



حقيبة تدريبية في تصميم مواقع الانترنت

إعداد
مركز التدريب وخدمة المجتمع



المحتويات

٣	الدرس الأول (مقدمة في الانترنت)
٨	الدرس الثاني (أساسيات لغة HTML)
١٤	الدرس الثالث (الألوان و أنماط الخطوط في لغة HTML)
٢٠	الدرس الرابع (الخطوط Fonts)
٢٧	الدرس الخامس (الصور Images والروابط Links)
٣٣	الدرس السادس (الصوت والفيديو)
٣٨	الدرس السابع (الجدول TABLES)
٤٥	الدرس الثامن (القوائم LISTS)
٥١	الدرس التاسع (الإطارات FRAMES)
٦٩	الدرس العاشر (النماذج FORMS)
٨٣	الدرس الحادي عشر (أوراق الأنماط المتعاقبة CSS)
٩٠	الدرس الثاني عشر (مدخل إلى لغة JavaScript)
٩٨	الدرس الثالث عشر (جمل التحكم في لغة JavaScript)
١٠٧	الدرس الرابع عشر (المصفوفات Arrays والدوال Function)



الدرس الأول

مقدمة إلى الانترنت

لمحة عن الإنترنت

شبكة الإنترنت *Internet* هي اختصار من الأحرف الأولى لعبارة "*International network*" وتعني الشبكة العالمية، ويتم الاتصال بين هذه الحواسيب ونقل المعلومات فيما بينها وفق قواعد وأنظمة محددة، تسمى بروتوكولات *Protocols*

ملكية الشبكة

لا يمكن لأي شخص أو شركة (حكومية أو غير حكومية) أن يدعي ملكية الإنترنت، وفي الوقت ذاته، تقوم شركات رائدة في مجال تكنولوجيا المعلومات بتنظيم خدمة الانترنت.

تاريخ الانترنت

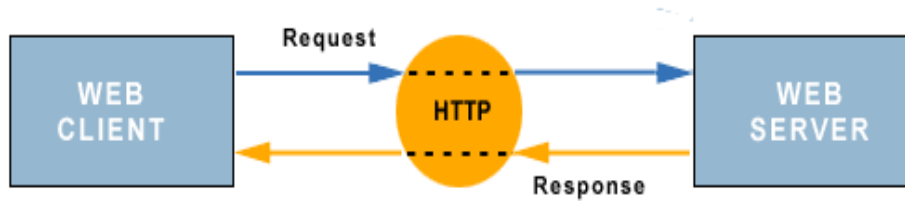
١. بدأت الإنترنت في منتصف الستينيات.
٢. في العام ١٩٦٩ م ربطت هذه الشبكة عدداً من الجامعات الأمريكية.
٣. في العام 1972 م ظهرت خدمة البريد الإلكتروني *Email*.
٤. دخولها مرحلة العالمية، في أوائل السبعينيات إثر ربطها ببعض الجامعات، ومراكز الأبحاث، في أوروبا،
٥. في أوائل الثمانينات، تمكن الناس في جميع أنحاء العالم من التعامل معها.
٦. في التسعينات تطورت شبكة الإنترنت، بشكل كبير، نظراً لظهور أول مستعرضات الويب الرسومية.

شبكة الويب العالمية *World Wide Web*

شبكة الويب العالمية *www* أو *w3* أو الويب (*web*)، هي مجموعة هائلة، من وثائق النص المترابط *Hypertext*، وسميت بالشبكة العنكبوتية.

المصطلح *HTTP*

هو بروتوكول نقل النص المترابط *Hyper Text Transfer Protocol*





المصطلح URL

محدد موقع المصدر **Uniform Resource Locator** الذي يحدد العنوان الحقيقي لموقع المعلومات المقابل للعنوان النصي .

المصطلح HTML

لغة النص التشعبي المترابط **Hyper Text Markup Language** يحدد كيفية عرض المعلومات في صفحات الويب

خدمات الانترنت

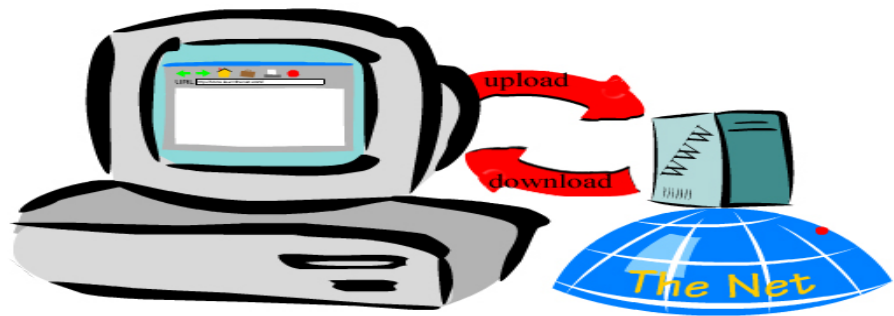
الويب **Web** والتي يعتبرها البعض الخدمة الوحيدة أو قد لا يفرقها عن الانترنت، وتبادل الرسائل **E-Mail**، والحوار **Chatting** ومؤتمرات الفيديو **Conferencing Video**، ونقل الملفات **FTP**، تبادل الأخبار **NNTP(Network News Transfer Protocol)** واستعراض الموقع **Browsing** والبحث **Searching**، والترفيه **Entertainment**، والتجارة الالكترونية **E-commerce**، وإذاعات الانترنت **Internet Radios** وغيرها الكثير .

البريد الإلكتروني E-mail

البريد الإلكتروني **Electronic Mail** هو نظام للتراسل الإلكتروني

نقل الملفات والبرامج FileTransfer Protocol (FTP)

حيث يستخدم البرتوكول **FTP** في رفع الملفات من الجهاز إلى الشبكة (**Upload**) ، أو تحميل الملفات من الشبكة إلى الجهاز (**Download**).





الويب Web

هي طريقة لتنظيم المعلومات،

الترفيه Entertainment

يحب الناس الكثير من التسلية والمتعة على الانترنت، إذ يمكن للمرء اللعب بالألعاب الموجودة على الشبكة، كما يمكنه مشاهدة المسرحيات الإذاعية والتلفزيونية، كذلك يمكن مشاركة اللعب مع أشخاص موجودون على الشبكة

إذاعات الإنترنت Internet Radio Stations

يمكنك أثناء الاستعراض أن تستمع إلى محطات البث والإذاعات العاملة عبر الإنترنت

الاتصال بالانترنت

لنتمكن من الاتصال بالإنترنت يجب أن يحوي جهازك على:

١. بروتوكول TCP/IP (Transmission Control Protocol/Internet Protocol)

الذي يقوم بتأسيس الاتصال وضمان نقل المعلومات من الإنترنت إلى جهازك وبالعكس.

٢. وحدة اتصال مودم Modem

٣. خط اتصال (خط هاتف أو الكبل أو الخطوط الرقمية أو الخطوط اللاسلكية).

٤. مزود خدمة الانترنت. ISP (Internet Service Provider).

٥. برمجيات خاصة بالإنترنت مثل (برنامج المتصفح)، (برنامج إدارة البريد الإلكتروني).

العنوان الإلكتروني

لكل موقع من مواقع الويب عنوان فريد يُسمى العنوان الإلكتروني، وهذا العنوان معني بتحديد مصدر الموقع ومساره، ويُدعى العنوان الإلكتروني بمحدد موقع المصدر - (Uniform Resource Locator - URL). كما ورد الحديث عنه سابقاً ..

النطاقات

مثال: www.google.com.sa

- www تعني أن الموقع مبني وفق معايير الويب .
- google هو اسم الجهة صاحبة الموقع.
- com تُبين مجال عمل الجهة صاحبة الموقع، والجدول التالي يقدم بعض من هذه



الاختصارات:

الاختصار	الشرح
edu	تعليمي
gov	حكومي
com	تجاري
mil	عسكري
net	شركة أو مؤسسة تتولى مجموعة أو شبكة أعمال
org	منظمات

sa : هي اختصار لاسم الدولة التي يتواجد فيها جهاز الخادم، وإليك بعض الاختصارات:

الاختصار	الشرح
sa	المملكة العربية السعودية
ae	الإمارات العربية المتحدة
sd	السودان
jo	الأردن
uk	المملكة المتحدة

صفحات الويب

هي نوع خاص من الملفات المكتوبة بلغة **HTML** ، يتم وضعها في مكان يتسنى للمستخدمين الدخول إليه عبر نظام الدخول لمواقع الشبكة **HTTP** وقراءة محتواها .

أنواع الصفحات.

١. صفحات مكتوبة بلغة (html, htm). **HTML**.
٢. ملفات صور (gif, jpeg, png) .
٣. ملفات وسائط متعددة (ram, mid, wav, swf, avi, mpeg, mpg, mpe) .
٤. ملفات نصوص (txt) مكتوبة في شكل (ASCII) .



لغات برمجة الإنترنت

مع ظهور الشبكة العالمية، نشأت لغات برمجة من أجل التعامل مع صفحات الويب، وتطورت بشكل كبير، وتنقسم هذه اللغات إلى قسمين :

١. لغات برمجة جانب المستضيف *Client Side Programming Languages*

وتُعنى هذه اللغات بجانب المستضيف (المستخدم) *Client Side* وتعمل عن طريق برنامج مستعرض الويب ومن أمثلتها

- *HTML ,Java script ,VBScript*

٢. لغات برمجة جانب الخادم *Server Side Programming Languages*

من أجل الحصول على المعلومات في صفحات مستعرضات الويب نحتاج إلى لغات من جانب الخادم ومن أمثلتها

- *ASP(Active Server Pages).*
- *PHP(Personal Home Page).*



الدرس الثاني

أساسيات لغة HTML

لغة HTML

وهي اختصار لـ **Hyper Text Markup Language** وتعني لغة النص التشعبي المترابط

تاريخ HTML

تعتبر لغة مطورة من لغة سابقة تعرف بـ **SGML** وهي اختصار لـ **Standard Generalized Markup Language**

وقد مرت لغة HTML بالمراحل التالية

- **HTML**: لم تستخدم بادئ الأمر على نطاق واسع إذ لم يتوفر إلا عدداً قليلاً من مغذيات الإنترنت في ذلك الوقت، ولكنها تعتبر أساساً لما تلاها من نسخ.
- **+HTML**: ظهرت في عام ١٩٩٣م، تطور بسيط عن النسخة الأولى.
- **HTML2.0**: ظهرت في عام ١٩٩٤م، وأضيف استخدام الصور داخل صفحات **HTML**.
- **HTML3.0**: ظهرت في عام ١٩٩٥م، ولكن لم تستخدم كثيراً لعدم توافقها في ذلك الوقت مع برامج التصفح المنتشرة حينها.
- **HTML3.2**: ظهرت عام ١٩٩٦م-١٩٩٧م، وفيها ظهر استخدام الجداول.
- **HTML4.0**: ظهرت في عام ١٩٩٧م-١٩٩٨م، فيها استخدمت الإطارات وأوراق الأنماط.
- **HTML4.01**: ظهرت في عام ١٩٩٩م، وهي النسخة الأحدث من هذه اللغة وتم فيها حل العديد من المشاكل التي ظهرت في النسخة السابقة، كما يطلق عليها البعض **HTML5**.

خصائص لغة HTML

لغة **HTML** ليست لغة برمجة بالمعنى المعروف للغات البرمجية الأخرى حيث أن هذه اللغة لا تحتاج إلى مترجم ونستطيع كتابة تعليماتها في أي محرر نصوص مثل برنامج المفكرة **Notepad**



الوسوم الأساسية لصفحة الويب Tags

لنأخذ الوسوم التالية:

وسم البداية	وسم النهاية
<code><HTML></code>	<code></HTML></code>
<code><HEAD></code>	<code></HEAD></code>
<code><TITLE></code>	<code></TITLE></code>
<code><BODY></code>	<code></BODY></code>

ماذا تلاحظ ؟ أن كل منها يتألف من زوج من الوسوم أحدهما وسم البداية ، والآخر وسم

النهاية. ويتميز وسم النهاية بوجود الرمز /

العناصر الأساسية في لغة HTML

<code><html></code>	
<code><head></code>	عنوان الصفحة ومعلومات خاصة عنها
<code><Title></code>	كلمات مفتاحية و الوصف واسم الكاتب والصفحة
<code></Title></code>	-----
<code></head></code>	-----
<code><body></code>	-----
	محتويات الصفحة بما فيها من نصوص وجداول ووصلات تشعبية ورسوم وصور
<code></body></code>	-----
<code></html></code>	



إذن فملف **Html** يبدأ دائماً بالوسم **<HTML>** وينتهي بالوسم **</HTML>**.
أما الوسم **<HEAD>** فيحدد بداية المقطع الذي يحتوي على المعلومات الخاصة بتعريف الصفحة.
كالعنوان الظاهر على شريط عنوان المتصفح. وهذا العنوان بدوره يحتاج لأن يوضع بين
الوسمين: **<TITLE> ... </TITLE>** وبالطبع يجب كتابة الوسم **<HEAD>** لكي ننهي هذا المقطع.
نأتي إلى الوسم **<BODY>** والذي يتم كتابة نصوص صفحة الويب ضمنه، بالإضافة إلى إدراج
الصور والجداول وباقي محتويات الصفحة. وهو أيضاً يحتاج إلى وسم الإنهاء **</BODY>**

سنبدأ بتطبيق هذه المعلومات بصورة عملية؟ قم بفتح برنامج المفكرة واكتب ما يلي:

```
<HTML>
<HEAD>
<TITLE>
    This is a Test Web Page
</TITLE>
</HEAD>
<BODY>
    I'm Writing My First Web Page
</BODY>
</HTML>
```

والآن سنقوم بحفظ ما كتبناه في ملف وبأي اسم نختاره. ولا تنسى أن الامتداد المستخدم في أسماء
ملفات **HTML** هو **.htm** أو **.html**. ومن الأفضل أن تقوم بإنشاء مجلد مستقل على القرص الصلب
لكي تحفظ به ملفاتك فهذا يسهل عليك عملية استرجاعها للعرض أو التحديث
ولمشاهدة نتيجة ما كتبناه. نقوم بتشغيل متصفح الإنترنت واختار الأمر **Open ...** من قائمة **File**.
ثم حدد المسار الذي يوجد به الملف.

بعض الملاحظات عند كتابة صفحات الويب

لا يوجد فرق بين كتابة الوسوم بالأحرف الإنجليزية الكبيرة **UPPERCASE** أو الأحرف الصغيرة
lowercase.

لذلك تستطيع الكتابة بأي شكل منهما أو حتى الكتابة بكليهما.

١. إن المتصفحات لا تأخذ بعين الاعتبار الفراغات الزائدة أو إشارات نهاية الفقرات أو بعبارة

أخرى فإنه باستطاعتك كتابة ملفك السابق بالشكل التالي:

```
<HTML><HEAD><TITLE> This is a Test Web Page </TITLE></HEAD><BODY>
    I'm Writing My First Web Page</BODY></HTML>
```

أو بالشكل التالي:

```
<HTML>
<HEAD>
<TITLE>
    This
```



```

is a
Test
Web Page
</TITLE>
</HEAD>
<BODY>
I'm
Writing
My First
Web Page
</BODY>
</HTML>

```

أو حتى بهذا الشكل:

```

<HTML> <HEAD> <TITLE>
This is a Test Web Page
</TITLE>
</HEAD>
<BODY>
I'm Writing My First Web Page
</BODY>
</HTML>

```

وفي كل الحالات ستحصل على نفس النتيجة

والآن قد نتساءل،

كيف يمكن التحكم بمقدار النص المكتوب في كل سطر وكيف يمكن تحديد نهاية الفقرة وبداية الفقرة التي تليها؟ **والإجابة عليه هي:**

سوف نستخدم الوسم **
** لتحديد النهاية للسطر. والبدء بسطر جديد

<input checked="" type="checkbox"/>	الوسوم مفردة وليس لها نهاية .
<input checked="" type="checkbox"/>	الوسم
 مفرد، أي ليس له وسم نهاية.

ونعود إلى المثال السابق، قم بتعديل الملف لكي يصبح بالشكل التالي

```

<HTML>
<HEAD>
<TITLE>
This is a Test Web Page
</TITLE>
</HEAD>
<BODY>
<BR> I'm Writing My First <BR>
Web Page
</BODY>
</HTML>

```

```

<HTML>
<HEAD>
<TITLE>

```

أما الفراغات فتعتبر رموزاً خاصة لذلك لا نستطيع التحكم بها وبعدها إلا باستخدام الـ **Non Breakable Space** (و الأحرف هي اختصار للعبارة **Non Breakable Space**). وإذا أردت إدخال عدة فراغات بين نص وآخر ما عليك إلا كتابة هذا الـ **Non Breakable Space** بنفس عدد الفراغات المطلوب. كما يجب عليك التقيد بالأحرف الصغيرة هنا.

إننا لنعد إلى المفكرة ونكتب ملفنا بالشكل التالي:

12



الدرس الثالث

التنسيق

الألوان و أنماط الخطوط في لغة HTML

الألوان Colors

خصائص الـ BODY

سنقوم بالتعرف على الخصائص التي يمكن إضافتها إلى الوسم **<BODY>** من أجل التحكم بالشكل العام للصفحة، وخصوصاً فيما يتعلق بالألوان. عندما لا نقوم نحن بتحديد إعدادات اللون الخلفية ولون الخط فإن المستعرض يعتمد خلفية رمادية وخط صغير نسبياً لونه أسود. كما ذكرنا في المحاضرة الماضية بأن محتويات الصفحة من الوان وصور وجداول تكتب ضمن الوسمين

<BODY> ...</BODY>

الوسم يقوم بإخبار المتصفح عن العمل الذي يجب القيام به أما الخاصية **Attribute** فتحدد الكيفية التي سيتم بها أداء هذا العمل.

انظر الى الشفرة التالية:

<BODY BGCOLOR = "FFFFFF">

...

</BODY>

لقد قمنا بإضافة الخاصية **BGCOLOR** إلى الوسم **<BODY>** ، وهي تقوم بتحديد لون الخلفية للصفحة. أما القيمة **FFFFFF** فهي القيمة التي تمثل اللون المختار وهو هنا اللون الأبيض، (لاحظ أنها مكتوبة بين إشارتي " ") ولو أردت تمثيل اللون الأسود لكتبت الرمز 000000. أو الرمز CC6699 للون الأزرق الفاتح..

تلاحظ أن القيم السابقة مكونة من ستة رموز، وهي مكتوبة بالصيغة التالية:-

FF	FF	FF
00	00	00
66	99	CC
RR	GG	BB
↑	↑	↑
رمزان يمثلان اللون الأحمر	رمزان يمثلان اللون الأخضر	رمزان يمثلان اللون الأزرق



هناك ثلاثة ألوان أساسية هي **الأحمر والأخضر والأزرق**، ولكل منها يوجد ٢٥٦ درجة لونية ويعبر عن هذه الدرجات بالأرقام من ٠٠٠ وحتى ٢٥٥. ومن خلال مزج هذه الألوان بدرجاتها اللونية المختلفة نحصل على الألوان الأخرى.

☑	إن أي لون هو مزيج -وبنسبة معينة من الدرجات- من هذه الألوان الثلاثة .
☑	بعض المتصفحات لا تتعرف على رموز الألوان إلا بوضع إشارة # قبل هذه الرموز، لذلك من الأفضل استخدامها دائماً.

لكن من أي جاءت الرموز **FFFFFF** والتي عبرت عن اللون الأبيض ؟
إنها ببساطة أرقام... مكتوبة بالنظام السداسي عشري (نظام عددي أساسه الرقم ١٦ ويعبر عنه باستخدام الأرقام العادية من ٠ إلى ٩ والرموز **A,B,C,D,E,F**). فالرقم ٢٥٥ بالنظام العشري العادي يكافئه الرقم **FF** بالنظام السداسي عشري.
إذن فالرقم السداسي عشري **FF** على اليسار يمثل الدرجة ٢٥٥ للون الأحمر. والرقم **FF** في الوسط يمثل الدرجة ٢٥٥ من اللون الأخضر. والرقم **FF** على اليمين يمثل الدرجة ٢٥٥ من اللون الأزرق.

وعلى هذا المنوال يعبر عن اللون الأزرق الفاتح بالرقم السداسي : **CC ٦٦٩٩**
أما اللون الأسود فرقمه هو ٠٠٠٠٠٠.

[التعامل مع اسم اللون بشكل صريح](#)

	Black		Gray		Silver		White
	Yellow		Lime		Aqua		Fuchsia
	Red		Green		Blue		Purple
	Maroon		Olive		Navy		Teal



عودة إلى الوسوم و خصائصها

<BODY BGCOLOR="#FFFFFF" BACKGROUND="image path">

.....

</BODY>

تقوم الخاصية **BACKGROUND** بتحديد صورة خلفية (ورق جدران) للصفحة حيث

image path تمثل المسار الكامل للصورة

بعض خصائص الوسم BODY

ربما لاحظت خلال استخدامك للإنترنت أن معظم الوصلات التشعبية (**Links**) التي تنقر عليها لتنتقل إلى صفحات أو مواقع أخرى على الشبكة هي دائماً مميزة باللون الأزرق، وأن الوصلات التي قمت بزيارتها فعلاً قد تحول لونها إلى القرمزي. هذه هي الألوان الافتراضية التي تعتمد عليها المتصفحات. لكن قد لا يعجبك ذلك وتريد تغيير هذا النظام.

سنوضح الخصائص التي تقوم بالتحكم في ألوان النصوص بالجدول التالي:

الوظيفة	الخاصية
تحديد لون النص الأساسي للصفحة	TEXT="#rrggbb"
تحديد لون الوصلات التشعبية	LINK="#rrggbb"
تحديد لون الوصلات التشعبية التي تمت زيارتها visited links	VLINK="#rrggbb"
تحديد لون الوصلة التشعبية الفعالة أي عندما يتم النقر عليها active links	ALINK="#rrggbb"



أنماط الخطوط

فيما يلي الوسوم الخاصة بها متنوعة بمثال ونتيجته:

١. الخط الغامق (الأسود العريض)، ونستخدم له الوسوم التالية:

** ... **

** ... **

<i>This is Bold Text</i>	 Bold Text
<i>This is Strong Text</i>	 Strong Text

٢. الخط المائل *Italic*

<I> ... </I>

** ... **

<i>This is Italic Text</i>	<I> Italic Text </I>
<i>This is Emphasized Text</i>	 Emphasized Text

٣. الخط المسطر *Under line*

<U> ... </U>

<i>This is <u>Underlined Text</u></i>	<U> Underlined Text </U>
---------------------------------------	---

٤. الخط المرتفع

^{...}

<i>This is ^{Superscript Text}</i>	<SUP> Superscript Text </SUP>
--	--

٥. الخط المنخفض

_{...}

<i>This is _{Subscript Text}</i>	<SUB> Subscript Text </SUB>
--	--

٦. خط كبير

<BIG> ... </BIG>

<i>This is Big Text</i>	<BIG> Big Text </BIG>
-------------------------	--

٧. خط صغير

<SMALL> ... </SMALL>

<i>This is Small Text</i>	<SMALL> Small Text </SMALL>
---------------------------	--



٨. نص يعترضه خط

<STRIKE> ... </STRIKE>

<S> ... </S>

 ...

This is Striked Text	<STRIKE> Striked Text </STRIKE>
This is Striked Text	<S> Striked Text </S>
This is Striked Text	< del > Striked Text </ del >

٩. نص الآلة الطابعة TeleType

<TT> ... </TT>

This is Tele Type Text	<TT> Tele Type Text </TT>
------------------------	---------------------------

ونص الآلة الطابعة يعرف أيضاً بالنص موحد المسافات *Monospaced Text*.

ولتوضيح هذا المفهوم إليك المثال التالي:

إذا أخذنا الحرفين m, i وكتبنا كل منهما عشر مرات متتالية نلاحظ أن المساحة التي شغلها الحرف m هي أضعاف المساحة التي شغلها الحرف i

iiiiiiiiii

mmmmmmmmmm

أما عند استخدام الوسم <TT> ... </TT> فإن المساحة التي يشغلها كلا الحرفين تصبح موحدة

iiiiiiiiiiiiii
mmmmmmmmmm

مثال يجمع بين عدة تنسيقات

</><U>

This is a Bold, Italic and Underlined Text

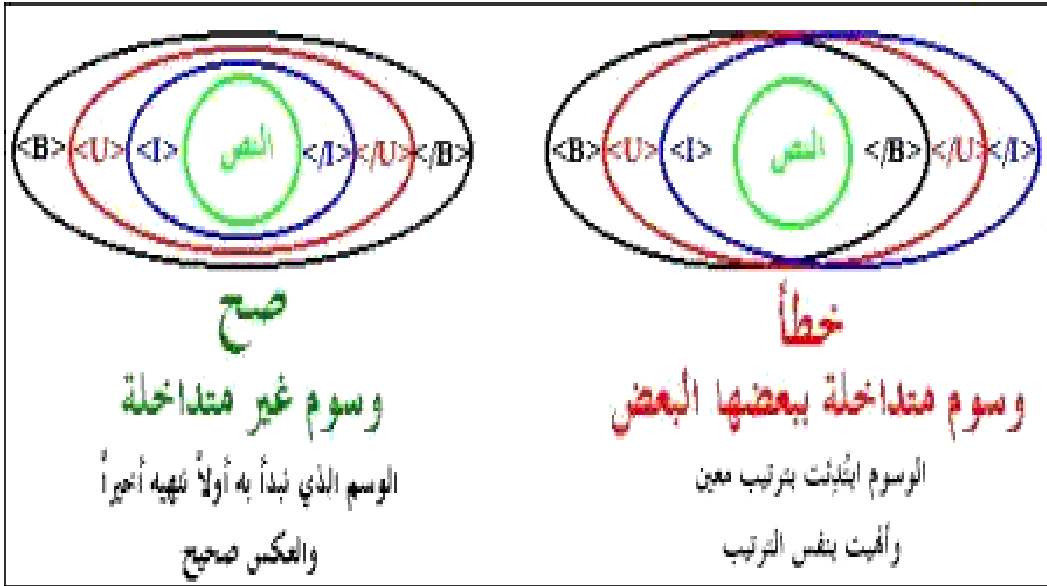
</U> </I>

النتائج

This is a Bold, Italic and Underlined Text

عند استخدام الوسوم المتعددة في مقطع واحد يجب مراعاة عدم التداخل بينها! ... كيف؟

أنظر إلى الشكل التالي:





الدرس الرابع

التنسيق في لغة HTML

الخطوط Fonts

محتويات المحاضرة

– الخطوط Fonts

- حجم الخط **Font Size**
- لون الخط **Font Color**
- اسم الخط **Font Face**
- تنسيق الحرف الأول (حرف استهلاكي)

– إضافات

- الوسم **<BASEFONT>**
- وسم المسطرة الأفقية **HR**
- وسم تحريك النص **marquee**

الخطوط Fonts

الإعدادات الافتراضية للخط هي خط عادي، نوعه **Times New Roman** وحجمه ٣ (بمقياس متصفحات الإنترنت).

الوسم الخاص بالخطوط هو

** ... **

وهو يقوم بالتحكم بالخطوط من حيث النوع واللون والحجم. أما الخصائص التي نستخدمها مع هذا الوسم



الوظيفة	الخاصية
تقوم هذه الخاصية بتحديد نوع الخط الذي نريده، وقد نقوم بتحديد أكثر من نوع معاً. وفي هذه الحالة إذا لم يتواجد الخط المحدد أولاً على جهاز الشخص الذي يتصفح الموقع يتم اعتماد الخط الثاني ... وهكذا ... Text ... 	Face
الوظيفة	الخاصية
أما هذه الخاصية فتحدد لون الخط. ... Text ... 	Color
أما هذه الخاصية فنقوم بتحديد حجم الخط ، هناك سبعة أحجام لأي خط تستطيع المتصفحات التعرف عليها. ونقوم بتحديد الحجم المطلوب بأسلوبين: ١. أولهما المباشر. حيث يتم كتابة رقم يتراوح ما بين ١-٧. أي أننا نختار الحجم الذي نريده مباشرة. ... Text ... ٢. أما الأسلوب الثاني فهو النسبي: حيث تكتب الأرقام من ١ إلى ٦ مرفقة إما بإشارة + أو بإشارة -. ... Text ... وفي هذه الطريقة فإن الأرقام ١-٦ تمثل درجات التكبير (+) أو التصغير (-) للخط وذلك نسبة إلى الحجم الافتراضي. فمثلاً الرقم ٤+ يعني تكبير الخط أربع درجات عن الحجم الافتراضي وهو ٣، أي أنه يصبح بالحجم ٧. بالمقابل فإن الرقم ١- يعني تصغير الخط درجة واحدة أي يصبح بالحجم ٢.	Size



لا تنسى أن تتأكد من كتابة أسماء الخطوط بالصورة الصحيحة هجائياً .	☑
في الأسلوب النسبي لا نستطيع الحصول على أكثر من سبعة أحجام للخطوط. حتى وإن حاولنا كتابة أرقام أكبر أو أصغر مثل +5 أو -3.	☑

سؤال يتبادر الى الذهن

سقول لقد ثبت حجم الخط على حده الأدنى عند الدرجة -2 وعلى حده الأعلى عند الدرجة +4.

إذن ما الفائدة من وجود الدرجات الأخرى الأقل من -2 والأكثر من +4؟
وأنا أجيبك بسؤال آخر: ماذا لو قمنا بتغيير الحجم الافتراضي للخط في كل الصفحة إلى 1 بدلاً من 3؟

ألا نحتاج في هذه الحالة إلى الدرجات من +1 إلى +6 لتمثيل الأحجام الأكبر منه؟
وإذا قمنا بتحديد 7 كحجم افتراضي

ألا نحتاج إلى الدرجات من -1 إلى -6 لتمثيل الأحجام الأصغر منه؟
إذن نحن نحتاج فعلاً إلى هذه الدرجات لكي نغطي جميع الاحتمالات الواردة.

أمثلة توضح كيفية استخدام الوسم FONT

خط من نوع Traditional

Arabic بحجم 6 ولون الخط أحمر

النتائج

بحجم 6 ولون الخط أحمر Traditional Arabic خط من نوع

This font is Times New Roman, Size is 5, Color is Blue

النتائج

This font is Times New Roman, Size is 5, Color is Blue



تنسيق الحرف الأول (حرف استهلاكي)

 C ustomize your font

النتائج

Customize your font

الوسم <BASEFONT>

وظيفته هي تحديد نوع الخط وخصائصه بالنسبة للصفحة كلها .
أي أنه يقوم بتعريف نوع الخط الأساسي الذي سيستخدم في الصفحة ويحدد لونه وحجمه .
وهو وسم مفرد ويكتب هذا الوسم مباشرة بعد وسم <BODY> . أما الخصائص المستخدمة معه فهي نفس الخصائص المستخدمة مع الاختلاف فقط في استخدام الخاصية (Name بدلاً من Face) .

مثال

<BASEFONT Name="Arial" COLOR="#FF0000" SIZE="5">

وبدراسة هذا المثال نستنتج أنه يقوم بتعديل الخط الافتراضي للصفحة بحيث يصبح نوعه Arial وحجمه 5 ولونه أحمر .

كل النصوص المكتوبة في تلك الصفحة سيطبق عليها هذا النمط من الخط. ما لم نقم طبعاً باستخدام الوسوم ... لتعديلها والتحكم بمظهرها والتي هي أكثر تحديداً وأكثر مرونة من الوسم <BASEFONT> .	☑
الخاصية Text التي تكتب مع الوسم <Body> لتحديد لون نص الصفحة لا يوجد تعارض بينها وخاصية Color في الوسم <BASEFONT> فأنت بكل بساطة تستطيع استخدام أي منهما في صفحتك. وإذا حدث واستخدمت كلاهما فإن اللون المحدد مع الوسم <BASEFONT> هو الذي سيطبقه المتصفح ويعتمده .	☑

وسم المسطرة الأفقية <HR> (Horizontal Rule)

يستخدم هذه الوسم لرسم خط افقي ويستخدم عادة لفصل الفقرات ، وهو وسم مفرد .

ويمتلك هذه الوسم عدة خصائص منها

١. خاصية سُمك الخط SIZE

مثلاً:

<HR SIZE="5">

<HR SIZE="1">

<HR SIZE="10">



٢. خاصية عرض الخط *WIDTH*

والتي من الممكن أن تأخذ قيمة مطلقة أو نسبية
مثلاً

- العرض كقيمة نسبية

```
<HR WIDTH="80%">
```

- العرض كقيمة مطلقة

```
<HR WIDTH="400">
```

- تغيير الحجم والعرض معاً

```
<HR SIZE="5" WIDTH="60%">
```

٣. خاصية المحاذاة *ALIGN*

والتي تأخذ القيم *center, left, right*

مثلاً:-

```
<HR WIDTH="80%" ALIGN="center">
```

```
<HR WIDTH="400" ALIGN="left">
```

```
<HR SIZE="5" WIDTH="60%" ALIGN="right">
```

٤. الخاصية *NOSHADE*

كما تلاحظ فإن هذا الخط يكتسب مظهراً غائراً ثلاثي الأبعاد وإذا أردت خطاً عادياً غير غائر

فقم بإضافة الخاصية *NOSHADE*

مثلاً:-

```
<HR SIZE="5" WIDTH="60%" ALIGN="center" NOSHADE>
```

٥. الخاصية *COLOR*

تستخدم لتغيير لون الخط وهي تعمل فقط مع مستكشف الانترنت *Internet Explorer*

مثلاً:-

```
<HR SIZE="5" WIDTH="60%" ALIGN="center" COLOR="#FF0000" NOSHADE>
```

وسم تحريك النص *marquee*

يستخدم الوسم *marquee* لجعل النص متحركاً

الصيغة العامة له

```
< marquee >...text...</ marquee >
```

وهو يمتلك العديد من الخصائص منها



١. الخاصية *behavior*

والتي تأخذ القيم التالية *alternate, slide, scroll*

٢. الخاصية *direction*

والتي تأخذ القيم التالية *right, left, up, down*

٣. الخاصية *scrollDelay*

تستخدم لتسريع مرور النص بمقدار القيمة المسندة.

٤. الخاصية *scrollAmount*

تستخدم لإبطاء مرور النص بمقدار الفترة المسندة.

٥. الخاصية *loop*

تحدد عدد مرات تكرار الحركة

مثال:-

اضافة نص متحرك مع تطبيق بعض خصائص النص المتحرك مثل
لون خلفية النص ازرق ، عدد مرات التكرار ٣ ، الاتجاه ناحية اليسار

```
<marquee behavior="scroll" direction="left" loop="3" bgcolor="#0066CC">
  In The Name Of Allah
</marquee>
```




الدرس الخامس

الصور *Images* والروابط *Links*

الصور *Images*

- يضيف استخدامك للصور والرسومات في صفحة الويب تأثيراً جيداً على من يتصفحون صفحتك على الشبكة.
- لقد اقتصر حديثنا عن الصور في المحاضرات الماضية على إضافة خلفيات للصفحات
- إن الوسم الرئيسي المستخدم لتعريف صورة ما داخل الصفحة هو `` وهو وسم مفرد والخاصية المستخدمة لتحديد موقع واسم الصورة هي `SRC`.
- يمكن إهمال المسار إذا كانت الصورة موجودة في نفس مكان الصفحة المستعرضة.
- يجب أن تكون الصورة من النوع `jpg` أو `gif` أو `png`.

أمثلة

``

الصيغة اعلاه تفترض أن الصورة موجودة في نفس الدليل الفرعي أو المجلد حيث يتواجد ملف **HTML** الذي تعمل عليه

``

الصيغة اعلاه تفترض أن الصورة موجودة في مجلد متفرع عن المجلد الموجود به ملف **HTML** وهو **images**

``

الصيغة اعلاه تفترض أن الصورة موجودة على موقع الكتروني

خصائص الصور

١. عرض وارتفاع الصورة *Image Height and Width*

كل صورة لها ارتفاع وعرض ، ولإخبار المستعرض عن حجمها ، نستخدم الخاصيتين

(a) خاصية الارتفاع *height*

(b) خاصية العرض *Width*

مثال:-

``

٢. الخاصية *ALIGN*



تستخدم هذه الخاصية لتحديد محاذاة الصورة مع النص المرافق لها أو لنقل بعبارة أخرى: تحديد موقع النص الذي يليها بالنسبة لها وهي تأخذ القيم:

BOTTOM, TOP, MIDDLE, LEFT, RIGHT

وأوضح لك تأثير كل قيمة كما يلي:

(a) في الحالة العادية وعندما لا نقوم بتحديد أي محاذاة فإن النص الذي يلي الصورة يظهر بمحاذاة الحافة السفلى لها. أما باقي النص فيمتد أسفلها.

وهذه هي الحالة الافتراضية لظهور الصور والتي تمثلها القيمة **BOTTOM**

(b) وعند تحديد القيمة **TOP** فإن السطر الأول من النص الذي يلي الصورة يقع بمحاذاة الحافة العليا لها. أما باقي النص فيمتد أسفلها.

(c) أما عند تحديد القيمة **MIDDLE** فإن السطر الأول من النص يقع بمحاذاة منتصف الصورة. كذلك فإن باقي النص يمتد أسفلها.

(d) القيمة **LEFT** تؤدي إلى محاذاة الصورة إلى أقصى اليسار. مع التقاف النص الذي يليها على الجهة اليمنى ولعدة أسطر حسب ارتفاع الصورة.

(e) أما القيمة **RIGHT** فتؤدي إلى محاذاة الصورة إلى أقصى اليمين. مع التقاف النص الذي يليها على الجهة اليسرى ولعدة أسطر حسب ارتفاع الصورة.



٣. الخاصية VSPACE

تستخدم لتحديد المسافة العمودية الفاصلة بين النص والحافتين العليا والسفلى للصورة.

مثال:

```
<IMG SRC="image.jpg" ALIGN="TOP" VSPACE="20" >
```

٤. الخاصية HSPACE

تستخدم لتحديد المسافة الأفقية الفاصلة بين النص والحافتين اليمنى واليسرى للصورة.

مثال:

```
<IMG SRC="image.jpg" ALIGN="RIGHT" SPACE="20">
```

٥. الخاصية BORDER

هذه الخاصية وظيفتها إضافة إطار حول الصور والتحكم بسُمكهِ. ويتم التحكم بالسُمك من خلال إسناد رقم يمثل السُمك بالبكسل. والقيمة الافتراضية له هي ٠ أي لا يوجد إطار حول الصورة.

مثال:

لإضافة إطار سُمكه ٥ بكسل نكتب الشفرة التالية:

```
<IMG SRC="image.jpg" BORDER="5">
```

٦. الخاصية ALT

تقوم هذه الخاصية بتحديد نصاً بديلاً يظهر مكان الصورة. وهذا النص يلاحظ خصوصاً عندما يكون خيار "إظهار الصور تلقائياً" غير فعال في المتصفح. كما تستطيع ملاحظته في الفترة التي تسبق تحميل الصور وخاصة في المواقع بطيئة التحميل.

```

```

أنواع الصور المستخدمة في صفحات الويب.

- تنسيق رسومات الإنترنت (GIF) Graphic Internet Format
- رابطة مجموعة خبراء التصوير (JPEG) Joint Photographic Experts Group
- رسومات الشبكة المحمولة (PNG) Portable Network Graphics
- وغيرها من التنسيقات الأخرى.



الروابط Links

الروابط هي الخيوط التي تؤلف حلقات الوصل بين الملايين من مواقع الشبكة العنكبوتية. فعندما تنقر على وصلة ما فنتنقل إلى صفحة أخرى في نفس الموقع... وتنقر على وصلة أخرى فنتنقل كلياً إلى أحد المواقع في الجانب الآخر من العالم... وصلة تجعلك تحمل ملفاً وأخرى تجعلك تشغل مقطعاً موسيقياً وثالثة تعرض لك صورة...

والسؤال الآن :- كيفية يتم إدراج الوصلات التشعبية ؟

هناك عدة خيارات للارتباطات التشعبية، منها أن يكون الارتباط لموقع آخر، أو أن يكون لصفحة أخرى داخل الموقع نفسه، ومنها أن يكون لمكان آخر في نفس الصفحة (إلى أعلى أو أسفل على سبيل المثال) أو أن يكون ارتباط لعنوان بريد إلكتروني *E-mail* وفي جميع الحالات فإن المبدأ واحد لكن تختلف بعض التفاصيل

نستخدم الوسوم

<A>

كوسوم أساسية لإدراج الارتباطات التشعبية، وهي اختصار لكلمة *Anchor*. وهي لا تعمل لوحدها بل تتطلب إضافة خصائص معينة

➤ الخاصية *HREF* (Hypertext Reference)

وهي تحدد عنوان *URL* الذي سينقلك إليه الرابط.

الشكل العام للخاصية *HREF* :

< a href="إليه الانتقال URL المراد الانتقال إليه" >

نص الارتباط

يجب أن يكتب عنوان الموقع المراد الانتقال إليه كاملاً.





أنواع الارتباطات

١. النوع الأول : إدراج ارتباط تشعبي يشير إلى موقع خارجي

لنقم بإدراج ارتباط تشعبي يشير إلى موقع جامعة الملك سعود. وعنوانه

<http://www.ksu.edu.sa>

في هذه الحالة يتم كتابة الشفرة بالشكل التالي:

```
<A HREF=" http://www.ksu.edu.sa ">
    King Saud University
</A>
```

ويظهر الارتباط كما يلي:

King Saud University

كما يمكنك إدراج صورة بدلاً عن الكلمات -كما تشاهد في الكثير من المواقع .
وكل ما عليك فعله في هذه الحالة هو كتابة الوسم الخاص بإدراج الصورة بين الوسمين

<A> ...

بالشكل التالي:

```
<A href="http://www.ksu.edu.sa">
    
</A>
```

والذي يؤدي إلى ظهور الصورة المحددة في المسار كارتباط تشعبي لموقع جامعة الملك
سعود

حيث **Image Path** تمثل مسار الصورة .

وبشكل عام :

<p>يجب أن يكتب عنوان الموقع المراد الانتقال إليه كاملاً. فإن أي شيء يوضع بين الوسمين <A> ... سوف يكون الوسيلة أو العنوان الذي ينقلنا إلى الموقع المشار إليه في الارتباط التشعبي، سواء كان هذا الشيء نصاً أو صورة أو كلاهما معاً.</p>	<input checked="" type="checkbox"/>
---	-------------------------------------



٢. النوع الثاني: هو أن يشير الارتباط التشعبي إلى ملف موجود في نفس الموقع (أي ملف محلي) سواء كان ملف **HTML** أو صورة أو غير ذلك.
في هذه الحالة فإن ما يكتب مع الخاصية **HREF** هو اسم الملف المطلوب الوصول إليه.

(a) إنشاء ارتباط تشعبي يقودنا إلى الصفحة الرئيسية لموقع (ملف **HTML**)
نفرض ان ملف الصفحة الرئيسية هو **index.html** وبالتالي فان الشفرة الخاصة بالوصول الي الصفحة الرئيسة من أي صفحة اخرى داخل الموقع هي
Main Page
[Main Page](#)

(b) ارتباط يشير الى صورة

شعار الجامعة

٣. النوع الثالث: وهو ان تقوم بإدراج ارتباط تشعبي لعنوان بريد إلكتروني، يؤدي النقر عليه إلى إطلاق برنامج البريد الإلكتروني للزائر بشكل تلقائي. فالاختلاف الوحيد الذي يطرأ هنا هو كتابة كلمة **MAILTO** بعد خاصية **HREF** لكي تدل على أن العنوان الذي يلي هو عنوان **EMAIL** وليس أي عنوان آخر .

My Email

[My Email](#)

➤ الخاصية target
هذه الخاصية تقوم بتحديد المكان الذي ستفتح فيه النافذة الجديدة وتأخذ القيم التالية

_self وتقوم بفتح الارتباط في نفس النافذة الحالية

_blank وتقوم بفتح الارتباط في نافذة جديدة

Teachers College

الدرس السادس

الصوت والفيديو

Voice & Video



في البداية قم بتحميل ملفات الصوت والفيديو والFLASH الموجودة على موقعي الشخصي داخل مجلد باسم الصوت والفيديو على سطح المكتب .	✓
---	---

تضمين الصوت والفيديو لصفحات الويب

– يمكن إضافة قصاصات الصوت والفيديو إلى صفحة الويب باستخدام الوسم **embed** ، وتعتبر هذه الطريقة المعيارية لإضافة الصوت والفيديو .

– الخاصية **src** : تستخدم لتحديد موقع واسم الفيديو أو الصوت.

`<embed src="alarifi.wmv " >`

مثال يوضح كيفية ادراج الفيديو في صفحة **html**

```
<html dir="rtl" >
<head>
<title>الصوت والفيديو</title>
</head>
<body>
<embed src=" Wildlife.wmv">
<p>
انقر فوق تشغيل من أجل سماع الصوت، ومن أجل التوقف انقر فوق إيقاف مؤقت/إيقاف
</p>
</body>
</html>
```

خصائص الوسم Embed المتعلقة بطريقة العرض

٧. الخاصية **width**

تستخدم للتحكم في عرض مشغل الفيديو

٨. الخاصية **height**

تستخدم للتحكم في ارتفاع مشغل الفيديو

٩. الخاصية **hidden**

تستخدم لإخفاء مشغل الفيديو وسماع الصوت فقط

مثال:- يوضح كيفية استخدام خصائص العرض للفيديو

```
<html dir="rtl" >
<head>
<title>الصوت والفيديو</title>
</head>
<body>
<embed src=" Wildlife.wmv " width="300" height="350" >
<p>
انقر فوق تشغيل من أجل سماع الصوت، ومن أجل التوقف انقر فوق إيقاف مؤقت/إيقاف
```



```
</p>
</body>
</html>
```

خصائص الوسم Embed المتعلقة بعمل الفيديو

١. الخاصية **autostart** هذه الخاصية تأخذ قيمتين

(a) **true** تعني تمكين خاصية التشغيل التلقائي وهي الوضع الافتراضي

(b) **false** تعني تعطيل خاصية التشغيل التلقائي

٢. الخاصية **loop** هذه الخاصية تأخذ قيمتين

(a) **true** تعني تكرار تشغيل الفيديو

(b) **false** عدم تكرار تشغيل الفيديو وهي الوضع الافتراضي

٣. الخاصية **volume** تستخدم للتحكم في حجم الصوت

مثال:- يوضح كيفية استخدام خصائص عمل الفيديو

```
<html dir="rtl" >
<head>
<title>الصوت والفيديو</title>
</head>
<body>
<embed src=" Wildlife.wmv " autostart="true" volume="50">
<p>يتم قراءة الملف تلقائياً دون أن يتكرر وبحجم قدره ٥٠</p>
</body>
</html>
```

مثال:- يوضح كيفية تكرار تشغيل الفيديو

```
<html dir="rtl">
<head>
<title>الصوت والفيديو</title>
</head>
<body>
<embed src=" Wildlife.wmv " loop="true">
<pre>
Loop سيتم تكرار الفيلم باستمرار بسبب الخاصية
</pre>
</body>
</html>
```




☑	<p>الوسم <code><pre></code> هو وسم زوجي ، والذي سيفيدك كثيرا و يريحك كثيرا ، وظيفة هذا الوسم هي جعل لغة ال <code>HTML</code> أن تنتبه إلى الأسطر والمسافات وعدم تجاهلها ، أي أن هذا الوسم يغنيك عن الوسم الفردي <code>
</code> والرمز <code>&nbsp;</code></p>
---	--

إضافة الفلاش

– يمكن أن نحدد أبعاد الفلاش كقيمة مطلقة أو نسبية البيكسل كوحدة قياس.

مثال

```
<embed src="flashname.swf" width="360" height="135">
```

مثال: – يوضح كيفية تشغيل الفلاش

```
<html dir="rtl">
<head>
<title>الفلاش</title>
</head>
<body>
<embed src="computer-intro_exam.swf" height="90%" width="80%">
</body>
</html>
```

إضافة الصوت كخلفية

الوسم `<bgsound>`

يستخدم لإضافة الصوت كخلفية ويعمل تلقائياً عند فتح الصفحة المدرج فيها كخلفية كما

يمكن التحكم في عدد مرات تكراره باستخدام الخاصية `loop`

مثلاً :

```
<bgsound src="Al-Fatihah.mp3" loop="2">
```

إضافة ملفات الوسائط باستخدام الوسم `<object>`

– الخاصية `data` تستخدم لتحديد اسم وموقع ملف الوسائط

– الخاصية `type`: لتحديد نوع الوسائط

▪ للصور `"image/jpeg", "image/gif"`

▪ للصوت `"audio/wav", "audio/basic"`



▪ للفديو. "video/mpeg", "video/quicktime".

— الخاصية **align**:

تستخدم لتحديد موقع النص الذي يلي ملف الوسائط وتأخذ القيم التالية

"bottom" | "left" | "middle" | "right" | "top"

— الخاصية **width**: تستخدم لتحديد العرض

— الخاصية **height**: تستخدم لتحديد الارتفاع

— الخاصية **hspace**: تستخدم لتحديد المسافة الأفقية بين النص وملف الوسائط من الناحيتين

اليمنى واليسرى

— الخاصية **vspace**: تستخدم لتحديد المسافة العمودية بين النص وملف الوسائط من الناحيتين

العليا والسفلى

مثال:- يوضح كيفية استخدام الوسم **object**

```
<html dir="rtl" >
<body>
<html>
<body>
<object data=" computer-intro_exam.swf " width="90%" height="90%" >
</object>
</body>
</html>
```



الدرس السابع

الجدول

Tables

تعد الجداول من أقوى الأدوات التي تتضمنها لغة **HTML** حيث لا يخلو اي موقع في الإنترنت من استخدامها.

الوسوم الأساسية الخاصة بالجدول

<code><TABLE>...</TABLE></code>	وسوم تعريف الجدول
<code><TR>...</TR></code>	وسوم تعريف الصف في الجدول <i>Table Row</i>
<code><TD> Cell Data </TD></code>	وسوم تعريف الخلايا في الصف <i>Table Data</i>

مثال:- انشاء جدول مكون من ثلاثة صفوف وعمودين

```
<TABLE>
<TR>
  <TD> Data </TD>
  <TD> Data </TD>
</TR>
<TR>
  <TD> Data </TD>
  <TD> Data </TD>
</TR>
<TR>
  <TD> Data </TD>
  <TD> Data </TD>
</TR>
</TABLE>
```

طريقة التعامل مع الجداول

تتم على ثلاثة مستويات:

1. مستوى الجدول ككل .
2. مستوى الصفوف ككل أو كل واحد على حده .
3. مستوى الخلايا ككل أو كل واحدة على حده.

ولكل من هذه المستويات خصائصه التي ينفرد بها كما أن هناك خصائص مشتركة تستخدم مع كل الوسوم.

1. الخصائص التي تستخدم مع الوسوم `<TABLE>...</TABLE>` (على مستوى الجدول ككل)



الوظيفة	الخاصية
تقوم هذه الخاصية بإضافة حدود للجدول وتحديد سماكتها، والقيمة الافتراضية لها هي صفر أي لا حدود <TABLE BORDER="5"> <TABLE BORDER="0">	BORDER
نستخدم هذه الخاصية لتحديد عرض الجدول ككل. وهناك أسلوبين لتحديد العرض: المطلق والنسبي. <TABLE WIDTH="600"> <TABLE WIDTH="80%">	WIDTH
لتحديد ارتفاع الجدول وهناك أسلوبين لتحديد الارتفاع: المطلق والنسبي. <TABLE HEIGHT="500"> <TABLE HEIGHT="100%">	HEIGHT
لتحديد المسافة بين كل خلية من خلايا الجدول <TABLE CELSPACING="10">	CELLSPACING
لتحديد المسافة الفاصلة بين الحدود وبداية النص في كل خلية. أو لنقل: تحديد حجم الهوامش لخلايا الجدول. <TABLE CELLPADDING="10">	CELLPADDING
لتحديد محاذاة الجدول أفقياً على الصفحة يميناً أو وسطاً أو يساراً. وهو يأخذ القيم <i>left, center, right</i> <TABLE ALIGN="Left"> <TABLE ALIGN="Right">	ALIGN
ويستخدم لتحديد لون الخلفية للجدول <TABLE BGCOLOR="#00FFFF">	BGCOLOR

٢. الخصائص التي تستخدم مع الوسوم <TR>...</TR> (على مستوى الصف)

الوظيفة	الخاصية
لتحديد محاذاة النص أفقياً داخل الخلايا التي يتكون منها الصف، والقيم المحتملة لها هي <i>Right, Left, Center</i> والقيمة الافتراضية هي <i>Left</i>	ALIGN



<p>لتحديد المحاذاة العمودية للنص داخل خلايا الصف، وذلك إما للأعلى أو للأسفل أو في المنتصف. وقيمها على التوالي هي: Middle, Bottom, Top</p>	VALIGN
<p>لتحديد لون الخلفية للخلايا التي يتكون منها الصف. وهنا يتم تجاهل لون الخلفية المحدد ضمن وسم <TABLE> ويتم تطبيق اللون المحدد هنا.</p>	BGCOLOR

مثال يوضح استخدام خصائص الوسوم **<TR> ... </TR>**

```

<TABLE BORDER="5" HEIGHT="300">
<TR ALIGN="Left " BGCOLOR="#808080" >
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR ALIGN="Right " BGCOLOR="#C0C0C0" >
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR ALIGN="Center">
<TD> Data </TD>
<TD> Data </TD>
</TR>
</TABLE>

```

٣. الخصائص التي تستخدم مع الوسوم **<TD> </TD>** (على مستوى الخلية)

الخاصية	الوظيفة
ALIGN	تحدد محاذاة النص الموجود في الخلية أفقياً، والقيم المستخدمة هي Left, Center, Right
VALIGN	تحدد المحاذاة العمودية للنص، وهو يأخذ القيم Top, Middle, Bottom
WIDTH	تحدد عرض الخلية، وذلك بكتابة القيمة المباشرة للعرض المطلوب بالبكسل، أو بكتابة رقم يمثل النسبة المئوية. ويكفي تحديد العرض للخلايا في أحد الصفوف لكي يتم تطبيقه على كل الخلايا في كل الصفوف.



تحدد الإرتفاع المطلوب للخلية في الصف، وذلك بالطرق المباشرة أو النسبية. وقيامك بتحديد ارتفاع إحدى الخلايا في الصف يؤدي إلى تطبيقه على كل الخلايا فيه.	HEIGHT
تحدد لون خلفية الخلية .	BGCOLOR
يقوم بدمج الخلية الحالية مع العدد المطلوب من الخلايا التي تليها أفقياً <TD COLSPAN="n"> حيث n هو عدد الخلايا التي سيتم دمجها	COLSPAN
يقوم بدمج الخلية الحالية مع العدد المطلوب من الخلايا التي تليها عمودياً (أي أسفلها). <TD ROWSPAN="n"> وبالطبع n هو عدد الخلايا التي سيتم دمجها	ROWSPAN

<ul style="list-style-type: none"> أولوية التنفيذ للخصائص التي تتكرر مع جميع الوسوم. <p>خذ مثلاً الخاصية BGCOLOR. كيف يتم التعامل معها إذا كررت مع جميع الوسوم؟ بكل بساطة يتم تطبيق اللون المحدد مع وسم الخلية، فإذا لم يكن محدداً يطبق اللون المحدد مع وسم الصف، فإذا لم يوجد يطبق اللون المحدد مع وسم الجدول. وإذا لم يكن هذا محدداً بدوره يتم اعتماد لون خلفية الصفحة المحدد في الوسم <BODY>.</p>	☑
---	---

الخصائص WIDTH, HEIGHT أفضل طريقة للتعامل معها هي قيامك بتحديد العرض و الإرتفاع للجدول ككل من خلال الوسم <TABLE> .	☑
إذا أردت أن تحتوي بعض الصفوف في الجدول على عدد من الخلايا أقل من باقي الصفوف، فلا يكفي أن تقوم بحذف وسوم الخلايا منها. كما في المثال التالي	☑

```
<TABLE BORDER="5">
<TR>
<TD> Data </TD>
</TR>
```

```
<TR>
<TD> Data </TD>
<TD> Data </TD>
```



</TR>

<TR>
<TD> Data </TD>
</TR>
</TABLE>

لقد بقي مكان الخلايا المحذوفة محجوزاً كما لو أنها لم تحذف. أما الخلايا الباقية فظلت محتفظة بنفس خصائصها، أي أننا لم نستفد من عملية الحذف. والحقيقة أن الطريقة المثلى لذلك هي أن تقوم بدمج الخلايا معاً وذلك باستخدام الخصائص **COLSPAN, ROWSPAN**. والآن نعيد كتابة شفرة الجدول السابق مع خاصيتي الدمج

```
<TABLE BORDER="5">
<TR>
  <TD COLSPAN="2"> Data </TD>
</TR>

  <TR>
    <TD> Data </TD>
    <TD> Data </TD>
  </TR>
  <TR>
    <TD COLSPAN="2"> Data </TD>
  </TR>
</TABLE>
```

لاحظ أن العدد ٢ هو عدد الخلايا التي قمنا بدمجها. ولاحظ أيضاً أنني لم أقم بإعادة وسوم الخلايا المحذوفة لأننا أصلاً لا نحتاج لها بعد أن قمنا بالدمج. وكقاعدة أساسية: كل خلية يتم دمجها يجب بالمقابل حذف وسوم التعريف الخاصة بها. ما عدا تعريف الخلية الأساسية بالطبع.

مثال آخر: لنقم بدمج الخلايا الموجودة في العمود الأول

```
<TABLE BORDER="5">
<TR>
  <TD ROWSPAN="3"> Data </TD>
  <TD> Data </TD>
</TR>

  <TR>
    <TD> Data </TD>
  </TR>

  <TR>
    <TD> Data </TD>
  </TR>
</TABLE>
```



الوسوم الأخيرة المستخدمة في الجداول هي `</CAPTION>` ... `<CAPTION>` وهي تختص بإضافة عنوان رئيسي للجدول ككل. لذلك فهي عندما تكتب يتم وضعها مباشرة بعد الوسم `<TABLE>` وبصورة مستقلة وليس ضمن وسوم الصفوف أو الخلايا.

مثلاً:

```
<TABLE BORDER="5">  
<CAPTION> Table Caption </CAPTION>
```




الدرس الثامن

القوائم

Lists

تحتوي لغة **HTML** على مجموعة من الوسوم الخاصة بتنظيم البيانات في قوائم وباستخدام عدة خيارات.

وهناك نوعين من القوائم :

٤. القوائم المتسلسلة (المرتبة) **Ordered Lists**.

٥. القوائم غير المتسلسلة (غير المرتبة) **Unordered Lists**.

عند التعامل مع القوائم بنوعيهما نحتاج إلى وسوم خاصة بتحديد بداية ونهاية القائمة ووسوم تحدد عناصر هذه القائمة.

** ... **

بالنسبة للقوائم المتسلسلة (المرتبة) نستخدم الوسوم

** ... **

أما بالنسبة للقوائم غير المتسلسلة (غير المرتبة) فنستخدم

ولتعيين كل عنصر من عناصر القائمة نستخدم الوسم **** وهو وسم مفرد يكتب في

بداية السطر الخاص بكل عنصر في القائمة **List Item**.

مثال علي القوائم المتسلسلة

```
<html dir="rtl">
```

```
<body>
```

بعض مدن المملكة

```
<OL>
```

```
<LI>الرياض
```

```
<LI>الدمام
```

```
<LI>مكة
```

```
<LI>جدة
```

```
<LI>ابها
```

```
</OL>
```

```
</body>
```

```
</html>
```

مثال علي القوائم غير المتسلسلة (غير المرتبة)

```
<html dir="rtl">
```

```
<body>
```



امثلة على بعض الجامعات السعودية

 جامعة الملك سعود

 جامعة الملك خالد

 جامعة الملك عبد العزيز

 جامعة الملك فهد

</body>

</html>

والخاصية الوحيدة التي تستخدم مع هذه الوسوم هي **TYPE** ووظيفتها تحديد شكل الرمز الظاهر مع عناصر القائمة، وعادة تستخدم مع وسوم بداية القوائم أو وبذلك نحدد رمزاً واحداً لكل القائمة. ولكن نستطيع استخدامها أيضاً مع وسم العناصر لإعطاء تحكم أكبر في مظهر القائمة من خلال تحديد رمز مختلف لكل عنصر. فعند وضعها ضمن تعريف القوائم المتسلسلة تأخذ القيم:

A, a, I, i التي تغير رموز الترقيم من الأرقام العادية الافتراضية (والتي رمزها ١) إلى ترقيم باستخدام الأحرف اللاتينية الكبيرة أو الصغيرة، أو باستخدام الأرقام الرومانية كما ترى في الجدول التالي:

<UL TYPE="square">	<UL TYPE="circle">	<OL TYPE="i">	<OL TYPE="I">	<OL TYPE="a">	<OL TYPE="A">
تكون ▪	تكون ○	.i	.I	.a	.A
علي ▪	علي ○	.ii	.II	.b	.B
شكل ▪	شكل ○	.iii	.III	.c	.C
مربع ▪	دائرة ○	.iv	.IV	.d	.D
		.v	.V	.e	.E

تغيير نقطة البداية للقوائم المرتبة

إذا اردت أن تبدأ الترقيم برقم غير الواحد كل ما عليك هو أن تضع الخاصية **Start** كما بالصيغة التالية وتحدد رقم البداية الذي تريده

<OL START="5">

هناك وسوماً أخرى تستخدم لإنشاء القوائم غير المتسلسلة، وبنفس الطريقة المستخدمة مع

... وهي :-



```
<DIR> ... </DIR>
<MENU> ... </MENU>
```

مثال علي القوائم غير المتسلسلة (غير المرتبة)

```
<html dir="rtl">
<body>
انواع القوائم
<DIR>
    القوائم المرتبة <LI>
    القوائم غير المرتبة <LI>
    قوائم التعريف <LI>
</DIR>
</body>
</html>
```

قوائم التعريف Definition Lists

هناك نوع خاص من القوائم يدعى قوائم الشرح أو التعريفات وكما يدل الاسم تستخدم عندما نريد إدراج قائمة من المصطلحات يتبع كل واحد منها شرح أو تعليق. ولإنشاء هذه القوائم نحتاج إلى ثلاثة وسوم هي :

١. **<DL> ... </DL>** لتعريف بداية ونهاية القائمة
٢. **<DT>** ويوضع قبل كل مصطلح لتحديده، وهو وسم مفرد.
٣. **<DD>** وهو وسم الشرح أو التعليق وهو أيضا مفرد.

مثال علي قوائم التعريف

```
<html >
<body>
<DL>

<DT>HTML <DD>Hyper Text Markup Language

<DT>WWW <DD>World Wide Web

<DT>FTP <DD>File Transport Protocol

</DL>
</body>
</html>
```



والنتيجة هي :-

HTML

Hyper Text Markup Language

WWW

World Wide Web

FTP

File Transport Protocol

العناوين (Headings)

وسم العنوان بقسم بطريقة متتالية بحيث يوجد ٦ مستويات مختلفة من العناوين من أكبر مستوى ويأخذ الوسم **<H1>** إلى أصغر مستوى ويأخذ الوسم **<H6>**

مثال

الوسم	التأثير
<H1>Heading 1</H1>	Heading 1
<H2>Heading 2</H2>	Heading 2
<H3>Heading 3</H3>	Heading 3
<H4>Heading 4</H4>	Heading 4
<H5>Heading 5</H5>	Heading 5
<H6>Heading 6</H6>	Heading 6

كل ما عليك فعله هو أن تضع نص العنوان داخل الوسم وعندها يظهر النص على حسب المستوى المختار .



مثال علي العناوين

```
<html >
<body>
<html >
<body>
```



<DL>

<DT><H1>HTML</H1> <DD><H3>Hyper Text Markup Language</H3>

<DT><H1>WWW</H1> <DD><H3>World Wide Web</H3>

<DT><H1>FTP</H1> <DD><H3>File Transport Protocol</H3>

</DL>

</body>

</html>

والنتيجة هي :-

HTML

Hyper Text Markup Language

WWW

World Wide Web

FTP

File Transport Protocol



الدرس التاسع

الاطارات Frames

- تعتبر الإطارات **frames** من الأساليب القوية لتخطيط صفحات الويب، حيث تسمح الإطارات بعرض أكثر من مستند في نافذة متصفح واحدة .

- قبل أن نبدأ..... سنقوم بإنشاء خمسة ملفات بسيطة لكي نستخدمها في تطبيق الأمثلة وهي كما يلي

frame1.html, frame2.html, frame3.html, frame4.html, frame5.html

- ونبدأ الآن بتعريف الملف الرئيسي الذي سيضم كافة الإطارات والملفات. وهو ملف ذو حالة

خاصة حيث نقوم **باستخدام الوسوم** **<FRAMESET> ... </FRAMESET>**
بدلاً من الوسوم **<BODY> ... </BODY>**

إذن لملف الرئيسي للإطارات لا يتضمن تعريفاً باستخدام BODY	<input checked="" type="checkbox"/>
---	-------------------------------------

```
<HTML>
<HEAD>
<TITLE>Master File</TITLE>
</HEAD>

<FRAMESET>
</FRAMESET>

</HTML>
```

خصائص الوسم **FRAMESET**

١. الخاصية **COLS**

وهي تعرف عدد وأحجام الإطارات العمودية للصفحة. وتُحدد الأحجام بطريقتين الطريقة المباشرة أو الطريقة النسبية... أو كلاهما معاً.
 أمثلة توضح هذه الخاصية :



<pre><FRAMESET COLS="50%,50%"> </FRAMESET></pre>	<p>يحدد إطارين عموديين حجم كل منهما ٥٠% من حجم الشاشة</p>
<pre><FRAMESET COLS="20%,50%,30%"> </FRAMESET></pre>	<p>يحدد ثلاثة إطارات أحجامها ٢٠% و ٥٠% و ٣٠% على التوالي من حجم الشاشة</p>
<pre><FRAMESET COLS="200,300,*"> </FRAMESET></pre>	<p>يحدد ثلاثة إطارات عمودية الأول حجمه ٢٠٠ بيكسل، والثاني ٣٠٠ بيكسل، ما الثالث * أي أنه غير محدد بحجم معين ولكنه سيكون بالحجم المتبقي من الشاشة (طالما أننا لا نعرف استبانة الشاشة التي يستخدمها زائر الموقع)</p>
<pre><FRAMESET COLS="200,*,15%,20%"> </FRAMESET></pre>	<p>يحدد أربعة إطارات حجم الأول هو ٢٠٠ بيكسل، والثالث ١٥% من حجم الشاشة، والرابع ٢٠% من حجم الشاشة أما الثاني فسيكون حجمه بما تبقى من الشاشة.</p>
<pre><FRAMESET COLS="150,*,2*"> </FRAMESET></pre>	<p>يحدد ثلاثة إطارات الأول حجمه ١٥٠ بيكسل.... أما المساحة المتبقية فتقسم على أساس أن الإطار الثالث حجمه هو ضعفي (٢*) حجم الإطار الثاني (*)</p>

٢. الخاصية ROWS

وهي تحدد عدد وحجم الإطارات الأفقية (الصفوف) داخل الصفحة. وذلك بنفس الأسلوب المتبع مع الأعمدة، أي إما باستخدام الطريقة النسبية أو المطلقة.



امثلة توضح هذه الخاصية :

<code><FRAMESET ROWS="50%,50%"></code> <code></FRAMESET></code>	يحدد إطارين أفقيين ارتفاع كل منهما ٥٠% من ارتفاع الشاشة
<code><FRAMESET ROWS="20%,50%,30%"></code> <code></FRAMESET></code>	يحدد ثلاثة إطارات أفقية ارتفاعاتها ٢٠% و ٥٠% و ٣٠% على التوالي من ارتفاع الشاشة
<code><FRAMESET ROWS="50,120,*"></code> <code></FRAMESET></code>	يحدد ثلاثة إطارات أفقية الأول ارتفاعه ٥٠ بيكسل، والثاني ١٢٠ بيكسل، والثالث سيكون بالارتفاع المتبقي من الشاشة
<code><FRAMESET ROWS="50,*,15%,20%"></code> <code></FRAMESET></code>	يحدد أربعة إطارات أفقية ارتفاع الأول هو ٥٠ بيكسل، والثالث ١٥% من ارتفاع الشاشة، والرابع ٢٠% من ارتفاع الشاشة أما الثاني فسيكون ارتفاعه بما تبقى من ارتفاع الشاشة.
<code><FRAMESET COLS="*,2*"></code> <code></FRAMESET></code>	يحدد إطارين الثاني ارتفاعه ضعفي ارتفاع الأول

الوسم <FRAME>

وهو وسم مفرد أي ليس له وسم نهاية تماماً مثل . وفيه نقوم بتحديد مصدر وخصائص كل ملف نريد إظهاره داخل أحد الإطارات. ويتم استخدام هذا الوسوم عدة مرات بنفس عدد الإطارات المذكورة داخل <FRAMESET>. وسنستخدم الخاصية SRC لتحديد مصدر الملف.



الآن سنقوم بإتمام الشيفرة لبعض الأمثلة المذكورة أعلاه.

المثال الأول:

```
<FRAMESET COLS="50%,50%">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
</FRAMESET>
```

المثال الثاني:

```
<FRAMESET COLS="200,400,*">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
  <FRAME SRC="frame3.html">
</FRAMESET>
```

المثال الثالث:

```
<FRAMESET ROWS="50*,15%,20%">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
  <FRAME SRC="frame3.html">
  <FRAME SRC="frame4.html">
</FRAMESET>
```

المثال الرابع:

```
<FRAMESET COLS="*,2*">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
</FRAMESET>
```

وبالإضافة إلى ما ذكر، نستطيع إدراج صورة مباشرة داخل الإطار وباستخدام **FRAME** **>** **SRC** تماماً كما ندرجها باستخدام **IMG SRC** كما بالمثال التالي:

```
<FRAMESET COLS="50%,50%">
  <FRAME SRC="frame1.html">
  <FRAME SRC="Jellyfish.jpg">
</FRAMESET>
```

كيفية تقسيم الصفحة الى اطرار افقية وعمودية معاً

نوضح ذلك من خلال المثال التالي

صفحة مكونة من صفين، الثاني منهما مقسم بدوره إلى عمودين

بما أن الصفحة تحتوي على صفين نقوم بتعريفهما أولاً حسب الارتفاعات المرغوب بها:

```
<FRAMESET ROWS="100,*">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
</FRAMESET>
```



لكن الصف الثاني مقسم إلى عمودين وهنا يعتبر بمفهوم لغة HTML وكأنه صفحة إطارات جديدة لذلك نعاود استخدام تعريف الصفحات! أي <FRAMESET> مرة أخرى.

```
<FRAMESET ROWS="100,*">
  <FRAME SRC="frame1.html">
  <FRAMESET>
  </FRAMESET>
</FRAMESET>
```

وبما أن الصف الثاني (أو لنقل الإطار الثاني) مقسم إلى عمودين، إذن بقي علينا إضافة تعريف لهذه الأعمدة. وبذلك تكون الشفرة النهائية كالتالي:

```
<FRAMESET ROWS="100,*">
  <FRAME SRC="frame1.html">

  <FRAMESET COLS="200,*">
    <FRAME SRC="frame2.html">
    <FRAME SRC="frame3.html">
  </FRAMESET>
</FRAMESET>
```

خصائص إضافية للوسم <FRAMESET>

١. الخاصية FRAMEBORDER

وهي تقوم بتحديد ظهور أو عدم ظهور الحدود حول الإطارات وتأخذ القيم ١ للظهور، و ٠ لعدم الظهور. وهذا مثال عليها:

```
<FRAMESET ROWS="50,*,15%,20%" FRAMEBORDER="0">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
  <FRAME SRC="frame3.html">
  <FRAME SRC="frame4.html">
</FRAMESET>
```

٢. الخاصية BORDER

تحدد سمك الحدود الظاهرة حول الإطارات وهي تأخذ قيمةً بالعكس.

BORDER="n"

٣. الخاصية BORDERCOLOR

BORDERCOLOR="rrggbg"

لإضافة لون للحدود

٤. الخاصية FRAMESPACING



لتحديد مسافات فارغة إضافية حول الإطارات وتأخذ قيمةً بالبيكسل
FRAMESPACING="n"

خصائص الـ <FRAME>

١. الخاصية **MARGINHEIGHT**

تحدد مقدار المسافة الفارغة المتروكة للهوامش العلوية والسفلية للإطار (بالبيكسل).
MARGINHEIGHT="n"

٢. الخاصية **MARGINWIDTH**

تحدد مقدار المسافة الفارغة المتروكة للهوامش اليمنى واليسرى للإطار (بالبيكسل).
MARGINWIDTH="n"

٣. الخاصية **SCROLLING**

تحدد إمكانية ظهور أو عدم ظهور أشرطة التصفح الأفقية والعمودية على جوانب أو أسفل الإطار. وتأخذ القيم **yes** للظهور. **no** لعدم الظهور. و **auto** التي تحدد ظهور الأشرطة أو عدمه تلقائياً بحسب الحاجة إليها. تماماً كما يحدث في معظم تطبيقات **Windows**

SCROLLING="yes"
SCROLLING="no"
SCROLLING="auto"

٤. الخاصية **NORESIZE**

عند إضافة هذه الخاصية يتم منع عملية التحكم بحجم الإطار بالتصغير أو التكبير من خلال السحب والإفلات. وهي لا تأخذ أي قيم.

أمثلة على خصائص الـ <frame>

<FRAMESET COLS="50%,50%">
<FRAME SRC="Jellyfish.jpg" MARGINHEIGHT="40">
<FRAME SRC="frame2.html">
</FRAMESET>

<FRAMESET COLS="50%,50%">
<FRAME SRC="Jellyfish.jpg" MARGINHEIGHT="40" MARGINWIDTH="30">
<FRAME SRC="frame2.html">
</FRAMESET>



```
<FRAMESET COLS="50%,50%">
  <FRAME SRC="Jellyfish.jpg" MARGINHEIGHT="40" MARGINWIDTH="30"
    SCROLLING="yes">
  <FRAME SRC="frame2.html">
</FRAMESET>
```

```
<FRAMESET COLS="50%,50%">
  <FRAME SRC="Jellyfish.jpg" MARGINHEIGHT="40" MARGINWIDTH="30"
    SCROLLING="yes" NORESIZE>
  <FRAME SRC="frame2.html">
</FRAMESET>
```

الوسم <NOFRAMES> ... <NOFRAMES>

يستخدم هذا الوسوم لتوفير بديل معين عن صفحة الإطارات في حالة قيام أحد الزوار بدخول الموقع مستخدماً متصفحاً لا يدعم الإطارات مثل الإصدارات القديمة لـ **Netscape, MS Explorer** فإذا أردت أن تمنح زوار موقعك الذين لا يستخدمون متصفحاً حديثاً فرصة مشاهدة موقعك، فكل ما عليك فعله هو إدراج هذا الوسوم في نهاية الملف الرئيسي للإطارات والبدء بكتابة صفحتك كما لو كانت صفحة ويب عادية.

```
<HTML>
<HEAD>
<TITLE>Main File</TITLE>
</HEAD>
<FRAMESET ROWS="50,*,15%,20%" FRAMEBORDER="0">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
  <FRAME SRC="frame3.html">
  <FRAME SRC="frame4.html">
</FRAMESET>
<NOFRAMES>
<BODY>
```

أكتب صفحتك بالصورة الإعتيادية هنا

```
</BODY>
</NOFRAMES>
</HTML>
```



الخاصية name للوسم <frame>

هذه الخاصية تعتبر من أهم خصائص الوسم <FRAME> لأنها تحدد طريقة تنسيق العمل بين الإطارات والصفحات وأسلوب عرضها. قبل تطبيق هذه الخاصية نستعرض شفرات الملفات السابقة التي قمنا بإنشائها وهي كما يلي

شفرة الملف الرئيسي الذي أسميته masterfile.html

```
<html >
<head>
<title>Master File</title>
</head>
<frameset rows="40%,30%,30%" frameborder="1" border="5"
bordercolor="green" >
<frame src="frame1.html" >
<frameset cols="25%,25%" frameborder="1">
<frame src="frame2.html" SCROLLING="yes" >
<frame src="frame3.html">
</frameset>
<frameset cols="25%,25%" frameborder="1">
<frame src="frame4.html" >
<frame src="frame5.html">
</frameset>
</frameset>
</html>
```

شفرة الملف frame1.html

```
<html dir="rtl" >
<head>
<title>صفحة عرض الاطارات</title>
</head>
<body>
<p align="center">
<b>
<font color="blue" size="6">
يتم عرض الاطارات في هذه الصفحة
</font>
</b>
</p>
</body>
</html>
```

شفرة الملف frame2.html

```
<html dir="rtl">
<head>
<title>كلية المعلمين</title>
</head>
<body bgcolor="aqua">
<basefont size="5" >
```



```
<marquee behavior="scroll" direction="right" loop="5" bgcolor="sky blue">
مرحباً بكم في كلية المعلمين --- جامعة الملك سعود
</marquee>
<pre>
<font face="Traditional Arabic" size="5" >
تحتوي الكلية على الاقسام الاكاديمية التالية:
قسم الحاسب الالى
قسم الدراسات القرانية
قسم اللغة الانجليزية
قسم العلوم
قسم الرياضيات
</font>
</pre>
</body>
</html>
```

[شفرة الملف frame3.html](#)

```
<html dir="rtl">
<head>
<title>ادراج الصور</title>
</head>
<body bgcolor="aqua">
<basefont size="5" >
<b>
<font face="Andalus" size="6" >
<center>
ادراج الصور يتم باستخدام الوسم img
<br>

</center>
</font>
</b>
</body>
</html>
```

[شفرة الملف frame4.html](#)

```
<html dir="rtl">
<head>
<title>التعداد النقطي والرقمي</title>
</head>
<body >
<font face="Simple Indust Shaded" size="6" color="red">
<b>
<p align="center">
```



<u>

مكونات الحاسب

</u>

</p>

تنقسم الى خمسة اقسام وهي

وحدات الادخال ومن امثلتها

<ol type="a">

لوحة المفاتيح

الفأرة

القلم الضوئي

الماسح الضوئي

وحدات الاخراج ومن امثلتها

<ol type="a">

الطابعة

<ol type="circle">

طابعة الليزر

طابعة الرسوم الهندسية

السماعات

وحدة المعالجة المركزية ومن امثلتها

<ol type="a">

وحدة التحكم

وحدة الحساب والمنطق

وحدة الذاكرة الرئيسية ومن امثلتها

<ol type="a">

ذاكرة الوصول العشوائي



```

</li>ذاكرة القراءة فقط</li>
</ol>
</font>
<li>وحدات التخزين الثانوية ومن أمثلتها</li>
<font color="red">
<ol type="a">
<li>القرص الصلب</li>
<li>القرص المرن</li>
<li>القرص المدمج</li>
</ol>
</font>
</ol>
</font>
</b>
</body>
</html>

```

شيفرة الملف الذي يحتوي على الوصلات التشعبية والذي أسميته [frame5.html](#)

```

<html>
<body>
<table border="10" align="center" bgcolor="white" width=500
bordercolor="#3574EC">
<tr height=60 align="CENTER" valign="middle"
bgcolor=#56A5EC >
<td bgcolor="white" align="CENTER" valign="middle" colspan="3">
<font face="Segoe Print" color=black size=4><b>
الارتباطات التشعبية
</b></font>
</td>
<tr height=30 align="right" valign="middle"
bgcolor=#56A5EC >
<td>
<font face="Traditional Arabic"color="black" size="5">
<a href="frame2.html" >
كلية المعلمين
</a>
</font>
</td>
<tr height=30 align="right" valign="middle"
bgcolor=#56A5EC >
<td >
<font face="Traditional Arabic"color="black" size="5">
<a href="frame3.html" >
ادراج الصوت والفيديو
</a>

```




```

</font>
</td>
</tr>
<tr height=30 align="right" valign="middle"
bgcolor=#56A5EC >
<td >
<font face="Traditional Arabic"color="black" size="5">
<a href="frame4.html">
التعداد الرقمي التفصيلي
</a>
</font>
</td>
</tr>
</table>
</body>
</html>

```

- والآن تبقى الإطار الذي نريد إظهار الملفات فيه. وهو الصفحة المسماة **frame1.html**
- وهنا يأتي دور الخاصية **NAME** والتي نقوم من خلالها بتعيين اسم ما للإطار -أي الإطار الذي نرغب أن تظهر به الملفات عندما نقوم بالنقر على الوصلات التشعبية- حيث سيتم فيما بعد استخدام هذا الاسم من أجل استهداف هذا الإطار من قبل الوصلات التشعبية في الإطارات الأخرى أو حتى في الصفحات الأخرى.
- و الآن حان الوقت لذكر خاصية جديدة من خصائص الوسم **<A> ... ** (وهو وسم الوصلات التشعبية) و هي **TARGET**.
- الخاصية **NAME**؟ تستخدم مع الوسم الخاص بتعريف الإطار الذي ستظهر به الملفات. وبافتراض أننا نريد تسمية هذا الإطار بالاسم **mainwindow**. وهو اسم يحدده المستخدم حسب ما يريد .

[إذن تصبح شفرة الملف الأساسي masterfile.html هي:](#)

```

<html >
<head>
<title>Master File</title>
</head>
<frameset rows="40%,30%,30%" frameborder="1" border="5"
bordercolor="green" >
<frame src="frame1.html" name="mainwindow">
<frameset cols="25%,25%" frameborder="1">
<frame src="frame2.html" SCROLLING="yes" >

```



```
<frame src="frame3.html">
</frameset>
<frameset cols="25%,25%" frameborder="1">
<frame src="frame4.html" >

<frame src="frame5.html">
</frameset>
</frameset>
</html>
```

وبهذا نكون قد انتهينا من إعداد الملف الرئيسي ليكون ملف إطارات فعال. ونستطيع الآن استهداف الإطار من أي مكان باستخدام الاسم الذي عرفناه به.

المرحلة الاخيرة

هي تعريف الوصلات التشعبية بطريقة تجعل الملفات المتعلقة بهذه الوصلات تظهر في الإطار المحدد. وهنا يأتي دور الخاصية **TARGET** لكي نقوم بأداء هذه الوظيفة. وسوف نعمل الآن على الملف المسمى **frame5.html**. هو الملف الذي يحتوي على الوصلات التشعبية؟! ليصبح بالشكل التالي:

```
<html>
<body>
<table border="10" align="center" bgcolor="white" width=500
bordercolor="#3574EC">
<tr height=60 align="CENTER" valign="middle"
bgcolor=#56A5EC >
<td bgcolor="white" align="CENTER" valign="middle" colspan="3">
<font face="Segoe Print" color=black size=4><b>
```

الارتباطات التشعبية

```
</b></font>
</td>
</tr>
<tr height=30 align="right" valign="middle"
bgcolor=#56A5EC >
<td>
<font face="Traditional Arabic"color="black" size="5">
<a href="frame2.html" target="mainwindow">
```

كلية المعلمين

```
</a>
</font>
</td>
</tr>
<tr height=30 align="right" valign="middle"
bgcolor=#56A5EC >
<td >
<font face="Traditional Arabic"color="black" size="5">
<a href="frame3.html" target="mainwindow">
```

ادراج الصوت والفيديو



```

</a>
</font>
</td>
</tr>
<tr height=30 align="right" valign="middle"
bgcolor=#56A5EC >
<td >
<font face="Traditional Arabic" color="black" size="5">
<a href="frame4.html" target="mainwindow">
التعداد الرقمي التفصيلي
</a>
</font>
</td>
</tr>
</table>
</body>
</html>

```



الدرس العاشر

النماذج

FORMS

النماذج تعتبر من المواضيع المتقدمة في لغة **HTML** حيث أن معظم مواقع الويب تكاد لا تخلو من وجودها.

وذلك لعدة أسباب من أهمها

- إيجاد إمكانية للتفاعل بين الموقع وصاحبه من جهة والزوار من جهة أخرى
- أحيانا قد تحتاج كمصمم لموقع ويب أن تعرف آراء زوار موقعك في معرفة أشياء محددة
- إمكانية تنظيم البيانات المدخلة من خلالها وسهولة استخدامها من قبل زوار الموقع.

ومن أبرز الأمثلة على النماذج في مواقع الويب

- دفاتر الزوار .
- صفحات البحث عن الكلمات أو العبارات داخل المواقع.

تكمّن صعوبة النماذج في التداخل بينها وبين لغات برمجة الانترنت مثل JavaScript مما يجعلها تختلف عن سابقتها من الوسوم أو العناصر الأخرى. خاصة إذا احتجت إلى بعض المقاطع البرمجية من هذه اللغات ضمن نماذجك.	☑
--	---

ادراج النماذج في صفحات الويب يتم باستخدام الوسوم

<FORM> ... </FORM>

خصائص الوسم FORM

توجد ثلاثة خصائص لهذا الوسم هي :

٣. الخاصية ACTION

تحدد العنوان الذي سيتم إرسال بيانات النموذج إليه لتتم معالجتها بالصورة المطلوبة. وعادة يكون هذا عنواناً لبريد إلكتروني **Email** سوف يتم إرسال بيانات النموذج إليه. أو قد يكون عنواناً لبرنامج **CGI** موجود على الكمبيوتر الخادم **Server** الذي تتواجد عليه صفحة الويب .

<FORM ACTION="mailto:someone@domain.com">

</FORM>



```
<FORM ACTION="name_and_address_of_CGI_script">
```

```
</FORM>
```

برنامج CGI هو اختصار Common Gateway Interface وهو يخدم صفحات الأنترنت و يربط بينها و بين البرمجيات التي لا تفاهم بلغات الإنترنت.	<input checked="" type="checkbox"/>
---	-------------------------------------

٤. الخاصية METHOD

تحدد الطريقة التي سيتم بها التعامل مع العنوان المحدد في الخاصية السابقة **ACTION**. وهناك قيمتين لهذه الخاصية هما:

a. GET تستخدم عندما تكون المعالجة داخلية أي تتم داخل الخادم **Server** نفسه.

مثل نموذج البحث عن كلمة في الموقع.

b. Post تستخدم عندما تكون عملية المعالجة خارجية كأن يتم إرسال البيانات إلى عنوان بريد إلكتروني.

```
<FORM ACTION="mailto:someone@domain.com" METHOD="post">
```

```
</FORM>
```

```
<FORM ACTION="name_and_address_of_CGI_script" METHOD="get">
```

```
</FORM>
```

٥. الخاصية ENCTYPE

هذه الخاصية تحدد طريقة الترميز التي سيتم إرسال البيانات وفقاً لها. وهي تأخذ

القيمتين التاليتين:

a. application/x-www-form-urlencoded

b. text/plain

يجب أن تكتب هذه القيم كما هي نصاً وحرفاً .	<input checked="" type="checkbox"/>
الدافع لاستخدام أي من القيمتين هو طبيعة عملية المعالجة التي ستجرى على البيانات أو طبيعة برنامج البريد الإلكتروني الذي ستستقبل هذه البيانات من خلاله (إذا كان يدعم MIME أم لا، وهي اختصار للعبارة Multi-purpose Internet Mail Extensions وهي من المعايير السائدة في الإنترنت والتي تتعلق بنقل جميع أنواع البيانات من صوت وصورة وليس فقط النصوص من خلال البريد الإلكتروني).	<input checked="" type="checkbox"/>



أشكال البيانات

نستخدم الوسم `<INPUT>` لتعريفها والحقيقة أن هذه الأشكال هي مجرد خصائص أو بالأحرى قيم لخصائص تابعة لهذا الوسم....

خصائص الوسم `input`

١. الخاصية `type`

تستخدم لإدراج مجموعة من الأشكال حسب القيمة المستخدمة مع الخاصية والجدول التالي يوضح كل قيم هذه الخاصية .

الشكل	النتيجة
<code><input type="text"></code>	<input type="text" value="hozaifa"/>
<code><input type="password"></code>	<input type="password" value="....."/>
<code><input type="hidden"></code>	<input type="hidden"/>
<code><input type="radio"></code>	<input type="radio"/>
<code><input type="checkbox"></code>	<input checked="" type="checkbox"/>
<code><input type="submit"></code>	<input type="submit" value="إرسال استعلام"/>
<code><input type="reset"></code>	<input type="reset" value="إعادة تعيين"/>
<code><input type="button"></code>	<input type="button"/>

٢. الخاصية `NAME`

تستخدم لتسمية حقل البيانات بحيث يمكن استخدامه فيما بعد للحاجات البرمجية وضرورات المعالجة إن وجدت من قبل البرامج التي قد تضيفها كمصمم للموقع. وحتى عندما تريد أن يُرسل النموذج إليك بالبريد فإن حقوله تعرّف بالاسم الذي أدرجته لها من خلال هذه الخاصية.

`<FORM >`

Please enter your address:

`<INPUT TYPE="text" NAME="address">`
`</FORM>`



☑	<p>العبارة Please enter your address فهي مجرد عبارة توضيحية أضفتها ليعرف الزائر ما الذي يجب عليه كتابته وتستطيع صياغة هذه العبارة كما تريد. ففي كل الأحوال ليس لها علاقة بجوهر النموذج نفسه بعكس الخاصية NAME.</p>
---	--

٣. الخاصية **value**

تستخدم لإسناد قيمة افتراضية للحقل وذلك في الحالات التي نتوقع فيها كتابة قيمة متكررة من قبل معظم الزوار وذلك للتسهيل عليهم مع توفر إمكانية حذفها وكتابة ما نريده بدلاً منها.

```
<FORM ...>
  Please enter your address :
  <INPUT TYPE="text" NAME="address" VALUE="Riyadh, Saudi Arabia">
</FORM>
```

٤. الخاصية **size**

تستخدم لتحديد حجم الحقل

```
<FORM ...>
  Please enter your address :
  <INPUT TYPE="text" NAME="address" VALUE="Riyadh, Saudi Arabia"
    SIZE="40">
</FORM>
```

☑	<p>لا يوجد للخاصية size أي صفة تحكمية بالنسبة لحجم المدخلات التي يمكن للزائر أن يكتبها داخل الحقل. وبعبارة أخرى: صحيح أننا حددنا حجم الحقل لكن ذلك يسري فقط على مظهره على الشاشة ، ولا يوجد ما يمنع الزائر من الكتابة بحيث يتجاوز النص حجم الحقل المحدد .</p>
---	--

٥. الخاصية **MAXLENGTH**

تستخدم للتحكم بالحد الأقصى للنص المدخل.

```
<FORM ...>
  Please enter your address :
  <INPUT TYPE="text" NAME="address" VALUE="Riyadh, Saudi Arabia"
    SIZE="40" MAXLENGTH="30">
</FORM>
```

الحقل **password**



يشبه الحقل **text** من حيث الخصائص تماماً غير أن مدخلاته تظهر على شكل ***** مهما كانت، وهو الفرق الوحيد بينهما. ويستخدم عندما يوجد حاجة لإدخال كلمة سر من قبل الزائر في النموذج

```
<FORM ...>
Please enter your name :
<INPUT TYPE="text" NAME="the name" VALUE="" SIZE="40" MAXLENGTH="30">
Please enter your password :
<INPUT TYPE="password" NAME="the password" VALUE="" SIZE="40"
MAXLENGTH="30">
</FORM>
```

الحقل hidden

وهو الحقل المخفي. وكما نستنتج من اسمه فهو لن يظهر ضمن النموذج.

والمثال التالي يوضح الحقل المخفي

```
<FORM ...>
Please enter your name :
<INPUT TYPE="text" NAME="the name" VALUE="" SIZE="40" MAXLENGTH="30">

<INPUT TYPE="hidden" NAME="my forms" VALUE="form1">

Please enter your password :
<INPUT TYPE="password" NAME="the password" VALUE="" SIZE="40"
MAXLENGTH="30">
</FORM>
```

هذا الحقل مثل عدمه بالنسبة لمظهر النموذج، وأن الزائر لن يتعامل معه.

والسؤال هنا: ما الفائدة من وجود حقل مخفي؟

ولكي أجيب على هذا السؤال دعني أطرح لك مثلاً أو حالة قد تواجهكم مصمم صفحات

ويب...

لنفرض أن لديك ثلاث صفحات تتضمن كل منها نموذجاً ما وأن هذه النماذج متشابهة. وتحتوي على نفس الحقول. وعندما تصلك البيانات كيف ستستطيع تمييز أي من هذه النماذج استخدم لإرسال البيانات؟

بإمكانك إضافة هذا الحقل (الوهمي) وإسناد أي اسم وأي قيمة له في كل نموذج.

في النموذج الأول ...

```
<INPUT TYPE="hidden" NAME="my forms" VALUE="form1">
```

في النموذج الثاني ...

```
<INPUT TYPE="hidden" NAME="my forms" VALUE="form2">
```




في النموذج الثالث ...

`<INPUT TYPE="hidden" NAME="my forms" VALUE=" form3">`

وبذلك عندما تصلك البيانات المرسله من قبل أي زائر استخدم أي من النماذج الثلاثة سيصلك أيضا حقل إضافي قمت أنت بتعبئته سلفاً عندما صممت النموذج وذلك بأحد الأشكال التالية:
`my forms=form1` أو `my forms=form2` أو `my forms=form3`

من أجل إظهار النموذج بصورة مرتبة ومنسقة والتحكم بموقع الحقول فيه فمن الأفضل دائماً وضعه داخل جدول مع جعل الجدول بلا حدود.	<input checked="" type="checkbox"/>
---	-------------------------------------

```
<FORM ...>
<TABLE BORDER="0">
<TR>
<TD>Please enter your name : </TD>
<TD>
<INPUT TYPE="text" NAME="the name" VALUE="" SIZE="40" MAXLENGTH="30">
</TD></TR>
<TR>
<TD>Please enter your password : </TD>
<TD>
<INPUT TYPE="password" NAME="the password" VALUE="" SIZE="40"
MAXLENGTH="30">
</TD></TR>
</TABLE>
</FORM>
```

الحقول الخاصة بالاختيار من متعدد

وهي ثلاثة انواع

النوع الاول زر الخيار Radio

يستخدم في حالة الاختيار من متعدد (خيار واحد فقط)

مثال :-

لنفرض اننا نريد أن نسأل الزائر عن اصدار المتصفح الذي يستخدمه من مجموعة الخيارات التالية :-

Internet Explorer 5.0
 Internet Explorer 6.0
 Internet Explorer 7.0
 Internet Explorer 8.0

والسؤال الان كيف ننشئ هذه القائمة يتم ذلك كما يلي

الخطوة الاولى :-

بما أن هناك أربعة مدخلات، إذن نحتاج إلى أربعة وسوم



```
<FORM>
<INPUT TYPE="radio"> <BR>
<INPUT TYPE="radio"> <BR>
<INPUT TYPE="radio"> <BR>
<INPUT TYPE="radio"> <BR>
</FORM>
```

الخطوة الثانية : -

نحتاج إلى تسمية هذه المدخلات، أي أننا سنستخدم الخاصية **NAME** معها. أما الاسم المعطى فمن الأفضل أن يكون مرتبطاً نوعاً ما بموضوع السؤال لأنه ضروري للشخص الذي يقوم باستقبال البيانات المرسله من خلال النموذج، وبالتالي من الأفضل أن يوجد عنوان معبر للبيانات بغرض التمييز. وبما أننا هنا نتحدث عن المتصفحات فليكن هذا الاسم هو **browser**

```
<FORM>
<INPUT TYPE="radio" NAME="browser"> <BR>
<INPUT TYPE="radio" NAME="browser"> <BR>
<INPUT TYPE="radio" NAME="browser"> <BR>
<INPUT TYPE="radio" NAME="browser"> <BR>
</FORM>
```

الخطوة الثالثة : -

تعريف كل حقل باسم صريح يوضح محتواه.

```
<FORM>
<INPUT TYPE="radio" NAME="browser" > Internet Explorer 5.0 <BR>
<INPUT TYPE="radio" NAME="browser" > Internet Explorer 6.0 <BR>
<INPUT TYPE="radio" NAME="browser" > Internet Explorer 7.0 <BR>
<INPUT TYPE="radio" NAME="browser" > Internet Explorer 8.0 <BR>
</FORM>
```

النوع الثاني مربع الاختيار CHECKBOX

يستخدم لاختيار أكثر من حقل في نفس الوقت! وهذا هو الفرق الأول بين **CHECKBOX** و **RADIO** ففي **RADIO** يمكن اختيار حقل واحد فقط ليس أكثر.

خطوات انشاء مربع الاختيار

١. نحدد القيمة **checkbox** للخاصية **TYPE** [TYPE="checkbox"]
٢. نعطي كل حقل في القائمة اسماً مميزاً في الخاصية **NAME** يختلف عن باقي الحقول.
٣. كتابة الأسماء التعريفية لكل حقل.



مثال :-

انشئ مربعات الاختيار التالية

من مميزات الحاسب

☐ السرعة

☐ الغباء

☐ عدم المرونة

☐ الدقة

<FORM>

<INPUT TYPE="checkbox" NAME="speed" >

السرعة

<INPUT TYPE="checkbox" NAME="thickness " >

الغباء

<INPUT TYPE="checkbox" NAME="non flexibility " >

عدم المرونة

<INPUT TYPE="checkbox" NAME="accuracy " >

الدقة

</FORM>

اوجه الاختلاف بين زر الخيار **Radio** ومربع الاختيار **CheckBox**

١. في **RADIO** نستطيع اختيار حقل واحد فقط أما في **CHECKBOX** فنختار أكثر من حقل.

٢. في **RADIO** تكون أسماء الحقول موحدة والقيم مختلفة، أما في **CHECKBOX** فتكون الأسماء مختلفة والقيم موحدة.



النوع الثالث قوائم الاختيار

هذا النوع يقودنا إلى وسوم جديدة من وسوم التعريف والتي تستخدم بدلاً من <INPUT> وهي

<SELECT>

<OPTION>

<OPTION>

<OPTION>



.....

 </SELECT>

حيث أن <SELECT> ... <SELECT/> تحددان بداية ونهاية القائمة، والوسم <OPTION> الذي يوضع دائماً بينهما يستخدم لتحديد كل عنصر من عناصر القائمة. هناك خصائص مشتركة عرفناها في الأشكال السابقة سيتم استخدامها هنا أيضاً كما يوجد خصائص جديدة تتعلق فقط بهذا الشكل من حقول البيانات. فبالنسبة لـ <SELECT> يوجد الخاصية **NAME** وهي تحدد اسم القائمة. كما توجد الخاصية **SIZE** التي تحدد ارتفاع القائمة، وبالتالي عدد البيانات الظاهرة فيها. وهي تأخذ أي قيمة عددية صحيحة.

مثال :-

انشئ قائمة الاختيار التالية

Department:

- Computer
- Mathematics
- Science
- English

```
<FORM>
Department:
<SELECT NAME="browser" SIZE="1">
  <OPTION> Computer
  <OPTION> Mathematics
  <OPTION> Science
  <OPTION> English
</SELECT>
</FORM>
```

ايضاً هناك إمكانية لجعل اختيار البيانات من هذه القائمة متعدداً وليس فقط قيمة واحدة بإضافة الخاصية **MULTIPLE**. لنقم الآن بعرض جميع القيم (لدينا أربعة قيم، إذن القيمة المكتوبة مع **SIZE** يجب أن تكون ٤)، ومن ثم لنتيح المجال أمام الزائر لاختيار أكثر من قيمة واحدة في القائمة.

```
<FORM>
Department:
<br>
<SELECT NAME="browser" SIZE="4" MULTIPLE >
  <OPTION> Computer
  <OPTION> Mathematics
  <OPTION> Science
  <OPTION> English
```



</SELECT>
</FORM>

الحقل TEXTAREA (صندوق الرسائل)

وهو المستخدم عادة لكتابة التعليقات الحرة في النموذج ويتم إدراجه بكتابة الوسوم

<TEXTAREA> ... </TEXTAREA>

ومن خصائصه

١. الخاصية NAME

تستخدم لإعطاء صندوق الرسائل اسم معين .

أي نص يكتب بين الوسمين <TEXTAREA> ... </TEXTAREA> يتم عرضه داخل الحقل بصورة تلقائية .	<input checked="" type="checkbox"/>
--	-------------------------------------

<TEXTAREA NAME="comments">
Hello, please write your comments here :-)
</TEXTAREA>

٢. خاصيتي العرض والارتفاع

a. خاصية العرض COLS تستخدم لتحديد العرض

b. خاصية الارتفاع ROWS تستخدم لتحديد الارتفاع

<TEXTAREA NAME="comments" COLS="30" ROWS="4">

Hello, please write your comments here

</ TEXTAREA >



٣. الخاصية WRAP

تحدد طريقة التفاف النص المكتوب داخل الحقل وهناك ثلاثة قيم تأخذها وهي على النحو التالي:

a. virtual : التي تعني أن النص سيلتف على عدة أسطر عند كتابته ولكنه سيصلك عند إرساله على شكل سطر واحد متتابع (حاول الكتابة داخل الحقول وأنظر كيفية تأثير هذه الخاصية على كل منها)

```
<TEXTAREA NAME="comments" COLS="30" ROWS="6" WRAP="virtual">
</TEXTAREA>
```

b. physical : تعني أن النص سيلتف على عدة أسطر وسيصلك أيضاً على هذا النحو عند إرساله

```
<TEXTAREA NAME="comments" COLS="30" ROWS="6" WRAP="physical">
</TEXTAREA>
```

c. off : تعني أن النص لن يلتف بصورة تلقائية على عدة أسطر لكنه على أية حال سيصلك بنفس الشكل الذي تم إدخاله به

```
<TEXTAREA NAME="comments" COLS="30" ROWS="6" WRAP="off">
</TEXTAREA>
```

الحقل Submit

يستخدم لإرسال البيانات التي تم ملؤها في النموذج.

```
<INPUT TYPE="submit" VALUE="Press here to send the form">
```

الحقل Reset

يستخدم لمسح ما كتب في النموذج وإلغاء الأمر بنفس طريقة التعريف والخصائص المستخدمة مع **submit**.

```
<INPUT TYPE="reset" VALUE="Forget about it">
```

الحقل button

يقوم بإنشاء زر ضمن النموذج، وهو مرتبط بالنماذج التي تحتوي على نصوص برمجية (أو برامج مكملة) من لغات متقدمة مثل **JavaScript** كونه يستخدم لتشغيل هذه البرامج وإطلاقها. وطبعاً هناك طرق معينة لربطها مع البرامج. لكن مبدئياً أقول إن طريقة الإدراج والتعريف هي ذاتها المستخدمة مع **reset, submit**.

```
<INPUT TYPE="button" VALUE="This is a sample button">
```



الدرس الحادي عشر

أوراق الأنماط المتعاقبة CSS Cascading Style Sheets

الهدف من استخدام CSS

الهدف الرئيسي لأوراق الانماط المتعاقبة هو جمع كل ماله علاقة بالتصميم وفصله تماماً عن المحتويات.

ولذلك فوائد عديدة منها:

- اختصار في كتابة الشفرة.
- مزيداً من التحكم بالتأثيرات الخاصة بالنصوص.
- سهولة التحديث وتغيير مظهر الصفحة.
- حجم ملفات أقل.

الصيغة العامة لـ CSS



- **القاعدة** : تخبر المستعرض بكيفية رسم عنصر معين على الصفحة.

- **المحدد** : يحدد العناصر التي ستتأثر بالقاعدة.

- **التصريح** : تخبر المستعرض بكيفية رسم العناصر المحددة.

عادة ما يكتب المحدد بحروف كبيرة والتصريح بحروف صغيرة .	<input checked="" type="checkbox"/>
--	-------------------------------------

مثال :

P { color : blue }

القيمة الخاصية



يمكن الجمع بين أكثر من محدد وذلك بالفصل بينهما بفاصلة:

SELECTOR1, SELECTOR2 { property : value }

مثال :

H1,P { color : blue }

يمكن الجمع بين أكثر من تصريح وذلك بالفصل بينهما بفاصلة منقوطة:

SELECTOR { property1 : value1 ; property2 : value2 }

مثال:

B { color : green ; text-align: center }

طرق اضافة CSS

١. الأنماط المباشرة **Inline CSS**
٢. الأنماط الداخلية **Internal CSS**
٣. الأنماط الخارجية **External CSS**

الأنماط المباشرة **Inline CSS**

يتم ادراج قواعد اوراق الأنماط قبل النص المراد تنسيقه مباشرة من خلال خاصية **Style**.

مثال:

```
<html>
<body>
<p style="background:blue; color: white; font-size:30 ;font-family:Impact" >
King Saud University
</p>
<b style="background: yellow; color: green">
Teachers College
</b>
<br>
<h1 style=" background-image:url(Desert.jpg); color:green">
Computer Department
</h1>
</body>
</html>
```

الأنماط الداخلية **Internal CSS**

توضع جميع أوراق الأنماط بين وسمي **<style>** و **</style>** في جزء الرأس من الصفحة.



مثال:

```
<html dir="rtl">
<head>
<style type="text/css">
  B { background :yellow; color :green}
  A:link { color :red ; text-decoration :none }
  UL { margin-right :75px ; color :#ff0000 }
  UL UL { margin-right :15px ; color :#0000ff }
</style>
</head>
<body>
<b>بسم الله الرحمن الرحيم</b>
<br> <a href="http\www.ksu.edu.sa">
  جامعة الملك سعود
</a> <br>
مقررات قسم الحاسب الالى
<ul>
<li>مقدمة في الحاسب</li>
<li>مقدمة الى الخوارزميات</li>
<li>برمجة الانترنت</li>
</ul>
<li>مبادئ</li>
<li>نشر الكتروني</li>
</ul>
<li>برمجة قواعد البيانات</li>
<li>شبكات الحاسب</li>
</ul>
</body>
</html>
```



الأنماط الخارجية External CSS

عندما نريد تطبيق نفس قواعد CSS على مجموعة من الصفحات نضع قواعد CSS في صفحة مستقلة وكل صفحة تربط بها عن طريق شفرة تكتب في جزء الرأس.

```
<head>
<link rel="stylesheet" type="text/css"
href="style.css">
</head>
```

حيث : **style.css** هي صفحة قواعد CSS

مثال :

انشي مجلد باسم الانماط الخارجية وضع بداخله الملفات التالية

الملف الاول يحتوي على قواعد **css** ويحفظ بالامتداد **css** مثلاً **style.css**

```
em { font-size : 25 ; font-weight : bold }
a:link { color: yellow ; text-decoration: none }
ol { margin-right: 75px ; color : green }
p { background: blue; color: white }
b { color: aqua; text-decoration: underline }
marquee { background: red; color: yellow }
body { background-image: url(Desert.jpg); background-position: center right;
background-repeat: no-repeat }
```

الملف الثاني

```
<html dir="rtl">
<head>
<link rel="stylesheet" type="text/css"
href="style.css">
</head>
<body>
<marquee>بسم الله الرحمن الرحيم</marquee>
<br>
<b>محتويات الصفحة</b>
<br>
<a href="http://www.ksu.edu.sa">
جامعة الملك سعود
</a>
<br>
<p>كلية المعلمين</p>
```



```
</body>
```

```
</html>
```

وأحفظه باسم **page1.html**

[الملف الثالث](#)

```
<html dir="rtl">
```

```
<head>
```

```
<link rel="stylesheet" type="text/css"
```

```
href="style.css">
```

```
</head>
```

```
<body>
```

```
<em>بسم الله الرحمن الرحيم</em>
```

```
<br>
```

```
بعض الجامعات السعودية
```

```
<ol>
```

```
<li>جامعة الملك سعود</li>
```

```
<li>جامعة الملك خالد</li>
```

```
<li>جامعة الملك فهد</li>
```

```
</body>
```

```
</html>
```

وأحفظه باسم **page2.html** ثم انسخ الصورة **Desert.jpg** من مجلد الصور وضعها داخل

مجلد الانماط الخارجية

تطبيق CSS مع الفئة Class

تستخدم الفئة **class** عندما نريد تطبيق قواعد مختلفة على عناصر من نفس النوع .

فمثلا : يمكن وضع عدة فئات لعنصر واحد كعنصر **H1**

كما يلي :

```
<html dir="rtl">
```

```
<head>
```

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

```
H1.one {color : green}
```

```
H1.two {color : red}
```

```
</style>
```



```
</head>
<body>
<h1> النص الأول </h1>
<h1 class="one"> النص الثاني </h1>
<h1 class="two"> النص الثالث </h1>
</body>
</html>
```

الفئات العامة

من المزايا في CSS إمكانية وضع فئة عامة تستخدم مع أي عنصر من عناصر **html**، ويكون ذلك بوضع اسم الفئة بعد نقطة فقط دون كتابة اسم العنصر أو المحدد، كالمثال التالي:

```
<html dir="rtl">
<head>
<style type="text/css">
.bb {color : red}
</style>
</head>
<body>
<b class="bb"> النص الاول </b>
<br>
<u class="bb"> النص الثاني </u>
</body>
</html>
```

توريث الخصائص

عند تداخل عنصرين فإن الداخلي سيرث خصائص العنصر الخارجي إذا لم تتنافى مع خصائصه.

مثال:

```
<html>
<head>
<style type="text/css">
i {color : red}
b {background: yellow}
</style>
</head>
<body>
<i> Cascading <b> Style </b> Sheets </i>
</body>
</html>
```



الدرس الثاني عشر

مدخل الى لغة JavaScript

الفرق بين لغة Java و لغة JavaScript

- لغة Java هي لغة برمجة صممت بواسطة شركة (Sun Microsystems) أما لغة JavaScript فقد صممت بواسطة الأشخاص المنتجين لمتصفح (Netscape) .
- البرنامج الذي يتم تصميمه باستخدام لغة Java هو برنامج قائم بذاته أما برامج لغة JavaScript غير قائمة بذاتها حيث أنها لا تعمل إلا من داخل صفحة الويب .
- صفحة الويب التي تحتوي علي لغة JavaScript لا بد أن يتم تصفحها من خلال متصفح يفهم لغة JavaScript
- لغة JavaScript تستخدم أسلوب البرمجة بالكائنات (Object Oriented Programming) الذي يسمح بالتعامل مع البيانات علي شكل كائنات (Objects) مثل document و window ... ولكل كائن مناهج (Methods) وخصائص (Properties) خاصة به .

كتابة برامج لغة JavaScript

- لغة JavaScript تكتب داخل مستند لغة HTML إما في الرأس (head) أو في الجسم (body) حيث يتم حفظها كنص مدمج داخل مستند لغة HTML .

<p>الفارق الجوهرى والرئيسي بين اللغة HTML و لغة JavaScript هو أن المسافات الفارغة بين الكلمات أو السطور لا تعني شيئاً بالنسبة للغة HTML أما بالنسبة للغة JavaScript فهي مهمة جداً ويجب أن تؤخذ في الاعتبار وإلا سوف تحدث أخطاء في البرنامج كما سنلاحظ فيما بعد .</p>	<input checked="" type="checkbox"/>
--	-------------------------------------

طرق كتابة كود JavaScript

١. الطريقة الأولى :

```
<script type = " text/javascript">
Write Code Here
</script>
```

٢. الطريقة الثانية :

```
<script language="JavaScript">
Write Code Here
</script>
```



اوامر لغة JavaScript

١. أمر الطباعة :

- الطريقة الأولى : طباعة جملة لا يقع عليها تأثير اي وسم من وسوم (html)
`document.write("هنا الجملة المراد طباعتها");`

- الطريقة الثانية : طباعة جملة يقع عليها تأثير وسم
ال (html)

`document.write("<h1>هنا الجملة المراد طباعتها </h1>");`

- الطريقة الثالثة : طباعة جملة يقع عليها تأثير وسمي html و style
`document.write("<h1 style='color : red'>هنا الجملة المراد طباعتها</h1>");`

- الطريقة الرابعة : طباعة قيمة متغير ...

`document.write(sum);`

- الطريقة الخامسة : طباعة قيمة متغير يسبقه جملة نصية ..

`document.write (" +sum " الناتج هو :);`

- الطريقة السادسة : طباعة قيمة متغير يسبقه ويعقبه جملة نصية ..

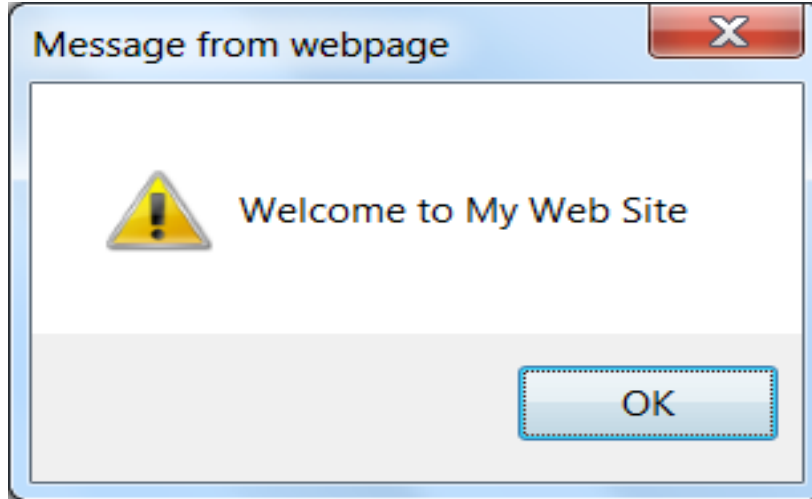
`document.write(" : الناتج هو " + sum + " العملية ");`

- | | |
|---|--|
| <ul style="list-style-type: none"> نستطيع ان نستخدم (<code>document.writeln</code>) بدلاً من (<code>document.write</code>) حيث تترك مسافة بين كل كلمة والآخرى في جملة الطباعة. طباعة جملة في سطرين اي " النزول الى سطر جديد " نضع بداخل جملة الطباعة (<code>
</code>) قبل الجملة التي تريد ان تجعلها في سطر جديد .
مثلاً : | |
|---|--|

`document.write(" Computer
Department ");`

٢. خروج نافذة للمستخدم نحدد نحن ما يكتب بها وهي خاصة في كائنات النوافذ Window

Object مثل النافذة التالية



وتسمى هذه النافذة (*alert*) ويكتب كودها بهذه الطريقة
`window.alert("Welcome to My Web Site");`

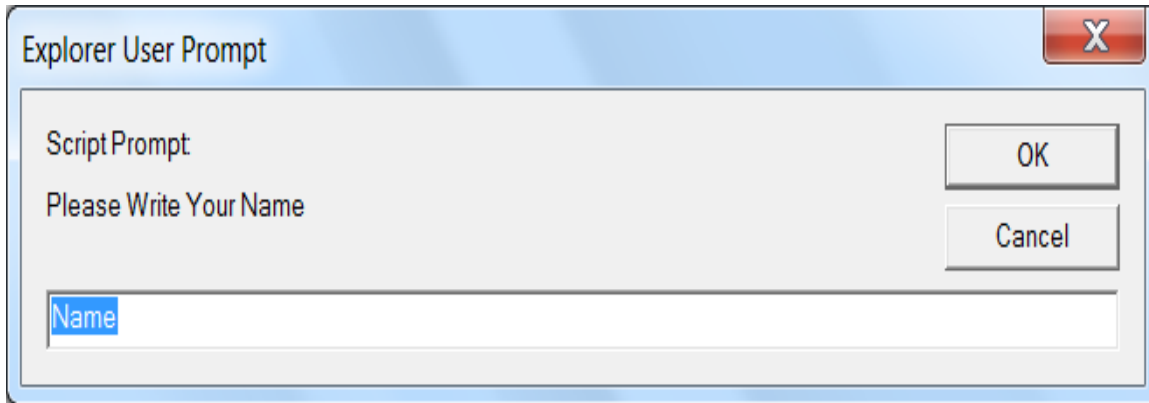
بعض الاوامر التي تستخدم داخل كود الـ (*alert*)

الامر	الوصف	مثال
<code>\n</code>	سطر جديد	<code>window.alert("hi \n all");</code>
<code>\t</code>	لتترك مسافة <i>tab</i>	<code>window.alert("hi \t all");</code>
<code>\r</code>	لوضع كل كلمة بسطر ولكن بإختلاف الموقع	<code>window.alert("hi \r all");</code>
<code>\\</code>	لكتابة رمز \	<code>window.alert("hi \\ all");</code>
<code>\"</code>	لكتابة رمز "	<code>window.alert("hi \" all");</code>
<code>'</code>	لكتابة رمز '	<code>window.alert("hi \' all");</code>



٣. خروج نافذة للمستخدم يحدد هو ما يكتب بها " اي نافذة قراءة من لوحة المفاتيح " وهي

خاصة بكائنات النوافذ **Window Object**



وتسمى هذه النافذة (**prompt**) ويكتب كودها بهذه الطريقة

```
window.prompt("Please Write Your Name", "Name");
```

المتغيرات في الـ **JavaScript**

المتغيرات في أي لغة برمجة تعتبر ذات فائدة كبيرة فبواسطتها نستطيع التحكم في البرنامج .

حيث تعرف في **JavaScript** بالكلمة المفتاحية **var**

مثال :

```
var name ;
```

أذا اردنا أن يكون المتغير قيمة رقمية (integer Or float) "أعداد صحيحة أو أعداد ذات أرقام عشرية" فإننا نقوم بتحويلها .	<input checked="" type="checkbox"/>
---	-------------------------------------

■ التحويل للأعداد الصحيحة

;(أسم المتغير هنا) **parseInt**

مثال

```
var number = 55 ;
parseInt (number) ;
```

■ التحويل للأعداد ذات الارقام العشرية

;(أسم المتغير هنا) **parseFloat**

مثال

```
var number = 5.4 ;
parseFloat (number) ;
```

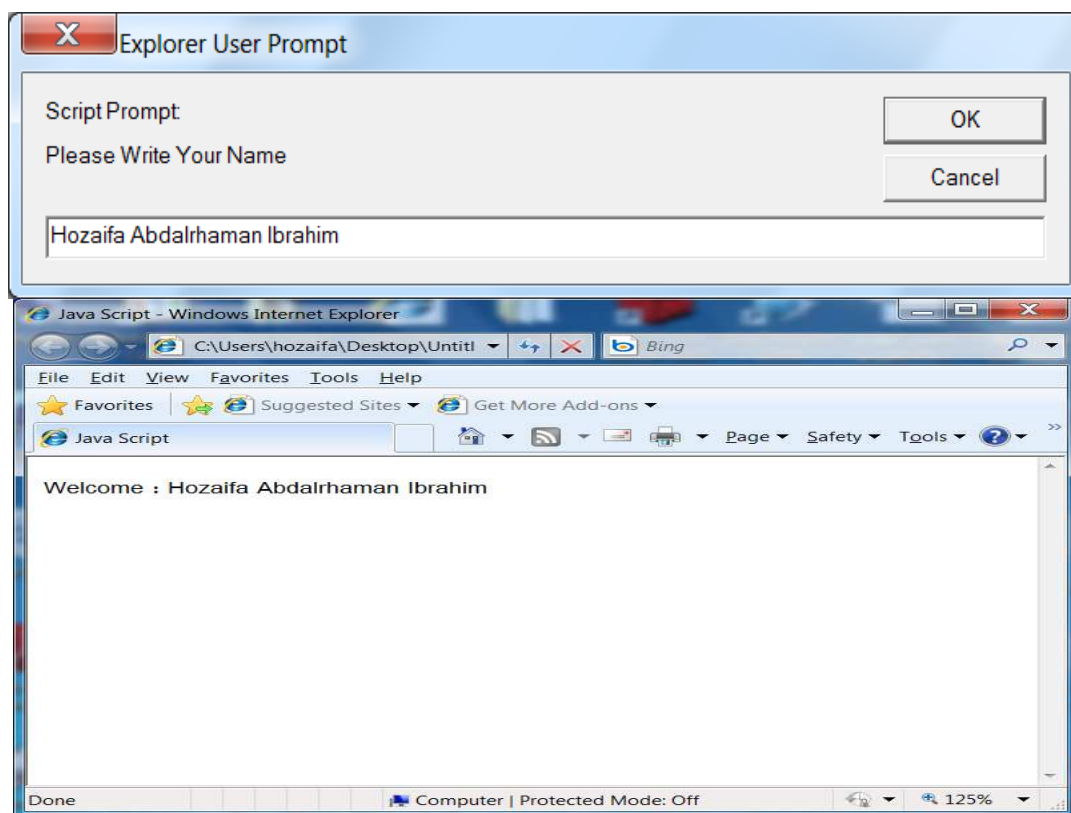
يجب الالتزام بحالة الاحرف في الكلمات المفتاحية .	<input checked="" type="checkbox"/>
--	-------------------------------------



أمثلة

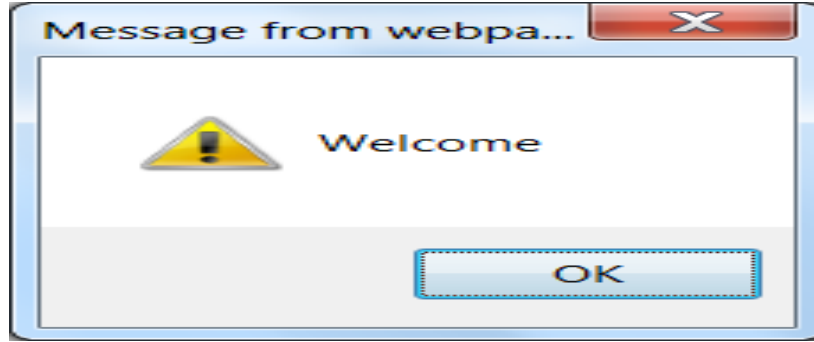
المثال الاول

```
<html>
<head><title> Java Script</title>
<script>
var name ;
name = window.prompt ( "Please Write Your Name"," Your Name");
document.write( " Welcome : " + name ) ;
</script>
</head>
<body> </body>
</html>
```



المثال الثاني

```
<html>
<head><title> Java Script</title>
<script type = "text/javascript" >
window.alert( "Welcome");
</script>
</head>
<body></body>
</html>
```



أنواع المؤثرات في لغة JavaScript

١. المؤثرات الحسابية *Arithmetic Operators* :

العملية الحسابية	في الجبر	في JavaScript	مثال
الجمع	$a+b$	$a+b$	$c=a+b$
الطرح	$a-b$	$a-b$	$c=a-b$
القسمة	$\frac{a}{b}$	a/b	$c=a/b$
الضرب	$a \times b$	$a*b$	$c=a*b$
باقي القسمة		$a\%b$	$c=a\%b$

٢. مؤثرات المقارنة *Relational Operators*

ويكون ناتجها إما صحيحاً *True* أو خاطئاً *False*

معنى المؤثر	مؤثر المقارنة	مثال
$x == y$	يساوي	$==$
$x != y$	لا يساوي	$!=$
$x > y$	أكبر من	$>$



$x < y$	أصغر من	$<$
$x \geq y$	أكبر من أو يساوي	\geq
$x \leq y$	أصغر من أو يساوي	\leq

٣. المؤثرات المنطقية *logical operators*

وهي تستخدم لربط أكثر من تعبير منطقي بسيط مع بعضها البعض و هي :

الاسم	الاسم بالعربي	رمز العملية
<i>and</i>	و	$\& \&$
<i>or</i>	أو	$\ $
<i>not</i>	النفى	$!$

٤. مؤثرات التعيين المركبة *compound assignation operators*

$(+=, -=, *=, /=, \%=)$

يتوفر في لغة *JavaScript* عدداً من عمليات الإسناد. فبالإضافة لعملية الإسناد العادية التي تستخدم فيها الإشارة = يوجد طرق مختصرة أخرى كما في الأمثلة التالية:

- عملية الإسناد $b=b+c$ تكافئ عملية الإسناد المختصرة $b+=c$
- عملية الإسناد $b=b-c$ تكافئ عملية الإسناد المختصرة $b-=c$
- عملية الإسناد $b=b*c$ تكافئ عملية الإسناد المختصرة $b*=c$
- عملية الإسناد $b=b/c$ تكافئ عملية الإسناد المختصرة $b/=c$
- عملية الإسناد $b=b\%c$ تكافئ عملية الإسناد المختصرة $b\%=c$

٥. مؤثرات التزايد و التناقص *Increase and decrease operators*

مؤثر التزايد هو ++ ويعني زيادة قيمة المتغير بمقدار واحد
ومؤثر التناقص هو -- ويعني إنقاص قيمة المتغير بمقدار واحد



مثال :- العبارة JS++

تعني زيادة قيمة المتغير JS بمقدار واحد وهي مكافئة للتعبير التالية

JS +=1

JS = JS +1

وينطبق هذا أيضا في حالة النقصان (--)

الدرس الثالث عشر

جمل التحكم في لغة JavaScript

- هي عبارة عن جمل تتحكم في تنفيذ خطوات البرنامج وتنقسم الى نوعين جمل شرطية وجمل تكرارية .

- الجمل الشرطية تتحكم في تنفيذ خطوات البرنامج من خلال شرط او مجموعة من الشروط .

- الجمل التكرارية تقوم بتكرار تنفيذ تعليمة او مجموعة من التعليمات اكثر من مرة .

الجمل الشرطية Conditional Statements

الجملة الشرطية هي مقارنة بين قيمتين قد يتطابقوا وعندها يكون الجواب صحيح (True) أو قد لا يتطابقوا وعندها يكون الجواب خاطئ (False) .

انواع الجمل الشرطية

٣. جملة if

■ العبارة الشرطية البسيطة if statement

تستخدم العبارة البسيطة في اختبار شرط إذا تحقق الشرط يتم تنفيذ تعليمة أو مجموعة من التعليمات

والشكل التالي يوضح الصيغة العامة للعبارة الشرطية البسيطة في حالة تنفيذ تعليمة واحدة .

صيغتها:

```
if (condition)
    statement;
```

مثال :-

```
<html dir="rtl">
<head>
<script type="text/javascript">
var pass_2="omar";
var user_pass;
user_pass = window.prompt("ادخل كلمة السر","كلمة السر");
```



```
if (pass_2 == user_pass)
document.write (" أهلا بك ");
</script> </head>
<body></body>
</html>
```

■ العبارة الشرطية الكاملة if else statement

تستخدم العبارة الشرطية الكاملة في اختبار شرط إذا تحقق الشرط فإن جملة أو مجموعة من الجمل تنفذ وإذالم يتحقق الشرط فإن جملة أو مجموعة من الجمل تنفذ أيضاً

صيغتها:

```
if(condition)
statement1;
else
statement2;
```

مثال :-

```
<html dir="rtl">
<head>
<script type = "text/javascript">
var pass_2="omar";
var user_pass;

user_pass = window.prompt("ادخل كلمة السر","كلمة السر");
if (pass_2 == user_pass)
document.write (" أهلا بك ");
else
document.write (" كلمة السر غير صحيحة ");
</script> </head>
<body></body>
</html>
```

■ العبارة الشرطية المتدرجة (متعددة الشروط)

لو افترضنا انه قد طلب منك - كمبرمج - عمل برنامج يمكنه احتساب التقديرات اعتمادا على مجموع

الطالب، في هذه الحالة نستخدم عبارة شرطية أيضا ولكن بها عدد من الشروط و عدد مناظر من النتائج .



أو ما يطلق عليه العبارة الشرطية المتدرجة أو متعددة الشروط.
والشكل التالي يوضح الصيغة العامة للعبارة الشرطية المتدرجة

صيغتها:

```
if (condition1)
    statement1;
else if(condition2)
    statement2;
else if(condition3)
    statement3;
:
else if(condition n)
    statement n;
else
    statement;
```

مثال :-

```
<html dir="rtl">
<head>
<script type = "text/javascript">
    var degree;

    degree = window.prompt("ادخل الدرجة","الدرجة");
if (degree>100 || degree<0 )
    document.write(" الدرجة المدخلة غير صحيحة ");
else
{
    if (degree>=90)
        document.write(" ممتاز ");
    else if (degree>=80)
        document.write(" جيد جداً ");
    else if (degree>=70)
        document.write(" جيد ");
    else if (degree>=60)
        document.write(" مقبول ");
    else
        document.write("ضعيف");
}
</script> </head>
<body></body>
</html>
```



٤. جملة Switch

تستخدم هذه الجملة الشرط المغلق **close condition** أي الشرط الذي يأخذ قيمة محددة (رقمية أو حرفية) ولهذا يمكن القول بأن هذه الجملة تتعامل مع متغير واحد يمكن أن يأخذ في كل مرة قيمة محددة من خلال مجموعة من القيم .

الصيغة العامة للجملة :

```
switch(variable)
{
case value 1: statement; break;
case value 2: statement; break;
:
:
case value n: statement; break;
default :statement;
}
```

هذه الصيغة تمثل الاحتمالات التي يمكن أن يأخذها المتغير **variable** من **[value1----- value n]** كل احتمال يتم وضعه داخل جملة باستخدام العبارة **case** ثم بعد ذلك كتابة الحدث الذي يمكن أن يكون تعليمة أو مجموعة تعليمات ولا بد أن ينتهي الحدث بواسطة العبارة **break** التي تمثل نهاية الاحتمال .

أما الاحتمال **default** يمثل الحدث الملازم للحالة خارج النطاق من **[value1-----value n]**

```
<html dir="rtl">
<head>
<script type = "text/javascript">
var country ;

country=window.prompt( "أدخل الدولة لتعرف عاصمتها", "" );
switch ( country )
{
case "فلسطين" :

document.writeln("<h3>عاصمتها القدس الشريف</h3>");
break ;

case "العراق" :

document.writeln("<h3>عاصمتها بغداد</h3>");
break;

case "السودان" :

document.writeln("<h3>عاصمتها الخرطوم</h3>");
```



```

break ;
case "السعودية" :
    document.writeln("<h3>عاصمتها الرياض </h3>");
    break ;
default :
    document.writeln("<h3>الدولة التي أدخلتها ليست من ضمن الخيارات
    </h3>");
}
</script>
</head>
<body></body>
</html>

```

الحلقات التكرارية (جمل التكرار) Iteration Statements

تستخدم في تكرار تنفيذ تعليمة أو مجموعة من تعليمات البرنامج أكثر من مرة وهي نوعان :

انواع الحلقات التكرارية

١. حلقة for

صيغتها:

for(counter=initial value;condition;step)

حيث:

condition: هو عبارة عن جملة تقارن بين قيمة العداد الحالية والقيمة النهائية

Step: تعبير رياضي يوضح التغير في العداد في كل دورة.

مثال :- اكتب برنامج لطباعة الاعداد من ١ الى ١٠

```

<html dir="rtl">
<head>
<script type = "text/javascript">
    document.write("<br>الاعداد هي");
    for (var i = 1 ; i <=10; i++ )
        document.write(i+"<br>");
</script>
</head>
<body></body>
</html>

```

تدريب عدل البرنامج اعلاه لطباعة الاعداد من ١٠ الى ١ (أي بشكل تنازلي)

٢. حلقة while

صيغتها:



```
while (condition)
{
statement 1 ;
.....
statement n ;
}
```

مثال :- اكتب برنامج لطباعة الاعداد من ١ الى ١٠

```
<html >
<head>
<script type = "text/javascript">
var i = 1 ;
while (i <= 10 )
{
document.write ( i+"<br>" ) ;
i++ ;
}
</script>
</head>
<body></body>
</html>
```

٣. حلقة do-while :

هذه الجملة مشابهة لجملة while وتختلف عنها في أمرين :

- تبدأ بالعملية أولاً ويتم التحقق من الشرط في أسفل الجملة.
- لا بد من تنفيذ الجمل الموجودة بين do و while مرة واحدة على الأقل حتى وان كان الشرط

غير صحيح

صيغتها:

```
do
{
statement 1;
.....
statement n ;
} while(condition)
```

مثال :- اكتب برنامج لطباعة الاعداد من ١ الى ١٠

```
<html >
<head>
```



```
<script type = "text/javascript">
  var i = 1 ;
  do {
    document.write( i+"<br>" );
    i++;
  }
  while ( i <= 10 )
</script>
</head>
<body></body>
</html>
```

مثال :- مستخدماً لغتي *html* و *JavaScript*

انشي نموذج لإدخال تاريخ الميلاد كما بالشكل التالي :

Date of Birth :	Day 1 ▾	, Month 1 ▾	, Year 1960 ▾
-----------------	---------	-------------	---------------

علماً بأن

اليوم <i>Day</i>	يأخذ القيم من ١ الى ٣١
الشهر <i>Month</i>	يأخذ القيم من ١ الى ١٢
السنة <i>Year</i>	يأخذ القيم من 1960 الى 2013

```
<html>
<body>
<form>
<table border="1 align="center" bgcolor="aqua">
  <tr>
    <td ><b>Date of Birth :</b></td>
    <td>
      Day
      <select name="Day" size="1" >
        <script language="JavaScript">
          for(i=1;i<=31;i++)
            document.writeln("<option value=i>" + i + "</option>");
        </script>
      </select>&nbsp;  ,
    </td>
  </tr>
</table>
</form>
</body>
</html>
```



```
Month
<select name="Month" size="1" >
  <script language="JavaScript">
    for(i=1;i<=12;i++)
      document.writeln("<option value=i>" + i + "</option>");
  </script>
</select>&nbsp; ,
Year
<select name="Year" size="1" >
  <script language="JavaScript">
    for(i=1960;i<=2013;i++)
      document.writeln("<option value=i>" + i + "</option>");
  </script>
</td>
</tr>
</table>
</form>
</body>
</html>
```



الدرس الرابع عشر

المصفوفات Arrays والدوال Functions

المصفوفات Arrays

المصفوفة هي عبارة منطقة في الذاكرة تتكون من عدد محدد من المواقع المتجاورة .

تعريف المصفوفة الاحادية

هي عبارة عن صف أو عمود يحتوي علي عدة عناصر من البيانات .

var name = new Array(size)

حيث : **name** اسم المصفوفة و **size** حجم المصفوفة (عدد عناصرها)

<p>التخزين في المصفوفات يبدأ من الموقع ٠ حيث ان العنصر الاول يخزن في الموقع 0 والعنصر رقم n يخزن في الموقع $n-1$</p> <p>يمكن للمصفوفة أن تحتوي علي أي نوع من البيانات بمعنى اخر ليس من الضروري ان تكون البيانات متجانسة (اي من نفس النوع) .</p>	☑
--	---

بناء المصفوفة :-

يمكن وضع العناصر في المصفوفة بأكثر من طريقة :-

الطريقة الأولى

```
var family = new Array (4)
family [0] = "father"
family [1] = "mother"
family [2] = "brother"
family [3] = "sister"
```

الطريقة الثانية

```
Var family = new Array (4)
family = ["father", "mother", "brother", "sister"]
```

الطريقة الثالثة

```
var family = new Array ("father", "mother", "brother", "sister")
```

طباعة عناصر المصفوفة :-

يتم عن طريق دالة الطباعة (**document.write** ()) مع تحديد موقع العنصر المراد طباعته.

مثال :-

```
<html>
<head>
<script type="text/javascript">
var family = new Array (4);
family [0] = "father";
family [1] = "mother";
```



```
family [2] = "brother";
family [3] = "sister";

for(i=0;i<٤;i++)
document.write(family[i]+"<br>");
</script>
</head>
<body>
</body></html>
```

اما في حالة طباعة عناصر المصفوفة على شكل سطر واحد يفصل بين كل عنصر والأخر بفاصلة فيكفي كتابة اسم المصفوفة فقط
مثلاً كما في المثال السابق فيكفي كتابة الجملة التالية :

```
document.write(family);
```



مثال :- اكتب برنامج لإدخال n من العناصر في مصفوفة ثم طباعتها على الشاشة

```
<html>
<head>
<script type="text/javascript">
var i,n;

n=window.prompt("ادخل عدد عناصر المصفوفة");
var x=new Array(n);
for(i=0;i<n;i++)
{
var y=i+1;

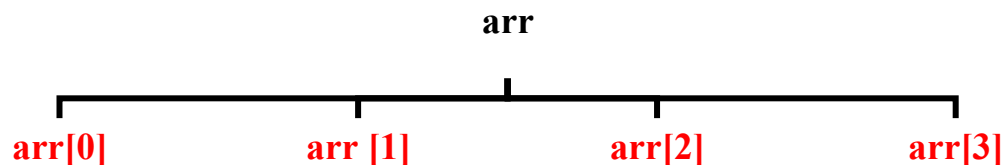
x[i]=window.prompt("ادخل العنصر رقم "+y);
}
for(i=0;i<n;i++)
document.write("<b>" + x[i] + "</b><br>");
</script>
</head>
<body>
</body>
</html>
```

تعريف المصفوفة الثنائية

هي عبارة عن عدة صفوف وعدة أعمدة (أي جدول).

```
var arr=new Array(4)
```

اعلان عن مصفوفة مكونة من اربعة صفوف

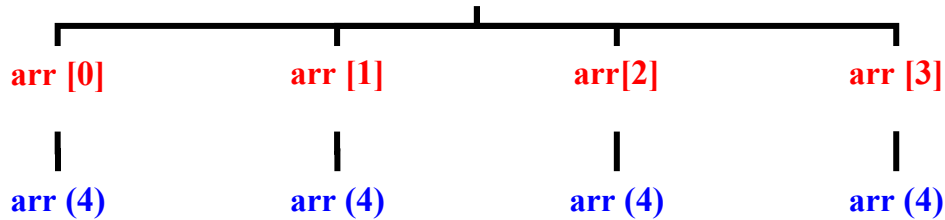




```
var arr = new Array(4)
for (i=0; i < 4; i++)
  arr[i] = new Array(4)
```

وكل صف من المصفوفة مكون من أربعة أعمدة

arr



مثال :- برنامج لإدخال عناصر مصفوفة ثنائية مكونة من أربعة صفوف وأربعة أعمدة ثم طباعتها على

شكل المصفوفة الثنائية

```
<html>
<head>
<script type="text/javascript">
var arr = new Array(4);
var i,j;

document.writeln("<h2>طباعة مصفوفة ثنائية الأبعاد</h2>");
for (var i = 0; i < 4; i++)
{
  arr[i] = new Array(4);
  for (var j = 0; j < 4; j++)
  {
    arr[i][j]=window.prompt("ادخل عناصر المصفوفة","");
  }
}
for (i=0 ;i< 4;i++)
{
  for (j=0 ;j< 4;j++)
  document.writeln(arr[i][j] + " ");
  document.writeln("<br>");
}
</script>
</head>
<body>
</body>
</html>
```

مثال :-

برنامج لإدخال بيانات n من الموظفين تشمل الرقم والاسم والعنوان والراتب ثم طباعتها على شكل جدول

```
<html dir="rtl">
```



```

<head>
<script type="text/javascript">
var i,j,n;
var employee = new Array(n);
n=window.prompt("ادخل عدد الموظفين","");
for (var i = 0; i < n; i++)
{
var x=i+1;
window.alert("ادخل بيانات الموظف "+x);
employee[i] = new Array(4);
for (var j = 0; j < 4; j++)
{
if(j==0)
employee[i][j]=window.prompt("ادخل رقم الموظف","");
else if(j==1)
employee[i][j]=window.prompt("ادخل اسم الموظف","");
else if(j==2)
employee[i][j]=window.prompt("ادخل عنوان الموظف","");
else if(j==3)
employee[i][j]=window.prompt("ادخل الراتب","");
}
}
document.write("<table width='50%' bgcolor='green' border='4'>");
document.write("<tr><td align='center' colspan='4'>بيانات الموظفين</td></tr>");
document.write("<tr><td>الرقم</td><td>الاسم</td>");
document.write("<td>العنوان</td><td>الراتب</td></tr>");
for (i=0 ;i< n;i++)
{
document.write("<tr>");
for (j=0 ;j< 4;j++)
document.write("<td>"+employee[i][j]+"</td>");
document.write("<tr>");
}
document.write("</table>");
</script>
</head>
<body>
</body>
</html>

```

الدوال Functions

الدالة هي عبارة عن برنامج فرعي يتم تعريفه ليقوم بوظيفة معينة ويتم استدعاؤها داخل البرنامج من أجل تنفيذ التعليمات الموجودة بداخلها

الفائدة من استخدام الدوال

- سهولة متابعة وإصلاح البرنامج .



- المساعدة في فهم البرنامج.
- عدم تكرار الجمل المراد إعادة تنفيذها .

الصيغة العامة للدالة

```
function function _ name (par1,par2,...)
{
    function statements;
    return(expression);
}
```

- **function** كلمة محجوزة تستخدم للإعلان عن الدالة
- **function name** : اسم الدالة (اسم يحدد من قبل المستخدم)
- **(par1,par2...)** : معاملات أو وسائط الدالة
- **function statements** : تعليمات الدالة
- **return** : ترجع قيمة التعبير **expression** إلى مكان استدعاؤها.

يتم استدعاء الدالة عن طريق اسمها مع مراعاة نوع وعدد الوسائط التي تستقبلها .	✓
---	---

مثال :-

برنامج يحتوي على دالة تقوم بإيجاد حاصل جمع عددين مدخلين من قبل المستخدم

```
<html>
<head>
  <script type = "text/javascript">
    var x,y,z ;
    x=parseInt(window.prompt("ادخل العدد x",""));
    y=parseInt(window.prompt("ادخل العدد y",""));
    document.write ( x ) ;
    document.write ( "<br>" + y ) ;
    z = summation ( x , y ) ;
    document.write ( " <br> " + z ) ;
    function summation ( a , b )
    {
      var sum = 0 ;
      sum= a + b ;
      return sum ;
    }
  </script>
</head>
<body></body>
</html>
```




مثال :-

برنامج يحتوي على دالتين (دالة تقوم باستدعاء داله بداخلها)
الدالة الاولى تقوم بإدخال عدد ثم استدعاء الدالة الاخرى التي تقوم بحساب مكعب العدد

```
<html>
<head>
<script type = "text/javascript">
function output ( )
{
var x,t;

x=parseInt(window.prompt("ادخل العدد x",""));
t = cube (x) ;
document.writeln ( t ) ;
}

function cube ( y )
{
return y*y*y ;
}
</script>
</head>
<body onload = " output ( ) "></body>
</html>
```

✓	<p>معلومة مهمة جداً</p> <p>onload = " output () "</p> <p>التعليمة اعلاه كتبت داخل الوسم body ووظيفتها استدعاء الدالة output () عند تحميل أو تشغيل الصفحة.</p>
---	--

مثال :- ربط النماذج مع لغة **Java Script**

برنامج تطبيقي يحتوي على

١. نموذج بالشكل التالي :

الاسم :	حذيفة عبدالرحمن ابراهيم
العنوان :	جامعة الملك سعود -كلية المعلمين -قسم الحاسب الالى
المهنة :	استاذ جامعي
ملاحظات :	مقرر برمجة الانترنت النماذج والدوال في [java script]
<input type="button" value="اعرض"/> <input type="button" value="مسح"/>	



٢. يقوم المستخدم بتعبئة النموذج و الضغط على زر "اعرض" عندها تظهر البيانات المدخلة في النموذج على جدول كما بالشكل التالي :

المعلومات التي سجلتها لدينا

الاسم :	حنيفة عبدالرحمن ابراهيم
المهنة :	استاذ جامعي
العنوان :	جامعة الملك سعود -كلية المعلمين -قسم الحاسب الالى
ملاحظات :	مقرر برمجة الانترنت النماذج والذوال في java script

علماً بان البيانات المدخلة في النموذج تقوم دالة معرفة من قبل المستخدم بتخزينها في الجدول

نص البرنامج :

```
<html dir="rtl">
<head>
<script type="text/javascript">
function information( )
{
var name=info.name.value;
var job=info.job.value;
var add=info.address.value;
var comm=info.comments.value;
document.writeln("<table cellpadding='5' border='1' width='80%' align='center'");
document.writeln("<dir='rtl' bgcolor='sky blue' bordercolor='red'>");
document.writeln("<caption><h3>المعلومات التي سجلتها لدينا</h3></caption>");
document.writeln("<tr><td width='30'><b>الاسم :</b></td><td> " + name +
"</td></tr>");
document.writeln("<tr><td width='30'><b>المهنة :</b></td><td> " + job +
"</td></tr>");
document.writeln("<tr><td width='30'><b>العنوان :</b></td><td> " + add +
"</td></tr>");
document.writeln("<tr><td width='30'><b>ملاحظات :</b></td><td> " + comm +
"</td></tr>");
document.writeln("</table>");
}
</script>
</head>
<body >
<form name="info" action="">
<table bgcolor="green" border="3" width="50%" align="center">
<tr>
<td>الاسم :</td>
<td><input type="text" name="name" value="" size="31"></td>
</tr>
```



```

<tr>
  <td>العنوان:</td>
  <td><input type="text" name="address" value="" size="45"></td>
</tr>
<tr>
  <td>المهنة:</td>
  <td><input type="text" name="job" value="" size="20"></td>
</tr>
<tr>
  <td>ملاحظات:</td>
  <td>
    <textarea name="comments" rows="4" cols="50"></textarea>
  </td>
</tr>
<tr>
  <td colspan="2" align="center">
    <input type="button" name="show" value="اعرض" onclick="information( )">
    <input type="reset" name="del" value="مسح" >
  </td>
</tr>
</table>
</body>
</html>

```