



دبلوم الامن السيبراني

# مبادئ برمجة صفحات الإنترنت



## ❖ الأهداف التفصيلية للمقرر

بنهاية هذا المقرر ستكون المتدربة قادرة وبكفاءة على أن:

- تدوين احتياجات المستخدم
- تحديد خطوات تصميم صفحات الانترنت
- معرفة أساسيات عمل شبكة الانترنت
- معرفة مكونات صفحة الانترنت وطريقة استعراضها
- كتابة وسوم لغة HTML
- استخدام برنامج لتحرير وإنشاء صفحات الانترنت
- معرفة المتغيرات في البرنامج JavaScript
- كتابة الأوامر البرمجية
- معرفة طريقة عمل نماذج الانترنت Forms
- تتبع مدخلات المستخدم والتحقق من صحتها

## ❖ فهرس الكتاب

### الفصل الأول: مقدمة عن الشبكة العالمية للمعلومات Internet

٩	نشأة الإنترنت وتطورها .....
١٩	خدمات الإنترنت internet services .....
٢٠	نموذج خادم / عميل ( client/server model ) .....

### الفصل الثاني: مبادئ تصميم مواقع الانترنت

٢٤	أنواع مواقع الشبكة العنكبوتية العالمية WWW .....
٢٨	تحديد أهداف الموقع site goals .....
٢٩	تحديد مكونات الموقع site components .....
٣٢	خطة بناء الموقع site construction plan .....

### الفصل الثالث: إنشاء الصفحات باستخدام لغة الترميز HTML

٣٩	الفرق بين HTML و XHTML .....
٤٠	البنية المعيارية XHTML والتحقق من الوثائق .....
	تنسيق النصوص .....
٤٩	تنسيق الصور .....
٥١	الروابط Hypertext Links .....
٥٣	القوائم Lists .....
٥٦	الجداول .....

### الفصل الرابع: مهارات متقدمة في XHTML

٦٢	النماذج Forms .....
	HTML5 .....
	الفيديو والصوت Video and Audio .....
	Web Forms 2.0 .....

### الفصل الخامس: أسلوب الصفحات المتتالي CSS

٨١	مقدمة .....
٨١	مستويات أسلوب الصفحات .....
٨٢	تنسيق تحديد الأسلوب .....
٨٣	أشكال المحددات .....
٨٧	الخاصية والقيمة Value, Property .....
٨٩	خصائص الخط .....
٩٢	خصائص القوائم .....
	خصائص أساليب العناصر الأخرى .....
١٠٦	المؤثرات <SPAN> <DIV> .....

### الفصل السادس: أساسيات لغة JavaScript

١١٤	مقدمة .....
-----	-------------

١١٥	الخصائص العامة والشكل
١١٦	التعليقات
١١٦	الأنماط والعمليات والعناصر
١٢٤	الإدخال والإظهار
١٢٧	تعليمات التحكم
١٣٠	المصفوفات Arrays
١٣٣	الوظائف Functions
١٣٤	مطابقة النماذج Pattern Matching

### الفصل السابع: التفاعل بين HTML و JavaScript

١٤٠	التعامل مع عناصر HTML من خلال JavaScript
١٤٣	الإحداثيات Events
١٥١	التحقق من الإدخال
١٥٦	مثال تطبيقي

### الفصل الثامن: الوثائق الديناميكية مع Java Script

	طرق تموضع العناصر
١٦٦	تحريك العناصر
	التعامل مع العناصر (الإظهار والإخفاء وتغيير لون الخط)
	المحتوى الديناميكي
	تكديس العناصر
	مؤشر الفأرة
	التحكم بحركة العناصر

### الفصل التاسع: أساسيات JQuery

	أساسيات JQuery
	محددات العناصر Selectors
	الواصفات والأساليب في JQuery
	أسلوب الصفحات المتتالي CSS
	مهارات JQuery
	التعامل مع الأحداث

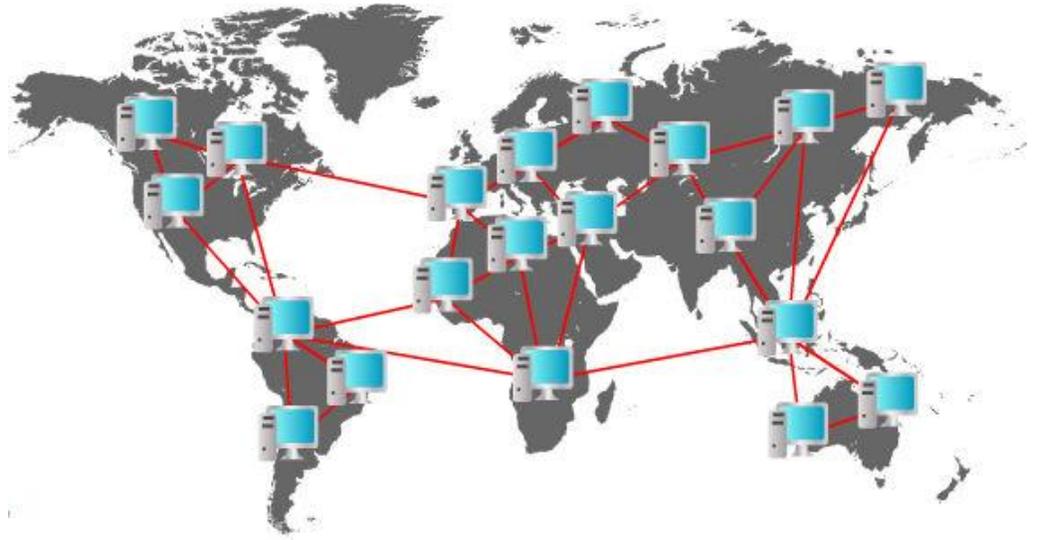
# أولاً: مقدمة عن الشبكة العالمية للمعلومات Internet

في هذا الفصل سنتعرف على المواضيع التالية:

- نشأة الانترنت وتطورها
- خدمات الانترنت **Internet Services**
- نموذج خادم العميل **client / server model**

## ❖ نشأة الانترنت وتطورها

أصبحت شبكة الانترنت العصب الرئيسي المحرك لعالمنا الرقمي المتطور ولا يخلو جزء من يومنا تقريباً من تأثره أو استخدامه المباشر لشبكة الانترنت، ولكن هل تساءلت من قبل عن ماهية الانترنت؟ تتألف الانترنت من مجموعة ضخمة من الحواسيب المرتبطة مع بعض عبر شبكات الاتصالات، تختلف هذه الحواسيب عن بعضها بحجمها ومصنعها وأنظمتها. بشكل مُضطرد تزداد كمية البيانات التي تنتقل عبر شبكة الإنترنت بسرعات من مرتبة الغيغابت في الثانية اعتماداً على شبكات الألياف البصرية، وقد شكل استحواذ شبكة الإنترنت على المشهد العالمي للاتصالات، ظاهرة تستحق الملاحظة فقد كان حجم المُعطيات المنقولة على شبكة الإنترنت يشكل ٥١% من إجمالي حجم المُعطيات المنقول عالمياً في العام ١٩٩٣، ثم ارتفعت النسبة لتصل إلى ٩٧% في العام ٢٠٠٧، وتجاوز ٩٧% في العام ٢٠٠٧، تواصل شبكة الإنترنت النمو مدفوعةً بكميات أكبر من المعلومات الناتجة عن التجارة والترفيه وشبكات التواصل الاجتماعي.



## • محطات تاريخية

يبدأ تاريخ شبكة الإنترنت مع تطور الحواسيب الرقمية في الخمسينيات من القرن الماضي، طوّرت المفاهيم الأولية لشبكات الحاسوب في العديد من مختبرات علوم الحاسوب في الولايات المتحدة والمملكة المتحدة وفرنسا، منحت وزارة الدفاع الأمريكية عقوداً في وقت مبكر من الستينيات لتطوير أنظمة شبكات تبادل الرزم، وتضمّن ذلك تطوير شبكة الأربانت، تم إرسال أول رسالة عبر شبكة الأربانت بإشراف البروفسور ليونارد كلينروك من جامعة كاليفورنيا في لوس أنجلوس إلى معهد ستانفورد للأبحاث، وكانت هناك تواريخ سطرت في عالم الإنترنت تاريخاً لا ينسى، وباتت علامات مهمة على مدى تطور الإنترنت والويب وصولاً إلى عالم الهواتف الذكية.

- ١٩٥٨ شركة، بل تخرع أول مودم يسمح بنقل البيانات الرقمية عبر خط الهاتف.
- ١٩٦١ ليونارد كلينروك من معهد ماساتشوستس للتقنية ينشر أول بحث عن نظرية استعمال تحويل الطرود لنقل البيانات.
- ١٩٦٢ تشريع (إحدى وكالات وزارة الدفاع الأميركية) في أبحاثها لتدشين شبكة "أربانت".
- ١٩٦٤ ليونارد كلينروك ينشر كتاباً بشأن استعمال تحويل الطرود في إنشاء شبكة.
- ١٩٦٧ أول مؤتمر حول شبكة "أربانت".
- ١٩٦٩ ربط أوائل الحواسيب من ٤ جامعات أميركية وفقاً لـ 'واجهة معالجة الرسائل لصاحبها ليونارد كلينروك.
- ١٩٧١ ٢٣ حاسوباً مرتبطاً بشبكة "أربانت" يرسل أول بريد إلكتروني.
- ١٩٧٢ ميلاد أول تنظيم مكلف بإدارة الإنترنت.
- ١٩٧٣ إنجلترا والنرويج ترتبطان بالإنترنت بحاسوب لكل منهما.
- ١٩٧٩ استحداث مجموعات (News Groups) منتديات نقاش من طرف طلبة أميركيين.
- ١٩٨٢ تعريف بروتوكول TCP/IP وظهور كلمة "إنترنت".
- ١٩٨٣ أول خادم نطاقات (DNS).
- ١٩٨٤ ١٠٠٠ حاسوب مرتبط بالشبكة.
- ١٩٨٧ ١٠,٠٠٠ حاسوب مرتبط بالشبكة.
- ١٩٨٩ ١٠٠,٠٠٠ حاسوب مرتبط بالشبكة.
- ١٩٩٠ اختفاء شبكة "أربانت".
- ١٩٩١ الإعلان للجمهور عن World Wide Web، وظهور لغة كتابة صفحات الإنترنت (HTML).
- ١٩٩٢ مليون حاسوب مرتبط بالشبكة.
- ١٩٩٣ ظهور أول متصفح انترنت NCSA Mosaic.

- ١٠١٩٩٦ مليون حاسوب مرتبط بالشبكة.
- ٢٠٠٠ انتشار الإنترنت على مستوى العالم.

وبعد عام ٢٠٠٠ أصبحت البنية التحتية للإنترنت شبه ثابتة، وأصبح التطور واضحاً أكثر على مستوى البرمجيات والخدمات التي تقدمها شبكة الإنترنت، فتطورت صفحات الإنترنت الثابتة إلى صفحات ديناميكية تعتمد على لغات البرمجة المختلفة، التي مهدت إلى ظهور جيل جديد من خدمات الإنترنت ومواقع التواصل الاجتماعي في فيسبوك وتويتر وقوقل بلس.

### • ماهي الإنترنت وألية عملها

الانترنت Internet بكل بساطة هو شبكة كأي شبكة من شبكات الحاسوب، ولكنه شبكة كبيرة بعض الشئ يتصل بها أكثر من ٢١ مليار جهاز حول العالم، هذا وفقاً لإحصائية في العام ٢٠٢٠، فهو البنية التحتية الضخمة التي يتصل بها كافة الأشخاص حول العالم، والتي يقومون من خلالها بالتواصل مع بعضهم البعض وارسال الملفات والإيميلات ومشاهدة أفلامهم المفضلة وحتى متابعة حالتهم الصحية، حيث إن بداخل شبكة الويب الضخمة التي نقرأ هذا المقال من خلالها هناك العديد من الشبكات الأخرى التي تربط بين الأفراد، والشركات والمنظمات والحكومات.

### • وإليك بعض النقاط المهمة حول الإنترنت:

- الإنترنت هو شبكة لامركزية أي لا يوجد من يتحكم بها، ويمكن لأي جهاز كمبيوتر الدخول في هذه الشبكة.
- يمكن لأي مجموعة من أجهزة الكمبيوتر أن تنشئ شبكتها الخاصة، ويمكن أن تتصل بالشبكة الأم الإنترنت أم لا حسب الرغبة، لذلك يمكننا أن نطلق على الإنترنت أنه شبكة الشبكات أو Network of Networks.
- في الحقيقة يمكن استخدام الإنترنت من الكثير من أنواع الأجهزة وليس أجهزة الكمبيوتر التقليدية فقط، ولذا مؤخراً ظهر مفهوم إنترنت الأشياء، وهو الذي يمكن أي جهاز من دخول الشبكة وربطه مع أجهزة أخرى في شبكة واحدة خاصة.
- الإنترنت يعمل وفقاً لمجموعة من البروتوكولات، أشهرها IP, TCP, FTP، وهي عبارة عن مجموعة من القوانين والقواعد التي يعمل وفقاً لها الإنترنت، ويجب أن يوافق عليها كل جهاز متصل بالشبكة.
- كل جهاز يتصل بشبكة الإنترنت يجب أن يمتلك IP ADDRESS خاص وغير مكرر، ومن خلاله يتم تنظيم عملية سير ومشاركة البيانات بين الأجهزة بدون اختلاط أو ارتباك، وهذا واحد من أهم بروتوكولات الإنترنت.
- تشبه شبكة الويب بتطورها الحالي الشبكة العنكبوتية، فهي عبارة عن مجموعة هائلة من الخيوط (الاسلاك) المتصلة معا.
- وتدمير جزء من هذه الشبكة لا يعني عطل النظام كله، لأن عملية الاتصال هنا سوف تأخذ مسارات أخرى حول الجزء المدمر لكي تصل لغايتها المنشودة.

## • المفاهيم والمصطلحات الأساسية بالإنترنت

### متصفح Browser

الأداة أو البرنامج الأساسي لكل من يريد تصفح مواقع الإنترنت، وبدونها فإن عملية الدخول للإنترنت ستكون مستحيلة، وهو برنامج يتم تثبيته على أجهزة الكمبيوتر والأجهزة النقالة بشكل مسبق في عند بعض أنظمة تشغيل الحاسوب، مع وجود متصفحات أخرى مستقلة يمكن تحميلها من الإنترنت، المتصفحات عبارة عن حزمة برامج مجانية، أو تطبيقات للهواتف المحمولة تتيح لك عرض صفحات الويب، والرسومات، ومعظم المحتوى عبر الإنترنت، تتضمن متصفحات الويب الأكثر شعبية جوجل كروم وفايرفوكس وإنترنت إكسبلورر وسفاري، ولكن هناك العديد من المتصفحات الأخرى ذات الشعبية المنخفضة.

### صفحة ويب Webpage

صفحة الويب هي ما تراه في المتصفح عندما تكون متصلاً بالإنترنت، يمكنك التفكير في صفحة الويب كصفحة في مجلة ورقية، سوف ترى النصوص والصور والرسوم البيانية، وصلات، والإعلانات وأكثر على أي صفحة تقوم بعرضها، وهي بالضبط مبدأ التصفح على الإنترنت.

في كثير من الأحيان، تنقر أو تنقر على منطقة معينة من صفحة ويب لتوسيع المعلومات أو الانتقال إلى صفحة ويب ذات صلة، يؤدي النقر على رابط – وهو مقتطف من النص الذي يظهر بلون مختلف عن بقية النص – إلى نقلك إلى صفحة ويب مختلفة، إذا كنت ترغب في العودة، يمكنك استخدام الأسهم الموجود في أعلى المتصفح.

## URL

وهي اختصاراً لعبارة "عناوين المواقع الموحدة" Uniform Resource Locators وتعني عناوين متصفح الويب لصفحات الإنترنت والملفات، باستخدام عنوان URL يمكنك تحديد موقع صفحات وملفات محددة ومراجعتها عبر متصفح الويب.

يمكنك إيجاد عنوان URL في كل مكان مثل إدراجها في الجزء السفلي من بطاقات العمل، أو على شاشات التلفزيون خلال الفواصل الإعلانية، أو مكتوبة في الوثائق التي تقرأها على شبكة الإنترنت، أو التقاطها من قبل واحدة من محركات البحث على الإنترنت.

**وتكون صيغة عنوان الموقع مثل:**

`http://www.examplewebsite.com/mypage`

**ويمكن اختصارها هكذا:**

`www.examplewebsite.com/mypage`

في بعض الأحيان تكون أطول وأكثر تعقيداً، ولكنها تتبع جميع القواعد المعترف بها لتسمية عناوين

URL.

**تتكون عناوين URL من ثلاثة أجزاء رئيسية لمعالجة صفحة أو ملف على الإنترنت:**

- بروتوكول الإنترنت: protocol وهو الجزء الذي ينتهي بعلامة // معظم صفحات الويب تستخدم بروتوكول http أو https ، مع وجود بعض البروتوكولات الأخرى.
- نطاق المضيف: Domain الذي غالباً ما ينتهي بـ .com أو .net أو .edu أو .org.
- اسم الملف أو اسم الصفحة نفسه Webpage .

## نطاق Domain

اسم النطاق Domain Name هو جزء فريد من عنوان URL يستند إلى الترتيب الأبجدي، يمكن تسجيل اسم نطاق موقعك رسمياً مع مسجل نطاق من قبل شخص أو مؤسسة تجارية أو مؤسسة غير ربحية، ويتكون اسم النطاق من جزأين:

– الكلمة أو العبارة الأبجدية الفعلية؛ على سبيل المثال “Arageek”

– اسم نطاق المستوى الأعلى الذي يحدد أي نوع من الموقع هو؛ على سبيل المثال، .com. “للنطاقات التجارية” .org، “للمنظمات”، .edu. “للمؤسسات التعليمية”

عند وضع هذين الجزأين معاً Arageek.com يكون هذا هو اسم الموقع أو النطاق على الإنترنت.

## SSL

وهو اختصار مأخذ التوصيل الآمنة Secure Sockets Layer وهو عبارة عن بروتوكول أو طبقة تشفير آمنة يتم استخدامها لجعل البيانات آمنة عند نقلها عبر الإنترنت.

يستخدم SSL بشكل خاص على مواقع التسوق للحفاظ على البيانات المالية آمنة، ولكن يستخدم أيضاً على أي موقع يتطلب بيانات حساسة “كلمات المرور”.

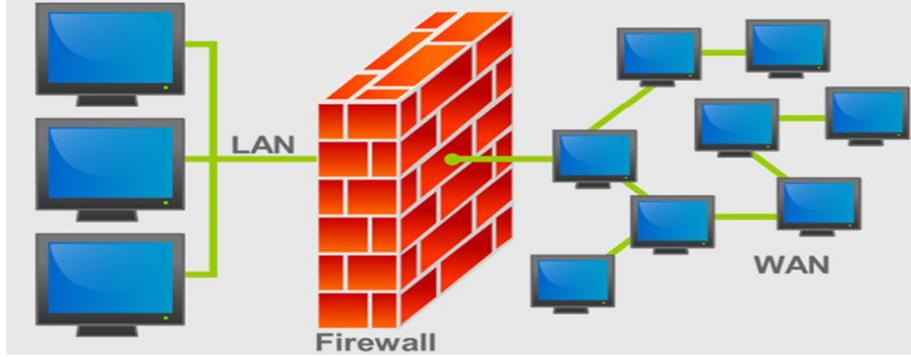
يمكنك التأكد من أنّ الموقع المعني يستخدم هذه الطبقة من التشفير عن النظر للعنوان URL ورؤية HTTPS قبل عنوان الموقع.

## مخدم بروكسي Proxy Server

الخادم الوكيل هو خادم ويب يعمل كدرع لمستخدمي الويب، وإخفاء المعلومات ذات الصلة مثل (عنوان الشبكة، والموقع، وما إلى ذلك) من مواقع الويب وغيرها من المستخدمين، في سياق الإنترنت، يتم استخدام خوادم بروكسي للمساعدة في التصفح المجهول، حيث يعمل الخادم الوكيل كمخزن مؤقت بين الباحث وموقع الويب المقصود، مما يسمح للمستخدمين بعرض المعلومات دون تعقبها.

## جدار الحماية Firewall

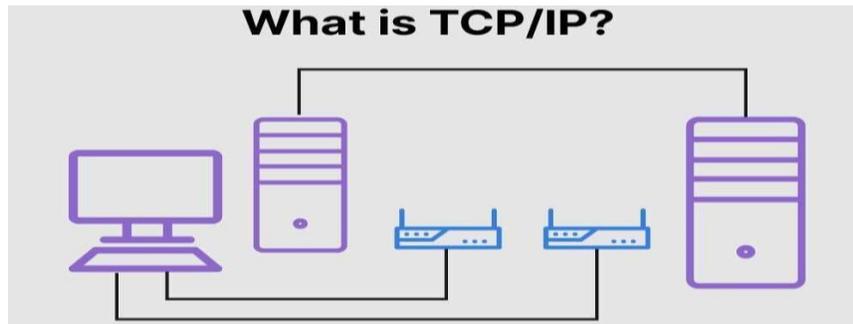
جدار الحماية هو إجراء أمني مصمم للحفاظ على أجهزة الكمبيوتر والمستخدمين والشبكات غير المصرح لهم من الوصول إلى البيانات على كمبيوتر آخر أو شبكة أخرى، جدران الحماية ذات أهمية خاصة لمستخدمي الويب؛ لأنها يمكن أن يحمي المستخدم من برامج التجسس الخبيثة والقراصنة أثناء الاتصال بالإنترنت.



جدار الحماية وتعامله مع شبكات الانترنت

## TCP/IP

بروتوكول التحكم في الإرسال (TCP) Transmission Control Protocol ، وبروتوكول الإنترنت (IP) Internet Protocol، هما بروتوكولات شبكات الكمبيوتر القياسية، والبروتوكول عبارة عن مجموعة متفق عليها من الإجراءات والقواعد، عند اتصال جهازي كمبيوتر بنفس البروتوكولات – نفس مجموعة القواعد – يمكن أن يفهم كل منهما الآخر ويتبادلان البيانات، ومع ذلك، فإنّ TCP و IP يشيع استخدامها معاً، مع أنّ TCP/IP أصبح مصطلحاً معيارياً للإشارة إلى مجموعة البروتوكولات المستخدمة في الاتصال بالإنترنت.



بروتوكول التحكم في الإرسال (TCP) Transmission Control Protocol

## ملفات الإنترنت المؤقتة Temporary Internet Files

مهمة جداً في سياق البحث على الويب، كل صفحة ويب يقوم المستخدم بزيارة البيانات المخزنة (صفحات، ملفات الفيديو، والصوت، وما إلى ذلك) في مجلد ملف معين على القرص الصلب للكمبيوتر الخاص به، يتم تخزين هذه البيانات مؤقتاً بحيث في المرة التالية التي يزور فيها المستخدم صفحة الويب هذه، سيتم تحميلها بسرعة وكفاءة؛ نظراً لأن الكثير من البيانات قد تم تحميلها بالفعل عبر ملفات الإنترنت المؤقتة بدلاً من خادم موقع الويب.

يمكن لملفات الإنترنت المؤقتة في نهاية المطاف أن تأخذ مساحة كبيرة جداً من الذاكرة على جهاز الكمبيوتر الخاص بك، لذلك، من المهم مسحها كل فترة والأخرى.

## HTTP و HTTPS

HTTP هو اختصار "بروتوكول نقل النص التشعبي" Hypertext Transfer Protocol ، ومعيار الاتصالات ونقل البيانات عبر صفحات الويب، عندما تحتوي صفحة الويب على هذه البادئة، يجب أن تعمل الروابط والنصوص والصور بشكل صحيح في متصفح الويب.

HTTPS هو اختصار "بروتوكول نقل النص التشعبي الآمن" Hypertext Transfer Protocol Secure ، ويشير إلى أن صفحة الويب لديها طبقة خاصة من التشفير المضافة لإخفاء المعلومات الشخصية وكلمات السر الخاصة بك من الآخرين.

عند تسجيل الدخول إلى حسابك المصرفي عبر الإنترنت أو موقع التسوق الذي تدخل فيه معلومات بطاقة الائتمان، ابحث عن "HTTPS" في شريط عنوان URL فهذا يعني أن الموقع آمن ويمكنك الثقة فيه.

## غير متصل بالإنترنت OFFLINE

يشير هذا المصطلح إلى عدم الاتصال بالإنترنت، بمعنى أنه لا يوجد ما يربط بينك وبين الأجهزة الأخرى على الشبكة، وعدم تحميل الصفحات والمواقع لقراءتها وتصفحها، بعض المواقع والتطبيقات تتيح لك العمل بوضع الأوفلاين Offline ، وعند الانتهاء من عملك وعند معاودة الاتصال Online يتم مزامنة ما أنجزته بشكل تلقائي على شبكة الإنترنت.

## ما هو الارتباط التشعبي Hyperlink

والمعروف باسم لبنة البناء الأساسية للشبكة العالمية most basic building block of the World Wide Web، هو رابط من وثيقة واحدة، صورة، كلمة، أو صفحة ويب التي تربط المواقع ببعضها البعض على شبكة الإنترنت، تتيح لنا تصفح المواقع وقراءة المعلومات على شبكة الإنترنت بسرعة وسهولة، يمكن اختصارها بأنّها البنية التي تم بناء الويب عليها.

## استضافة المواقع الإلكترونية Web Hosting

مضيف الويب هو نشاط تجاري/شركة توفر مساحة وتخزين واتصال لتمكين مستخدمي الإنترنت من مشاهدة مواقع الويب، استضافة المواقع عادةً ما تشير إلى أعمال استضافة مساحة لمواقع ويب نشطة، وهي خدمة توفر مساحة على خادم الويب، فضلاً عن اتصال إنترنت مباشر، وبالتالي فإنّ الموقع يمكن أن ينظر إليها والتفاعل معها من قبل أي شخص لديه اتصال بالإنترنت.

هناك أنواع مختلفة من استضافة المواقع، من موقع أساسي من صفحة واحدة يحتاج فقط قليلاً من المساحة، إلى فئة العملاء الذين يحتاجون إلى مراكز بيانات كاملة لخدماتها.

العديد من شركات استضافة المواقع توفر لوحة القيادة للعملاء تسمح لهم بالتحكم في جوانب مختلفة من خدمات استضافة المواقع الخاصة بهم، وهذا يشمل بروتوكول نقل الملفات FTP ، وتثبيت نظام إدارة المحتوى المختلفة CMSI ، وملحقات حزمة الخدمات .SPE

## التشفير والمصادقة Encryption & Authentication

**التشفير Encryption** هو الخط الرياضي للبيانات بحيث تكون مخفية عن المستخدمين، يستخدم التشفير الصيغ الرياضية المعقدة لتحويل البيانات الخاصة إلى نصوص أو رموز لا معنى لها، ويمكنك أنت فقط فكها باستخدام البرامج المخصصة لذلك.

التشفير هو الأساس لكيفية استخدام الإنترنت كخط دفاع لإجراء أعمال موثوق بها، مثل الخدمات المصرفية عبر الإنترنت، وشراء بطاقات الائتمان عبر الإنترنت.

**ترتبط المصادقة Authentication** مباشرةً بالتشفير، المصادقة Authentication هي الطريقة المعقدة التي تتحقق بها أنظمة الكمبيوتر من أنك شخص طبيعي وليس روبوتاً.

## حوسبة سحابية Cloud Computing

بدأت الحوسبة السحابية كمصطلح لوصف البرامج الموجودة على الإنترنت، بدلاً من شرائها وتثبيتها على جهاز الكمبيوتر الخاص بك، البريد الإلكتروني المستند إلى الويب هو مثال على الحوسبة السحابية، يتم تخزين كل البريد الإلكتروني للمستخدم والوصول إليها في سحابة على الإنترنت من أي مكان.

خدمات الإنترنت (Internet service) هي الوسيلة التي يتم من خلالها نقل البيانات Data من المصدر الرئيسي (موفر الخدمة) إلى الأجهزة العامة للأشخاص في العالم، ومن خلال خدمة الإنترنت يمكن للأشخاص الاتصال بالإنترنت والاستفادة من الخدمات عن طريق تبادل الاتصال بين الأجهزة العامة، وتضم خدمات الإنترنت بشكل عام: حزم البرامج والتطبيقات على اختلاف أنواعها، وحسابات البريد الإلكتروني، ومواقع الويب الشخصية أو الصفحات الرئيسية.

### • أهمية توفير خدمات الإنترنت

١. ترك أثر إيجابي على الاقتصاد: ساهمت خدمات الإنترنت وانتشارها على مستوى واسع في رفع نسب التسويق التجاري العام وترك أثر إيجابي على المؤسسات والمجتمع والدولة.
٢. توفير الجهد والوقت والمال: وفرت خدمات الإنترنت على كل من أرباب العمل والعمليين الوقت والجهد، وخفضت من بعض التكاليف والإنفاقات، نظراً لتسهيلها عملية الاتصال والتواصل.
٣. ترك أثر إيجابي على الثقافات والمجتمع: مكّنت خدمات الإنترنت الأفراد الاطلاع على الثقافات، مما زاد من نسبة الانفتاح الاجتماعي وعملية التبادل الثقافي.

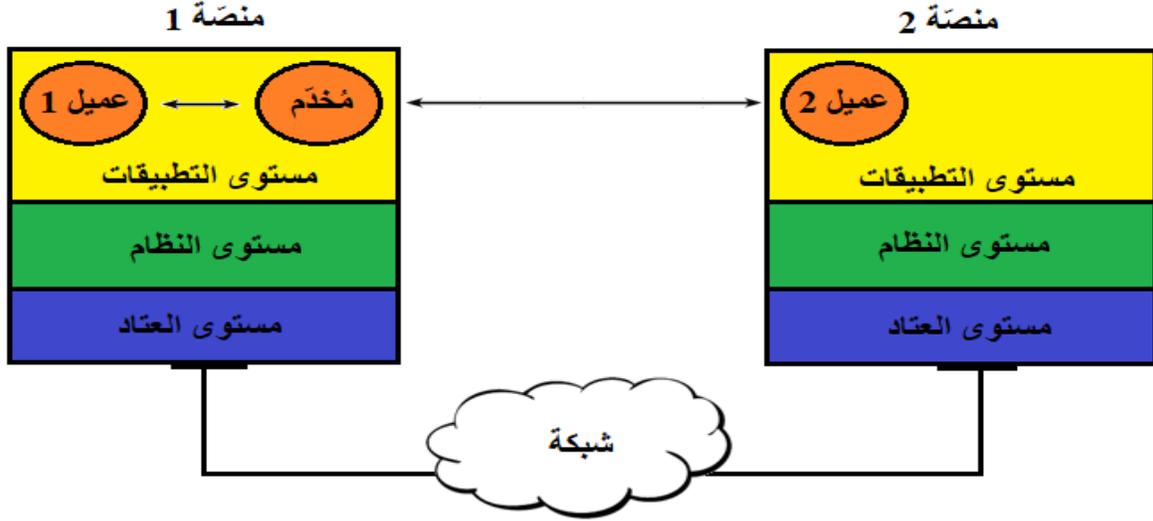
### • أنواع خدمات الإنترنت

خدمات الإنترنت العامة واحدة في مختلف أرجاء العالم، وأهم أنواع هذه الخدمات:

- الوصول لخدمة الإنترنت Internet acces
- الوصول للبريد الإلكتروني email access
- استضافة المواقع web hosting
- مركز البيانات data center services
- الاتصالات Telecommunication
- استرجاع البيانات Information retrieval
- خدمات الويب web service

## ❖ نموذج خادم العميل client / server model

نموذج خادم العميل هو إطار اتصالات موزع لعمليات الشبكة بين طالبي الخدمة والعملاء ومقدمي الخدمات. حيث يسمح بتأسيس اتصال عميل الخادم من خلال شبكة أو الإنترنت.



غالباً ما يتصل المُخدّم مع العميل عبر شبكة حواسِب، حيث يعمل كل منهما على منصة مُنفصلة، ولكن يُمكن أن يتواجد المُخدّم والعميل ضمن نفس النظام.

يُمكن للمُخدّم أن يُشغّل برنامجاً واحداً أو أكثر من البرامج الخاصة بطرف المُخدّم لتقديم خدمة واحدة أو أكثر أو مُشاركة الموارد مع عميل واحد أو أكثر، أمّا العميل فلا يُشارك موارده مع أحد، ولكّنه يطلب الخدمة أو الموارد من المُخدّم، لذلك يبدأ العملاء بإنشاء قنوات اتصال مع المُخدّمات التي تنتظر مبادرتهم وتعمل على تقديم الخدمة المُناسبة لهم.

من الأمثلة عن التطبيقات التي تعتمد هذا النموذج: تطبيقات البريد الإلكتروني والطباعة عبر الشبكة وتطبيقات الويب.

يعد نموذج خادم العميل مفهوماً أساسياً للحوسبة على الشبكة، كما يعمل أيضاً على إنشاء وظائف لتبادل البريد الإلكتروني والوصول إلى قاعدة البيانات / الويب.

## تقنيات وبروتوكولات الويب المبنية حول طراز خادم العميل هي:

- بروتوكول نقل النص التشعبي (HTTP)
- نظام اسم المجال (DNS)
- بروتوكول نقل البريد البسيط (SMTP)
- بروتوكول تيلنت (TelNet - Teletype Network)

### • دور العميل ودور المُخدّم

إنّ العلاقة بين العميل والمُخدّم هي علاقة تعاون بين برنامجين ضمن نفس التطبيق، في هذه العلاقة يقوم المُخدّم بالوظيفة أو يُقدّم خدمة لعميل واحد أو أكثر ممن يطلبون ذلك، يجري تصنيف المُخدّمات بحسب الخدمات التي يقدمونها، فمثلاً مُخدّم الويب يُخدّم صفحات الويب، ومُخدّم الملفات يقدم خدمته للملفّات، إنّ مشاركة الموارد تعتبر أيضاً شكلاً من اشكال الخدمة التي يمكن أن تقدم ويشمل ذلك البرامج والمعطيات ومساحة التخزين وقدرات المعالجة.

يُمكن للحاسب أن يكون عميلاً لخدمة ما ومُخدّماً لخدمة أخرى أو كلاهما معاً في نفس الوقت، كما يُمكن أن يكون عميلاً ومُخدّماً لنفس لخدمة بنفس الوقت، ويتعلق الأمر بالبرامج التي يتمّ تشغيلها فيه،

لايتواصل العملاء فيما بينهم من أجل الحصول على الخدمة، بل يقومون بإرسال طلباتهم إلى المُخدّمات، ينتظر المُخدّم استقبال طلبات العملاء، وهو قادر على تقديم الخدمة لأكثر من عميل بنفس الوقت، يمكن أن تتواصل المُخدّمات التي تقدّم نفس الخدمة مع بعضها البعض من أجل مُزامنة قواعد بياناتها وصُولاً إلى قاعدة بيانات مُشتركة، تسمّى هذه العملية الاتصال بين

المُخدّمات (Inter-Server Communication).

## • الاتصال بين العميل والمُخدّم

بشكل عام، الخدمة هي شكل تجريديّ يُعبّر عن الموارد التي يمكن للمُخدّم أن يقدّمها للعملاء إنّ نظام طلب الخدمة مُصمّم بحيث يكون العملاء غير معنيين بالآلية التي يقوم فيها المُخدّم بالاستجابة للطلبات، ويتوجب عليهم فقط أن يكونوا قادرين على فهم الردّ الوارد من المُخدّم، ويحصل ذلك عن طريق تشغيل العميل لأحد البروتوكولات الخاصة بالخدمة المطلوبة.

يتبادل العميل والمُخدّم الرسائل ضمن نمط الطلب/الرد (Request/Response) ، حيث يرسل العميل طلباً فيقوم المُخدّم بالردّ عليه، إنّ هذا النمط هو مثال عن عملية اتصال بين العمليات، لنجاح الاتصال يجب أن يدعم المُخدّم والعميل نفس بروتوكولات الاتصالات، حيث تحدد هذه البروتوكولات مجموعة القواعد الخاصة بتنسيق وإعداد ونقل البيانات فيما بينهما بحيث تحصل العمليات السابقة بطريقة مفهومة لطرفي الاتصال.

تعمل كل بروتوكولات طلب الخدمة في أقرب طبقة للتطبيقات بحسب نموذج الاتصال المستعمل، على أيّ حال تسمى هذه الطبقة بطبقة التطبيق في نموذج الإنترنت وفي نموذج الاتصال المعياري أيضاً، ويُعرّف كل من هذه البروتوكولات أنماطاً مُميزة خاصة به من رسائل الطلب/الرد، كما يُمكن أن يدعم المُخدّم واجهة برمجة تطبيقات خاصة به، وذلك بهدف تخصيص أنماط مُتبادلة، ويسهّل ذلك من عملية مُزامنة قواعد البيانات.

قد يستقبل المُخدّم طلبات من العديد من العملاء في فترة قصيرة، ويمكن أن يُشكل هذا ضغطاً على موارده، لذلك تعتمد المُخدّمات إلى الاعتماد على نظام جدولة لطلبات العملاء، بحيث يتمّ تصنيف هذه الطلبات بحسب أهميتها، قد تُشكّل محدودية موارد المُخدّم ثغرة أمنيّة لهجوم إيقاف الخدمة، حيث يعتمد المهاجمون إلى استغلال التزام المُخدّم بالرد على طلبات العملاء فيقومون بإرسال الطلبات بمعدّل مُفرط.

## ثانياً: مبادئ تصميم مواقع الانترنت websites

في هذا الفصل سنتعرف على المواضيع التالية:

- أنواع مواقع الشبكة العنكبوتية العالمية WWW
- تحديد أهداف الموقع site goal
- تحديد مكونات الموقع site components
- خطة بناء الموقع site construction plan

## ❖ أنواع مواقع الشبكة العنكبوتية العالمية www

أنواع المواقع الإلكترونية كثيرة جداً، وفي عالم سريع التطور اليوم يتم تحديث أنواع المواقع على شبكة الإنترنت واتجاهات التصميم يوماً بعد يوم وعموماً بعد عام.

### • أنواع المواقع بحسب وظيفتها

١. **محركات البحث:** إذ يوجد على الإنترنت الكثير من محركات البحث كجوجل وباهو وبينج وغيرها من محركات البحث التي تساعدك على العثور على المعلومات من خلال كلمات رئيسية تدخلها فيها وتحيلك إلى المواقع الإلكترونية.

٢. **البوابات الإلكترونية:** البوابة الإلكترونية نوع من أنواع المواقع الإلكترونية يجمع لك معلومات مختلفة سواء إخبارية، أو سياسية، أو تعليمية، أو تكنولوجية وتتسم بطابع موسوعي نوعاً ما، وهناك بوابات إلكترونية متخصصة في موضوع معين كالموضة، أو الرياضة، أو الإلكترونيات، أو غيرهم.

٣. **مواقع الشركات الكبرى:** وهي أماكن فيها ملفات الشركات أو المؤسسات التي تعرفك بآخر أنشطتها ووسائل التواصل معها وبالتالي تقدم هذه المواقع لمن يبحث عن الشركة معلومات عامة حول نشاطها وخصائص مهمة أخرى كفرص العمل والأحداث الأخيرة وبالطبع عرض أحدث منتجاتها وخدماتها إضافة إلى كونها تمثل بيئة اتصال بين الموظفين، وبعضهم، والموظفين، ومديرهم.

٤. **المواقع الحكومية:** والغرض من هذه المواقع نشر المعلومات ذات الصلة للمواطنين، وكذلك الحملات الإعلامية والتقارير الخاصة بالشفافية وأنشطة الوزارات والمؤسسات كما يمكن أن تقدم المواقع الحكومية للسياح خريطة بالأماكن التي يمكن أن يزورها من مطاعم وأماكن جذب سياحي ووسائل اتصال في وقت الشعور بالخطر وتقدم بعض المواقع الإلكترونية الحكومية لمواطنيها آليات لمشاركتهم الرأي والمشورة.

٥. **المواقع التعليمية:** وتتمثل مهمة هذا النوع من أنواع المواقع الإلكترونية في نشر محتوى تعليمي من خلالها، إما بتوفير محتوى مجاني لتعليم الأطفال، إلى أكثر المحتويات الاحترافية المرتبطة بمؤسسات جامعية.

٦. **وسائل التواصل الاجتماعي:** مواقع التواصل الاجتماعي باتت اليوم وسيلة من وسائل التواصل وتبادل الخبرات والمعلومات في كل مكان في العالم، مهما اختلفت صورتها والغرض منها، سيعرض الناس فيها اهتماماتهم الشخصية، أو السفر، أو الحفلات، أو اجتماعاتهم بالأصدقاء، أو إنجازاتهم المهنية، إلخ.

٧. **المواقع الإخبارية:** إذ انتقلت الصحف وحتى قنوات التلفاز من المجالات التقليدية والصحافة المطبوعة إلى التركيز على التواجد على الويب ببث مباشر يعرف الناس ما يدور حولهم من أحداث أولا بأول، بل ويستقبل من المواطنين أخبارا.

٨. **المدونات:** وهي مواقع تتيح لأي شخص كتابة المقالات ونشرها فيها من خلال الويب، سواء كانت هذه المدونات شخصية أو تتعلق بمجال معين ونقل خبرات معينة في الأسفار أو الصحة أو اللياقة، والغالب حاليا في هذا الاتجاه التدوين الصوتي والمكتوب.

٩. **مواقع الخدمات المصرفية:** أحد أبرز أنواع المواقع الإلكترونية اليوم المواقع التي تقدم خدمات نقل الأموال إما عن طريق البنوك على الأرض أو البنوك التي لا مقرات لها وإنما فقط هي بنوك إلكترونية ويتسم هذا النوع من المواقع بدرجة أمان مرتفعة عن غيرها.

١٠. **المتاجر على الانترنت:** هي أماكن للبيع على الإنترنت يعرض فيها الناس بضاعتهم وعروضهم وتوصياتهم وحتى تعليقات الناس الذين اشترو هذه السلع، ويتم الدفع بأساليب مختلفة في نفس الموقع والشحن عن طريق المتجر نفسه أو طرف ثالث.

١١. **تطبيقات الويب:** وهذا النوع من أنواع المواقع الإلكترونية يقدم لك تطبيقات مساعدة وخدمات خاصة بك، إما في إدارة مشروع، أو تخزين ملفات، أو تحرير مستندات، أو غيرها من الأشياء التي تساعدك في إنجاز أعمالك.

١٢. **مواقع الألعاب:** وهي مواقع على الإنترنت متخصصة في التفاعل واللعب بعدد كبير من السمات التي تجعل كثيرا من الناس يجربونها وخصوصا فئات المراهقين.

## • أنواع المواقع الالكترونية حسب البنية

تتكون انواع المواقع الالكترونية حسب البنية من نوعين أساسيين هما أهم محدد لأي موقع الكتروني عند بنائه وتصميمه وذلك من خلال الإجابة على سؤال هل تريد موقع الكتروني تفاعلي ومستجيب مع العملاء أم لا.

١. **المواقع الالكترونية الثابتة:** هي المواقع الالكترونية التي يتم فيها تقديم بيانات عن العمل أي كان ما هو دون السماح للعملاء بالتفاعل معه أو تغيير وإدخال بيانات بها والوحيد القادر على إجراء التغييرات فيه هو صاحب الموقع الالكتروني وكذلك المبرمج والذي يقوم بإدارة المحتوى به.
٢. **المواقع الالكترونية المستجيبة:** هي المواقع الالكترونية التي يتفاعل معها العملاء ويقومون بإجراء التغييرات والتفاعلات فيها ويعتبر من أشهر هذه انواع المواقع الالكترونية هي مواقع التواصل الاجتماعي مثل تويتر وفيسبوك.

## • انواع المواقع الالكترونية حسب الجهات المالكة لها

١. **مواقع أفراد شخصية:** ويتم امتلاك هذه انواع من المواقع الالكترونية من قبل فرد واحد أو مجموعة من الأشخاص ولعل من أشهر انواع المواقع الالكترونية من هذا النوع هي موقع طبي خاص بطبيب أو موقع إلكتروني قد قام بعمله مهندس ما للتعريف بنفسه بأعماله وغيرها الكثير من الأمثلة على هذا النوع.
٢. **مواقع مؤسسات وشركات:** أصبح من السهل الآن عليك إنهاء معظم المعاملات الحكومية وأنت جالس في منزلك بعيداً عن زحام الطوابير أمامها ومضيعة الوقت والمجهود ، ويعتبر هذا النوع من انواع المواقع الالكترونية المملوكة للمؤسسات في دولتك أو دولة ما هو نوع مواقع مؤسسات وشركات، وقد ازدهر وانتشر هذه النوع من المواقع الالكترونية بصورة كبيرة وذلك بعدما وجدت فيه الشركات والمؤسسات قناة تواصل ودعاية مجانية لها بجانب الكثير من المميزات الأخرى التي تكتسبها هذه الشركات والمؤسسات من إنشائها.

## • أنواع المواقع الالكترونية حسب نوع المحتوى

تقدم المواقع الالكترونية محتوى إخباري لحظة بلحظة إليك بينما تقوم المتاجر الالكترونية بتقديم خدمات البيع لك لمنتجات وخدمات،

ولعل من أهم أنواع المواقع الالكترونية حسب المحتوى هما نوعين أساسيين:

١. **خدمات ومنتجات ملموسة**: وفي هذا النوع من أنواع المواقع الالكترونية يتم تقديم

خدمات ومنتجات ملموسة من أغذية وأجهزة كهربائية وكذلك الملابس والماركات

العالمية إلى آخره، ويعتبر موقع الالكتروني نمشي على سبيل المثال نوع من انواع

هذه المواقع الالكترونية

٢. **خدمات ومنتجات رقمية غير ملموسة**: في هذا النوع من انواع المواقع الالكترونية يتم

تقديم محتوى غير ملموس مثل المواقع الالكترونية الإخبار، وكذلك مواقع التعليمية

المعروفة والمشهورة مثل Coursera أو Udemy وغيرها الكثير من انواع المواقع

الالكترونية التي تقدم خدمات غير ملموسة.

## ❖ تحديد أهداف الموقع site goal

أول خطوة لتطوير المواقع هي تحديد الأهداف المرغوبة في الموقع الإلكتروني، حيث يساهم ذلك في تحديد المحتوى والتصميم العام والوظائف البرمجية المطلوبة، وتختلف أهداف إنشاء المواقع من مشروع لآخر بحسب متطلبات مالك الموقع واحتياجاته ويمكن سرد بعضها مثل:

- ١- إيصال رسالة: دعائية – سياسية – طبية – دينية وغيرها.
- ٢- تحسين صورة العمل: فحتى أصغر الأعمال يمكن أن يكون لها حضور أمام أكبر الشركات.
- ٣- زيادة إنتاجية العمل: بوضع اجابات للأسئلة الشائعة والمتكررة عن العمل بالموقع.
- ٤- إنشاء رسالة إعلامية متجددة دائما وسهلة التحديث بدون الحاجة للأوراق أو الأخبار أو المطابع وغيرها
- ٥- التكلفة المجدية للموقع: فتكلفة متابعة الموقع وتحديثه تعتبر قليلة نسبياً ويمكن تحملها مقارنة مع ما يدفع لوسائل الاعلان الاخرى.
- ٦- استعمال البريد الالكتروني: ارسال رسالة لمائة عميل معاً بتكلفة اقل من ارسال رسالة لعميل واحد بالبريد العادي، من اهم اهداف انشاء الموقع الالكتروني
- ٧- الترويج للعمل على مدار الساعة: حيث لا توجد وسيلة اعلامية اخرى تقدم هذا العرض وبهذه المرونة.
- ٨- تنفيذ الاعمال على الانترنت والبيع والشراء عبر بطاقات الائتمان.
- ٩- استلام المعلومات من مواقع أخرى لأعمال مشابهة والتزود بالجديد ومقارنة الاسعار والمواصفات بسهولة.
- ١٠- ايجاد خدمة متواصلة على مدار ٢٤ ساعة.

## ❖ تحديد مكونات الموقع site components

### • تنظيم صور وملفات موقع الانترنت

يتكون أي موقع انترنت من مجموعة من صفحات الانترنت والتي تخزن على هيئة ملف نصي بصيغة html وتعتبر هذه الصفحات هي المكون الرئيسي لمواقع الانترنت،

#### وتحتوي الصفحات على التالي:

- الصور المعروضة في الصفحة
- الروابط، وتنقسم الى روابط توصل الى صفحات انترنت اخرى داخلية او خارجية او توصل الى ملفات مختلفة الصيغ.
- التطبيقات المختلفة والتي تؤدي مهام او وظائف معينة
- في حال كون الصفحات تحوي تطبيق معين فإن صيغة تخزين الصفحة تختلف باختلاف البرنامج الذي تم تطوير التطبيق به فإذا كان التطبيق مطور بلغة PHP يكون صيغة الملف php وإذا طور بتقنية ASP.net تكون الصيغة .aspx.

### • تنظيم صور وملفات موقع الانترنت

إذا كان الموقع المراد إنشائه صغير فيكفي إنشاء مجلد واحد لحفظ الصور التي ستستخدم في بناء الموقع.

إذا كان الموقع كبير وبه العديد من الصور يجب بناء هيكلية بحيث تكون مجلدات الصور مقسمة حسب طبيعة الصفحات أو أي تصنيف آخر يراه مطور موقع الانترنت، وذلك لسهولة إدارة الصور الموجودة في الموقع.

كذلك يجب إنشاء مجلدات مستقلة لكل نوع من الملفات التي سيتم تخزينها على الموقع الالكتروني (word – PDF – PowerPoint – Video – Audio) وكما يمكن إنشاء مجلدات فرعية لكل من انواع الملفات السابقة لخزن الملفات الخاصة بمحتوى معين في موقع الانترنت.

تعتمد الالية السابقة في معظم الاوقات على حجم واحتياجات الموقع وكذلك الاسلوب الذي يتبعه المطور في تنظيم الملفات.

## • مبادئ تصميم صفحات الانترنت

لتجنب إنشاء صفحات الانترنت بشكل بشع او غير متناسق يجب تعلم المبادئ الاساسية للتصميم.

### المبادئ الاساسية هي:

- المحاذاة لمحتويات الصفحة
- تقارب محتويات الصفحة
- تناغم وتكرار التصميم
- التباين
- التحقق الجانب اللغوي لمحتوى الصفحة

## • محاذاة محتويات الصفحة

المحاذاة: هي اسلوب تنسيق محتويات (نصوص، صور، ...) الصفحة جنباً الى جنب بطريقة تظهر الصفحة بشكل احترافي ومرتب.

### ○ أنواع المحاذاة:

- المحاذاة الافقية Horizontal alignment
- المحاذاة العمودية Vertical alignment

يجب اختيار محاذاة افقية واحدة وعمودية واحدة وتطبيقها على جميع صفحات الموقع الالكتروني (يجب تغيير المحاذاة في صفحات الموقع).

## • تقارب محتويات الصفحة

تقارب المحتويات: هي كيفية تكوين العلاقات لمحتويات الصفحة من حيث تقاربها او تباعدها من بعضها البعض.

يجب ان يكون هناك وعي في عملية توزيع محتويات الصفحة:

- تجميعها مع بعضها البعض
- او إبعادها ووضع مسافات فيما بينها.

مثال لتقارب محتويات الصفحة هو توزيع الحقول التي يتمكن من خلالها المستخدم من ادخال البيانات:

- تجميع ومقاربة حقول المعلومات الشخصية مع بعضها البعض يليها مسافة او فراغ
- ومن ثم تجميع معلومات الاتصال مع بعضها البعض.

## • تناغم وتكرار التصميم

**تناغم التصميم:** هو ربط العناصر من خلال تكرار الخصائص او الترتيبات المشتركة فيما بينها مثل (توحيد الخطوط وطريقة عرض الصور)

**تكرار التصميم:** هو تكرار بعض العناصر عبر كل صفحات موقع او تطبيق الانترنت.

**العناصر المكررة والتي توحد المظهر العام لموقع الانترنت:**

- ايقونات التنقل بين الصفحات او قائمة التنقل بين الصفحات.
- الالوان
- تنسيق تقسيم الصفحة
- خصائص تنسيق المحتويات

## • التباين Contrast

-التباين هو ما يجذب نظر المتصفح لصفحات الموقع.

-التباين يحقق جذب النظر لعناصر الصفحة من خلال إنشاء بنية هرمية متدرجة تمكن المتصفح من مسح محتويات الصفحة والوصول الى المعلومة التي يحتاجها بسرعة.

## • التحقق الجانبي اللغوي لمحتوى الصفحة

-الاطفاء الاملائية والنحوية يمكنها ان تدمر المظهر الاحترافي للصفحة كما يمكن ان تضع المستخدم في موقف عدم تأكد او تعارض في فهم المقصود من المحتوى.

-استخدام المصحح اللغوي الالي يمكن ان يقلل من الاخطاء اللغوية بنسبة كبيرة.

## ○ تنسيق الصفحة Web Page Layout

تنسيق الصفحة هو امر بالغ الاهمية لتطوير موقع الكتروني متميز.

**التنسيق السيء للصفحة يتسبب في:**

- صعوبة التنقل فيما بين صفحات الموقع.
- صعوبة ايجاد المعلومات التي يحتاجها المتصفح او المستخدم على الصفحة.
- غير جاذب للرؤية.

## ❖ خطة بناء الموقع site construction plan

### • الهيكلية بناء موقع الانترنت

الهيكلية هي التنظيم الاساسي للنظام ويتضمن مكوناته والعلاقة فيما بين هذه المكونات وكذلك البيئة والمبادئ التي تتحكم في تصميمه وتطوره، حيث تربط الهيكلية مراحل تطوير موقع الانترنت.

المتطلبات الخاصة بتطبيق الانترنت ترسم كمكونات والعلاقات فيما بين هذه المكونات.

الهيكلية هي مجموعة قرارات التصميم والتي تبقي المطورين والقائمين على الصيانة بعينين عن ممارسة الابداع غير اللازم (بحيث يكون هناك مجموعة من المهام والنشاطات المدروسة والخطوات التي يتم تنفيذها لضمان جودة الانتاج واستمرارية المنتج النهائي بشكل صحيح) ، وهي تجعل تطبيقات الانترنت متكاملة وقابلة للتحكم بها، وذلك من خلال مرورها بمراحل التهيئة والتخطيط والتنفيذ والصيانة بطريقة مدروسة ومنظمة.

### • المتطلبات التي يجب مراعاتها عند هيكلية موقع الانترنت

#### ○ المتطلبات الوظيفية، وتشمل:

١- المتطلبات التي تخص المالك؛ مثل المتطلبات التي تمثل الجهة التي ترغب

بتطوير التطبيق لغرض وهدف محدد سواء كان ربحي او غير ربحي.

٢- المتطلبات التي تخص المستخدمين؛ وتمثل متطلبات الافراد او المجموعات

التي ستقوم باستخدام التطبيق لأداء مهام محددة سواء كانوا مستخدمين

من داخل او خارج المنظمة المطورة للتطبيق.

#### ○ المتطلبات الغير وظيفية، وتشمل:

١- الأداء

٢- امكانيات التطوير والتعديل

٣- امكانيات الاستخدام السهل

#### ○ المتطلبات التشغيلية:

١- انظمة التشغيل وبرامج التطوير

٢- البرامج الوسيطة او البرامج المساعدة في عملية التطوير (مثل: برامج التصميم،

برامج رفع واستضافة الموقع)

## • جوانب الهيكلية

### ○ الجانب الدراسي والتجهيزي، وتشمل:

- ١- تحديد الاهتمامات وتقسيمها، بحيث يتم دراسة الاهتمامات العامة التي سيغطيها تطبيق الانترنت لوضعها في الهيكلية.
- ٢- عدد المستخدمين المبدئي والذين سيتم خدمتهم من خلال التطبيق.
- ٣- الاحتياجات العامة والمشاركة للتطبيق (مثل: الاحتياجات الامنية، احتياجات الصيانة)

### ○ جانب البيانات، وتشمل:

- ١- نوع البيانات التي سيتم التعامل معها
- ٢- الحجم وطبيعة البيانات
- ٣- حساسية البيانات
- ٤- قاعدة البيانات المستخدمة لخرن البيانات
- ٥- متطلبات معدل نقل البيانات

### ○ جانب البنية التحتية لتطبيقات الانترنت، وتشمل:

- ١- المفاهيم العامة لعمليات واحتياجات المنظمة المطورة لتطبيق الانترنت وكيفية تطبيقها
- ٢- كيفية تكامل العمليات التي تتم بشكل تقليدي في المنظمة مع العمليات التي سيتم تنفيذها من خلال تطبيق الانترنت
- ٣- اختيار البرامج التي سيتم من خلالها تطوير تطبيق الانترنت.
- ٤- المكونات على الشبكة (جدار الحماية، البروكسي، موازنة التحميل، الشهادات الرقمية، أساليب التشفير)
- ٥- المكونات على الشبكة الداخلية (مزود صفحات الانترنت، مزود التطبيقات، مزود قواعد البيانات).

## • الخطة التطويرية لتطبيقات الانترنت

تطور أساليب العمل في مختلف المجالات وخاصة الإدارة والتقنية منها، وتوجهها نحو أساليب العمل الممنهج الذي يعتمد على دراسة وافية لجميع النواحي التي تخص العمل المراد تنفيذه، أصبح تنفيذ أي مهمة أو أحداث أي تطوير يتم التعامل معه كمشروع مستقل له هدف أو مجموعة من الاهداف ومتطلبات وتكلفة ومدة زمنية لتنفيذه، كذلك هو الحال مع تطوير وإنشاء تطبيقات الانترنت، التعامل مع عملية التطوير هذه كمشروع مستقل له العديد من المتطلبات ويستلزم إنشاء خطة عمل مفصلة للخروج بموقع الكتروني يلبي احتياجات الاطراف المستفيدة ويعطي صورة احترافية عن المؤسسة أو المنظمة التي يمثلها تطبيق الانترنت.

هناك عدد كبير جدا من الخطوات والمراحل لعملية تطوير وتصميم المواقع الالكترونية بدءاً من جمع المعلومات وإنشاء الموقع ووصولاً الى صيانة الموقع الالكتروني والحفاظ عليه محدثاً وخالي من الاخطاء والاعطال.

فقبل البدء بالعمل على نشر الموقع، ينبغي أن تكون فكرة كاملة عن المحتوى الذي سيعرض حالياً وأن يخطط للمحتوى الذي سيعرض لاحقاً.

وبناء على المحتوى يجب أن يبني الموقع على هيئة تسويقية للجهة التي يمثلها الموقع بحيث يكون مميزاً وربطه بالذهن أمراً سهلاً وكذلك جعل المحتوى واضح مع التأكد من سهولة الاستخدام وإمكانية الوصول الى المحتوى بشكل سهل وسريع.

عملية التطوير تختلف من مطور الى اخر وكذلك على الهدف من تطبيق الانترنت وطبيعته والجهة التي يمثلها، ولكن الخطوات الاساسية عادة ما تكون هي نفسها، وهي كالتالي:

- جمع المعلومات
- التخطيط
- التصميم
- التطوير
- الاختبار والنشر
- الصيانة

## • جمع المعلومات

الخطوة الاولى في تصميم موقع الكتروني ناجح هي جمع المعلومات، حيث يتم وضع العديد من الاسئلة في هذا الشأن لفهم الاحتياجات والهدف من الموقع الالكتروني، **أهم الاسئلة التي يجب طرحها هي:**

○ ما هو الغرض من الموقع الالكتروني؟

○ الاهداف المرغوب الوصول لها من الموقع الالكتروني

ما هو المأمول تحقيقه من إنشاء الموقع الالكتروني؟ الاهداف الاكثر شيوعا هي تحقيق ربح مادي او مشاركة المعلومات.

○ ما هي الفئات المستهدفة والمرغوب الوصول؟

○ محتوى الموقع الالكتروني

ما هو نوع المعلومات التي سيتم عرضها للفئة او الفئات المستهدفة والتي ستزور الموقع؟ وما هي الخدمات التي سيوفرها الموقع الالكتروني لزواره؟

## • التخطيط

○ إنشاء خريطة الموقع:

خريطة الموقع هي قائمة لكل المواضيع الرئيسية والفرعية في الموقع الالكتروني، حيث تعمل كدليل للمحتوى الذي سيتوفر على الموقع، وهي اساسية لتطوير موقع الكتروني متكامل، وسهل الاستيعاب والاستعراض.

يجب وضع المستخدمين في الحسبان عند تصميم الموقع الالكتروني من حيث خبراتهم وقدرتهم على الوصول الى المحتوى، من خلال التخطيط يمكن التعرف على التقنيات والادوات التي يلزم استخدامها في تطوير تطبيق الانترنت ومختلف صفحات الموقع.

## • التصميم

في هذه المرحلة يتم تحديد المظهر الذي سيقدم به الموقع الالكتروني، الفئات المستهدفة عبر تطبيق الانترنت والجهة التي يمثلها التطبيق هي واحدة من اهم الامور التي يجب وضعها بعين الاعتبار.

ويجب الالتزام بوضع معيار لصفحات الموقع كالشعار والالوان المستخدمة لتقوية هوية الجهة التي يمثلها التطبيق والموقع الالكتروني.

يتم تصميم نموذج مبدئي للهيئة التي سيبدو عليها الموقع الالكتروني **والتي تفيد في:**

- تمكين القائمين على المشروع من التعبير عن اعجابهم او عدم اعجابهم بتصميم الصفحات.
- مناقشة التعديلات الممكنة قبل نشر الموقع لكي لا يبذل جهد كبير في تغيير التصميم العام.

## • التطوير

هي المرحلة التي يتم فيها إنشاء تطبيق الانترنت، ويتم الاعتماد على التصميم النهائي الذي تم الاتفاق عليه لبناء الموقع الفعلي.

يتم تطوير الموقع بالبداية بصفحة البداية او الصفحة الرئيسية، بعد ذلك يتم تطوير الصفحة النموذجية او صفحة القالب

### ○ الصفحة النموذج او القالب الذي يتم استخدامه ل:

- محتويات الصفحات الداخلية او الصفحات الفرعية للموقع الالكتروني.
- يكون محتوى القالب العام نفسه لكل الصفحات كرأس الصفحة وقائمة التنقل بين صفحات الموقع وتعيير المحتوى لكل صفحة على حدا.

تنفذ العناصر التفاعلية او التطبيق نفسه في هذه المرحلة بحيث يتم استخدام لغة البرمجة والمحددة مسبقا في مرحلة هيكلية تطبيق الانترنت لتطوير تطبيقات او تطبيقات الانترنت التي يحتويها الموقع.

## • الاختبار والنشر

هذه المرحلة يتم فيها وضع اللمسات الاخيرة واختبار جميع تفاصيل الموقع الالكتروني، حيث سيتم

### اختبار:

- عمل تطبيق الانترنت وجميع العمليات التي يجب ان ينفذها التطبيق.
- قبول المدخلات وتخزينها بشكل صحيح، وظهور المخرجات بشكل صحيح.
- ظهور المحتوى بنفس الاسلوب وعدم وجود اختلافات بين الصفحات.
- اختبار عمل الموقع على العديد من المتصفحات والاصدارات المختلفة من المصفحات وخصوصا الاخيرة منها.
- التحقق من عمل جميع الروابط الداخلية التي تنقل المستخدم من صفحة الى اخرى والخارجية التي تربط بصفحات خارجية

بعد التحقق من جميع ما سبق يتم نشر الموقع الالكتروني على الانترنت ثم يتم عمل اختبار وتحقق اخير للأمر التي تم سردها سابقاً.

## • الصيانة

بمجرد الانتهاء من نشر الموقع لا يعني ذلك الانتهاء من العمل على الموقع الالكتروني لان المواقع الالكترونية بطبيعتها تتطلب تحديث محتواها والاضافة الدائمة لمحتوى جديد متطور. يجب عمل خطة صيانة يومية واسبوعية وشهرية لصيانة الموقع الالكتروني لتحقيق الاهداف السابق ذكرها والحفاظ على ديمومة الموقع، حيث إن تحديث المحتوى سيلزمه صيانة دورية للمحتوى القديم وللتحقق من عمل الموقع بشكل مثالي.

## ثالثاً: إنشاء الصفحات باستخدام لغة الترميز HTML

في هذا الفصل سنتعرف على المواضيع التالية:

- الفرق بين HTML و XHTML
- البنية المعيارية XHTML والتحقق من الوثائق
- تنسيق النصوص
- تنسيق الصور
- الروابط Hypertext Links
- القوائم
- الجداول

### تطور HTML وXHTML

–اشتقت اللغة HTML من اللغة المعيارية<sup>1</sup> SGML والتي هي معيار لتوصيف تنسيق النصوص وفق المعايير الموضوعية من قبل المنظمة الدولية للمعايير<sup>2</sup> ISO.

–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 تستخدم على حواسيب مختلفة ومتصفحات متنوعة، ولقد أغنى أسلوب الصفحات المتتالي<sup>3</sup> CSS والذي ظهر في أواخر التسعينات تفاصيل الإظهار التي يُمكن تحديدها من خلال هذه اللغة.

### إصدارات HTML وXHTML

تعددت إصدارات HTML انتهاءً بالنسخة ٥ عام ٢٠١٢. أما المعيار ١,٠ XHTML فقد تمت الموافقة عليه في عام ٢٠٠٠. تم إنشاء هذا المعيار بإعادة تشكيل اللغة HTML وبهدف تكون لغة تأشير موسعة extensible Hypertext Markup Language XML

وفي عام ٢٠٠١ تمت الموافقة على النسخة ١,١ XHTML من قبل منظمة الويب<sup>4</sup> W3C والتي دورها مراقبة تطور لغات الويب، ثم النسخة ٢ XHTML عام ٢٠١٠.

### HTML و XHTML

يمكن 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 .

## الشكل الأساسي Syntax

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

## أساسيات تأشير النص

### 1- 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
  everywhere 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 everywhere that Mary, the lamb was sure to go.

### 2- التحقق من الوثائق XHTML

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

<http://validator.w3.org>

يُحدد الأمر DOCTYPE المعيار المطلوب.

### ٣- 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
  everywhere that
  Mary went, the lamb
  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

## ٤-الترويسات Headings

تتوفر ستة ترويسات مختلفة تحددتها المؤثرات من <h1> إلى <h6>.

تستخدم الترويسات ١ و ٢ و ٣ أحجاماً للخط أكبر من الخط الافتراضي، أما الترويسة ٤ فتستخدم الخط الافتراضي، وتستخدم الترويسات ٥ و ٦ 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 (h) </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)
```

## ٥- 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.
```

## ٦-المؤثر <em>

يُستخدم الوضع كلمة أو جملة معينة بشكل مختلف.

مثال:

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

يظهر المتصفح:

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

## ٧-الخطوط Fonts

تتوفر المؤثرات التالية والتي تسمح بتنسيق الخط:

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

والتي يبينها 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 تظهر:

The sleet in **Crete**

**lies** completely in the street

Monospace font

$$x_2^3 + y_1^2$$

## ٨-المحارف الخاصة

يبين الجدول التالي مجموعة المحارف الخاصة التي يمكن استخدامها:

Character	Entity
&	;amp&
<	;lt&
>	;gt&
“	;quot&
’	;apos&
(space)	;nbsp&
1/4	;frac14&
1/2	;frac12&
3/4	;frac34&
°	;deg&

فمثلاً تظهر الفقرة:

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

كما يلي:

$$1/4+1/4= 1/2$$

## ٩-المسطرة الأفقية

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

## ١٠-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 **JavaScript و XHTML** :

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

## ١١-الصور Images

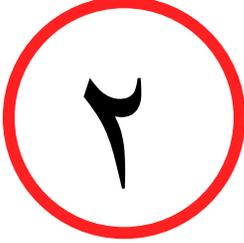
تحسن الصور مظهر صفحات الويب بشكل كبير، مع ملاحظة أنها تُبطئ استعراض الوثائق من قبل المستخدمين الذين لا يملكون سرعات وصول عالية للإنترنت، تخزن الصور في ملفات مستقلة وتُستدعى من قبل XHTML.

## ❖ تنسيق الصور Image Formats

يوجد بشكل أساسي ثلاثة أشكال لتنسيق ملفات الصور:

### • الملفات (GIF (Graphic Interchange Format

تستخدم 8-bits لكل بيكسل pixel (أي ٢٥٦ لون) يُمكن أن تكون هذه الصور شفافة transparent يفضل استخدامها للرسوم المرسومة باستخدام برامج حاسوبية.



### • الملفات (JPEG (Joint Photographic Experts Group

تستخدم 24-bits لكل بيكسل pixel (أي ١٦ مليون لون) لا يمكن أن تكون شفافة، يُفضل استخدامها للصور الطبيعية photo.



### • الملفات (PNG (Portable Network Graphics

ظهر هذا الشكل عام ١٩٩٦ ليجمع بين مميزات الشكلين السابقين (شفافية JIF وعدد ألوان JPEG)، ومن المتوقع أن يصبح الشكل الأوسع على الانترنت، لاسيما وأنه مصدر حر free.



### المؤثر <img>

- يُستخدم المؤثر <img> إدراج صورة في الوثيقة.
- تحدد الوصفة src اسم ملف الصورة.
- تحدد الوصفة alt نص الصورة أي النص الذي سيظهر في حال عدم تحميل الصورة أو عند وضع مؤشر الفأرة فوق الصورة وهي واصفة مطلوبة. required
- تحدد الوصفتان width و height أبعاد الصورة.

يبين 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>
```

حيث يُظهر المتصفح:



## الروابط Hypertext Links ❖

يتصرف الرابط كمؤشر إلى مورد آخر يمكن 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 تغني مواقع الويب.

يُستخدم المؤثر `< a >` لتحديد ارتباط، يملك هذا المؤثر الوصفة `href` التي تحدد عنوان الهدف أما محتوى المؤثر فهو الذي يُشكل الرابط (أي الذي يجب النقر عليه للانتقال إلى الهدف) يمكن أن يكون المحتوى نص أو صورة أو عنوان.

يبين 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.
- تستخدم الواجهة id لإعطاء معرف هدف، فمثلاً، إذا أعطينا ترويسة معرف

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

فيمكن استخدام هذا المعرف في الرابط (يجب سبقه 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>
```

- يُمكن 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

### 1-القوائم غير المرتبة

تكون القائمة محتوى المؤثر <ul>، وحيث يُعرّف كل عنصر من عناصر القائمة باستخدام المؤثر <li>. يُمكن 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 يظهر المتصفح:

- Syria

## ٢-القوائم المرتبة

تعرف في هذه الحالة القائمة باستخدام المؤثر <ol> تُرقم عناصر القائمة تلقائياً بشكل تسلسلي. كما يبين 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

### ٣-قوائم التعريف

يستخدم هذا النوع من القوائم عادة لسرد مجموعة من التعاريف تكون القائمة محتوى المؤثر <dl> يستخدم المؤثر <dt> لعنوان كل تعريف، والمؤثر <dd> لمحتوى التعريف كما يبين 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

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

### 1- مؤثرات الجدول الأساسية

يكون الجدول محتوي المؤثر `<table>` لهذا المؤثر الواصفة `border` يُمكن وضع قيمة رقمية في هذه الواصفة لتحديد عرض حدود الجدول بالبكسل، أو وضع قيمتها تساوي "border" فيكون عرض حدود الجدول العرض الافتراضي، وعند عدم وضع هذه الواصفة يبقى الجدول بلا حدود.

يُسبق الجدول عادة بعنوان وذلك باستخدام المؤثر `<caption>` يُعرف كل سطر في الجدول باستخدام المؤثر `<tr>` أما كل خلية في السطر فتعرّف 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

## ٢- امتداد الخلايا

يمكن استخدام الـ `colspan` الواسفة لتحديد عدد الأعمدة التي تمتد خلية عليها، والواسفة `rowspan`

لتحديد عدد الأسطر التي تمتد خلية عليها، كما يبين 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	٠	١	0

### ٣-المحاذاة

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

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

يبين المثال التالي استخدام هذه الواصلات:

```
<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 يظهر المتصفح:

The align and valign attributes

	Column Label	Another One	Still Another One
align	Left	Center	Right
Valign	Default	Top	
			Bottom

تحدد الواصفة cellspacing للمؤثر <table> تباعد خلايا الجدول عن بعضها، أما الواصفة cellpadding فتحدد بعد محتوى الخلية عن حدودها، كما يبين 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
----------------	---------------

# رابعاً: مهارات متقدّمة في XHTML

في هذا الفصل سنتعرف على المواضيع التالية:

○ النماذج Forms

○ HTML5

○ Video and Audio

○ Web Forms 2.0

## ❖ النماذج Forms

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

توفر لغة XHTML مجموعة من المؤثرات التي تقوم بخلق أغراض تحقق 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.

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

### ١- المؤثر <form>

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

تحدد الوصفة method طريقة إرسال البيانات، تأخذ هذه الوصفة إحدى القيمتين التاليتين:

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

### ٢- المؤثر <input>

يُستخدم هذا المؤثر لإنشاء ٦ 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.

## صندوق النص textbox

- يكون الحجم الافتراضي ٢٠ حرف.
- يمكن تحديد حجم معين باستخدام الوصفة size.
- في حال كتابة محارف أكثر من الحجم المحدد يظهر تلقائياً شريط انزلاق.
- يمكن تحديد عدد محارف أعظمي لا يمكن تجاوزه باستخدام الوصفة maxlength.
- يجب إعطاء اسم لصندوق النص باستخدام الوصفة name.

:Text  Input

## صندوق كلمة السر password

يمثل صندوق النص في واصلاته، 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.
- يمكن استخدام الوصفة checked = "checked" لجعل صندوق التحقق محدداً.

Checkbox checked

Checkbox unchecked

## زر الخيار radio button

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

Radio button 1

Radio button 2

Radio button 3

## زر الإرسال submit button

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

## زر الإعادة reset button

يعيد هذا الزر عناصر التحكم في النموذج إلى قيمها الابتدائية.

يبين 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="over 50"/> 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 ترسل 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 المخفية في النموذج تظهر في العنوان واضحة).

### ٣- المؤثر <select>

– يمكن استخدام القوائم menu عوضاً عن صناديق التحقق أو أزرار الخيار لاسيما إذا كان عدد الخيارات كبيراً.

– يستخدم المؤثر <select> لإنشاء قائمة، أما الخيارات فيستخدم المؤثر <option> لكل خيار.

– يمكن 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>، يمكن استخدام الوصفة size لتحديد

عدد العناصر الظاهرة.

– يمكن جعل خيار محدد افتراضياً بوضع الوصفة "selected="selected" للمؤثر <option>.

يبين المثال التالي إنشاء قائمة:

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

حيث يُظهر المتصفح:

eggs	✓
milk	
bread	
eggs	
cheese	

## ٤-المؤثر <textarea>

يُستخدم المؤثر <textarea> لإنشاء صندوق نص متعدد الأسطر، تحدد الواصفة rows عدد الأسطر والواصفة Cols عدد الأعمدة، عند تجاوز قيم هذه الواصفات تظهر تلقائياً أشرطة انزلاق موافقة.

كما يبين 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:

Please provide your employment aspirations

**مثال:** يبين 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.3.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>

```

**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

## المؤثرات الجديدة في HTML5

قدمت 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 type="date">
- النوع number للمؤثر <input type="number">
- النوع url للمؤثر <input type="url">
- النوع email للمؤثر <input type="email">
- المؤثر <output>
- الوصفة <placeholder>
- الوصفة <autofocus>
- الوصفة <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:



## واصفات المؤثر <video>

الخاصية	الوصف
autoplay	حال وضع هذه الواصفة، سوف يتم تشغيل الفيديو تلقائياً.
controls	حال وضع هذه الواصفة، سوف يتمكن المستخدم من بالفيديو (التشغيل، الإيقاف، الصوت، ...).
height	التحكم بارتفاع العنصر.
loop	في حال وضع هذه الواصفة، سوف يتم تشغيل الفيديو بشكل حلقة متواصلة.
poster	تستخدم 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.

يبين 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.way" controls autoplay>
    Your browser does not support the Audio element.
  </ audio >
</body>
</html>
```

حيث يقوم المتصفح بتشغيل ملف الصوت الموافق.

## واصفات المؤثر <audio>

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

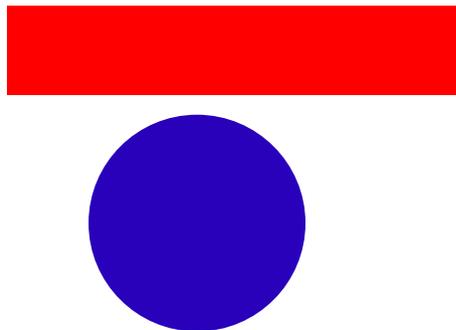
## المؤثر <svg>

يُستخدم المؤثر <svg> لإنشاء الرسوم المتجهة، حيث اختصار لـ Scalable Vector Graphics والتي هي لغة لتوصيف الرسومات ثنائية البعد هي 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



## HTMT5\_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 للأنواع المعروفة السابقة (text, password, radio, ...) تمت إضافة مجموعة من الأنواع الجديدة من أهمها:

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

## النوع date للمؤثر <input>

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

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

حيث يظهر المتصفح تقويم لمساعدة المستخدم في إدخال التاريخ:

Date:

January, 2015 ▾

Sun	Mom	Tue	Web	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>

يبين 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 <inputs>

يبين المثال التالي استخدام النوع email لإدخال قيمة بريد إلكتروني 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>
</body>
</html>
```

## النوع url للمؤثر <input>

يبين 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 >

يستخدم هذا المؤثر الجديد لإظهار قيمة نتيجة ما على الصفحة، يبين المثال التالي استخدام المؤثر <output > الذي يظهر النتيجة على الصفحة بالاعتماد على الإجراءات المكتوبة بلغة Java Script<sup>2</sup>

```
<!DOCTYPE html>
</html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>output</ title>
  <script type="text javascript ">
    Function show Result (){
      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>
<head>
  <form action="/cgi-bin/html.cgi" method="get">
    Enter email :<input type="email" name="newinput"
    Place holder="email@example.com"/>
    <input type="submit" value="submit"/>
    <output name="result"/>
  </form>
</body ></html>
```

حيث يظهر المتصفح:

Enter email:

### الواصفة <autofocus>

تقوم هذه الواصفة بوضع التركيز على العنصر.

## الواصفة <required>

تجعل هذه الوصفة قيمة العنصر واجبة الإدخال.

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

## اقتراحات وتمارين

قم بإنشاء استمارة التسجيل التالية:

### Registration Form

First Name:	<input type="text"/>
Middle Initial:	<input type="text"/>
Last Name:	<input type="text"/>
City:	<input type="text"/>
State/Province:	<input type="text" value="Please Choose....."/>
Country:	<input checked="" type="radio"/> United States <input type="radio"/> Canada <input type="radio"/> Other
Zip:	<input type="text"/>
Email:	<input type="text"/>
Username:	<input type="text"/>
Password:	<input type="text"/>
Repeat Password:	<input type="text"/>
Comments:	<input type="text"/>

I agree to the site terms

submit

Reset Form

# خامساً: أسلوب الصفحات المتتالي CSS

في هذا الفصل سنتعرف على المواضيع التالية:

- مقدمة
- مستويات أسلوب الصفحات
- تنسيق تحديد الأسلوب
- أشكال المحددات
- الخاصية والقيمة Value, Property
- خصائص الخط
- خصائص القوائم
- خصائص أساليب العناصر الأخرى
- المؤثرات <SPAN> <DIV>

## ❖ 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.

سمي 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 اللاصقة css يتم استخدام المؤثر <link> في رأس الوثيقة لتحديد ملف أسلوب خارجي:

```
<link rel="stylesheet" type="text/css"
Herf="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 الواصفة type=text/css / \* ... \* / لوضع تعليقات ضمن القواعد.

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

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

- يمكن 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 ستطبق على كل ظهور لهذا المؤثر في الوثيقة.
- يُمكن أيضاً أن يكون المحدد مجموعة من المؤثرات المفصولة بفواصل، وفي هذه الحالة، ستطبق قيم الخصائص على كل ظهور لهذه المؤثرات في الوثيقة.
- إذا وضعنا \* في المحدد، فإن قيم الخصائص ستطبق 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 الخط لمحتوى غامق ومائل هو ٣٠

يبين 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 bi {font-size: 30pt;}
</style>
</head>
<body>
<h1> This is heading 1</h1>
<h2> This is heading 2</h2>
<h1> This is heading 2</h2>
</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;}
```

تستخدم الوصفة class مع المؤثر لتحديد الصف المطلوب، كما يبين 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

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

مثلاً:

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

تستخدم 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>
```

## ❖ الخاصية والقيمة Property-Value

### أشكال خاصية - قيمة Property-Value

- يوجد حوالي ٦٠ خاصية تتوزع على ٧ فئات:

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

- يمكن أن تكون قيم الخصائص كلمات مفتاحية:

- large
- medium
- small

مثلاً:

font-size: large

- قيم رقمية مع الوحدات التالية:

- Pixels- px
- 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 نسبة مئوية من القيمة السابقة. فمثلاً إذا وضعنا:

font-size: 70%;

يُصبح حجم الخط 70% من القيمة السابقة لحجم الخط.

- تأخذ قيم محددات المصدر الشكل:

url (...)

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

- يمكن تحديد قيمة لون:

- باستخدام اسم اللون:

Color: white

- بوضع قيمة اللون بالنظام السداسي عشر:

Color: #FFFFFF

- باستخدام الوظيفة rgb:

Color: rgb (255,255,255)

(للوظيفة rgb ثلاث معاملات تأخذ قيمها بين ٠ و ٢٥٥ وتحدد بالترتيب درجة الأحمر، الأخضر، الأزرق)

## ❖ 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 ومحصورة بين 100 و900 حيث القيمة 400 تكافئ normal والقيمة 700 تكافئ bold:

font-weight:800

### وزن الخط font-weight

تكون قيمها italic, normal

### الخاصية font

يمكن 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

يبين 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
```

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

## يبين المثال التالي استخدام ملف خارجي CSS

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head> <title> External style sheet </title>
    <link rel=" style sheet "= 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}
```

وتظهر 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 List Properties

- يمكن في القوائم تحديد:
  - شكل الرسمة التي تسبق العنصر في القوائم غير المرتبة
  - طريقة ترقيم القوائم المرتبة
- تحدد الخاصية `list-style-type` شكل الرسمة التي تسبق العنصر وتكون قيمها:  
`disk, circle, square, none`
- يمكن وضع هذه الخاصية للمؤثر `<ul>` وبالتالي يكون تأثيرها على جميع عناصر القائمة، أو وضعها للمؤثر `<li>` فيكون تأثيرها على العنصر المعني فقط.  
كما يبين 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

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

مثال	نوع الترقيم	قيمة الخاصية
١, ٢, ٣, ٤	Arabic numerals	decimal
A, B, C, D	Uc letters	upper-alpha
a, b, c, d	Lc letters	lower-alpha
I, I, III, IV	Uc Roman	upper-roman
i, ii, iii, iv	Lc Roman	lower-roman

يُبين المثال التالي استخدام الخصائص السابقة:

```
<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
```

```
<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 >
</ li >
</ ol >
< body > </ html >
```

### **Aircraft Types**

#### I. General Aviation (piston-driven engines)

##### A. Single-Engine Aircraft

I. Tail wheel

Γ. Tricycle

##### B. Dual-Engine Aircraft

I. Wing-mounted engines

Γ. Push-pull fuselage-mounted engines

#### II. Commercial Aviation (jet engines)

##### A. Dual-Engine

I. Wing-mounted engines

Γ. Fuselage-mounted engines

##### B. Tri-Engine

I. Third engine in vertical stabilizer

Γ. Third engine in fuselage

## اللون Color

- [الألوان الـ ١٦ الأساسية](#)

هي الألوان التي تظهر على أي متصفح:

black	000000	green	008000
silver	C0C0C0	lime	00FF00
gray	808080	olive	000٨٠٨
white	FFFFFF	yellow	FFFF00
maroon	٨00000	navy	0000٨0
red	FF0000	blue	0000FF
purple	800080	teal	008080
fuchsia	FF00FF	aqua	00FFFF

- [لوحة ألوان الويب المؤلفة من ٢١٦ لون](#)

الألوان التي تظهرها معظم المتصفحات يُمكن توليدها من تركيب الأرقام: هي

**00,33,66,99,CC, FF** لتحديد الأحمر والأخضر والأزرق (الأحمر مثلاً FF0000).

- [جميع الألوان الـ ١٦ مليون \(bits24\)](#)

في حال عدم تمكن المتصفح من إظهار نفس اللون فإنه يُظهر أقرب لون له.

تستخدم الخاصية color لتحديد اللون الأمامي والخاصية background-color لتحديد لون الخلفية.

كما يبين 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 يظهر المتصفح:

To really make it stand out,  
use a red background

## محاذاة النص

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

أما الخاصية text-align فتحدد محاذاة النص وتأخذ إحدى القيم التالية:

left, center, right, justify.

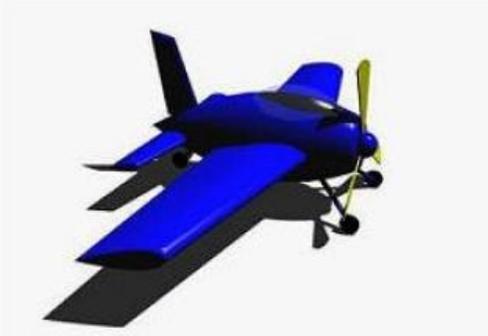
تستخدم الخاصية float غالباً مع الصور والجداول لتحديد كيفية تدفق النص حول الصورة أو الجدول، وتكون:

left, right, none .

يبين 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>
```

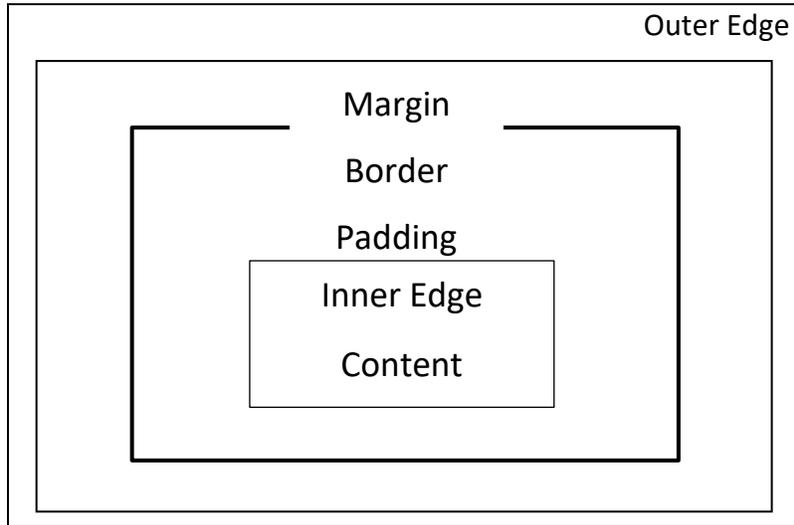
حيث يظهر المتصفح:

<p>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>	
--	--

## نموذج الصندوق The Box Model

يمكن لكل عنصر في وثيقة أن يكون له حدود تأخذ أنماطاً مختلفة كاللون والعرض، تدعى المسافة بين العنصر وحدوده بالبطانة padding أما المسافة بين العنصر والعناصر الأخرى المجاورة له فتدعى الهامش margin.

كما يبين الشكل التالي:



## نموذج الصندوق The Box Model

### الحدود

- تستخدم الخاصية `border-style` لتحديد نمط الحدود وتأخذ القيم

`solid, dotted, dashed, double, none`

- تحدد الخاصية `border-width` عرض الحد ويُمكن أن تأخذ القيم:

`thin, medium (default), Thick`

أو قيمة رقمية بالبركسل 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, borde-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 {borders -top-width: medium;
        borders -bottom-width: thick;
        borders -top-color: red;
        borders - bottom - color: blue;
        borders -top- style: dotted;
        borders - bottom - style: dashed;
      }
      p {borders - style: dashed; borders - width: thin;
        borders - color: green
      }
    </style>
  </head>
  <body>
    <table borders ="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>
</tr>
    <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	,	,	
Lunch		,	,
Dinner	,	,	

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

## البطانة والهوامش

- يمكن استخدام الخاصية margin لتحديد هوامش العنصر الأربعة أو استخدام الخصائص:

margin-top- margin- bottom, margin - left, margin -right

- يمكن استخدام الخاصية padding لتحديد بطانة العنصر، أو الخصائص

Padding-top, padding-bottom, padding-left, padding-right

يبين 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: 0.2 in;
        padding: 0.2 in;
        background -color: #c0c0c0;
        borders - style: solid;
      }
      P. two {margin: 0.1 in;
        padding: 0.3 in;
        background -color: #c0c0c0;
        borders - style: solid;
      }
      P. three {margin: 0.3 in;
        padding: 0.1 in;
        background -color: #c0c0c0;
        borders - style: solid;
      }
      P. four {margin: 0.4 in;
        background -color: #c0c0c0;
      }
      P. five {margin: 0.4 in;
        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 g = 0.2in ]

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

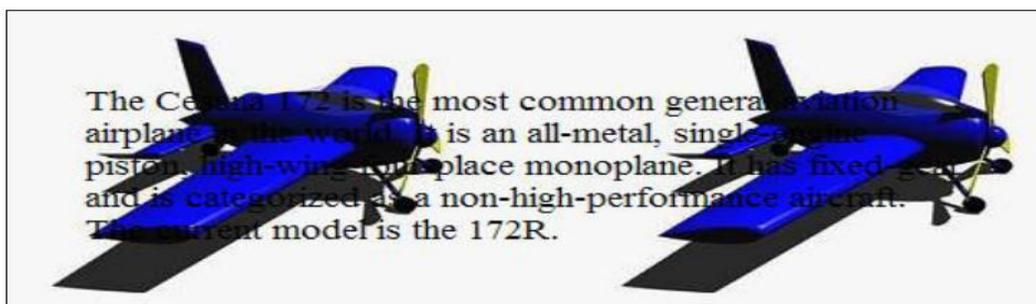
**Here is the last line**

## صور الخلفية

تستخدم الخاصية background-image لوضع صورة في الخلفية، كما يبين المثال التالي:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head <title> background </title>
    <style type="text/css">
      body {background -image: url (plane .png);}
      p {margin -top: 50px; margin -right: 30 px;
        margin -top: 50px; font -size: 14 pt;}
    </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 يظهر المتصفح:



يقوم المتصفح بتكرار الصورة بحيث تملأ مساحة العنصر، يمكن التحكم بعملية التكرار باستخدام الخاصية background-repeat والتي تأخذ القيم:

repeat(default), no-repeat, repeat-x, repeat-y.

تحدد القيمة repeat - x أن التكرار يكون أفقياً فقط، أما القيمة repeat - y فتحدد أن التكرار يكون عمودياً فقط.

يمكن استخدام الخاصية background-position والتي يمكن أن تأخذ عدد من القيم:

top, center, bottom, right, center لتحديد مكان وضع الصور.

## ❖ المؤثرات <span> و <div>

يوفر المؤثر <span> 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">
      .spamred { font - size: 24pt;
        font -family: Arial; color: red}
    </style>
  </head>
  <body>
    It sure is fun to be in <span class="spanred"> total </span>
    control of text
  </ p >
  < body >
</html>
```

يظهر المتصفح:

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

يمكن 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>
    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

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

h1, h2, h3 h4, h5, h6

٢. 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

٣. قم بإنشاء ست فقرات لكل فقرة ترويسة استخدم) من a إلى h6 تستخدم الفقرة الأولى والرابعة p1 تستخدم الفقرة الثانية والخامسة p2 تستخدم الفقرة الثالثة والسادسة p3 يجب أن تظهر الصفحة كما يلي:

## 1 Heading

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 6

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

I. ant farming

II. hang gliding

A. falling down

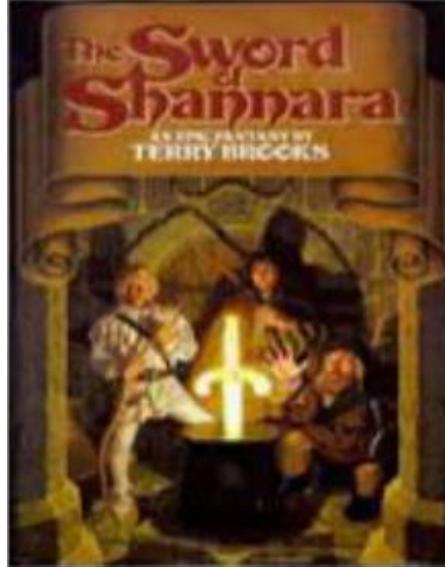
1. leaking blood

B. breaking things

III. drooling

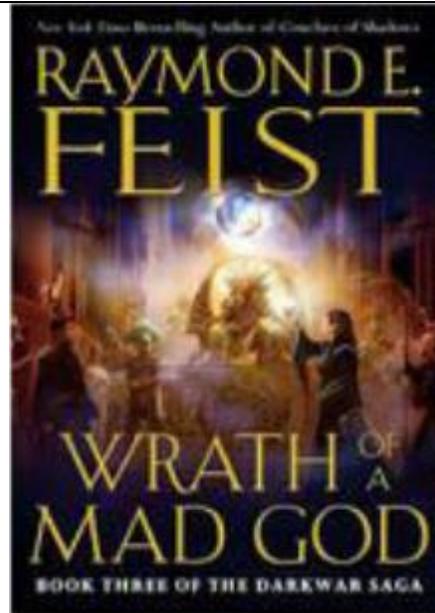
٥. قم بتعريف الأساليب اللازمة لإظهار الفقرتين التاليتين مع الصور الموافقة كما يلي:

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, Krontors 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.



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

### Some of Feist's books include



Magician



Silverthorn



A Darkness at Sethanon

# سادساً: أساسيات لغة JavaScript

في هذا الفصل سنتعرف على المواضيع التالية:

- مقدمة
- الخصائص العامة والشكل
- التعليمات
- الأنماط والعمليات والتعبير
- الإدخال والإظهار
- تعليمات التحكم
- المصفوفات Arrays
- الوظائف Functions
- مطابقة النماذج Pattern Matching

## ❖ مقدمة

ظهرت اللغة الخطاطية JavaScript عام ١٩٩٦ نتيجة تعاون مئمر بين الشركتين Sung Netscape

وأصبحت اليوم معيار عالمي معتمد iso-١٦٢٦٢.

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 يمكن أن تتعامل مع مخدم الويب كعمليات الوصول إلى قواعد البيانات مثلاً.

### البرمجة غرضية التوجه وJavaScript

لا تدعم JavaScript البرمجة غرضية التوجه، أي أنها لا تحوي صفوف classes، بل تعمل أغراضها objects كأغراض وكنموذج للأغراض في أن واحد models of object 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 والشكل

- يمكن تضمين الخططات مباشرة في ملف XHTML:

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

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

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

- يمكن وضع تعليمات اللغة ضمن تعليقات خاصة من الشكل التالي وذلك بهدف إخفاء التعليمات عن المتصفحات التي لا تدعم JavaScript:

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

### المعرفات

- 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

تحوي 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

يمكن 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 والتعبير

### 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:

`٧٢ , ٧٢ , .٧٢ , ٧٢. , ٧٢E2, 7e2, .7e2, 7.e2, 7.2E-2`

– تحاط 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"`

– لا يوجد 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، وتعامل كـ NaN عند معاملتها كرقم، وكـ false عند معاملتها كمتحول منطقي.
- يكون متغير undefined عندما يكون معرفاً ولم تسند له قيمة.

فمثلاً إذا كتبنا التعليمات التالية:

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

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=٠,
```

```
today = "Monday",
```

```
false= flag;
```

## العمليات الرقمية

توفر JavaScript العمليات الرقمية:

++, --, +, -, \*, /, %

تراعي JavaScript أفضلية العمليات حيث تكون أفضلية %, /, \* أعلى من +, - وفي حال تساوي الأفضليات 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 />");
</scrip>
```

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\_VALVE, NaN, POSITIVE\_INFINITY, NEGATIVE\_INFINITY, PI**

– (مثلاً، تُعطي Number.Min\_VALUE أصغر قيمة ممكنة).

– تعيد عملية جبرية مع فيضان overflow القيمة NaN.

– تستخدم 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

تستخدم إشارة الجمع + لجمع السلاسل النصية:

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

## 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 />");  
  x =2007 + "August";  
  // x 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 ="10" * 3;  
  // x now is NaN  
  document. write ("t now is ", t, "<br />");  
</scrip>
```

## تحويل الأنماط الصريح

يمكن طلب التحويل بين الأنماط بشكل صريح كما يلي:

- يُستخدم الباني String للحصول على سلسلة نصية.
- يُستخدم الباني Number للحصول على رقم.
- تستخدم الطريقة toString() على رقم لتحويله إلى سلسلة نصية.
- يمكن 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 str = string (33.33);
// str1 now is "33.33"
document. write ("str1 now is ", str1, "<br />");
    Var num1 = 6.6;
    Var str2 = num1.to string ();
// 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
document. write ("num5 now is ", num5, "<br />");
// num5 now is 33.33
document. write ("num5 now is ", num5, "<br />");
</scrip>
```

## 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 تبين 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

تعيد typeof نمط متغير كما تبين 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

– يمكن التصريح عن متغير تاريخ بأخذ 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. –تستخدم الطريقة write للغرض document بشكل أساسي لخلق خطاطة خرج، كما يبين 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 يمكن 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");
```

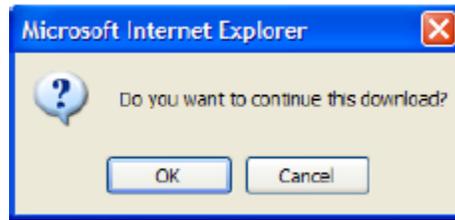


### • الطريقة confirm

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=
```

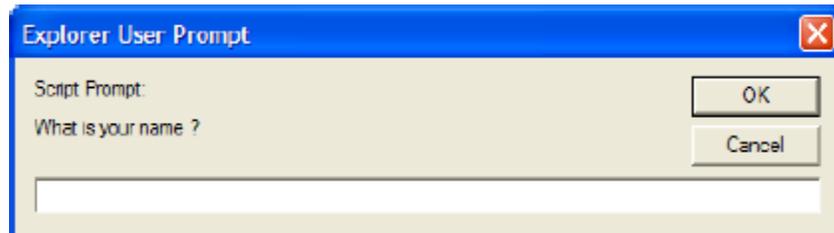
```
confirm ("Do you want to continue this download?");
```



### • 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?", "");
```



مثال: 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(bb 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

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

### التعبير المنطقية

تكون نتيجة تقويم تعبير منطقي القيمة true أو القيمة false.

يوجد ثلاثة أنواع من التعبيرات المنطقية:

- القيم الأساسية Primitive values
  - إذا كانت القيمة رقمية فهي تعتبر true ما لم تكن مساوية للصفر.
  - إذا كانت القيمة نصية تعتبر 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 تعبير 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

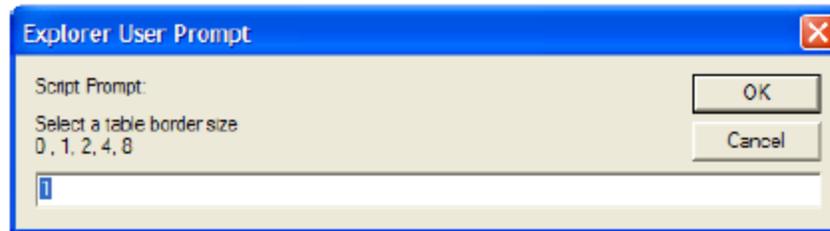
تأخذ التعليمة switch الشكل:

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

مثال: 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> 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 for و while يكونان 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 ❖

يتم تعريف المصفوفات في JavaScript إما باستخدام التعليمات new أو بإسناد عناصر المصفوفة مباشرة.

كما تبين الأمثلة التالية:

```
var myList1 = new Array ( ٢٤);
```

```
var myList2 = ["bread", 99, true];
```

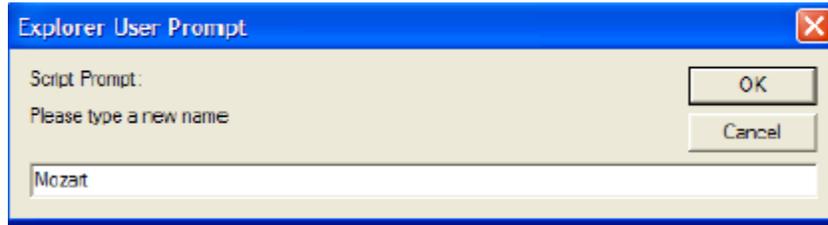
- يكون طول المصفوفة length هو فهرس آخر عنصر فيها زائد 1.
- يُمكن تغيير طول المصفوفة بإسناد قيمة معينة إلى الخاصية length:

```
my List. length = 150;
```

مثال: يوضح المثال التالي استخدام المصفوفات، نصرح 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 ("A1", "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 بفواصل محدد.
- reverse لعكس عناصر المصفوفة.
- slice للحصول على جزء من المصفوفة.

## مثال: يبين المثال التالي بعض طرق التعامل مع المصفوفات

```
<script type="text/javascript">
  a = new Array ("a", "b", "c", "d");
  n = new Array(٣ , ٢ , ١) ;
  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 ❖

- يُشبه شكل الوظائف في JavaScript شكل الوظائف في لغة C.
- يتم التصريح عن الوظائف ضمن المؤثر <head>
- لا يتم التحقق عند استدعاء الوظيفة من نمط المعاملات المممة ولا من عددها. حيث يتم تجاهل المعاملات الزائدة، أما المعاملات الناقصة فتعتبر undefined
- يتم 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 1parameter(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 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:

`\ | 0 [ ] { } ^ $ * + ? .`

(يمكن جعل محرف مترفع يُعامل كمحرف عادي بسبقه بـ \).

تعتبر 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 (تفهرس الأحرف اعتباراً من الصفر)، أو -1 في حال عدم وجوده.

كما يبين المثال التالي:

```
var str = "Rabbits are furry";  
var position = str.search(/bits/)
```

الجواب: 'bits' appears in position ٣

- يُطابق المحرف المترفع (.) أي محرف عدا السطر الجديد، فمثلاً يُطابق النموذج / snow. / كل من snowy, snowe, snowd (المطابقة نقطة سلسلة نصية يجب سبق النقطة في النموذج بـ .، فمثلاً، يُطابق النموذج \3\4\ السلسلة النصية ٣,٤ ولا يطابق ٣٧٤.
- 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. يمكن استخدام المحرف (^) لعكس المحارف المعينة، فمثلاً / [^abc] / تعني أي محرف ما عدا الأحرف a, b, c .

- 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  $\d\d\d\d$  أي رقم تليه نقطة يليها رقمين.
- يُطابق النموذج  $\D\d\D$  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 $x\{4\}z$ /السلسلة $xyyyyz$

- 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).

يبين 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("٤٤٤-٥٤٣٢");
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("٤٤-١٢٣٤");
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

يمكن 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 يطابق

"My pearls are pretty"

### تعديل النماذج

يمكن استخدام المحرف (i) لطلب تجاهل حالة الأحرف، فمثلاً يُطابق النموذج /ok/i كل من 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. replaces (/rab/g, "tim");
document. write ("str is ", str,"<br />");
```

تصبح str مساوية إلى "Some timbits are timid".

## الطريقة match

تستخدم 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].

# سابعاً: التفاعل بين HTML و JavaScript

في هذا الفصل سنتعرف على المواضيع التالية:

- التعامل مع عناصر HTML من خلال JavaScript
- الأحداث Events
- التحقق من الإدخال
- مثال تطبيقي

## ❖ التعامل مع عناصر HTML من خلال JavaScript

يمكن 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

- تكون 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 يمكن الوصول لأي عنصر في الوثيقة باستخدام elements [z] , document.forms[i]

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 أضفنا زر جديد قبل الزر السابق سيُصبح العنوان السابق عنوان الزر الجديد.

## استخدام أسماء العناصر

مثال:

المثال السابق يمكن كتابته:

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

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

## استخدام الطريقة getElementById

مثال:

يمكن 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" name="turnlton"/>
  </form>
  <script "type="text/javascript">
    document. getElementById("turnlton"). value="on";
  </script>
</body>
</html>
```

– نضع في غالب الأحيان لعناصر النموذج اسم name ومعرّف id نضع نفس القيمة عادة).

– نقوم عادة بإعطاء نقوم عادة بإعطاء نفس الاسم لمجموعة من أزرار الخيار أو مجموعة من صناديق التحقق، وعندها يكون هذا الاسم مصفوفة يُمكن من خلالها الوصول لكل عنصر في المجموعة.

مثلاً لو كان لدينا مجموعة من صناديق التحكم:

```
<div id = "Vehicle Group">
<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 ("Vehicle Group");

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

### استخدام الطريقة - getElementBy Name

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

```
var numChecked = 0;

var dom = document. getElementByName("vehicles");

for (index = 0; index < dom. length; index++)
    if (dom[index]. checked)
        numChecked++;
```

يُخبر الحدث بأن شيء محدد قد حصل بفعل المتصفح أو بفعل المستخدم، ندعو معالج الحدث Event handler الخطاطة التي تنفذ 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 الفأرة إلى العنصر	focus	onfocus	>a<
			>input<
			>textarea<
			>select<
التحميل	load	onload	>body<
ضغط زر الفأرة	mousedown	onmousedown	Most elements
تحريك الفأرة	mousemove	onmousemove	Most elements
مغادرة الفأرة العنصر	onmouseout	onmouseout	Most elements
توضع الفأرة فوق العنصر	mouseover	onmouseover	Most elements
تحرير زر الفأرة	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 لخاصية 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 خاصية 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

يعتبر حدثي التحميل load وإنهاء التحميل unload من أهم أحداث المؤثر <body> .

يبين 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 الواصفة onclick:

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

- أو باستخدام:

```
document.getElementById("freeButton"). onclick= free ButtonHandler ();
```

يبين 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;

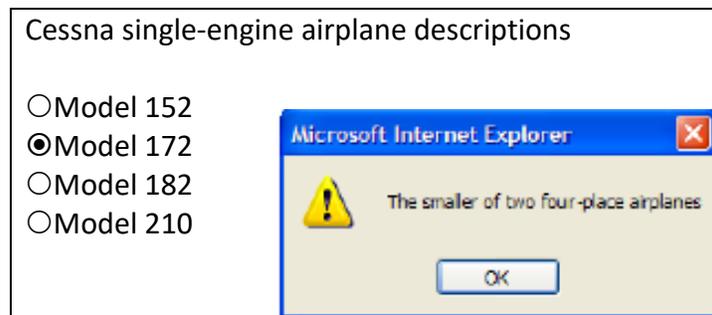
}

}

// -->
</script>
</head>
<body>
<h4> Cessna single-engine airplane descriptions </h4>
<form id="myForm" action="handler">
<p>
<input type="radio" name = "planeButton"
```

```
value = "152" onclick = "plane Choice (152)" /> Model 152  
<br />  
<input type="radio" name = "plane Button"  
value = "172" onclick = "plane Choice (172)" />  
Model 172<br />  
<input type="radio" name = "plane Button"  
value = "182" onclick = "planeChoice (182)" /> Model 182  
<br />  
<input type="radio" name = "plane Button"  
value = "210" onclick = "planeChoice (210)" /> Model 210  
<p />  
</from>  
</body>  
</html>
```

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



## معالجة أحداث صندوق النص وصندوق كلمة السر

يمكن لصندوق النص إنشاء أربعة أحداث مختلفة:

- blur
- focus
- change
- select

### حدث التركيز Focus

يبين 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 ستقوم بإخراج التركيز عن هذا الصندوق كلما قام المستخدم بالانتقال إليه.

Coffee Order Form		
Product Name	Price	Quantity
French Vanilla (1 lb.)	\$٣,٤٩	٢
Hazlenut Cream (1 lb.)	\$٣,٩٥	١
Columbian (1 lb.)	\$٤,٥٩	١

Total Cost

```
<!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 compute Cost () {

var french = document.getElementById("french").value
```

```

<!-- 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

<p>

  <input type="button" value = "Total Cost"

    onclick="compute Cost ();" />

  <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 =" Clean order Form "/>
</p>

</from>

</body>
</html>

```

## ❖ التحقق من الإدخال

تعتبر عمليات التحقق من الإدخالات التي يقوم بها المستخدم من العمليات الأساسية التي يُحبذ القيام بها على الزبون باستخدام JavaScript وذلك تلافياً للقيام بهذه التحقيقات 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:** يقوم المثال التالي بإظهار صندوق نص لإدخال كلمة السر وتأكيدها، 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 وتأكيدها،

وفي حال عدم تطابقهما، يقوم بإظهار رسالة خطأ موافقة ومن ثم يقوم بوضع التركيز على

كلمة السر وتحديد القيمة المدخلة.

Password Input

Reset

Submit Query

```

<!DOCTYPE html >
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> Illustrate password checking </title>
    <script type="text/javascript">
      <!--

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" />
      </br/> <br>

      </input type="reset" name="submit" />
      </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>

```

## مثال ٢:

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

### Customer Information

<input type="text" value="Alkhatib. Bassel. M."/>	Name (last name, first name, middle initial)
<input type="text" value="999-555-3333"/>	Phone number (ddd-ddd-dddd)

• 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> illustrates from input validation </ title >
<script type="text/javascript">
  <!--
  //The event handler function for the name text box
  function ChkName () {
    var my Name = document. getElementById ("Cust Name");
```

```
//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 = my Name. value. Search (
```

```

</script>
</head>
<body>
  <h3> Customer Information </h3>
  <form action="">
    <p>
      <input type="text" id="Cust Name"
        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 ("Cust Name"). onchange = ChkName

```

## ❖ مثال تطبيقي

تهدف الصفحة التالية إلى حساب سعر الإقامة في فندق وذلك وفق مجموعة من الخيارات التي يمكن 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/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 Kitchel	
How Much	٨٠٠	

ولكن مثلًا الأسعار التالية المعتمدة:

Stars		Where		Food		Extra	
Stars	Price	Sea	+50%	Breakfast	٥٠	TV	٥٠
*	١٠٠	Forest	+30%	Lunch	١٠٠	Pool	١٠٠
**	٢٠٠	Mountain	+20%	Dinner	٢٠٠	Kitchen	١٥٠
**	٣٠٠						
***	٤٠٠						
*****	٥٠٠						

يمكن في هذه المسألة البسيطة تثبيت الأسعار في قيم عناصر HTML للرجوع إليها في  
خطاطة 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 Change Flag()
{
document.getElementById ("imgFlag").src=
document.getElementById ("SelectFlag").value
}
function get Price ()
{
var x;
x=document.getElementById ("Select Stars").value;
if (document.getElementById ("RadWhere1"). checked)
x= x* document.getElementById ("RadWhere1").value;
else if (document.getElementById ("Rad Where2"). checked)
x= x* document.getElementById ("RadWhere2").value;
else if (document.getElementById ("Rad Where3"). 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);
}

```

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

```
<input id="RadWhere2" type="radio" name="RadWhere" title="Forest" value="1.3" /> Forest &nbsp; <br />
<input id="RadWhere3" type="radio" name="RadWhere" title="Mountain" value="1.2" /> Mountain&nbsp; <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>
  <td>
    < select id = " select Extra" multiple name = " select Extra " 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>

```

- 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 sr 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.

# ثامناً: الوثائق الديناميكية مع Java Script

في هذا الفصل سنتعرف على المواضيع التالية:

- طرق تموضع العناصر
- تحريك العناصر
- التعامل مع العناصر (الإظهار والإخفاء وتغيير لون الخط)
- المحتوى الديناميكي
- تكديس العناصر
- مؤشر الفأرة
- التحكم بحركة 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، تعتبر JavaScript من أكثر الأدوات ملائمة للقيام بذلك من جهة الزبون.

## تموضع العناصر

يُحدّد تموضع العنصر في الوثيقة بالخصائص الثلاث التالية للواصفة: **position, left, top: style**.

ويوجد ثلاث حالات لتموضع 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 تموضع العناصر الأخرى.

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

يمكن استخدام التوضع المطلق لترتيب نص معين فوق نص آخر مما يخلق ما يُشبه العلامة المائية، وذلك بتوزيع 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 */
. abs text {position: absolute; top: 25px; left: 50px;
font-family: Times; font-size: 24pt;
font-style: italic; letter-spacing: 1em;
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">
</p>
</body>
</html>
```

Apple is the common name for any tree of the genus Malus, of the family Rosaceae. Apple trees grow 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: 500 px;
    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

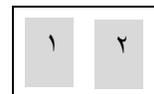
عندما يتم وضع position: relative دون تحديد للإحداثيات top, left يُعامل العنصر كأن لم تسند أي قيمة للخاصية position (إلا أنه سيتمكن تحريكه فيما بعد) أما إذ حددت قيم الإحداثيات left, top فيتم إزاحة العنصر بمقدار هذه الإحداثيات عن الموضع الذي كان سيأخذه فيما لو لم يتم تحديد هذه القيم.

#### مثال ١:

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 ٢:

بوضع إذا قمنا الخصائص:

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

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

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

فسيبتعد الزر الثاني عن الزر الأول بمقدار ٥٠ بكسل.



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

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>
```

Apples are **GOOD** for you.

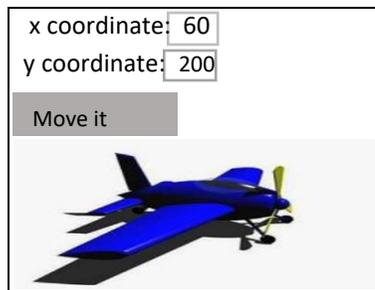
### التوضع الساكن Static

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

يمكن تحريك عنصر وذلك بتغيير إحداثياته left, top بشرط أن يكون موضعه 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 move It (movee, new Top, new Left) {
        dom = document. Get ElementByld (movee). style;
      // Change the top and left properties to perform the move
      // Note the addition of units to the input values
        dom. top = new Top + "px";
        dom. left = new Left + "px";
      }
      //*****
      // -->
```

```
    </script>
  </head>
  <body>
    <form action = "">
      <p>
        x coordinate: <input type="text" id="leftCoord"
          size="3"/>
        <br />
        y coordinate: <input type="text" id =top Coord"
          size="3"/>
        <br />
        <input type="button" value "Move it"
          onclick="move It ('plane',
            document. getElementByld('leftCoord').value,
            document. getElementByld ('topCoord').value"
        />
      </p>
    </form>
    <div id="plane" style="position: absolute;
      top: 115px; left: 0;">
      
    </div>
  </body>
</html>
```

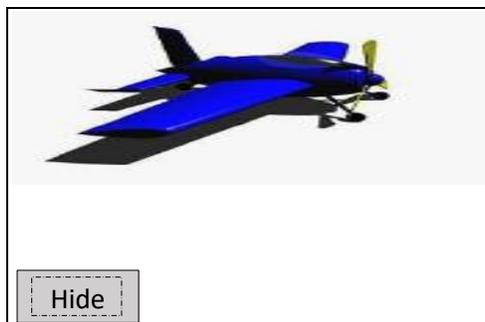


## إظهار وإخفاء العناصر

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

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>
```



## تاسعاً: أساسيات JQuery

في هذا الفصل سنتعرف على المواضيع التالية:

- أساسيات JQuery
- محددات العناصر Selectors
- الواصفات والأساليب في JQuery
- أسلوب الصفحات المتتالي CSS
- مهارات JQuery
- التعامل مع الأحداث

## ماهي JQuery

- هي مكتبة تحوي مجموعة من الوظائف المكتوبة بلغة JavaScript قام ببنائها John Resig عام ٢٠٠٦ بهدف الكتابة أقل الفعل أكثر Write less, do more.
- تُسهل هذه المكتبة التعامل مع مختلف مكونات صفحة HTML من أهم الميزات التي تقدمها هذها المكتبة:

### التعامل مع العناصر DOM manipulation

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

### معالجة الأحداث Event handling

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

### دعم Ajax support

تساعد في تطوير صفحات غنية وتفاعلية باستخدام تكنولوجيا Ajax.

### الاحياء Animations

توفر مجموعة من إمكانيات الإحياء الجاهزة.

### الخفة Lightweight

لا يتجاوز 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

يمكن تنزيل إصدار المكتبة الأخير مجاناً من الرابط [jquery.com](http://jquery.com).

(jquery-1.10.2.min.js).

## استخدام مكتبة JQuery

يمكنك 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 نضيف الأحداث عند جهوزية هذا النموذج. إذا أردت أن يعمل حدث في الصفحة، فيجب استدعائه داخل () Ready (document) 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:

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>
```

## المحددات Selectors

- توفر مكتبة JQuery إمكانية الوصول للعناصر في نموذج كائن الوثيقة (DOM) بسرعة وسهولة.
- يبدأ المحدد ب (\$) ويُمكن 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.

### تبيين الأمثلة التالية اهم إمكانيات التحديد:

\$('*')	تحديد جميع عناصر الوثيقة
\$('p')	تحديد جميع الفقرات
\$('div, p')	تحديد العناصر من النوع p أو div
\$('p > *')	العناصر جميع العناصر أبناء فقرة
\$("#specialID")	تحديد جميع 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.special Class")	تحديد الروابط لها المعرف specialID والصف special Class
\$("p a. specialClass")	تحديد الروابط من الصف special class 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

<code>\$("li: even")</code>	تحديد العناصر ذات الفهرس الزوجي في قائمة li
<code>\$("li: odd")</code>	تحديد العناصر ذات الفهرس الفردي في قائمة li
<code>\$("li: visible")</code>	تحديد العناصر الظاهرة li
<code>\$("li: hidden")</code>	تحديد العناصر المخفية li
<code>\$(": radio")</code>	تحديد جميع أزرار الراديو في نموذج
<code>\$(": checked")</code>	تحديد جميع صناديق التحقق التي لها القيمة checked
<code>\$(":text")</code>	تحديد صناديق النص
<code>\$('div, p')</code>	تحديد جميع العناصر من النوع div أو من النوع p
<code>\$('p. small')</code>	تحديد الفقرات من الصف small
<code>\$("a [@href* =domain.com]")</code>	تحديد جميع الروابط التي تحوي domain.com
<code>\$("input[@name^=myname]")</code>	تحديد العناصر من النوع input والتي اسمها يبدأ بـ myname
<code>\$("input[@name=myname]")</code>	تحديد العناصر من النوع input والتي اسمها يساوي myname

مثال: استخدام محدد المؤثر

عند تنفيذ المثال التالي، ستظهر ٣ رسائل على التوالي تظهر محتوى الفقرات.

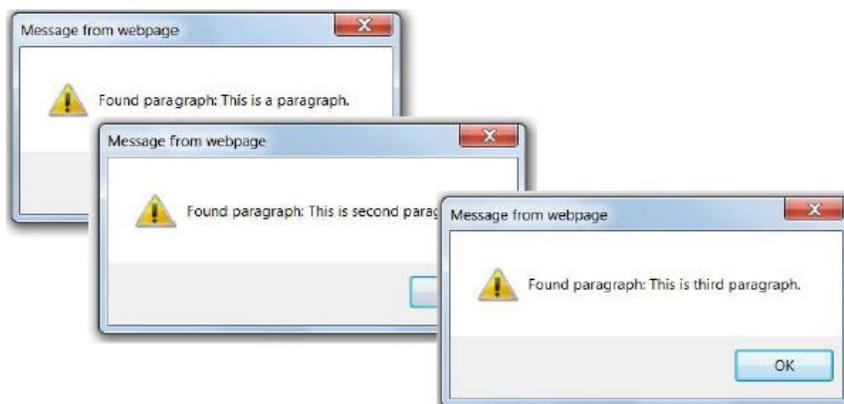
```
<!DOCTYPE HTML >
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>The jQuery Example</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready (function () {
      var pars = $("p");
      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 يظهر المتصفح:

This is a paragraph.

This is a 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="pid3">This is first paragraph. </p>
    <p class="para2" id="pid3">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 يظهر المتصفح:

This is first paragraph.

This is second paragraph.

This is third paragraph.

أكاديمية التعلم  
Academy Of Learning



المؤسسة العامة للتدريب التقني والمهني  
Technical and Vocational Training Corporation



تحت إشراف

9 2 0 0 0 3 1 3 7

a o l . e d u . s a



a o l k s a