="<%"$ ConnectionStrings:ConnectionStringHotel %>"
SelectCommand="SELECT * FROM [Meals]"></asp:SqlDataSource></td>
<td style="width: 100px"></td></tr>
<tr><td style="width: 100px">Extra:</td>
<td style="width: 100px">
<asp:ListBox ID="lstExtra" runat="server" Style="position: relative"
SelectionMode="Multiple" DataSourceID="SqlDataSourceExtras"
DataTextField="Extra" DataValueField="Price"></asp:ListBox>
<asp:SqlDataSource ID="SqlDataSourceExtras" runat="server"
ConnectionString="<%"$ ConnectionStrings:ConnectionStringHotel %>"
SelectCommand="SELECT * FROM [Extras]"></asp:SqlDataSource>
</td><td style="width: 100px"></td></tr>
<tr><td style="width: 100px">
<asp:Button ID="btnCalc" runat="server" OnClick="btnCalc_Click"
Style="position: relative"Text="How Much" /></td>
<td style="width: 100px">
<asp:TextBox ID="txtOut" runat="server" Style="position:
relative">0</asp:TextBox></td>
<td style="width: 100px"></td></tr>
</table></div>
<asp:HyperLink ID="HyperLink1" runat="server" NavigateUrl="~/Default.aspx"
Style="position: relative">Home</asp:HyperLink></form></body></html>

```

نقوم في الإجرائية Page_Load للصفحة بتحديد أول عنصر من قائمة خيارات الأماكن كخيار افتراضي:

```
protected void Page_Load(object sender, EventArgs e)
{
    if ( ! Page.IsPostBack)
        lstWhere.SelectedIndex = 0;
}
```

كما نقوم في الإجرائية PreRender لعنصر قائمة البلاد بإظهار علم أول بلد:

```
protected void lstCty_PreRender(object sender, EventArgs e)
{
    this.imgFlag.ImageUrl = "Images\\" + this.lstCty.SelectedValue;
}
```

وتصبح إجرائية حساب السعر:

```
protected void btnCalc_Click(object sender, EventArgs e)
{
    double x = 0;
    x = double.Parse(this.lstStars.SelectedValue);


    x = double.Parse(this.lstWhere.SelectedValue) * x;

    for (int i = 0; i < this.lstFood.Items.Count; i++)
        if (this.lstFood.Items[i].Selected)
            x = x + double.Parse(this.lstFood.Items[i].Value);
    for (int i = 0; i < this.lstExtra.Items.Count; i++)
        if (this.lstExtra.Items[i].Selected)
            x = x + double.Parse(this.lstExtra.Items[i].Value);

    this.txtOut.Text = x.ToString();
}
```

يُبين الشكل التالي إستدعاء للصفحة:

Hotel Example (DB)

Country:	France ▾	
Stars:	***** ▾	
Where:	<input checked="" type="radio"/> Sea <input type="radio"/> Mountain <input type="radio"/> Forest	
Meals:	<input checked="" type="checkbox"/> Breakfast <input type="checkbox"/> Lunch <input checked="" type="checkbox"/> Dinner	
Extra:	TV Pool Kitchen	
How Much	1050	

[Home](#)

استخدام عناصر البيانات

الأهداف التعليمية

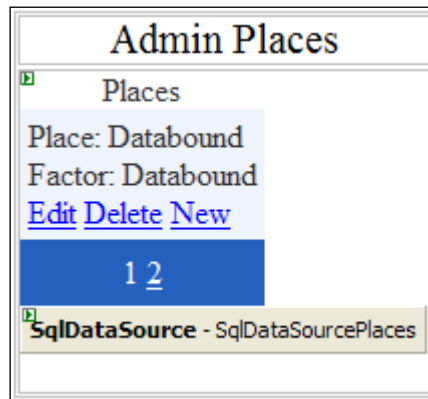
- استخدام Form View
- Grid View
- Details View

استخدام FormView

مثال:

- نقوم في الصفحة التالية AdminPlaces.aspx باستخدام العنصر FormView من أجل إضافة/تعديل/حذف بيانات جدول الأماكن Places.
- يسمح العنصر FormView بعرض البيانات بشكل نموذج. كما يسمح بإضافة وتعديل وحذف البيانات.
- يرتبط العنصر FormView بمصدر بيانات مثل SqlDataReader.

مثال:



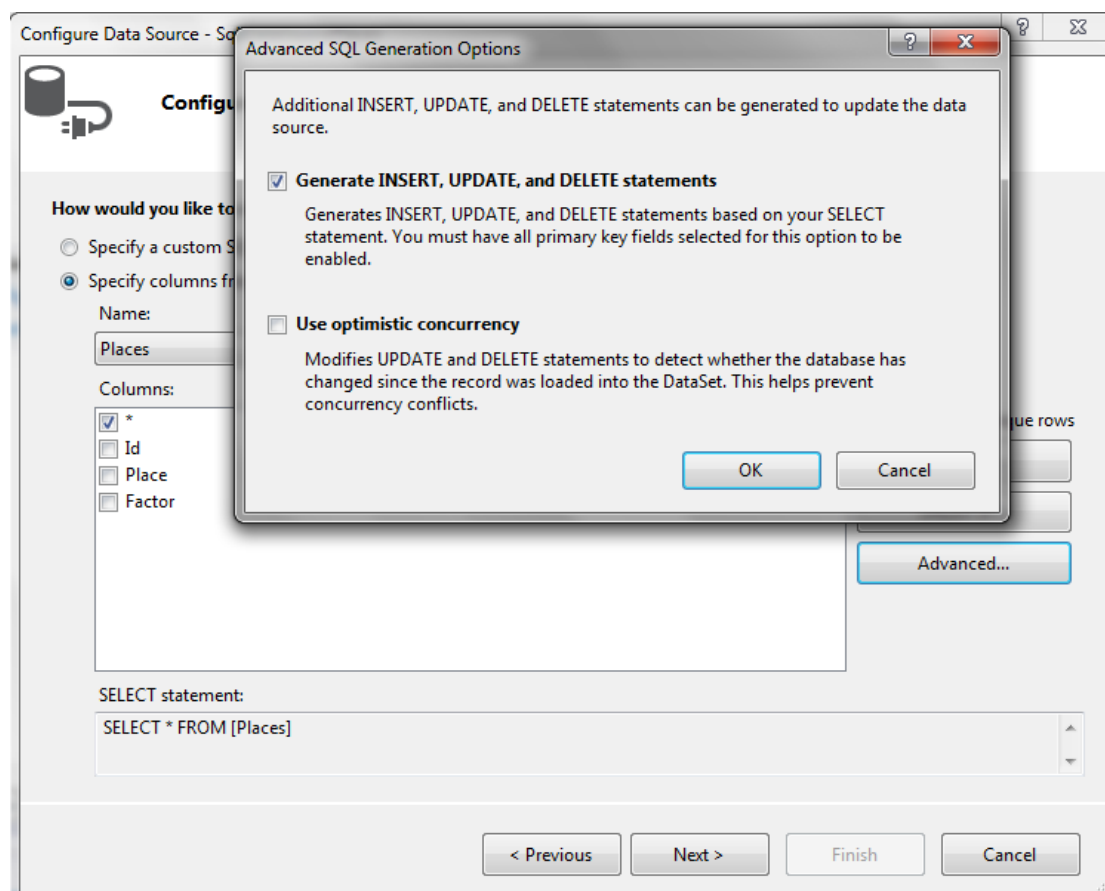
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="AdminPlaces.aspx.cs" Inherits="AdminPlaces" %>
<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server"><title>Admin Places</title></head>
<body>
<form id="form1" runat="server">
<table style="position: relative">
<tr>
<td colspan="2" style="text-align: center">
<span style="font-size: 16pt">Admin Places</span></td></tr>
<tr>
<td colspan="2">
<asp:FormView ID="FormView1" runat="server" AllowPaging="True"
Caption="Places" CellPadding="4"
DataKeyNames="Place" DataSourceID="SqlDataSourcePlaces" ForeColor="#333333"
Style="position: relative">
<FooterStyle BackColor="#507CD1" Font-Bold="True" ForeColor="White" />
<EditRowStyle BackColor="#2461BF" />
<EditItemTemplate>
Place:
<asp:Label ID="PlaceLabel1" runat="server" Text='<%# Eval("Place")
%>'></asp:Label><br />
Factor:
<asp:TextBox ID="FactorTextBox" runat="server" Text='<%# Bind("Factor")
%>'></asp:TextBox><br />
<asp:LinkButton ID="UpdateButton" runat="server" CausesValidation="True"
CommandName="Update"
Text="Update"></asp:LinkButton>
<asp:LinkButton ID="UpdateCancelButton" runat="server"
CausesValidation="False" CommandName="Cancel"
Text="Cancel"></asp:LinkButton></EditItemTemplate>
```

```

<RowStyle BackColor="#EFF3FB" />
<PagerStyle BackColor="#2461BF" ForeColor="White" HorizontalAlign="Center"
/>
<InsertItemTemplate>
Place:
<asp:TextBox ID="PlaceTextBox" runat="server" Text="<%# Bind("Place")
%>"></asp:TextBox><br />
Factor:
<asp:TextBox ID="FactorTextBox" runat="server" Text="<%# Bind("Factor")
%>"></asp:TextBox><br />
<asp:LinkButton ID="InsertButton" runat="server" CausesValidation="True"
CommandName="Insert"
Text="Insert"></asp:LinkButton>
<asp:LinkButton ID="InsertCancelButton" runat="server"
CausesValidation="False" CommandName="Cancel"
Text="Cancel"></asp:LinkButton></InsertItemTemplate>
<ItemTemplate>Place:
<asp:Label ID="PlaceLabel" runat="server" Text="<%# Eval("Place")
%>"></asp:Label><br />Factor:
<asp:Label ID="FactorLabel" runat="server" Text="<%# Bind("Factor")
%>"></asp:Label><br />
<asp:LinkButton ID="EditButton" runat="server" CausesValidation="False"
CommandName="Edit"
Text="Edit"></asp:LinkButton>
<asp:LinkButton ID="DeleteButton" runat="server" CausesValidation="False"
CommandName="Delete"
Text="Delete"></asp:LinkButton>
<asp:LinkButton ID="NewButton" runat="server" CausesValidation="False"
CommandName="New"
Text="New"></asp:LinkButton></ItemTemplate>
<HeaderStyle BackColor="#507CD1" Font-Bold="True" ForeColor="White"
/></asp:FormView>
<asp:SqlDataSource ID="SqlDataSourcePlaces" runat="server"
ConnectionString="<%%$ ConnectionStrings:ConnectionStringHotel %>"
DeleteCommand="DELETE FROM [Places] WHERE [Place] = @Place"
InsertCommand="INSERT INTO [Places] ([Place], [Factor]) VALUES (@Place,
@Factor)"
SelectCommand="SELECT * FROM [Places]" UpdateCommand="UPDATE [Places] SET
[Factor] = @Factor WHERE [Place] = @Place">
<DeleteParameters>
<asp:Parameter Name="Place" Type="String" /></DeleteParameters>
<UpdateParameters><asp:Parameter Name="Factor" Type="Double" />
<asp:Parameter Name="Place" Type="String" /></UpdateParameters>
<InsertParameters>
<asp:Parameter Name="Place" Type="String" />
<asp:Parameter Name="Factor" Type="Double" />
</InsertParameters>
</asp:SqlDataSource></td></tr></table>
<br />
<asp:HyperLink ID="HyperLink1" runat="server" NavigateUrl="~/Default.aspx"
Style="position: relative">Home</asp:HyperLink>
</form>
</body>
</html>

```

يرتبط العنصر sqlDataSource مع الجدول Places مع ملاحظة توليد عبارات Insert, Update, Delete.



ويكون تنفيذ الصفحة:



استخدام GridView و DetailsView

- تقوم في الصفحة التالية AdminStars.aspx باستخدام كل من عنصري البيانات GridView و DetailsView من أجل عرض/إضافة/تعديل/حذف تسجيلات جدول البلاد Countries.
- يسمح عنصر التحكم GridView بعرض البيانات بشكل جدولي. كما يسمح بتعديل أو حذف البيانات (إلا أنه لا يمكن إضافة تسجيلية جديدة باستخدام هذا العنصر). بينما يسمح العنصر DetailsView بإضافة/تعديل/حذف البيانات.
- تقوم بربط العنصرين وبحيث يتم إظهار تفصيل التسجيلية في العنصر DetailsView عند تحديدها في GridView:
 - نربط العنصر GridView بمصدر بيانات يرتبط مع الجدول Stars.
 - نربط العنصر DetailsView بمصدر بيانات آخر يرتبط أيضاً مع الجدول Stars مع تعريف معامل لاستعلام التحديد يأخذ قيمته من القيمة المحددة في العنصر GridView.
- يُمكن إعادة ربط العنصر GridView مع مصدر البيانات في كل مرة يتم فيها إضافة/تعديل/حذف تسجيلية في العنصر DetailsView وذلك باستدعاء الطريقة DataBind() بعد كل إضافة أو تعديل أو حذف من العنصر DetailsView.

مثال:

Admin Stars	
Stars	
Star	Price
Select	Databound Databound
Select	Databound Databound
Select	Databound Databound
Select	Databound Databound
Select	Databound Databound
Stars Details	
Star	Databound
Price	Databound
Edit	Delete New
SqlDataSource - SqlDataSourceDetailsStars	
SqlDataSource - SqlDataSourceStars	

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="AdminStars.aspx.cs"
Inherits="AdminStars" %>
<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Admin Stars</title></head>
<body>
    <form id="form1" runat="server">
        <div>&nbsp;<table style="position: relative">
<tr><td colspan="1" style="text-align: center">
Admin Stars</td></tr>
<tr><td style="width: 100px">
&nbsp;<asp:GridView ID="GridViewStars" runat="server"
AutoGenerateColumns="False"
Caption="Stars" CellPadding="4" DataKeyNames="Id"
DataSourceID="SqlDataSourceStars"
ForeColor="#333333" GridLines="None" Style="position: relative">
<FooterStyle BackColor="#507CD1" Font-Bold="True" ForeColor="White" />
<Columns>
<asp:CommandField ShowSelectButton="True" />
<asp:BoundField DataField="Id" HeaderText="Id" InsertVisible="False"
ReadOnly="True"SortExpression="Id" Visible="False" />
<asp:BoundField DataField="Star" HeaderText="Star" SortExpression="Star" />
<asp:BoundField DataField="Price" HeaderText="Price" SortExpression="Price"
/></Columns>
<RowStyle BackColor="#EFF3FB" />
<EditRowStyle BackColor="#2461BF" />
<SelectedRowStyle BackColor="#D1DDF1" Font-Bold="True" ForeColor="#333333"
/>
<PagerStyle BackColor="#2461BF" ForeColor="White" HorizontalAlign="Center"
/>
<HeaderStyle BackColor="#507CD1" Font-Bold="True" ForeColor="White" />
<AlternatingRowStyle BackColor="White" /></asp:GridView></td></tr>
<tr>
<td style="width: 100px">
&nbsp;<asp:DetailsView ID="DetailsViewStars" runat="server"
AutoGenerateRows="False"
Caption="Stars Details" CellPadding="4" DataKeyNames="Id"
DataSourceID="SqlDataSourceDetailsStars"
ForeColor="#333333" GridLines="None" Height="50px" Style="position:
relative"Width="125px">
<FooterStyle BackColor="#507CD1" Font-Bold="True" ForeColor="White" />
<CommandRowStyle BackColor="#D1DDF1" Font-Bold="True" />
<EditRowStyle BackColor="#2461BF" />
<RowStyle BackColor="#EFF3FB" />
<PagerStyle BackColor="#2461BF" ForeColor="White" HorizontalAlign="Center"
/>
<Fields>
<asp:BoundField DataField="Id" HeaderText="Id" InsertVisible="False"
ReadOnly="True"SortExpression="Id" Visible="False" />
<asp:BoundField DataField="Star" HeaderText="Star" SortExpression="Star" />
<asp:BoundField DataField="Price" HeaderText="Price" SortExpression="Price"
/>
<asp:CommandField ShowDeleteButton="True" ShowEditButton="True"
ShowInsertButton="True" /></Fields>
<FieldHeaderStyle BackColor="#DEE8F5" Font-Bold="True" />
<HeaderStyle BackColor="#507CD1" Font-Bold="True" ForeColor="White" />
<AlternatingRowStyle BackColor="White" /></asp:DetailsView>
</td></tr></table>

```

```

<asp:SqlDataSource ID="SqlDataSourceDetailsStars" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionStringHotel %>"
DeleteCommand="DELETE FROM [Stars] WHERE [Id] = @Id" InsertCommand="INSERT
INTO [Stars] ([Star], [Price]) VALUES (@Star, @Price)"
SelectCommand="SELECT * FROM [Stars] WHERE ([Id] = @Id)"
UpdateCommand="UPDATE [Stars] SET [Star] = @Star, [Price] = @Price WHERE
[Id] = @Id">
<DeleteParameters>
<asp:Parameter Name="Id" Type="Int32" /></DeleteParameters>
<UpdateParameters>
<asp:Parameter Name="Star" Type="String" />
<asp:Parameter Name="Price" Type="Double" />
<asp:Parameter Name="Id" Type="Int32" /></UpdateParameters>
<SelectParameters>
<asp:ControlParameter ControlID="GridViewStars" Name="Id"
PropertyName="SelectedValue" Type="Int32" /></SelectParameters>
<InsertParameters>
<asp:Parameter Name="Star" Type="String" />
<asp:Parameter Name="Price" Type="Double" /></InsertParameters>
</asp:SqlDataSource>
<asp:SqlDataSource ID="SqlDataSourceStars" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionStringHotel %>"
SelectCommand="SELECT * FROM [Stars]">
</asp:SqlDataSource><br />
</div></form></body></html>

```

Configure Data Source - SqlDataSourceDetailsStars

Define Parameters

The wizard has detected one or more parameters in your SELECT statement. For each parameter in the SELECT statement, choose a source for the parameter's value.

Name	Value
Id	GridViewStars.SelectedValue

Parameter source:

ControlID:

DefaultValue:

[Show advanced properties](#)

SELECT statement:

< Previous Next > Finish Cancel

```

protected void DetailsViewStars_ItemInserted(object sender,
DetailsViewInsertedEventArgs e)
{
    this.GridViewStars.DataBind();
}
protected void DetailsViewStars_ItemUpdated(object sender,
DetailsViewUpdatedEventArgs e)
{
    this.GridViewStars.DataBind();
}
protected void DetailsViewStars_ItemDeleted(object sender,
DetailsViewDeletedEventArgs e)
{
    this.GridViewStars.DataBind();
}

```

Admin Stars		
Stars		
	Star	Price
Select	*	100
Select	**	200
Select	***	300
Select	*****	500
Select	*****	500
Stars Details		
Star	***	
Price	300	
Edit Delete New		

ربط معاملات الاستعلام مع عناصر التحكم

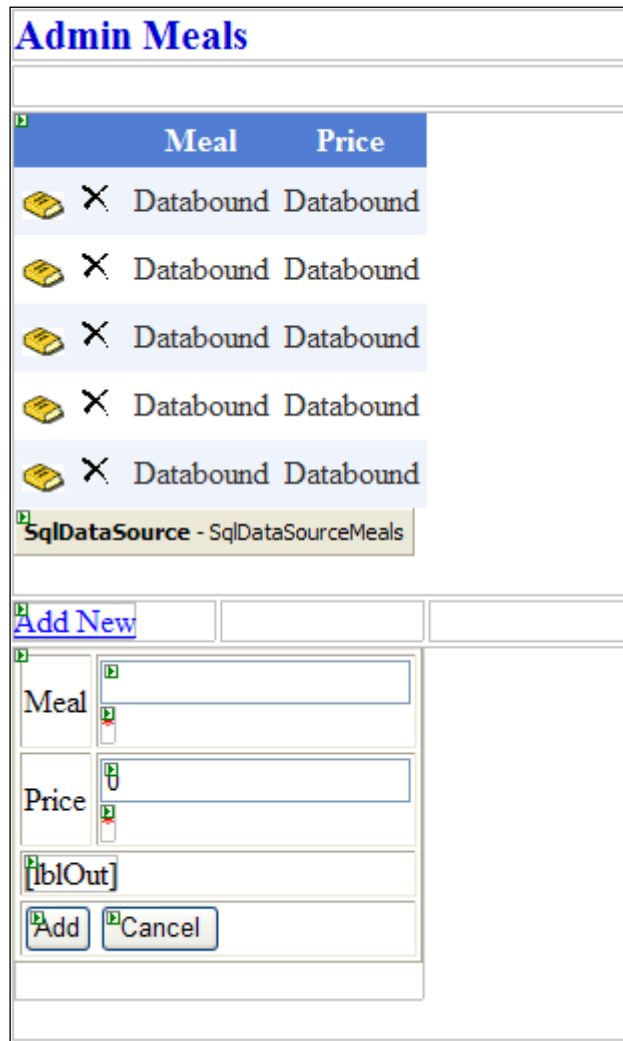
الأهداف التعليمية

- ربط معاملات الاستعلام مع عناصر التحكم.

ربط معاملات الاستعلام مع عناصر التحكم

نقوم في الصفحة التالية AdminMeals.aspx بعرض بيانات الجدول Meals وتعديلها وحذفها باستخدام عنصر GridView يرتبط مع مصدر بيانات sqlDataSource. أما لإضافة تسجيلة جديدة للجدول فنستخدم صندوق نص نربطها مع معاملات عبارة الإضافة InsertQuery لمصدر البيانات.

مثال:



```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="AdminMeals.aspx.cs"
Inherits="AdminMeals" %>
<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
<title>Admin Meals</title></head>
<body>
<form id="form1" runat="server">
<div style="position: relative">
<tr>
```

```

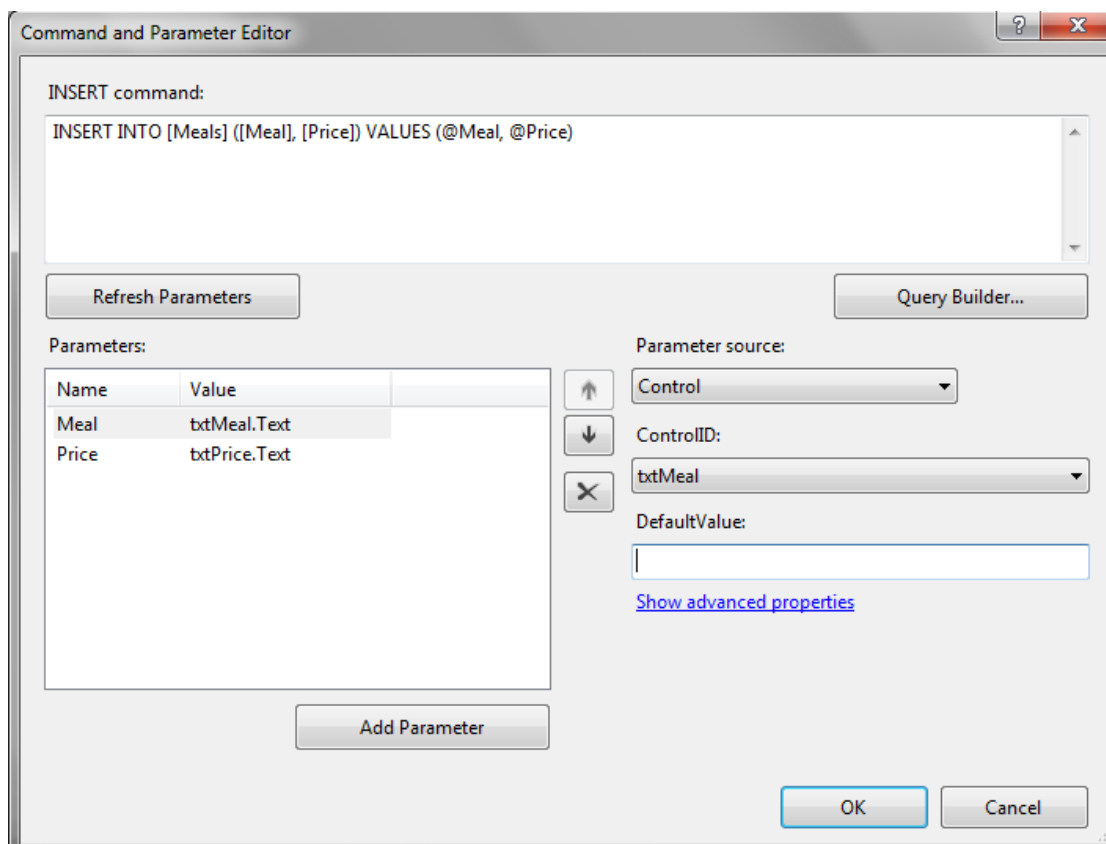
<td colspan="3">
<strong><span style="font-size: 16pt; color: #0000cc">Admin
Meals</span></strong></td></tr>
<tr><td colspan="3"></td></tr>
<tr><td colspan="3">
<asp:GridView ID="grdMeals" runat="server" AutoGenerateColumns="False"
CellPadding="4"
DataKeyNames="Id" DataSourceID="SqlDataSourceMeals" ForeColor="#333333"
GridLines="None"
Style="position: relative">
<FooterStyle BackColor="#507CD1" Font-Bold="True" ForeColor="White" />
<Columns>
<asp:CommandField ButtonType="Image" DeleteImageUrl="~/Images/delete.gif"
EditImageUrl="~/Images/edit.gif"
ShowDeleteButton="True" ShowEditButton="True"
CancelImageUrl="~/Images/cancel.gif" UpdateImageUrl="~/Images/update.gif"
/>
<asp:BoundField DataField="Id" HeaderText="Id" InsertVisible="False"
ReadOnly="True"
SortExpression="Id" Visible="False" />
<asp:BoundField DataField="Meal" HeaderText="Meal" SortExpression="Meal" />
<asp:BoundField DataField="Price" HeaderText="Price" SortExpression="Price"
/></Columns>
<RowStyle BackColor="#EFF3FB" />
<EditRowStyle BackColor="#2461BF" />
<SelectedRowStyle BackColor="#D1DDF1" Font-Bold="True" ForeColor="#333333"
/>
<PagerStyle BackColor="#2461BF" ForeColor="White" HorizontalAlign="Center"
/>
<HeaderStyle BackColor="#507CD1" Font-Bold="True" ForeColor="White" />
<AlternatingRowStyle BackColor="White" /></asp:GridView>
<asp:SqlDataSource ID="SqlDataSourceMeals" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionStringHotel %>"
DeleteCommand="DELETE FROM [Meals] WHERE [Id] = @Id" InsertCommand="INSERT
INTO [Meals] ([Meal], [Price]) VALUES (@Meal, @Price)"
SelectCommand="SELECT * FROM [Meals]" UpdateCommand="UPDATE [Meals] SET
[Meal] = @Meal, [Price] = @Price WHERE [Id] = @Id">
<DeleteParameters>
<asp:Parameter Name="Id" Type="Int32" /></DeleteParameters>
<UpdateParameters>
<asp:Parameter Name="Meal" Type="String" />
<asp:Parameter Name="Price" Type="Decimal" />
<asp:Parameter Name="Id" Type="Int32" /></UpdateParameters>
<InsertParameters>
<asp:ControlParameter ControlID="txtMeal" Name="Meal" PropertyName="Text"
Type="String" />
<asp:ControlParameter ControlID="txtPrice" Name="Price" PropertyName="Text"
Type="Decimal" /></InsertParameters>
</asp:SqlDataSource></td></tr>
<tr><td style="width: 100px">
<asp:LinkButton ID="btnAdd" runat="server" Style="position: relative"
OnClick="btnAdd_Click">Add New</asp:LinkButton></td>
<td style="width: 100px"></td>
<td style="width: 100px"></td></tr>
<tr><td colspan="3">
<asp:Panel ID="panelMeal" runat="server" Height="50px" Style="left: 0px;
position: relative;top: 0px" Visible="False" Width="192px">
<table border="1" style="position: relative">
<tr><td style="width: 100px">Meal</td>
<td style="width: 112px">

```

```




<asp:TextBox ID="txtMeal" runat="server" Style="position: relative; left:
0px; top: 0px;"></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
ControlToValidate="txtMeal"
ErrorMessage="Extra cannot be blank" Style="position: relative"
ToolTip="Extra cannot be blank">*</asp:RequiredFieldValidator></td></tr>
<tr>
<td style="width: 100px">Price</td>
<td style="width: 112px">
<asp:TextBox ID="txtPrice" runat="server" Style="position:
relative">0</asp:TextBox>
<asp:CompareValidator ID="CompareValidator1" runat="server"
ControlToValidate="txtPrice" Operator="DataTypeCheck" Style="position:
relative" ToolTip="Price must be number"
Type="Currency">*</asp:CompareValidator></td></tr>
<tr><td colspan="2">
<asp:Label ID="lblOut" runat="server" Style="position:
relative"></asp:Label></td></tr>
<tr><td colspan="2">
<asp:Button ID="btnAddExtra" runat="server" Style="position: relative"
Text="Add" OnClick="btnAddExtra_Click" />
<asp:Button ID="btnCancel" runat="server" CausesValidation="False"
Style="position: relative"
Text="Cancel" OnClick="btnCancel_Click" /></td></tr></table>
</asp:Panel></td></tr></table></div>
</form></body></html>

```



```
protected void btnAddExtra_Click(object sender, EventArgs e)
{
    SqlDataSourceMeals.Insert();
    txtMeal.Text = "";
    txtPrice.Text = "";
    grdMeals.DataBind();
    this.panelMeal.Visible = false;
}
```

Admin Meals

	Meal	Price
	✕ Breakfast	50.0000
	✕ Lunch	100.0000
	✕ Dinner	200.0000

Add New

Meal	<input type="text"/>
Price	<input type="text" value="0"/>
<input type="button" value="Add"/> <input type="button" value="Cancel"/>	

الفصل الرابع عشر:

التعامل مع قواعد البيانات برمجياً

الكلمات المفتاحية

.SqlConnection ، SqlCommand ، SqlDataReader ، SqlDataAdapter ، DataSet.

المُلخَص

نستعرض في هذا الفصل أهم طرق التعامل مع قواعد البيانات من خلال الكود البرمجي.

الأهداف التعليمية:

يتعرف الطالب في هذا الفصل على:

- الصف SqlConnection.
- الصف SqlCommand.
- الصف SqlDataReader.
- الصف SqlDataAdapter.
- الصف DataSet.
- عنصر البيانات Repeater.

المخطط

يضم فصل التعامل مع قواعد البيانات برمجياً 5 وحدات (Learning Objects) هي:

- التعامل مع قواعد البيانات برمجياً
- مثال تطبيقي
- عنصر التحكم Repeater
- مجموعات البيانات Datasets
- مثال تطبيقي

التعامل مع قواعد البيانات برمجياً

الأهداف التعليمية

- الصفوف الأساسية للتعامل مع قواعد البيانات.

التعامل مع قواعد البيانات برمجياً

يُمكن برمجياً الاتصال مع قواعد البيانات وتنفيذ العمليات الأساسية عليها (استعلام Select، إضافة Insert، حذف Delete، تعديل Update).

نعتمد في التعامل مع البيانات SQL Server على مجموعة من الصفوف التي توجد في كل من الفضائين System.Data و System.Data.SqlClient. لذا يجب استيراد هذين الفضائين في بداية ملف الكود الخلفي:
using System.Data;
using System.Data.SqlClient;

الصف SqlConnection

- يُستخدم غرض من هذا الصف لوصل غرض من الصف SqlCommand مع قاعدة بيانات معينة.
- من أهم خصائص هذا الغرض الخاصيةConnectionString والتي تُحدد سلسلة الإتصال مع قاعدة البيانات.
- من أهم طرق هذا الغرض الطريقة Open() التي تقوم بفتح اتصال مع قاعدة البيانات المحددة بسلسلة الاتصال، والطريقة Close() التي تقوم بإغلاق الاتصال المفتوح.
- يُستخدم غرض من هذا الصف لوصل غرض SqlConnection مع غرض SqlDataReader أو SqlDataAdapter.
- يسمح الغرض SqlCommand بتنفيذ عبارات SQL أو إجراءات مخزنة Stored Procedures على قاعدة البيانات. يتضمن هذا إعادة مجموعة تسجيلات (حيث نستخدم للوصول إليها غرض آخر كالغرض SqlDataReader أو الغرض SqlDataAdapter)، أو إعادة قيمة وحيدة، أو إعادة عدد العناصر المتأثرة بالاستعلامات التي لا تُعيد مجموعة تسجيلات.
- من أهم طرق هذا الغرض:
 - الطريقة ExecuteReader: تقوم بتنفيذ الأمر المعرف في الخاصية CommandText والذي يُعيد مجموعة من التسجيلات.
 - الطريقة ExecuteNonQuery: تقوم هذه الطريقة بتنفيذ الأمر المُعرّف في الخاصية CommandText (استعلام لا يُعيد صفوف مثل تعليمة Insert أو Update أو Delete) وتُعيد عدد التسجيلات التي تأثرت بالاستعلام.
 - الطريقة ExecuteScalar: تقوم بتنفيذ الأمر المعرف في الخاصية CommandText والذي يُعيد قيمة وحيدة.

- يوفر غرض من هذا الصف طريقة سريعة وفعالة للوصول إلى مجموعة من التسجيلات بشكل أمامي وللقراءة فقط.
- من أهم طرق هذا الغرض:
 - الطريقة Read: تقوم بتحريك مؤشر التسجيلة إلى التسجيلة التالية. وتُعيد false عند الوصول لنهاية التسجيلات.
 - الطريقة GetOrdinal("ColName"): تُعيد فهرس العمود الذي اسمه "ColName".
 - الطريقة GetString (index): تُعيد قيمة العمود ذو الفهرس index كسلسلة نصية.

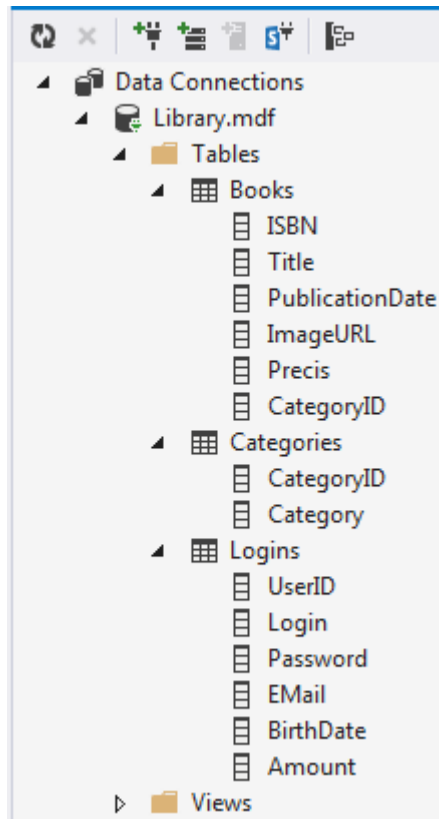
مثال تطبيقي

الأهداف التعليمية

- مثال على التعامل مع قواعد البيانات برمجياً.

ليكن لدينا قاعدة البيانات التالية Library.mdf والمنشأة باستخدام SQL Server Express. تحوي هذه القاعدة على ثلاثة جداول:

- جدول الكتب Books
- جدول فئات الكتب Categories
- جدول المستخدمين Logins



لنفرض أننا نريد قراءة بيانات جدول الكتب واستعراض عناوينها في قائمة برمجياً. يكون الملف .aspx في هذا المثال:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="DataReader.aspx.cs"
Inherits="DBCon" %>

<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>>Data Reader</title>
</head>
<body>
  <form id="form1" runat="server">
    <asp:Button ID="Button1" runat="server" OnClick="Button1_Click"
Style="z-index: 100;
  left: 80px; position: absolute; top: 350px" Text="Data Reader
2" Width="192px" />
    <asp:ListBox ID="ListBox1" runat="server" Height="184px" Style="z-
index: 101; left: 80px;
  position: absolute; top: 80px" Width="193px"></asp:ListBox>
```

```

        <asp:Button ID="Button3" runat="server" OnClick="Button3_Click"
Style="z-index: 102;
        left: 80px; position: absolute; top: 312px" Text="Data Reader
1" Width="192px" />
        <asp:Button ID="Button4" runat="server" OnClick="Button4_Click"
Style="z-index: 103;
        left: 80px; position: absolute; top: 472px" Text="Clear"
Width="192px" />
        <asp:Label ID="Label1" runat="server" Style="z-index: 104; left:
80px; position: absolute;
        top: 48px" Text="Books Titles:" Width="184px"></asp:Label>
        <asp:Button ID="Button2" runat="server" Style="z-index: 102;
        left: 72px; position: relative; top: 416px"
Text="ExecuteNonQuery" Width="192px" OnClick="Button2_Click" />
        <asp:Button ID="Button5" runat="server" Style="z-index: 102;
        left: -126px; position: relative; top: 376px"
Text="ExecuteScalar" Width="192px" OnClick="Button5_Click" />
        <asp:Label ID="Label2" runat="server" Style="left: -318px;
position: relative; top: 256px"
        Width="184px"></asp:Label>
    </form>
</body>
</html>

```

نقوم في الزر الأول (Data Reader 1) بإنشاء اتصال مع قاعدة البيانات وتعريف أمر Select، ومن ثم تنفيذ هذا الأمر باستخدام الطريقة ExecuteQuery وتضمين البيانات الناتجة في SqlDataReader. نقوم بعد ذلك بتنفيذ حلقة على هذا القارئ لقراءة عنوان كل كتاب وإضافته إلى القائمة.

```

protected void Button3_Click(object sender, EventArgs e)
{
    String ConStr ;
    ConStr = "Data
Source=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\Library.mdf;Integrated
Security=True";

    SqlConnection sqlCON = new SqlConnection();
    sqlCON.ConnectionString = ConStr;

    SqlCommand MySelectCommand = new SqlCommand();
    MySelectCommand.Connection = sqlCON;

    String sql ;
    sql = "SELECT * FROM Books";

    MySelectCommand.CommandText = sql;

    SqlDataReader DReader ;

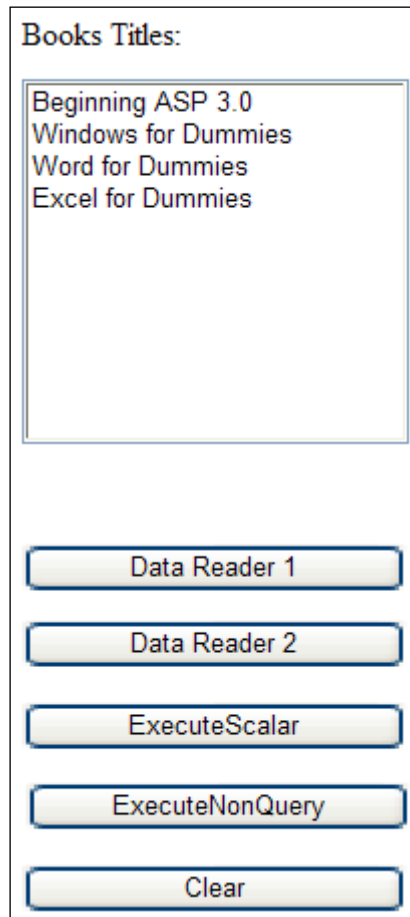
    sqlCON.Open();
    // Execute the SqlCommand and get the SqlDataReader.
    DReader = MySelectCommand.ExecuteReader();
    this.ListBox1.Items.Clear();
    String s;
    int i;
    while (DReader.Read())
    {
        i = DReader.GetOrdinal("Title");
        s = DReader.GetString(i);
    }
}

```

```

        this.ListBox1.Items.Add(s);
    }
    DReader.Close();
    sqlCON.Close();
}

```



يُمكن ربط القائمة مباشرة مع القارئ وذلك بإسناد اسم القارئ إلى الخاصية DataSource للقائمة، وإسناد اسم الحقل المعني إلى الخاصية DataTextField، ومن ثم تنفيذ الطريقة DataBind() على القائمة. كما تُبين الإجراءية التالية للزر Data Reader 2:

```

protected void Button1_Click(object sender, EventArgs e)
{
    String ConStr ;
    ConStr = "Data
Source=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\Library.mdf;Integrated
Security=True";

    SqlConnection sqlCON = new SqlConnection();
    sqlCON.ConnectionString = ConStr;

    SqlCommand MySelectCommand = new SqlCommand();
    MySelectCommand.Connection = sqlCON;
}

```

```

String sql ;
sql = "SELECT * FROM Books";

MySelectCommand.CommandText = sql;

SqlDataReader DReader ;

sqlCON.Open();
// Execute the SqlCommand and get the SqlDataReader.
DReader = MySelectCommand.ExecuteReader();
this.ListBox1.DataSource = DReader;
this.ListBox1.DataTextField = "Title";
this.ListBox1.DataBind();

DReader.Close();
sqlCON.Close();
}

```

تقوم إجرائية الزر ExecuteScalar بإظهار عدد الكتب الموجودة في الجدول:

```

protected void Button5_Click(object sender, EventArgs e)
{
    String ConStr ;
    ConStr = "Data
Source=(LocalDB)\v11.0;AttachDbFilename=|DataDirectory|\Library.mdf;Integrated
Security=True";

    SqlConnection sqlCON = new SqlConnection();
    sqlCON.ConnectionString = ConStr;

    SqlCommand MySelectCommand = new SqlCommand();
    MySelectCommand.Connection = sqlCON;

    String sql ;
    sql = "SELECT count(*) FROM Books";

    MySelectCommand.CommandText = sql;

    sqlCON.Open();

    int nL = int.Parse(MySelectCommand.ExecuteScalar().ToString());

    Label2.Text = nL.ToString() + " Books ";
    sqlCON.Close();
}

```

Books Titles:

Beginning ASP 3.0
 Windows for Dummies
 Word for Dummies
 Excel for Dummies

4 Books

Data Reader 1

Data Reader 2

ExecuteScalar

ExecuteNonQuery

Clear

كما تقوم إجرائية الزر ExecuteNonQuery بتنفيذ استعلام تحديث على جدول الكتب وإظهار عدد التسجيلات التي تم تحديثها:

```
protected void Button2_Click(object sender, EventArgs e)
{
    String ConStr ;
    ConStr = "Data
Source=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\Library.mdf;Integrated
Security=True";

    SqlConnection sqlCON = new SqlConnection();
    sqlCON.ConnectionString = ConStr;

    SqlCommand MySelectCommand = new SqlCommand();
    MySelectCommand.Connection = sqlCON;

    String sql ;
    sql = "Update Books set CategoryId=1";

    MySelectCommand.CommandText = sql;

    sqlCON.Open();

    int nL= int.Parse( MySelectCommand.ExecuteNonQuery().ToString());
    Label2.Text = nL.ToString() + " Records has been updated ";
    sqlCON.Close();
}
```

}

Books Titles:

Beginning ASP 3.0
Windows for Dummies
Word for Dummies
Excel for Dummies

4 Records has been updated

Data Reader 1

Data Reader 2

ExecuteScalar

ExecuteNonQuery

Clear

عنصر التحكم Repeater

الأهداف التعليمية

- استخدام عنصر التحكم Repeater.

استخدام عنصر التحكم Repeater

يُستخدم عنصر التحكم Repeater لعرض مجموعة من البيانات بشكل تكراري. يتم تحديد مظهر الإظهار بتعريف مجموعة من القوالب.

نقوم في المثال التالي بعرض كتب فئة معينة يُحددها المستخدم من قائمة الفئات باستخدام عنصر التحكم Repeater.

Search By Category:

Search Results:



C How to Program

ISBN: 0764544020

Published: February 2005

This book is for beginners who have no previous experience



Artificial Intelligence A Modern Approach

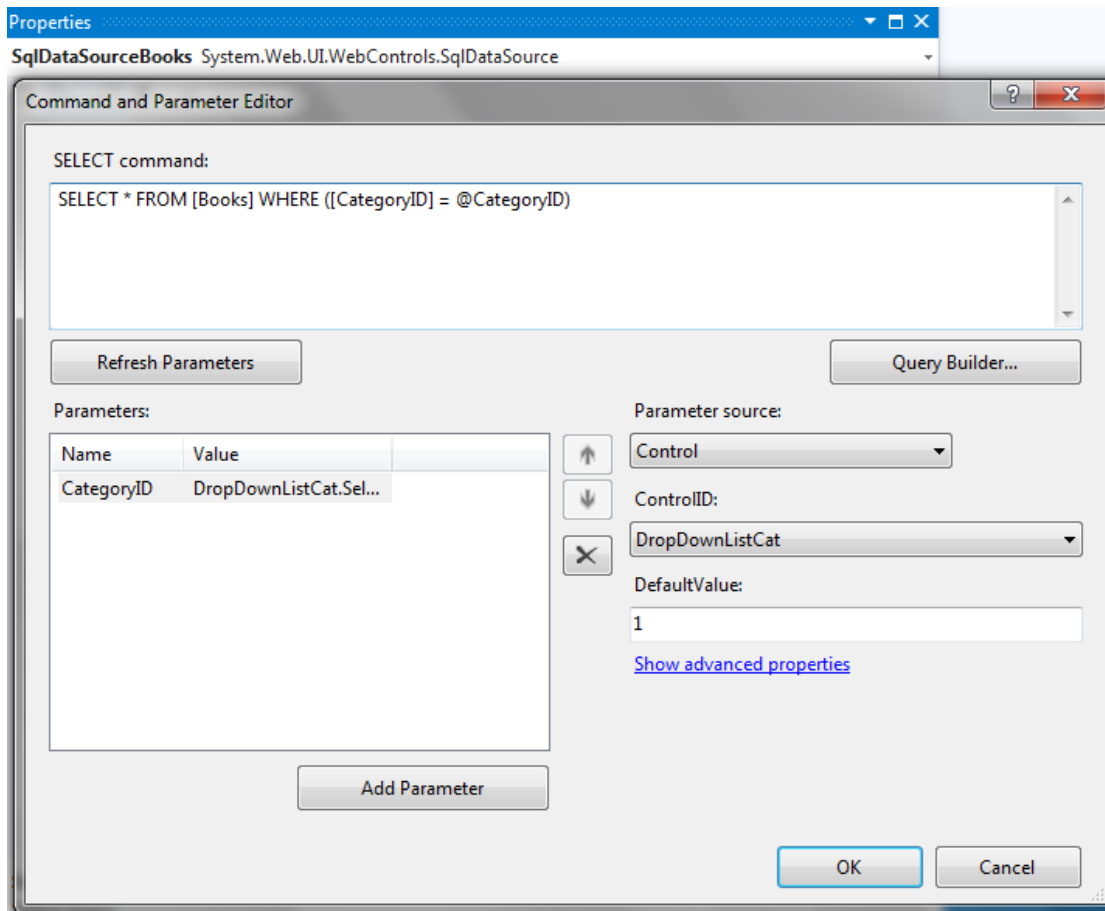
ISBN: 0764543636

Published: July 2006

This is a great book Most widely used in U.S. universities

For more information visit <http://www.svuonline.org>

ترتبط قائمة الفئات DropDownListCat في المثال السابق بمصدر بيانات مرتبط مع جدول الفئات. أما العنصر Repeater فيرتبط مع مصدر بيانات مرتبط مع جدول الكتب إنما مع شرط Where وبحيث أن معرف فئة الكتاب تساوي إلى فئة الكتاب المحددة من قائمة فئات الكتب:



يتم تحديد قوالب العنصر Repeater في الملف .aspx. وأهمها:

- قالب ترويسة العنصر HeaderTemplate
- قالب تذييل العنصر FooterTemplate
- قالب العنصر ItemTemplate
- قالب الفاصل بين العناصر SeparatorTemplate

لاحظ أننا نربط صورة غلاف الكتاب بصفحة أخرى BooksForm.aspx تحوي على عنصر FormView مرتبط مع جدول الكتب. نُمرر لهذه الصفحة معامل ISBN في العنوان كي تقوم بإظهار الكتاب المعني.

```
<div>
  <strong><span style="font-size: 14pt; color: #0000cc">
    Search By Category:</span></strong>
  <asp:DropDownList ID="DropDownListCat" runat="server"
    DataSourceID="SqlDataSourceCat"
    DataTextField="Category" DataValueField="CategoryID"
    Style="position: relative" AutoPostBack="True" Font-Bold="True" Font-
    Names="Tw Cen MT Condensed" Font-Size="Larger" ForeColor="DarkBlue">
  </asp:DropDownList><asp:SqlDataSource ID="SqlDataSourceCat"
    runat="server" ConnectionString="<%$
    ConnectionStrings:LibraryConnectionString %>"
    SelectCommand="SELECT * FROM [Categories]"></asp:SqlDataSource>
<br />
</div>
```

```

<div>

<asp:Repeater ID="Repeater1" runat="server"
DataSourceID="SqlDataSourceBooks">

<HeaderTemplate>
<div class="rHead">
Search Results: <br />

</div>
</HeaderTemplate>

<ItemTemplate>
<div class="rItem">
<a href="BooksForm.aspx?ISBN=<%# DataBinder.Eval(Container.DataItem,
"ISBN")%>" >
  <input type="image"
    src="images/<%# DataBinder.Eval(Container.DataItem, "ImageUrl")%>"
  />
</a>

<b><%# DataBinder.Eval(Container.DataItem, "Title")%></b><br/>
ISBN: <%# DataBinder.Eval(Container.DataItem, "ISBN")%> <br/>
Published: <%# DataBinder.Eval(Container.DataItem, "PublicationDate")%><br
/>
<%# DataBinder.Eval(Container.DataItem, "Precis")%>
</div><br/>
</ItemTemplate>

<SeparatorTemplate>

</SeparatorTemplate>

<FooterTemplate>

<div class="rFoot">
For more information visit
  <a href="http://www.svuonline.org">http://www.svuonline.org</a>
</div>
</FooterTemplate>

</asp:Repeater>

</div>
<asp:SqlDataSource ID="SqlDataSourceBooks" runat="server"
ConnectionString="<%$ ConnectionStrings:LibraryConnectionString %>"
SelectCommand="SELECT * FROM [Books] WHERE ([CategoryID] = @CategoryID)">
  <SelectParameters>
    <asp:ControlParameter ControlID="DropDownListCat" DefaultValue="1"
Name="CategoryID"
  PropertyName="SelectedValue" Type="Int32" />
  </SelectParameters>
</asp:SqlDataSource>
  &nbsp;
</form>
</body>
</html>

```

مجموعات البيانات Datasets

الأهداف التعليمية

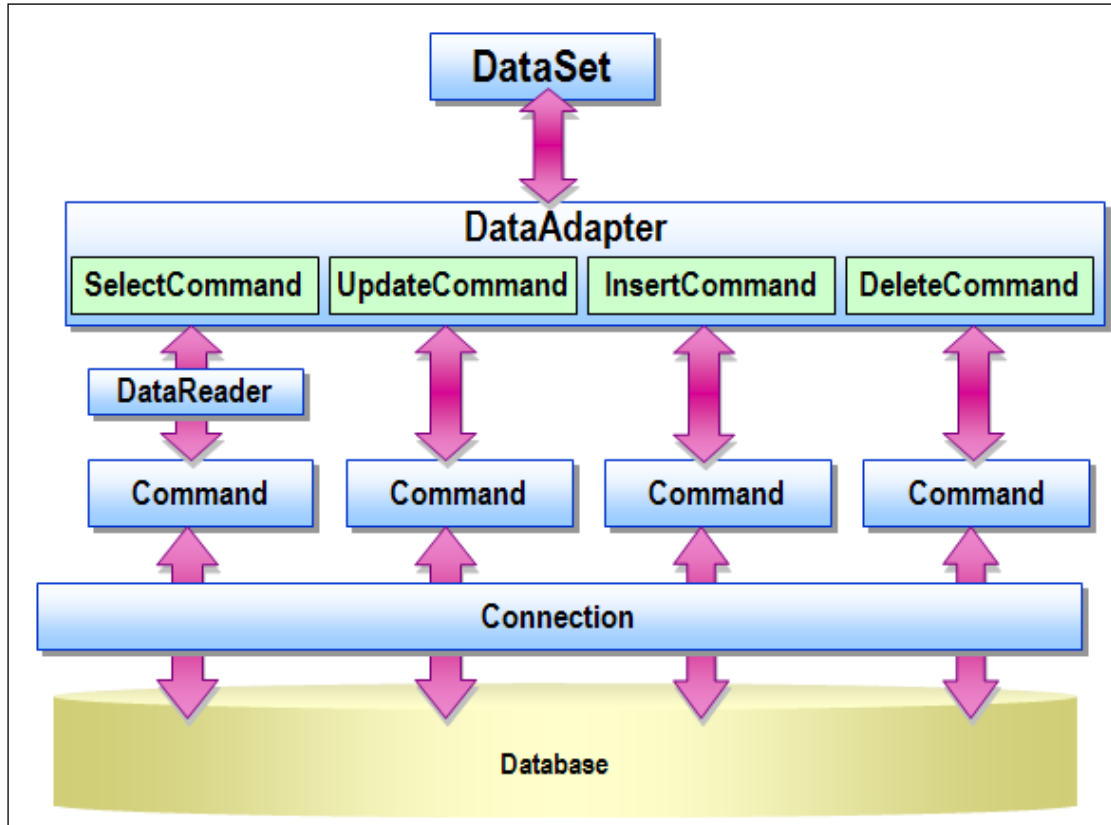
- مجموعات البيانات Datasets.

الصف SqlDataAdapter

- يقوم الغرض من هذا الصف بوصل غرض من الصف sqlCommand أو أكثر بغرض DataSet.
- يملك أربعة خصائص أساسية لتحديد الأوامر الأساسية: SelectCommand, InsertCommand, UpdateCommand, DeleteCommand.
- من أهم الطرق:
 - الطريقة Fill: تقوم بتنفيذ الأمر SelectCommand لتأهيل غرض DataSet.
 - الطريقة Update: تقوم باستدعاء الأوامر InsertCommand, DeleteCommand, UpdateCommand لكل تسجيلية تم إدراجها أو تعديلها أو حذفها ضمن مصدر البيانات DataSet الموافق.

الصف DataSet

- يُقدم هذا الصف مجموعة من الطرق للتعامل مع قواعد البيانات العلاقتية في بيئة غير متصلة.
 - نقوم أولاً بملء غرض الصف من قاعدة بيانات.
 - نتعامل مع هذا الغرض في بيئة غير متصلة.
 - نُعيد الاتصال مع قاعدة البيانات لعكس جميع التغييرات التي قُمنّا بها على البيانات.
- يُبين الشكل التالي مخطط التعامل مع قاعدة بيانات:



مثال:

نقوم في هذا المثال التعليمي بربط عنصر DataAdapter مع أربعة أوامر Select, Insert, Update, Delete. ومن ثم استخدام الطريقة Fill() لملء مجموعة بيانات DataSet. نقوم بعد ذلك بتعديل بيانات هذه المجموعة (إضافة، تعديل، حذف) من خلال الكود. ومن ثم تنفيذ هذه التعديلات على قاعدة البيانات باستخدام الطريقة Update. كما نربط عنصر تحكم من النوع GridView بمجموعة البيانات عن طريق الخاصية DataSource والطريقة .DataBind()

مثال:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="DataAdapter.aspx.cs" Inherits="DR" %>

<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Button ID="Button5" runat="server" Height="48px"
OnClick="Button5_Click" Style="z-index: 100;
                left: 168px; position: absolute; top: 240px" Text="DataAdapter"
Width="177px" />
            <asp:GridView ID="GridView1" runat="server" Style="z-index: 102;
left: 32px; position: absolute;
                top: 96px" Width="504px">
            </asp:GridView>

        </div>
    </form>
</body>
</html>
```

```
protected void Button5_Click(object sender, EventArgs e)
{
    String ConStr ;
    ConStr = "Data
Source=(LocalDB)\v11.0;AttachDbFilename=|DataDirectory|\\Library.mdf;Integrated
Security=True";

    SqlConnection sqlCON = new SqlConnection();
    sqlCON.ConnectionString = ConStr;

    SqlCommand SelectCom = new SqlCommand();
    SqlCommand UpCom = new SqlCommand();
    SqlCommand InsertCom = new SqlCommand();
    SqlCommand DelCom = new SqlCommand();

    SelectCom.Connection = sqlCON;
    UpCom.Connection = sqlCON;
    InsertCom.Connection = sqlCON;
    DelCom.Connection = sqlCON;
```

```

SelectCom.CommandText = "Select * from Logins";

UpCom.CommandText = "UPDATE [Logins] SET [Login] = @Login, [Password] =
@Password, [BirthDate] = @BirthDate, [Amount] = @Amount, [EMail] = @EMail
WHERE [UserID] = @UserID";
UpCom.Parameters.Add("@Login", SqlDbType.VarChar, 5, "Login");
UpCom.Parameters.Add("@Password", SqlDbType.VarChar, 100, "Password");
UpCom.Parameters.Add("@BirthDate", SqlDbType.DateTime, 100, "BirthDate");
UpCom.Parameters.Add("@Amount", SqlDbType.Int, 100, "Amount");
UpCom.Parameters.Add("@EMail", SqlDbType.VarChar, 100, "EMail");
UpCom.Parameters.Add("@UserId", SqlDbType.Int, 100, "UserId");

InsertCom.CommandText = "INSERT INTO [Logins] ([Login], [Password],
[BirthDate], [Amount], [EMail]) VALUES (@Login, @Password, @BirthDate,
@Amount, @EMail)";
InsertCom.Parameters.Add("@Login", SqlDbType.VarChar, 5, "Login");
InsertCom.Parameters.Add("@Password", SqlDbType.VarChar, 100, "Password");
InsertCom.Parameters.Add("@BirthDate", SqlDbType.DateTime, 100,
"BirthDate");
InsertCom.Parameters.Add("@Amount", SqlDbType.Int, 100, "Amount");
InsertCom.Parameters.Add("@EMail", SqlDbType.VarChar, 100, "EMail");

DelCom.CommandText = "Delete from [Logins] WHERE [UserID] = @UserID";
DelCom.Parameters.Add("@UserId", SqlDbType.Int, 100, "UserId");

SqlDataAdapter DBAdapter= new SqlDataAdapter() ;
DataSet ResultsDataSet = new DataSet();

DBAdapter.SelectCommand = SelectCom;
DBAdapter.UpdateCommand = UpCom;
DBAdapter.InsertCommand = InsertCom;
DBAdapter.DeleteCommand = DelCom;

DBAdapter.Fill(ResultsDataSet);

for (int i = 0 ; i<= ResultsDataSet.Tables[0].Rows.Count - 1;i++)
    ResultsDataSet.Tables[0].Rows[i].ItemArray[5] = i * 1000;

ResultsDataSet.Tables[0].Rows.Add(2, "tata", "789");
ResultsDataSet.Tables[0].Rows[0].Delete();

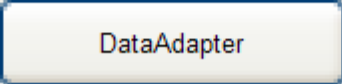
DBAdapter.Update(ResultsDataSet);

sqlCON.Close();

this.GridView1.DataSource = ResultsDataSet;
this.GridView1.DataBind();
}

```

UserID	Login	Password	EMail	BirthDate	Amount
10	titi	456	titi@titi.com	9/9/1990 12:00:00 AM	2000.0000
2	tata	789			



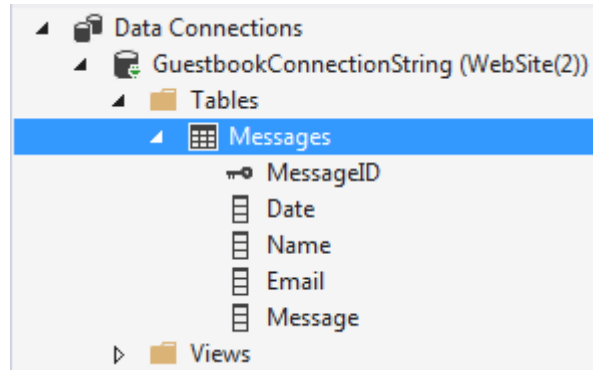
مثال تطبيقي

الأهداف التعليمية

- مثال تطبيقي.

ليكن المطلوب إنشاء صفحة في موقع يستطيع من خلالها زوار الموقع تسجيل ملاحظاتهم حول هذا الموقع. لتكن قاعدة البيانات Guestbook.mdf. تحوي هذه القاعدة جدول وحيد لتخزين تعليقات زوار الموقع Messages.

يُبين الشكل التالي حقول الجدول Messages:



لتكن الصفحة Guestbook.aspx التالية:

Please leave a message in our guestbook:

Your name:

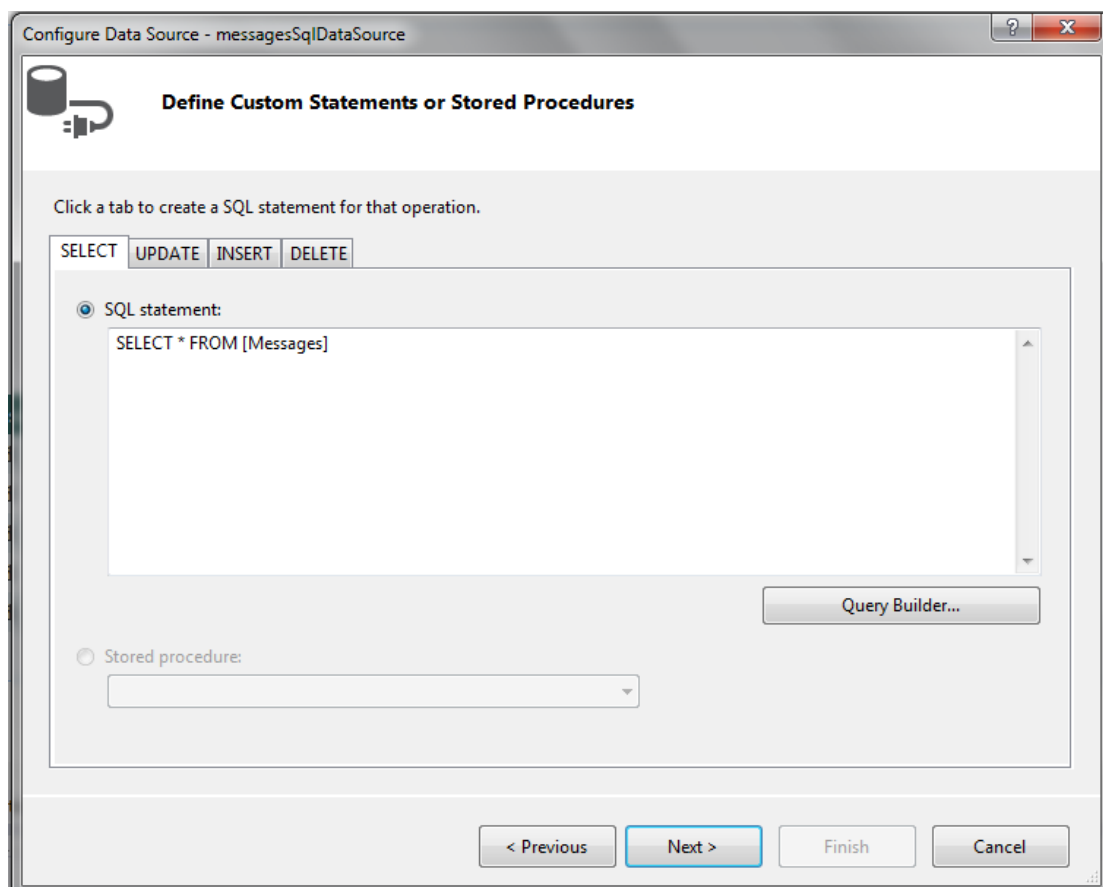
Your e-mail address:

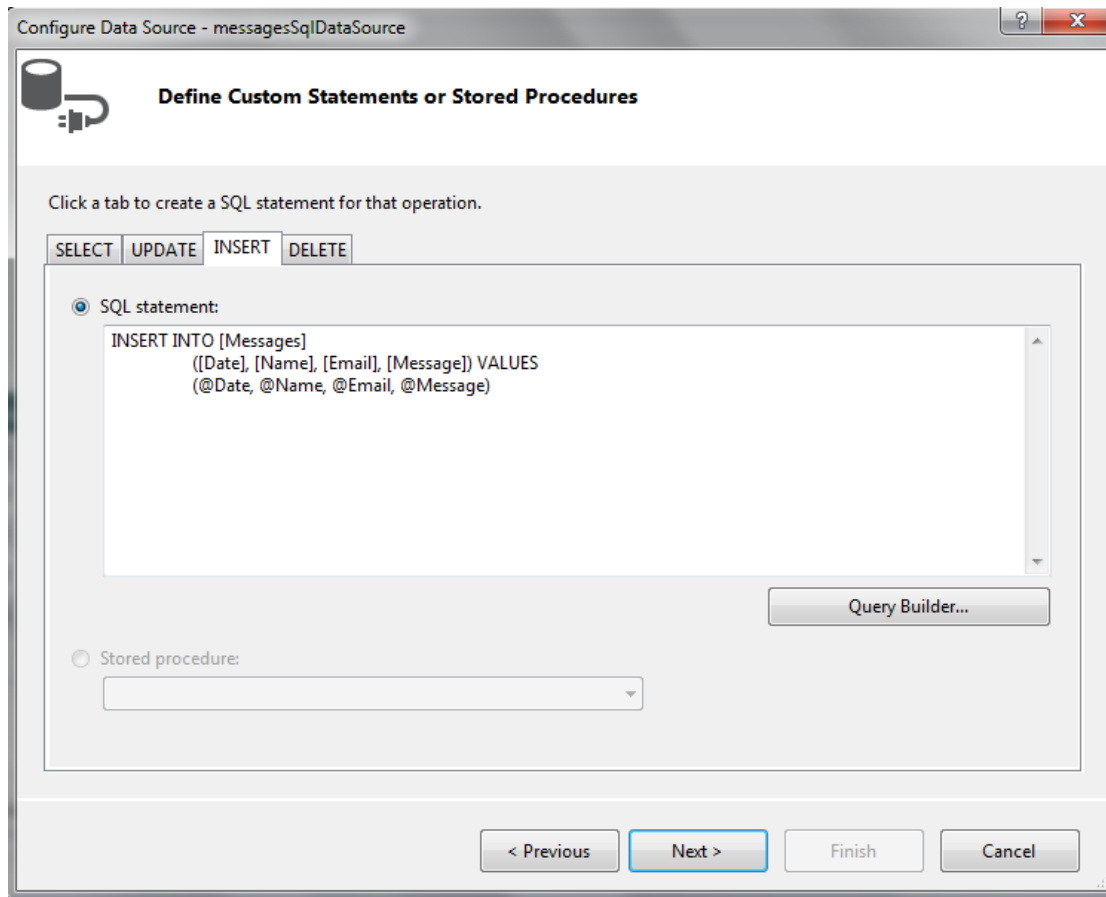
Tell the world:

Date	Name	Email	Message
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc

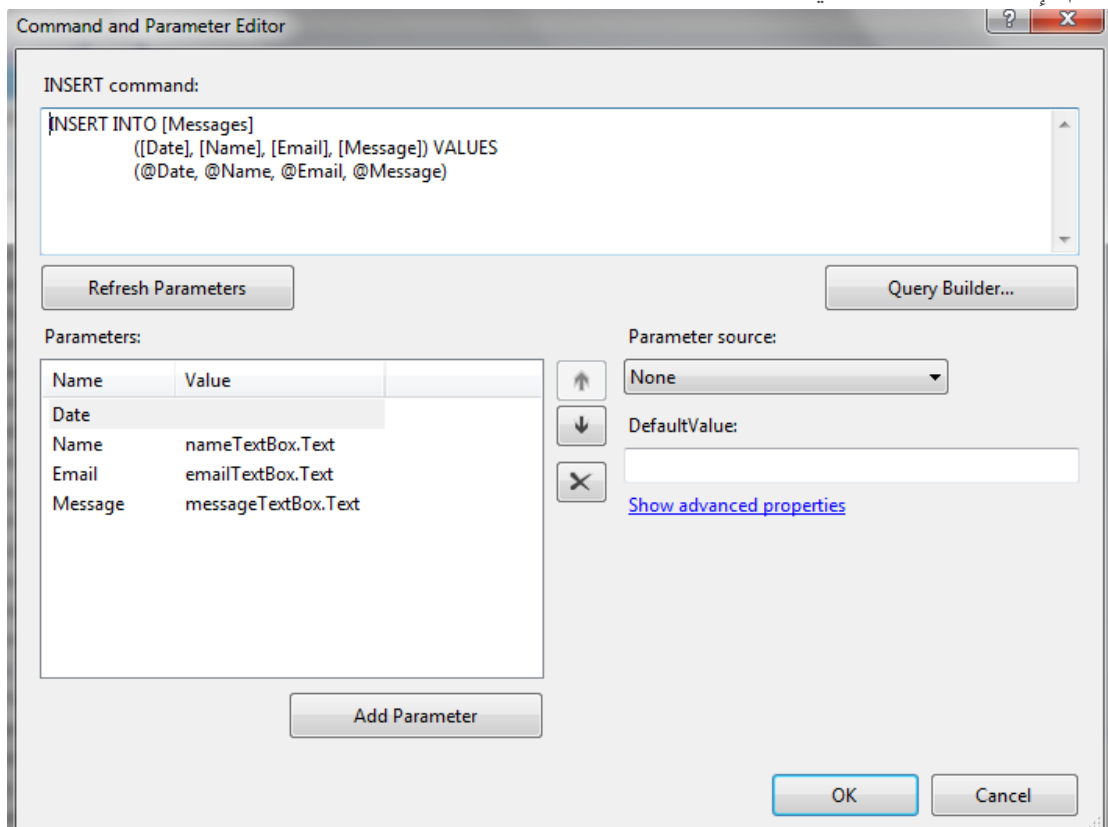
SqlDataSource - messagesSqlDataSource

يرتبط العنصر MessagesSqlDataSource من النوع sqlDataSource مع جدول الرسائل وذلك باستخدام المعالج Configure Data Source (بالنقر بالزر الأيمن على العنصر).





ترتبط معاملات العبارة السابقة بعناصر التحكم وفق ما يلي مع ملاحظة أن المعامل @Date لا يرتبط بعنصر تحكم وإنما سنعطيه قيمته في الكود:



```

<%-- Guestbook.aspx --%>
<%-- Guestbook Web application with a form for users to submit --%>
<%-- guestbook entiers and a GridView to view existing entries. --%>
<%@ Page Language="C#" AutoEventWireup="true"
    CodeFile="Guestbook.aspx.cs" Inherits="Guestbook" %>

<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml" >
  <head runat="server">
    <title>Guestbook</title>
  </head>
  <body>
    <form id="form1" runat="server">
      <div>
        <h2><span style="color: navy">
          Please leave a message in our guestbook:</span></h2>

        <table>
          <tr>
            <td style="width: 130px; height: 21px;"
                valign="top">
              Your name:<br />
            </td>
            <td style="width: 300px; height: 21px;"
                valign="top">
              <asp:TextBox ID="nameTextBox" runat="server"
                Width="300px"></asp:TextBox>
            </td>
          </tr>
          <tr>
            <td style="width: 130px;" valign="top">
              Your e-mail address:<br />
            </td>
            <td style="width: 300px;" valign="top">
              <asp:TextBox ID="emailTextBox" runat="server"
                Width="300px"></asp:TextBox></td>
          </tr>
          <tr>
            <td style="width: 130px;" valign="top">
              Tell the world:<br />
            </td>
            <td style="width: 300px;" valign="top">
              <asp:TextBox ID="messageTextBox" runat="server"
                Height="100px" Rows="8" Width="300px">
              </asp:TextBox>
            </td>
          </tr>
          <tr>
            <td style="width: 130px;" valign="top">
            </td>
            <td style="width: 300px;" valign="top">
              <asp:Button ID="submitButton" runat="server"
                Text="Submit" OnClick="submitButton_Click" />
              <asp:Button ID="clearButton" runat="server"
                Text="Clear" OnClick="clearButton_Click" />
            </td>
          </tr>
        </table>
      </div>
    </form>
  </body>
</html>

```

```

</table>
<br />

<asp:GridView ID="messagesGridView" runat="server"
    AutoGenerateColumns="False" CellPadding="4"
    ForeColor="#333333" GridLines="None"
    DataSourceID="messagesSQLDataSource" Width="600px"
    DataKeyNames="MessageID">
    <FooterStyle BackColor="#1C5E55" Font-Bold="True"
        ForeColor="White" />
    <RowStyle BackColor="#E3EAEB" />
    <PagerStyle BackColor="#666666" ForeColor="White"
        HorizontalAlign="Center" />
    <SelectedRowStyle BackColor="#C5BBAF" Font-Bold="True"
        ForeColor="#333333" />
    <HeaderStyle BackColor="#1C5E55" Font-Bold="True"
        ForeColor="White" />
    <EditRowStyle BackColor="#7C6F57" />
    <AlternatingRowStyle BackColor="White" />
    <Columns>
        <asp:BoundField DataField="Date" HeaderText="Date"
            SortExpression="Date" />
        <asp:BoundField DataField="Name" HeaderText="Name"
            SortExpression="Name" />
        <asp:BoundField DataField="Email"
            HeaderText="Email"
            SortExpression="Email" />
        <asp:BoundField DataField="Message"
            HeaderText="Message"
            SortExpression="Message" />
    </Columns>
</asp:GridView>

<asp:SqlDataSource ID="messagesSqlDataSource"
    runat="server"
    ConnectionString=
        "<%$ ConnectionStrings:GuestbookConnectionString %>"
    SelectCommand="SELECT * FROM [Messages]"
    DeleteCommand="DELETE FROM [Messages] WHERE
        [MessageID] = @original_MessageID"
    InsertCommand="INSERT INTO [Messages]
        ([Date], [Name], [Email], [Message]) VALUES
        (@Date, @Name, @Email, @Message)"
    UpdateCommand="UPDATE [Messages] SET [Date] = @Date,
        [Name] = @Name, [Email] = @Email, [Message] = @Message
        WHERE [MessageID] = @original_MessageID">
    <DeleteParameters>
        <asp:Parameter Name="original_MessageID"
            Type="Int32" />
    </DeleteParameters>
    <UpdateParameters>
        <asp:Parameter Name="Date" Type="String" />
        <asp:Parameter Name="Name" Type="String" />
        <asp:Parameter Name="Email" Type="String" />
        <asp:Parameter Name="Message" Type="String" />
        <asp:Parameter Name="original_MessageID"
            Type="Int32" />
    </UpdateParameters>
    <InsertParameters>
        <asp:Parameter Name="Date" Type="String" />
        <asp:ControlParameter ControlID="nameTextBox"

```

```

        Name="Name" PropertyName="Text" Type="String" />
        <asp:ControlParameter ControlID="emailTextBox"
        Name="Email" PropertyName="Text" Type="String" />
        <asp:ControlParameter ControlID="messageTextBox"
        Name="Message" PropertyName="Text" Type="String" />
    </InsertParameters>
</asp:SqlDataSource>
</div>
</form>
</body>
</html>

```

أما ملف الكود الخلفي:

```

// Guestbook.aspx.cs
// Code-behind file that defines event handlers for the guestbook.
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class Guestbook : System.Web.UI.Page
{
    // Submit Button adds a new guestbook entry to the database,
    // clears the form and displays the updated list of guestbook entries
    protected void submitButton_Click( object sender, EventArgs e )
    {
        // create a date parameter to store the current date

        System.Web.UI.WebControls.Parameter date =
            new System.Web.UI.WebControls.Parameter(
                "Date", TypeCode.String, DateTime.Now.ToShortDateString() );

        // set the @Date parameter to the date parameter

        messagesSqlDataSource.InsertParameters.RemoveAt( 0 );
        messagesSqlDataSource.InsertParameters.Add( date );

        // execute an INSERT SQL statement to add a new row to the
        // Messages table in the Guestbook database that contains the
        // current date and the user's name, e-mail address and message
        messagesSqlDataSource.Insert();

        // clear the TextBoxes
        nameTextBox.Text = "";
        emailTextBox.Text = "";
        messageTextBox.Text = "";

        // update the GridView with the new database table contents
        messagesGridView.DataBind();
    } // end method submitButton_Click

    // Clear Button clears the Web Form's TextBoxes

```

```
protected void clearButton_Click( object sender, EventArgs e )
{
    nameTextBox.Text = "";
    emailTextBox.Text = "";
    messageTextBox.Text = "";
} // end method clearButton_Click
} // end class Guestbook
```

ويكون التنفيذ:

Please leave a message in our guestbook:

Your name:

Your e-mail address:

Tell the world:

Date	Name	Email	Message
5/17/2007	Bassel ALKHATIB	t_balkhatib@svuonline.org	Great site!
5/17/2007	Sue Black	sblack@svuonline.org	I love the site! Keep up the good work!
5/18/2007	Liz White	lwhite@svuonline.org	Very useful information. Will visit again soon.

الفصل الخامس عشر: خدمات الويب

الكلمات المفتاحية

خدمات الويب، نشر خدمة ويب، استخدام خدمة ويب.

المُلخَص

نعرض في هذا الفصل لآلية بناء واستخدام خدمات الويب.

الأهداف التعليمية

يتعرف الطالب في هذا الفصل على:

- آلية عمل خدمات الويب.
- إنشاء ونشر خدمات الويب.
- استخدام خدمة ويب.

المخطط

يضم فصل خدمات الويب 4 وحدات (Learning Objects) هي:

- خدمات الويب
- إنشاء ونشر خدمات الويب
- مثال تطبيقي لخدمة ويب
- مثال: استخدام خدمة ويب

خدمات الويب

الأهداف التعليمية

- استخدام خدمات الويب.
- آلية عمل خدمات الويب.

خدمات الويب Web Services

- تسمح خدمة ويب Web Service بالقيام بالحسابات الموزعة وذلك من خلال السماح لحاسب من استدعاء طريقة method موجودة على حاسب آخر عبر صيغ بيانات مشتركة وبروتوكولات معيارية مثل XML و HTTP.

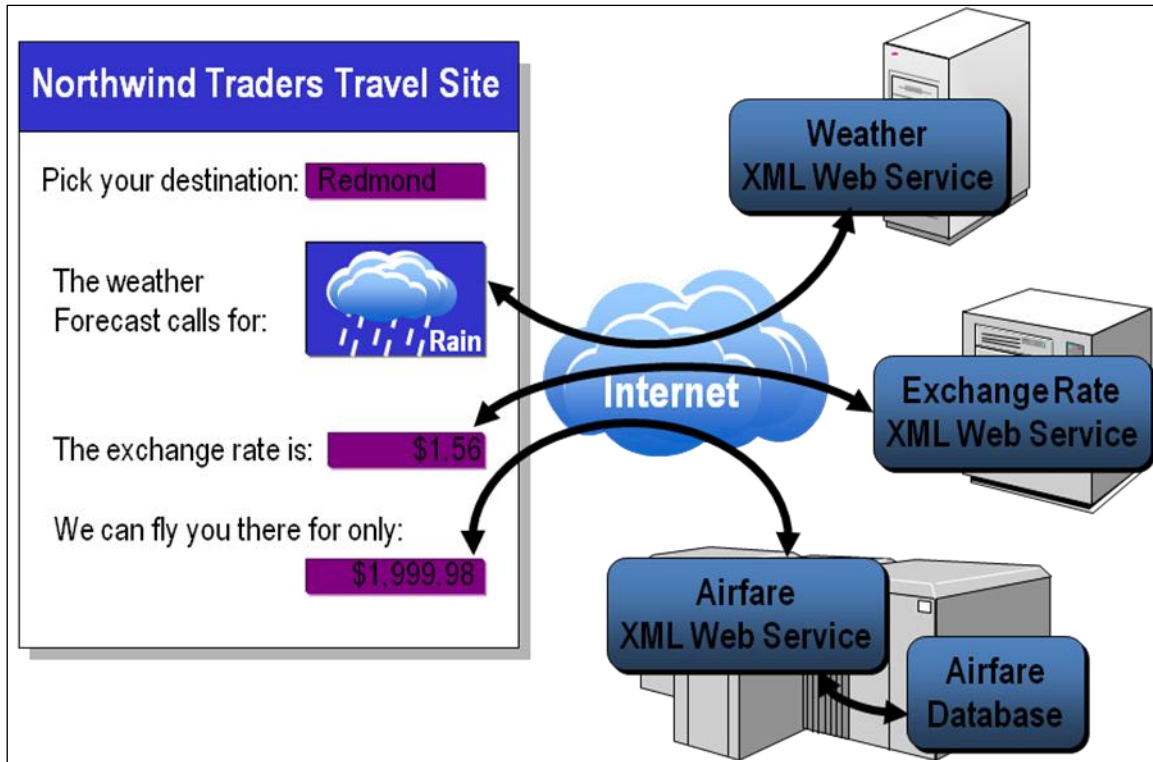
- تُشجع Microsoft مسوقي البرمجيات والأعمال الالكترونية على نشر وتشارك خدمات الويب. إذ طالما تتزايد يوماً بعد يوم المتصلين بالانترنت، فإن استدعاء التطبيقات للطرق عبر الانترنت يُصبح مريحاً.

- تسمح هذه التقنية بتسهيل عملية تعاون تطبيقات الأعمال الضخمة ونموها بشكل سريع. إذ بالتشارك في خدمات الويب الموافقة، ستمكن الشركات العاملة في تطوير المنتجات البرمجية من اختصار الزمن اللازم للتطوير والاتفات دائماً لتطوير خدمات جديدة.

لنأخذ كمثال بسيط، حالة شركة تبيع على الويب أقراص موسيقية كما تُعطي معلومات عن الفنانين. ولنفرض أن شركة أخرى تختص ببيع بطاقات الحفلات تُقدم خدمة ويب تسمح بمعرفة تواريخ الحفلات القادمة لمختلف الفنانين وتسمح للمستخدمين بشراء بطاقات هذه الحفلات. يُمكن للشركتين أن تتعاونوا وبحيث يتمكن مستخدمي موقع الشركة الأولى من شراء بطاقات الحفلات مما سيزيد من عدد زوار موقع الشركة الأولى. كذلك فإن الشركة الثانية ستزداد مبيعاتها إذ أنها ستستفيد من زبائن الشركة الأولى كما يُمكن لها أن تطلب عمولة من الشركة الأولى على البطاقات المباعة.

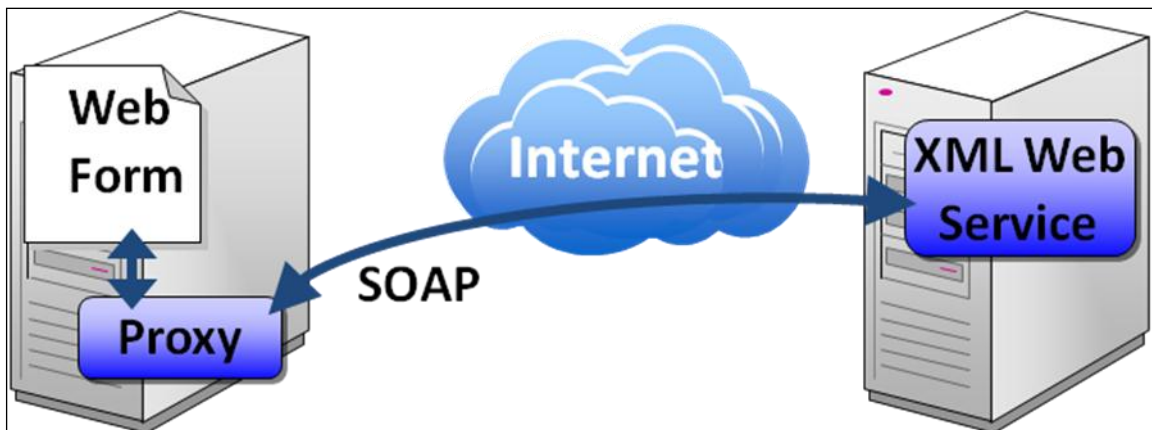
يُمثل الشكل التالي مثلاً صفحة ويب لشركة سفر وسياحة تستفيد من عدة خدمات ويب لمساعدة زبونها:

- خدمة ويب لمعرفة حالة الطقس للوجهة المسافر إليها.
- خدمة ويب لمعرفة سعر الصرف للعملة المحلية.
- خدمة ويب لمعرفة ثمن التذكرة.



آلية عمل خدمات الويب

- تكون خدمة الويب عبارة عن تطبيق موجود على حاسب ويحيث يُمكن لأي حاسب آخر على الشبكة من الوصول إلى هذا التطبيق. يُدعى عادة الحاسب الحاضن للتطبيق بالحاسب البعيد remote machine.
- يقوم التطبيق الذي يريد الاستفادة من خدمة الويب بإرسال طلب استدعاء لطريقة توفرها خدمة الويب للحاسب البعيد، والذي يقوم بدوره بمعالجة هذا الطلب وإرسال الجواب إلى التطبيق المستدعي.
- يتم نقل الطلبات والأجوبة باستخدام البروتوكول SOAP. مما يعني أن أي زبون قادر على توليد ومعالجة رسائل SOAP يستطيع استخدام خدمة الويب، وذلك بغض النظر عن اللغة التي كُتبت بها خدمة الويب. كما يُبين الشكل التالي:



- يُمكن لخدمة الويب أن تقتصر خدماتها على زبائن مخولين `authorized` فقط.
- بما أن خدمات الويب والبروتوكول `SOAP` مستقلين عن منصات التشغيل واللغات المستخدمة، فيمكن للشركات أن تتعاون إذاً عبر خدمات الويب دون أن تقلق لتوافقية البرمجيات أو العتاد أو تقانات الاتصال.
- تتعاون مثلاً الشركات مثل `Amazon` و `Google` و `eBay` عبر تشارك مجموعة من خدمات الويب. سنرى أن خدمة الويب هي في النهاية صف `Class` يحوي مجموعة من الطرق `Methods` ويتواجد على حاسب بعيد.
- تُستدعى الطرق الموجودة في خدمة ويب عن بعد باستخدام إجرائية الاستدعاء البعيدة `Remote Procedure Call (RPC)`. يتمّ تأشير هذه الطرق باستخدام الوصفة `WebMethod` كما سنرى لاحقاً.

إنشاء ونشر خدمات الويب

الأهداف التعليمية

- التعرف على آلية إنشاء ونشر خدمة ويب.

إنشاء خدمة ويب

تُبين الخطوات التالية الخطوات اللازمة لإنشاء خدمة الويب البسيطة Test باستخدام Visual Studio 2013 على مخدم الانترنت المحلي Internet Information Services IIS.

1. أنشئ موقع ويب وسمّه مثلاً TestHelloService، مع اختيار مكان التخزين:

<http://localhost/TestHelloService>

سوف يتم تخزين الموقع في المسار الافتراضي لمواقع الويب على الحاسب: C:\inetpub\wwwroot

2. أضف عنصر جديد إلى الموقع من النوع Web Service وسمّه مثلاً HelloWebService.

يقوم Visual Studio بإنشاء خدمة ويب بسيطة تتألف مما يلي:

- الصف HelloWebService.cs الذي يحوي الطريقة HelloWorld، وهو صف مشتق من الصف System.Web.Services.WebService.
- الملف HelloWebService.asmx والذي يحوي:

```
<%@ WebService Language="C#" CodeBehind="~/App_Code/HelloWebService.cs" Class="HelloWebService" %>
```

محدداً لغة البرمجة المستخدمة، مسار الكود الخلفي، واسم الصف.

3. قم بفتح IIS للتأكد من استضافة خدمة الويب المنشأة على الحاسب.

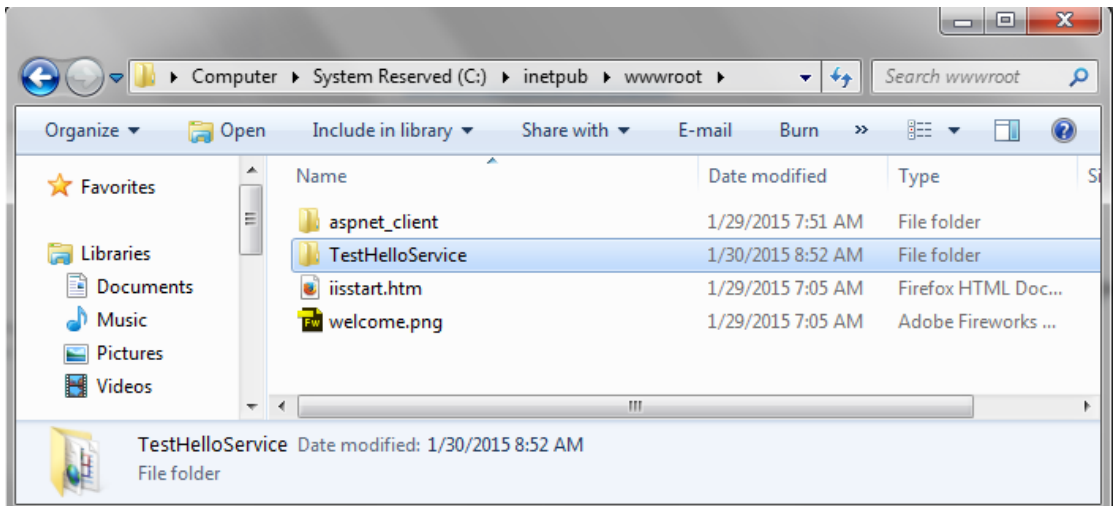
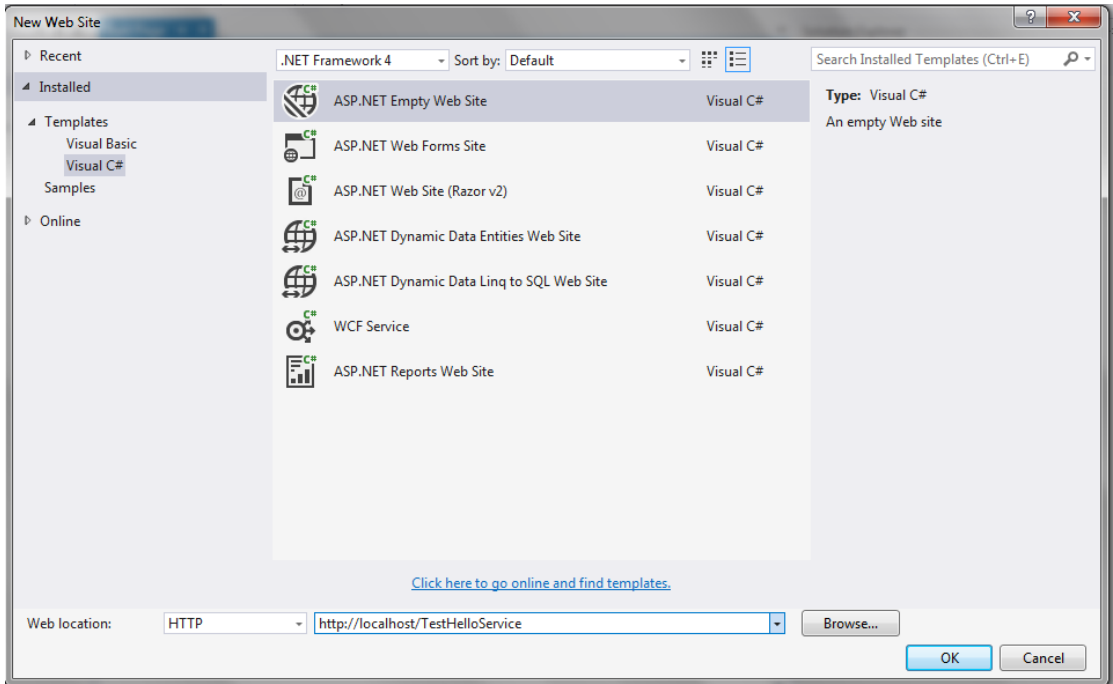
4. تجريب خدمة الويب لتجريب خدمة الويب

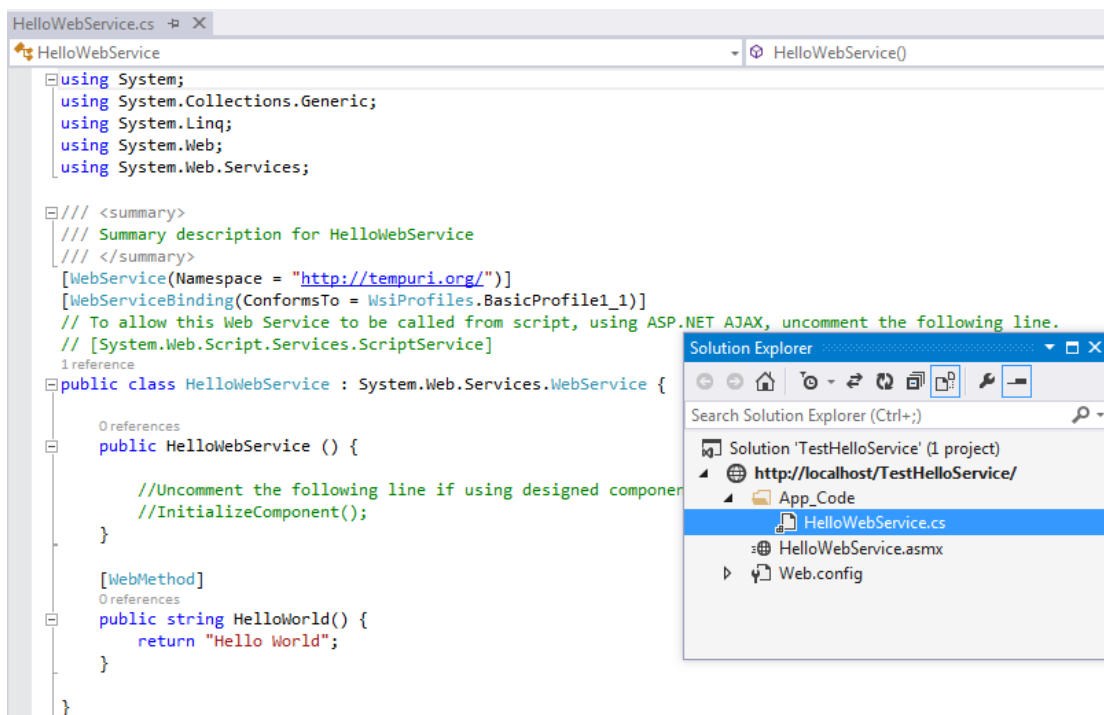
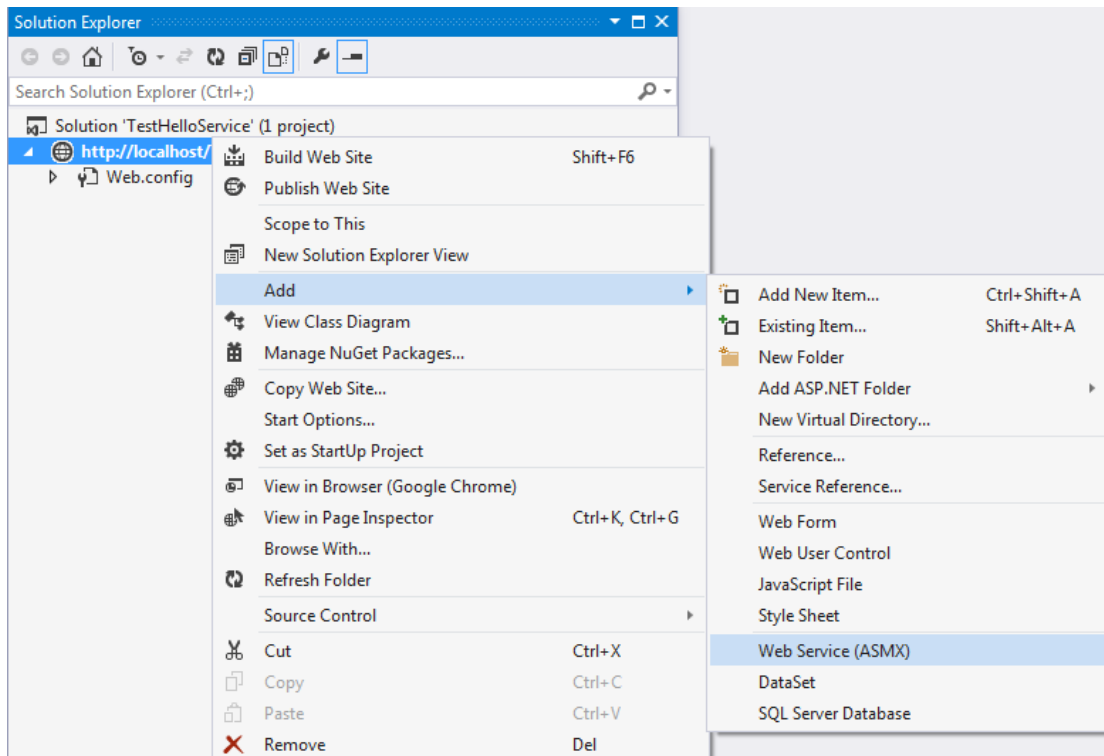
لتشغيل خدمة الويب: <http://localhost/TestHelloService/HelloWebService.asmx>

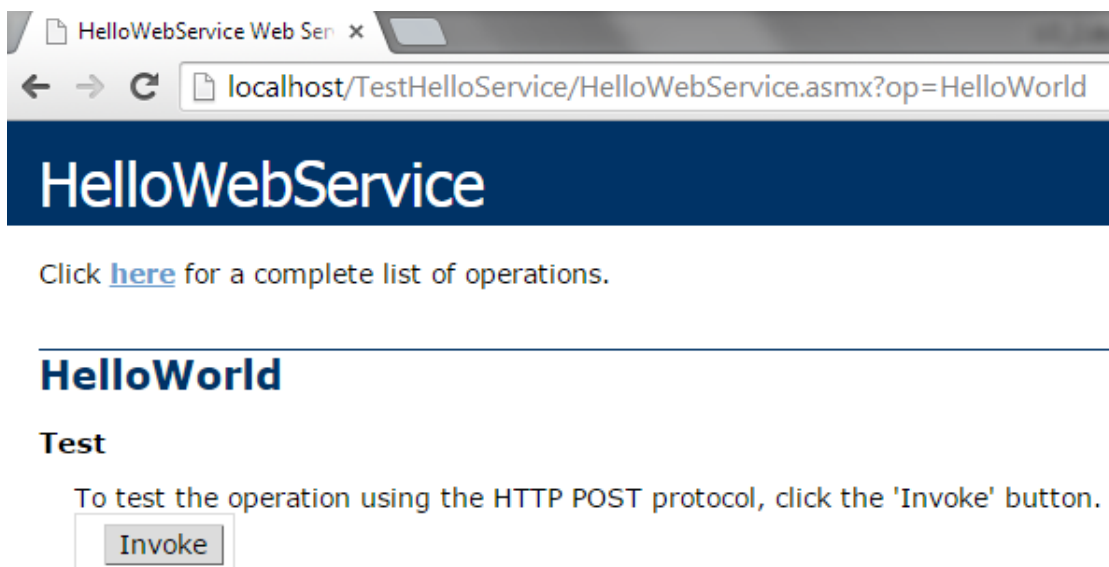
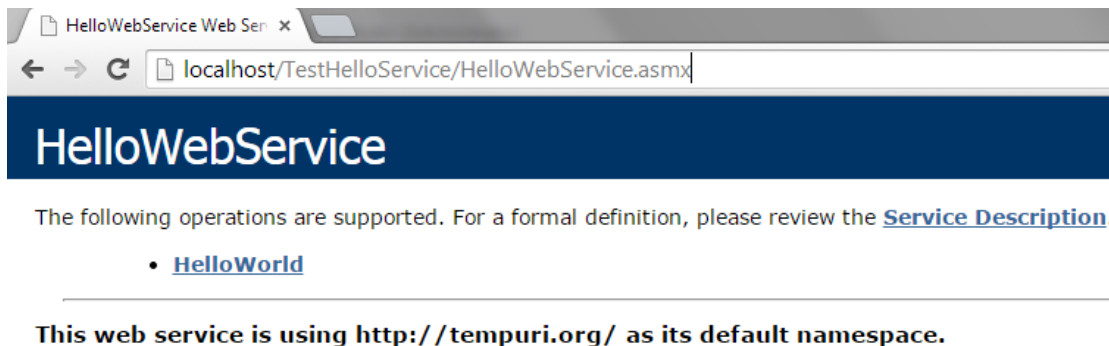
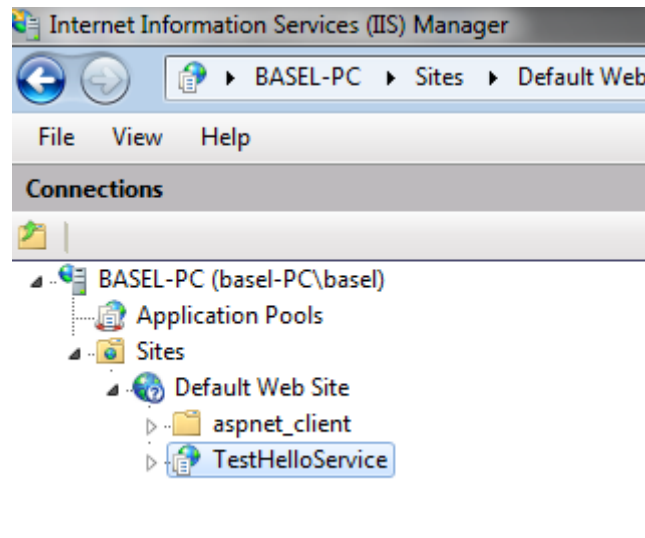
وبالنقر على الارتباط HelloWorld تظهر النافذة التالية:

وبالنقر على الزر Invoke يظهر ناتج تنفيذ طريقة الويب:

5. قم بتعديل الخدمة الأولية إلى الخدمة المطلوبة.







نشر خدمة الويب

- تكون خدمة الويب السابقة منشورة deployed تلقائياً لأننا أنشئناها على المخدم IIS مباشرة.
- يُمكن فتح خدمة الويب من المستعرض كما وضعنا سابقاً مع العنوان:
`http://localhost/TestHelloService/HelloWebService.asm`
- يُمكن للآخرين الموصولون عبر الشبكة من استخدام خدمة الويب المنشورة على IIS على حاسبك بكتابة العنوان التالي في المستعرض:
`http://host/TestHelloService/HelloWebService.asm`
حيث host هو اسم حاسبك أو عنوانك IP address على الشبكة.

مثال تطبيقي لخدمة ويب

الأهداف التعليمية

- مثال تطبيقي لخدمة ويب.

إنشاء خدمة ويب

سوف نقوم في المثال التطبيقي التالي أولاً بإنشاء خدمة ويب تسمح لزيون مسافر بإدخال المعلومات عن المقعد الذي يريد حجزه، ومن ثم تقوم هذه الخدمة بحجز المقعد إن كان متاحاً.

1. أنشئ موقع ويب جديد:

<http://localhost/ReservationServiceWebSite>.

2. أنشئ قاعدة بيانات ضمن الموقع Tickets.mdf وأنشئ الجدول Seats ضمنها.

3. يحوي الجدول Seats في قاعدة البيانات Tickets.mdf أربعة أعمدة:

▪ Number رقم المقعد.

▪ Type نوع المقعد (Window, Middle, Aisle).

▪ Class درجة المقعد (Economy, First).

▪ Taken محجوز (1) أو متاح (0).

تكون بيانات الجدول مثلاً كما يلي:

4. أضف مجموعة بيانات DataSet للمشروع وسمّها TicketsDataSet. أضف الجدول Seats إليها

(اسحب بالفأرة الجدول من نافذة Server Explorer إلى مجموعة البيانات).

5. أضف الاستعلام التالي لموائم الجدول والذي يقوم بإرجاع المقاعد الشاغرة من نوع وصف محدد:

```
SELECT Number
FROM Seats
WHERE (Taken = 0 ) AND (Type=@type) AND (Class=@class)
```

6. قم بتسمية الطرق كما يلي:

7. أضف استعلام التحديث التالي لموائم الجدول والذي يقوم بتعديل مقعد ذو رقم معطى ليُصبح شاغراً:

```
UPDATE Seats
SET Taken=1
WHERE (Number=@number)
```

أعط الاسم UpdateSeatAsTaken للطريقة الناتجة.

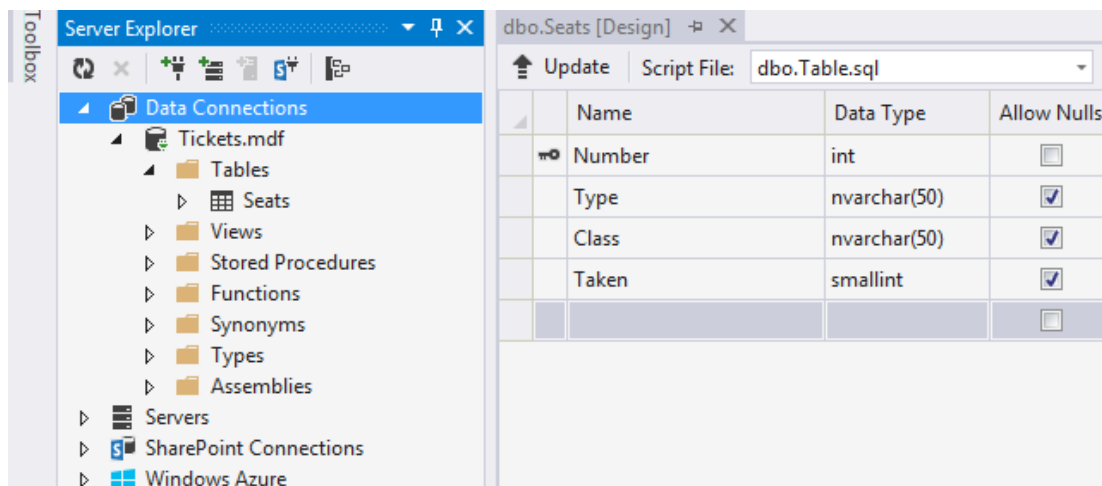
8. أضف خدمة وب ReservationService.aspx إلى الموقع.

9. افتح الملف ReservationService.cs واكتب الطريقة Reserve التي تقوم بالبحث في مقاعد قاعدة

البيانات Tickets.mdf لتحديد مقعد يطابق رغبة الزبون. إذا وجدت الطريقة مقعد تقوم بتعديل قاعدة

البيانات لحجزه و تُعيد True. أما إذا لم يتم الحجز فتُعيد False.

تمتلك الطريقة Reserve معاملي دخل، الأول هو سلسلة نصية string تُحدد نوع المقعد المطلوب: (Window, Middle, Aisle). والثاني سلسلة نصية تُحدد درجة المقعد: (Economy, First). تُعيد الطريقة Reserve القيمة true للدلالة على نجاح الحجز. أما إذا لم يكن هناك مقعداً متاحاً فإنها تُرجع false.



	Number	Type	Class	Taken
	1	Aisle	Economy	0
	2	Middle	Economy	0
	3	Window	Economy	0
	4	Aisle	Economy	0
	5	Middle	Economy	0
	6	Window	Economy	0
	7	Aisle	First	0
	8	Middle	First	0
	9	Window	First	0

```
// ReservationService.cs
// Airline reservation Web Service.

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

/// <summary>
/// Summary description for ReservationService
/// </summary>
[WebService(Namespace = "http://www.svuonline.org/", Description =
    "Service that enables a user to reserve a seat on a plane." ) ]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]

// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment
the following line.
```

```

// [System.Web.Script.Services.ScriptService]
public class ReservationService : System.Web.Services.WebService {
    private TicketsDataSet ticketsDataSet = new TicketsDataSet();

    // create SeatsTableAdapter for interacting with the database
    private TicketsDataSetTableAdapters.SeatsTableAdapter
SeatsTableAdapter =
        new TicketsDataSetTableAdapters.SeatsTableAdapter();
    public ReservationService () {
        // create TicketsDataSet object for caching data
        // from the Tickets database
    }
    [WebMethod(Description = "Method to reserve a seat.")]
    public bool Reserve( string seatType, string classType )
    {
        // fill TicketsDataSet.Seats with rows that represent untaken
        // seats that match the specified seatType and classType
        SeatsTableAdapter.FillByTypeAndClass(
            ticketsDataSet.Seats, seatType, classType);

        // if the number of seats returned is nonzero,
        // obtain the first matching seat number and mark it as taken
        if (ticketsDataSet.Seats.Count != 0)
        {
            int seatNumber = ticketsDataSet.Seats[0].Number;

            SeatsTableAdapter.UpdateSeatAsTaken(seatNumber);
            return true; // seat was reserved
        } // end if
        return false; // no seat was reserved
    } // end method Reserve
}

```

إنشاء نموذج ويب يستهلك خدمة الويب

1. أنشئ موقع ويب جديد:
(http://localhost/ReservationClientWebSite).
2. أضف مرجع لخدمة الويب السابقة:
3. أضف صفحة جديدة. يُبين الشكل التالي تصميم صفحة الويب والتي تسمح للمستخدم بتحديد كل من درجة المقعد (Economy, First) ونوع المقعد (Aisle, Middle, Window).
سوف تستخدم هذه الصفحة خدمة الويب السابقة لتأمين حجز المقعد.
يُبين الشكل التالي كود XML للصفحة ReservationClient.aspx السابقة:
 - تُعرّف هذه الصفحة قائمتين منسدلتين DropDownList: قائمة أنواع المقاعد، وقائمة درجة المقعد.
 - ينقر المستخدم زر الأمر reserveButton لإرسال طلبه للمخدم ليقيم بتنفيذ الإجراءات المرتبطة بالزر reserveButton_Click.
 - تُستخدم التسمية errorLabel لإظهار الرسائل الموافقة للمستخدم.
يُبين الشكل التالي الكود الخلفي ReservationClient.aspx.cs للصفحة السابقة:

يُبين الشكل التالي استعراض صفحة الويب:

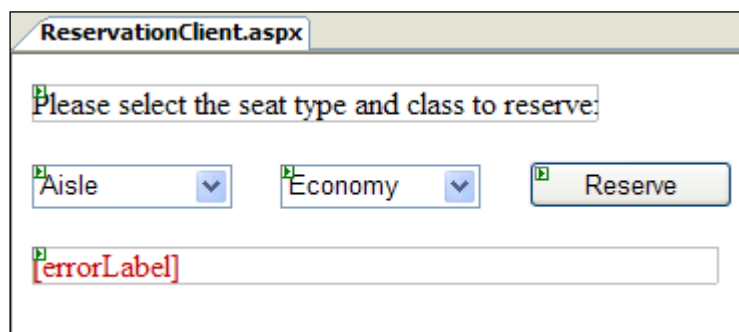
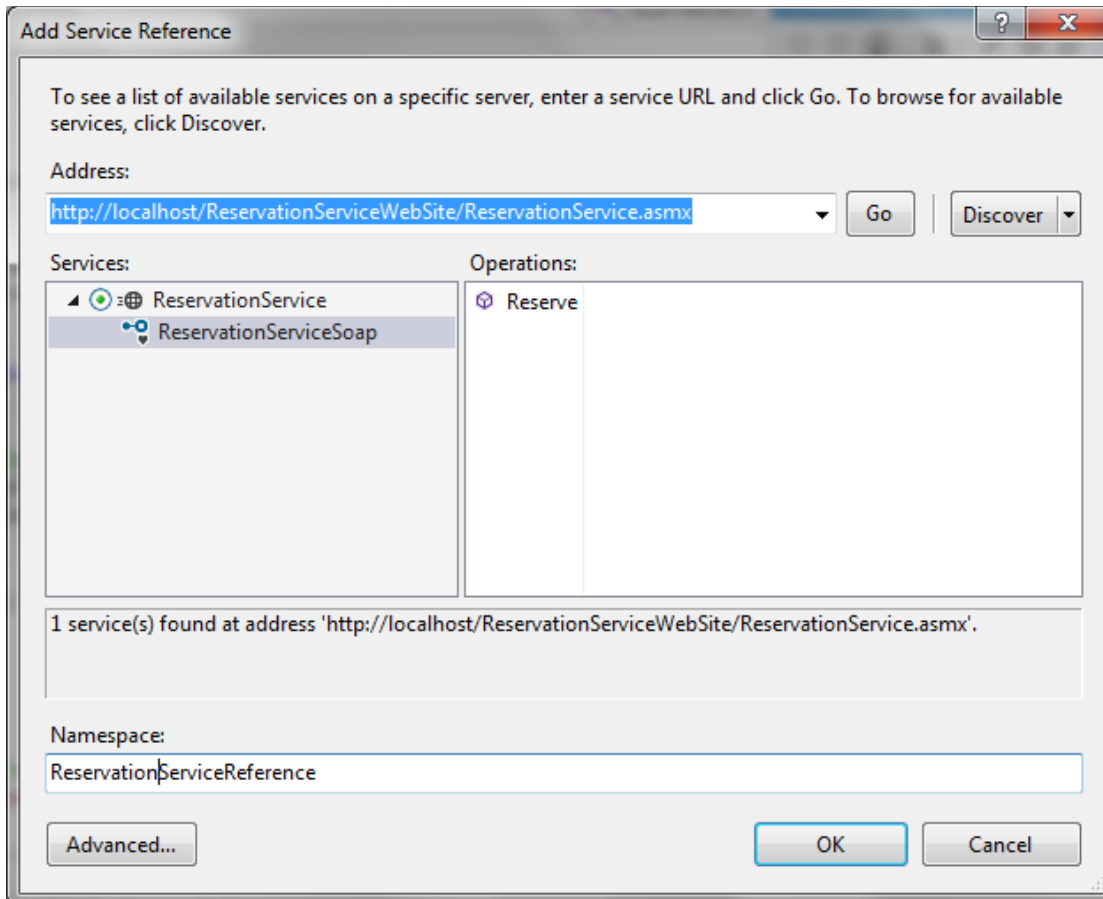
عند اختيار المستخدم لمقعد متاح تظهر الرسالة:

Your reservation has been made. Thank you.

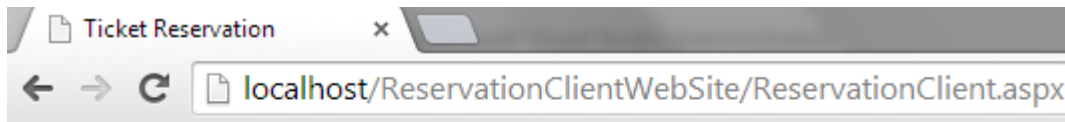
أما عند اختياره لمقعد غير متاح، فتظهر الرسالة:

This type of seat is not available.

Please modify your request and try again.




```
Response.Write( "Your reservation has been made. Thank you." );
} // end if
else // WebMethod returned false, so signal failure
{
    // display message in the initially blank errorLabel
    errorLabel.Text = "This type of seat is not available. " +
        "Please modify your request and try again.";
} // end else
} // end method reserveButton_Click
}
```



Please select the seat type and class to reserve:

مثال: استخدام خدمة ويب

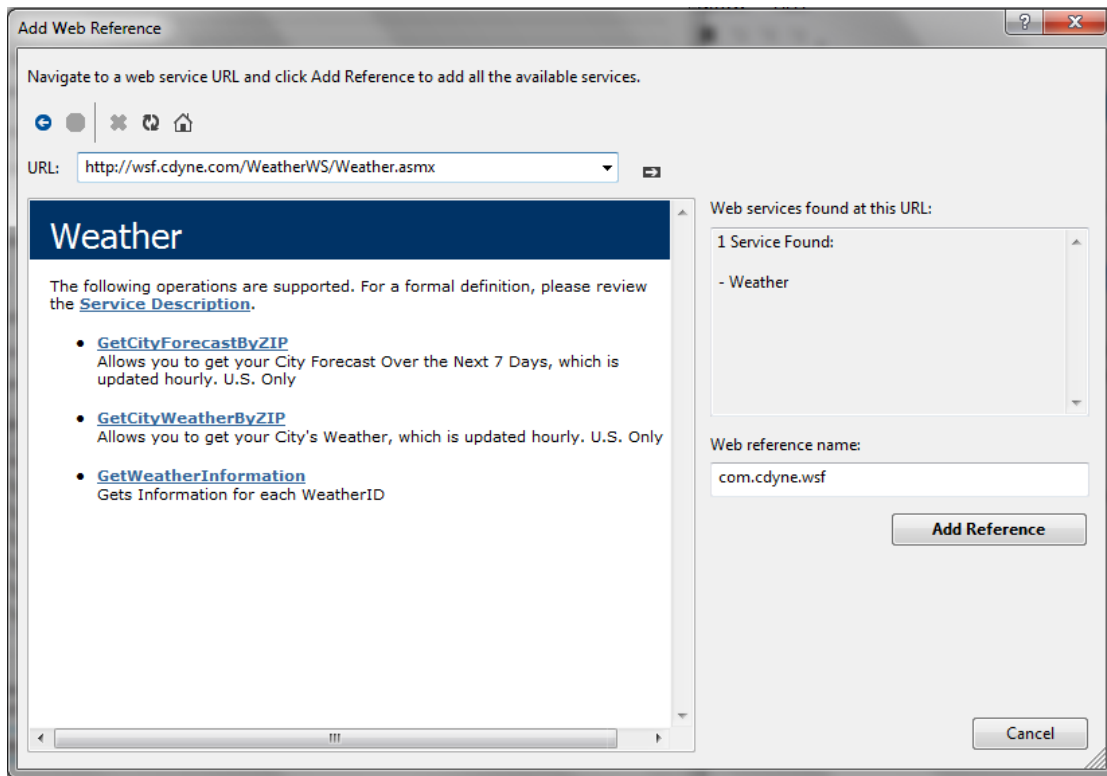
الأهداف التعليمية

- استخدام خدمة ويب.

استخدام خدمة ويب

يُبين المثال التالي استخدام خدمة ويب شهيرة تقوم بإعطاء حالة الطقس:

1. أنشئ موقع ويب.
2. قم بإضافة مرجع لموقع الويب كما يلي:
3. أنشئ صفحة ويب تحوي قائمة بأسماء بعض المدن.
4. قم باستدعاء خدمة الويب السابقة كما يلي:
5. قم بإختيار مدينة لمعرفة طقسها:



```
protected void Button1_Click(object sender, EventArgs e)
{
    //http://ws.cdyne.com/WeatherWS/Weather.asmx cdyneServiceWeather
    com.cdyne.wsf.Weather t = new com.cdyne.wsf.Weather();
    com.cdyne.wsf.ForecastReturn x = new com.cdyne.wsf.ForecastReturn();
    com.cdyne.wsf.Forecast[] r;
    x=t.GetCityForecastByZIP(DropDownList1.SelectedValue );
    r = x.ForecastResult;
    string s="<table border=1>";
    s = s + "<tr > <th colspan=2> " + x.City + " </th>";
    for (int i = 0; i < r.Length; i++)
    {
        s = s + "<tr> <td> " + r[i].Date.ToLongDateString() + " <td> " +
r[i].Description + " </tr>";
    }

    Label1.Text = s;
}
```

Using ws.cdyne.com Weather Web Service

Enter City: ▼

New York	
Saturday, September 20, 2014	Partly Cloudy
Sunday, September 21, 2014	Mostly Cloudy
Monday, September 22, 2014	Partly Cloudy
Tuesday, September 23, 2014	Sunny
Wednesday, September 24, 2014	Sunny
Thursday, September 25, 2014	Sunny
Friday, September 26, 2014	Sunny

الفصل السادس عشر:

مقدمة إلى XML

الكلمات المفتاحية

.XML Control ،XSL ،XML

المُلخَص

نستعرض في هذا أساسيات التعامل مع لغة التّأشير الموسعة XML.

الأهداف التعليمية

يتعرف الطالب في هذا الفصل على:

- لغة التّأشير الموسعة XML.
- لغة الأسلوب الموسعة XSL.
- مجموعات البيانات و XML.

المخطط

يضم فصل مقدمة إلى XML 3 وحدات (Learning Objects) هي:

- XML
- مثال تطبيقي
- مجموعات البيانات وXML

XML

الأهداف التعليمية

- أساسيات XML, XSL

مقدمة

- طورت لغة التأسيس الموسعة (XML (Extensible Markup Language) من قبل منظمة الويب World Wide Web Consortium's (W3C) عام 1996. وهي عبارة عن لغة تأسيس لتوصيف بيانات مهيكلة، أي أن المحتوى مفصول تماماً عن طريقة العرض بخلاف لغة HTML.
- بما أن XML تحوي بيانات فقط، فستكون من مهمة التطبيقات المختلفة تقرير كيفية عرض هذه البيانات.
- بخلاف لغة HTML، فإن لغة XML تسمح للمستخدمين بخلق مؤثراتهم tags لاستخدامها في التطبيقات المختلفة. مما يسمح بخلق لغات تأسيس جديدة لتوصيف أنماط معينة من البيانات مثل الصيغ الرياضية أو المركبات الكيميائية أو الموسيقى.
- وبما أن مؤثرات XML توصف البيانات، فسيكون بالإمكان البحث أو الفرز أو التنسيق لإظهار مستندات XML باستخدام تقانات مشابهة مثل لغة الأسلوب الموسعة Extensible Style Language(XSL).
- يُبين المثال التالي Contacts.xml ملف XML لبيانات بعض الأشخاص، حيث يمكن فتحه باستخدام المتصفح:

```
<?xml version="1.0" encoding="utf-8" ?>
<contacts>
  <contact>
    <lastName>Deitel</lastName>
    <firstName>Harvey</firstName>
  </contact>
  <contact>
    <lastName>Deitel</lastName>
    <firstName>Paul</firstName>
  </contact>
  <contact>
    <lastName>Nieto</lastName>
    <firstName>Tem</firstName>
  </contact>
</contacts>
```

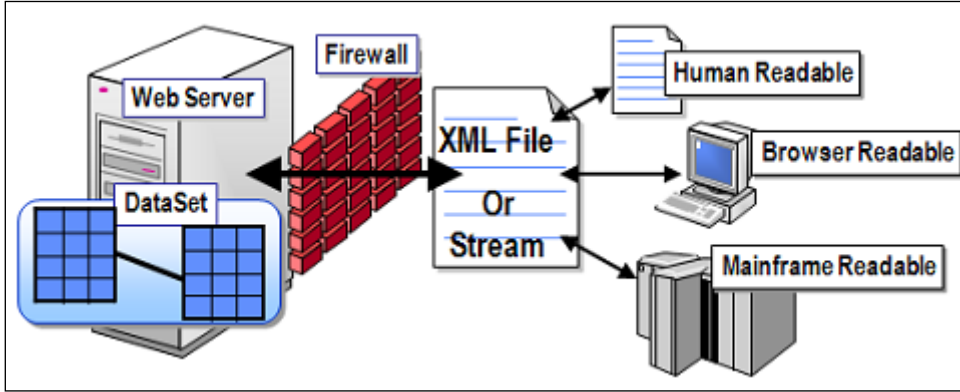


```

<?xml version="1.0" encoding="utf-8" ?>
- <contacts>
- <contact>
  <lastName>Deitel</lastName>
  <firstName>Harvey</firstName>
</contact>
- <contact>
  <lastName>Deitel</lastName>
  <firstName>Paul</firstName>
</contact>
- <contact>
  <lastName>Nieto</lastName>
  <firstName>Tem</firstName>
</contact>
</contacts>

```

- أصبحت اليوم XML الشكل الموحد لتبادل البيانات على الانترنت، لاسيما في تطبيقات تبادل مجموعات البيانات DataSets من وإلى الزبائن البعيدين Remote clients.



لغة الأسلوب الموسعة XSL

- تُعرّف لغة الأسلوب الموسعة (XSL) Extensible Stylesheet Language كيفية إظهار محتوى مستند XML. إشارة تولتية
- يُحدّد الملف Contacts.xsl طريقة إظهار بيانات الملف Contacts.xml السابق:

```

<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet type="text/xsl" href="Contacts.xsl" ?>
<contacts>
  <contact>
    <lastName>Deitel</lastName>
    <firstName>Harvey</firstName>
  </contact>
  <contact>
    <lastName>Deitel</lastName>
    <firstName>Paul</firstName>
  </contact>
</contacts>

```

1 يُمكن اعتبار العلاقة بين XML وXSL مثل العلاقة بين HTML وCSS حيث تُحدّد XSL قواعد إظهار (أو تنظيم) بيانات وثيقة XML.

```

<contact>
  <lastName>Nieto</lastName>
  <firstName>Tem</firstName>
</contact>
</contacts>

```

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <body>
        <table WIDTH = "200" CELLSPACING = "0" BORDER = "1">
          <Caption> My Contacts</Caption>
          <th WIDTH = "100" >Last Name</th>
          <th WIDTH = "100" >First Name</th>
          <xsl:for-each select = "contacts/contact" >
            <tr>
              <td WIDTH = "100" >
                <xsl:value-of select = "lastName" />
              </td>
              <td WIDTH = "100" >
                <xsl:value-of select = "firstName" />
              </td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>

```

My Contacts	
Last Name	First Name
Deitel	Harvey
Deitel	Paul
Nieto	Tem

مثال تطبيقي

الأهداف التعليمية

- مثال تطبيقي XML
- عنصر التحكم XML control.

ليكن الملف Books.xml والذي يحوي بيانات مجموعة من الكتب:

```
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet type="text/xsl" href="books.xsl" ?>
<database>
  <author>Deitel </author>
  <book>
    <title>C++ How to Program: Third Edition</title>
    <isbn>0-13-089571-7</isbn>
    <pages>1130</pages>
    <description>C++ programming textbook.
    </description>
    <image>Images/cplusplus.jpg</image>
  </book>
  <book>
    <title>Getting Started with Microsoft's Visual C++ 6
      with an Introduction to MFC</title>
    <isbn>0-13-016147-0</isbn>
    <pages>163</pages>
    <description>Introductory MFC programming textbook.
    </description>
    <image>Images/mfcvcplusplus.jpg</image>
  </book>
  <book>
    <title>Java How to Program: Third Edition</title>
    <isbn>0-13-012507-5</isbn>
    <pages>1200</pages>
    <description>Java Programming textbook.
    </description>
    <image>Images/javahttp.jpg</image>
  </book>
</database>
```

```
<?xml version = "1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match = "/">
    <html>
      <head>
        <title>
          <xsl:value-of select = "database/author"/>
        </title>
      </head>
      <body>
        <h1>
          <xsl:value-of select = "database/author"/>
        </h1>
        <div ID = "list">
          <xsl:apply-templates select = "database"/>
        </div>
      </body>
    </html>
  </xsl:template>

  <xsl:template match = "database">
    <table width = "300" cellspacing = "0" border = "1">
      <th width = "100" align = "center">Image</th>
```

```

<th width = "50" align = "center">Title</th>
<th width = "50" align = "center">ISBN</th>
<th width = "50" align = "center">Pages</th>
<th width = "50" align = "center">Description</th>

<xsl:for-each select = "book" >
  <xsl:sort select="title"/>
  <tr>
    <td width = "100" align = "center" valign = "top">
      <xsl:choose>
        <xsl:when test = "image[!='"']">
          <xsl:element name = "image">
            <xsl:attribute name = "src">
              <xsl:value-of select = "image"/>
            </xsl:attribute>
          </xsl:element>
        </xsl:when>
        <xsl:otherwise>
          n/a
        </xsl:otherwise>
      </xsl:choose>
    </td>

    <td width = "50" align = "left" valign = "top">
      <xsl:choose>
        <xsl:when test = "title[!='"']">
          <xsl:value-of select = "title"/>
        </xsl:when>
        <xsl:otherwise>
          n/a
        </xsl:otherwise>
      </xsl:choose>
    </td>

    <td width = "50" align = "center" valign = "top">
      <xsl:choose>
        <xsl:when test = "isbn[!='"']">
          <xsl:value-of select = "isbn"/>
        </xsl:when>
        <xsl:otherwise>
          n/a
        </xsl:otherwise>
      </xsl:choose>
    </td>

    <td width = "50" align = "center" valign = "top">
      <xsl:choose>
        <xsl:when test = "pages[!='"'] ">
          <xsl:value-of select = "pages"/>
        </xsl:when>
        <xsl:otherwise>
          n/a
        </xsl:otherwise>
      </xsl:choose>
    </td>

    <td width = "50" align = "left" valign = "top">
      <xsl:choose>
        <xsl:when test = "description[!='"']">
          <xsl:value-of select = "description"/>
        </xsl:when>
      </xsl:choose>
    </td>
  </tr>
</for-each>




```

```

</xsl:when>
<xsl:otherwise>
  n/a
</xsl:otherwise>
</xsl:choose>
</td>
</tr>
</xsl:for-each>
</table>
</xsl:template>
</xsl:stylesheet>

```

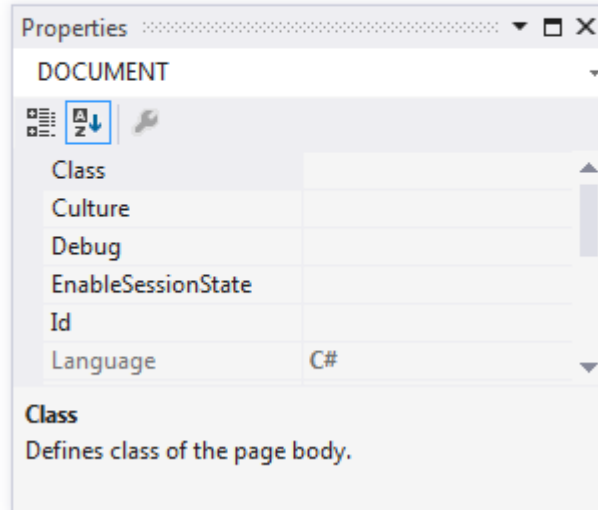
Deitel

Image	Title	ISBN	Pages	Description
	C++ How to Program: Third Edition	0-13- 089571- 7	1130	C++ programming textbook.
	Getting Started with Microsoft's Visual C++ 6 with an Introduction to MFC	0-13- 016147- 0	163	Introductory MFC programming textbook.
	Java How to Program: Third Edition	0-13- 012507- 5	1200	Java Programming textbook.

عنصر التحكم XML Control

- يرتبط عنصر التحكم XML Control مع ملف XML وملف XSL مما يسمح بعرض محتوى الملف XML في صفحة ASPX وفق الأسلوب المحدد في XSL.

Xml - Xml1
Use this control to perform XSL transforms.



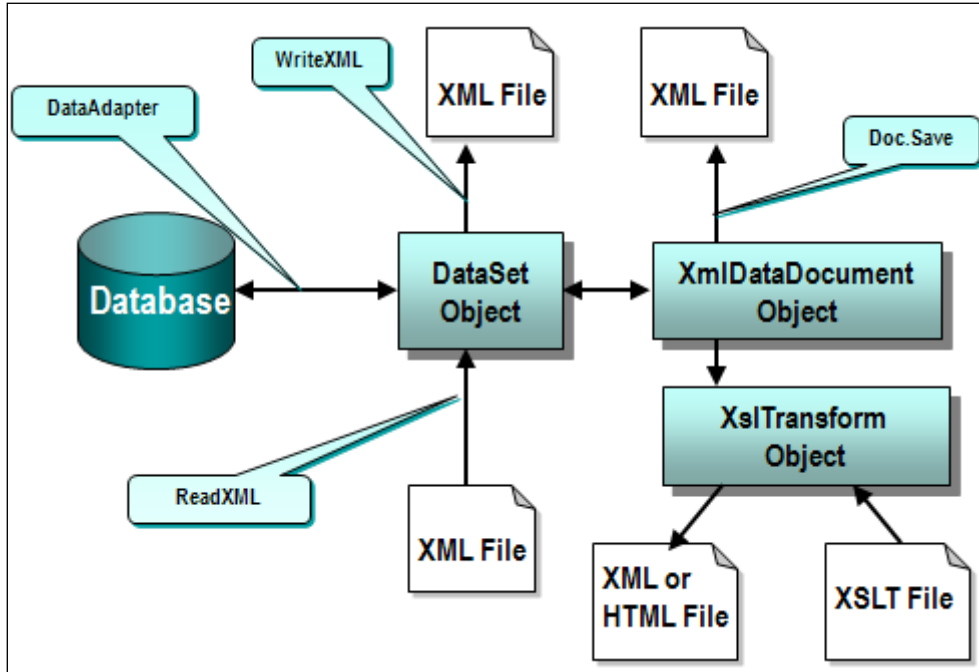
مجموعات البيانات وXML

الأهداف التعليمية

- الاستيراد والتصدير بين مجموعة بيانات وXML.

مجموعات البيانات XML و Datasets

- يُمكن استيراد و تصدير بيانات مجموعة بيانات إلى مستند XML كما يُبين الشكل التالي:



- نقوم في الإجراءية التالية بملء مجموعة بيانات من جدول الكتب ومن ثم كتابة البيانات في المستند :Library.xml

```

protected void Button1_Click(object sender, EventArgs e)
{
    String ConStr;
    ConStr = "Data
Source=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\Library.mdf;Integrated
Security=True;Connect Timeout=30";

    SqlConnection sqlCON = new SqlConnection();
    sqlCON.ConnectionString = ConStr;

    SqlCommand SelectCom = new SqlCommand();

    SelectCom.Connection = sqlCON;
    SelectCom.CommandText = "Select * from Books";

    SqlDataAdapter DBAdapter = new SqlDataAdapter();
    DataSet ResultsDataSet = new DataSet();

    DBAdapter.SelectCommand = SelectCom;

    DBAdapter.Fill(ResultsDataSet);

    ResultsDataSet.WriteXml(Server.MapPath("Library.xml"));
    Response.Redirect("Library.xml");
}
  
```

```

<?xml version="1.0" standalone="yes" ?>
- <NewDataSet>
  - <Table>
    <ISBN>0764544020</ISBN>
    <Title>Visual C# 2005 How to Program</Title>
    <PublicationDate>February 2005</PublicationDate>
    <ImageURL>C.jpg</ImageURL>
    <Precis>This book is for beginners who have no previous
      experience</Precis>
    <CategoryID>1</CategoryID>
  </Table>
  - <Table>
    <ISBN>0764543636</ISBN>
    <Title>Artificial Intelligence A Modern Approach</Title>
    <PublicationDate>July 2006</PublicationDate>
    <ImageURL>IA.png</ImageURL>
    <Precis>This is a great book Most widely used in U.S. universities</Precis>
    <CategoryID>1</CategoryID>
  </Table>
</NewDataSet>

```

- نقوم في الإجرائية التالية بملء مجموعة بيانات من جدول الكتب ومن ثم كتابة البيانات في مستند Library.xml ومن ثم توليد ملف Library.html بعد تطبيق الأسلوب :Library.xsl

```

protected void Button2_Click(object sender, EventArgs e)
{
    String ConStr;
    ConStr = "Data
Source=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\Library.mdf;Integrated
Security=True;Connect Timeout=30";

    SqlConnection sqlCON = new SqlConnection();
    sqlCON.ConnectionString = ConStr;

    SqlCommand SelectCom = new SqlCommand();
    SelectCom.Connection = sqlCON;
    SelectCom.CommandText = "Select * from Books";

    SqlDataAdapter DBAdapter = new SqlDataAdapter();
    DataSet ResultsDataSet = new DataSet();
    DBAdapter.SelectCommand = SelectCom;

    DBAdapter.Fill(ResultsDataSet);

    XmlDocument xmlDoc = new XmlDocument(ResultsDataSet);

    XslTransform xslTran = new XslTransform();
    xslTran.Load(Server.MapPath("Library.xsl"));
    XmlTextWriter writer = new
XmlTextWriter(Server.MapPath("Library.html"), System.Text.Encoding.UTF8);
    xslTran.Transform(xmlDoc, null, writer);
    writer.Close();
    Response.Redirect("Library.html");
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match = "/">
    <html>
      <head>
        <title>Coho Library</title>
      </head>
      <body>
        <table width = "100%" cellpadding = "0" border = "1">
          <th width = "200" align = "center">Image</th>
          <th width = "30%" align = "center">Title</th>
          <th width = "25%" align = "center">ISBN</th>
          <th width = "30%" align = "center">Description</th>

          <xsl:for-each select = "NewDataSet/Table" >
            <xsl:sort select="Title"/>
            <tr>
              <td width = "200" align = "left" valign = "top">
                <xsl:choose>
                  <xsl:when test = "ImageURL[!='']">
                    <xsl:element name = "image">
                      <xsl:attribute name = "src">
                        <xsl:text> Images/</xsl:text>
                        <xsl:value-of select = 'ImageURL' />
                      </xsl:attribute>
                    </xsl:element>
                  </xsl:when>
                  <xsl:otherwise>
                    n/a
                  </xsl:otherwise>
                </xsl:choose>
              </td>

              <td width = "25%" align = "left" valign = "top">
                <xsl:choose>
                  <xsl:when test = "Title[!='']">
                    <xsl:value-of select = "Title"/>
                  </xsl:when>
                  <xsl:otherwise>
                    n/a
                  </xsl:otherwise>
                </xsl:choose>
              </td>

              <td width = "10%" align = "left" valign = "top">
                <xsl:choose>
                  <xsl:when test = "ISBN[!='']">
                    <xsl:value-of select = "ISBN"/>
                  </xsl:when>
                  <xsl:otherwise>
                    n/a
                  </xsl:otherwise>
                </xsl:choose>
              </td>



              <td width = "60%" align = "left" valign = "top">
                <xsl:choose>
                  <xsl:when test = "Precis[!='']">
                    <xsl:value-of select = "Precis"/>
                  </xsl:when>

```

```

        <xsl:otherwise>
            n/a
        </xsl:otherwise>
    </xsl:choose>
</td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Image	Title	ISBN	Description
	Artificial Intelligence A Modern Approach	0764543636	This is a great book Most widely used in U.S. universities
	Visual C# How to Program	0764544020	This book is for beginners who have no previous experience

اقتراحات وتمارين

المشروع الأول: تصميم وتنفيذ موقع تحميل أغاني وموسيقى

المطلوب تصميم وتنفيذ موقع لتنزيل الأغاني والموسيقى يشابه مثلاً الموقع

www.melody4arab.com

وبحيث يقدم الخدمات الأساسية التالية:

- إمكانية البحث عن أغنية معينة وفق وذلك حسب اسم الأغنية أو اسم المغني.
- إمكانية استعراض الألبومات الموجودة لمغني معين.
- تظهر نتائج البحث بشكل جدولي يظهر لكل مغني: عنوان الألبوم، الأغاني، صورة الألبوم.
- وجود تصنيف للمغنيين والأغاني حسب الأحرف الابجدية بحيث عند النقر على حرف ابجدي معين على مستوى المغنيين أو الأغاني نحصل على جميع الأغاني أو المغنيين الذي يبدأ اسمهم بهذا الحرف (على الصفحة الرئيسية وباقي الصفحات لاحظ الأحرف الابجدية وإمكانية الوصول لاسم المغني أو الأغنية ضمن الموقع المذكور أعلاه). مثلاً: عند الضغط على حرف (ن) على مستوى المغنيين نحصل على جميع المغنيين الذي يبدأ اسمهم بحرف (ن). وكذلك بالنسبة للأغاني.
- عند النقر على عنوان الأغنية أو صورة الألبوم ننتقل إلى صفحة المغني والتي تحوي إضافة إلى المعلومات السابقة لمحة عن المغني وإمكانية تنزيل الأغنية.
- لا يستطيع الزائر تنزيل أي أغنية مالم يكن مسجلاً مسبقاً في الموقع وبالتالي يجب أن يكون لدينا صفحة sign up تحوي مجموعة من معلومات الزبون لإمكانية الدخول للموقع وتنزيل الأغاني بعد اضافتها الى عربة التسوق.
- إدارة عربة تسوق Shopping Basket والتي تحوي الأغاني التي تم طلبها مع إمكانية تعديل عدد الأغاني أو الحذف.
- صفحة طلبية Order تحوي بعض المعلومات عن المستخدم: الاسم، العنوان، البريد الإلكتروني، رقم بطاقة الاعتماد.
- يجب أن تكون صفحات الموقع كلها وفق MasterPage واحد.

المشروع الثاني: مشروع بناء تطبيق وب لتسجيل الطلاب لموادهم ضمن الجامعة الافتراضية السورية

توصيف المشروع:

- الهدف من هذا المشروع هو بناء تطبيق يسمح لطلاب الجامعة الافتراضية السورية بتسجيل موادهم بشكل مباشر على الانترنت وذلك حسب المواد المتاحة لهم في برنامجهم.
- يتألف المشروع من قسمين قسم الطالب وقسم الإدارة.

أولاً-قسم الإدارة

- يقوم مدير النظام بتعريف جميع البرامج وجميع المواد لكل برنامج (رمز المادة، اسم المادة، عدد الجلسات، سعر المادة).
- يُحدد مدير النظام من أجل كل برنامج في الجامعة المواد المفتوحة للفصل الحالي أي المفتوح التسجيل فيه.
- ومن أجل كل مادة مفتوحة: عدد الطلاب الأعظمي الممكن تسجيلهم في المادة للفصل الحالي.
- يُمكن لمدير النظام إظهار من أجل أي مادة في أي برنامج أسماء الطلاب المسجلين في المادة.

ثانياً- قسم الطالب

1. بداية يجب أن يقوم الطالب بالتسجيل ضمن الموقع المطلوب من خلال صفحة تحوي معلومات التسجيل الأساسية وهي: الاسم الكامل، اسم المستخدم، كلمة المرور، البريد الإلكتروني، البرنامج الذي يدرس فيه الطالب، ID الطالب (يتم توليده تلقائياً ويتم اعطائه للطلاب لاستخدامه فيما بعد).
 2. بعد عملية التسجيل يستطيع الطالب الدخول إلى الموقع من خلال صفحة يطلب منه إدخال اسم المستخدم وكلمة السر.
 3. بعد تسجيل الدخول يظهر للطالب صفحة تُظهر له المواد التي يُمكن له التسجيل عليها (مواد برنامجه المفتوحة حالياً).
- يكون لكل مادة المعلومات التالية: رمز المادة، اسم المادة، عدد الطلاب المسجلين بهذه المادة لغاية الآن، السعة القصوى للطلاب لهذه المادة، سعر المادة.
1. يقوم الطالب إذا كان من الممكن ان يسجل بمادة (أي لا يزال هناك شواغر) بالتسجيل بالضغط وحيث تُضاف المادة إلى قائمة تسجيل Registration Cart حيث يتم تجميع المواد مع المبلغ الإجمالي لكافة المواد التي قام الطالب بتسجيلها ضمن Registration Cart.
 2. لا يحق للطلاب التسجيل على أكثر من خمسة مواد.
 3. يُمكن للطلاب دائماً معاينة قائمة تسجيله وإجراء تعديل عليها إضافة/حذف.

ملاحظات:

- عندما يقوم الطالب بالتسجيل بمادة بالطبع فإن عدد المقاعد للمادة سينقص واحد.
- تنتهي عملية التسجيل بمادة بعملية الإضافة إلى Registration Cart والتي تُظهر المواد التي قام بتسجيلها والمبلغ الإجمالي.
- يُمكن للطلاب إلغاء تسجيله في مادة وعندها يُحرر المقعد المحجوز له.

المشروع الثالث: تصميم وتنفيذ موقع لشراء الكتب

المطلوب تصميم وتنفيذ موقع لعرض وشراء الكتب.

يُمكن الاستئناس بمواقع شهيرة مثل:

<http://www.amazon.co.uk>

يجب أن يقدم الموقع الخدمات الأساسية التالية للزائر:

- إمكانية البحث عن كتاب معين وذلك حسب فئة الكتاب و/أو اسم الكتاب و/أو المؤلفين المشتركين و/أو الناشر و/أو تاريخ النشر.
- عند النقر على عنوان الكتاب أو صورة الكتاب ننتقل إلى صفحة معلومات الكتاب (صورة غلاف الكتاب، الناشر، المؤلفون، لمحة، السعر).
- لا يستطيع الزائر شراء أي كتاب ما لم يكن مسجل مسبقاً في الموقع وبالتالي يجب أن يكون لدينا صفحة sign up تحوي مجموعة من معلومات الزبون (الاسم، البريد، ...).
- يمكن لكل مشترك تصنيف كتاب (1، 2، 3، 4، 5) وإضافة تعليق.
- يظهر في صفحة الكتاب عدد من النجوم (من 1 إلى 5) هي وسطي تقييمات المشتركين.
- يمكن في صفحة الكتاب قراءة تعليقات المشتركين.
- يحوي الموقع إدارة كاملة لعربة تسوق.
- يقوم الزبون بعد تعبئة عربة التسوق بإجراء submit فقط (أي لن ندخل في مسائل الدفع الإلكتروني).

أما بالنسبة لمدير النظام:

- إدارة فئات الكتب/ الكتب/ المؤلفين/ الناشرين.
- إصدار تقرير إحصائي يومي/ شهري بأرباح الموقع (مجموع قيم الكتب المشتراة).