



HTML5 + HTML المختصر المفيد في

جمع وإعداد :

أحمد إبراهيم

ahmedcia17@gmail.com

الطبعة الأولى

20 / 4 / 2014

فهرس الكتاب

3 المقدمة
4 الفصل الأول (كتابة النصوص وتنسيقاتها)
8 الفصل الثاني (الوسائط المتعددة)
13 الفصل الثالث (وسوم وخصائص)
16 الفصل الرابع (الروابط)
20 الفصل الخامس (الجداول والقوائم)
25 الفصل السادس (الإطارات)
29 الفصل السابع (النماذج)
43 الفصل الثامن (الرسم)
47 الفصل التاسع (XHTML)
49 الفصل العاشر (ملاحظات عامة)
55 الخاتمة

المقدمة

اجزاء لغة html

دائماً يجب أن تبدأ اللغة بالوسم `<html>` وتنتهي بالوسم `</html>` وتكون قسمين :- القسم الأول هو قسم الرأس (الجزء الرئيسي أو العنوان) ويكتب فيه كود العنوان `<title>` واكواد ال CSS واكواد الجافا سكربت , ويبدأ بالوسم `<head>` وينتهي بالوسم `</head>` , أما القسم الثاني هو قسم الجسم (هيكل البرنامج وما سيظهر داخل الصفحة) وتكتب فيه جميع الرموز وهو يبدأ بالوسم `<body>` وينتهي بالوسم `</body>` . والوسم `<title>` يكتب في القسم الأول `<head>` ويكتب داخله عنوان الـ class . مثال /

```
<html>
```

```
<head> <title> العنوان </title> </head>
```

```
<body>
```

```
</body>
```

```
</html>
```

الفصل الأول

(كتابة النصوص وتنسيقاتها)

كتابة النص

بما أننا سنكتب باللغة العربية نحتاج أن يكون النص من اليمين لليساار ولهذا نحتاج إلى خاصية نكتبها في داخل الوسم التعريفي `<html>` هكذا: `<html dir="rtl">` حيث إن `dir` هو اختصار لكلمة Direction ومعناها اتجاه , و `rtl` هو اختصار Right to Left وهي تعني إن النص سيكون من اليمين لليساار , وإذا أردنا أن يكون اتجاه النص من اليسار الى اليمين نضع القيمة `ltr` .

ملاحظة:- إن لغة html لا تفرق بين الحروف الكبيرة والصغيرة أي لا فرق بين DIR وبين dir .
هناك عدة وسوم للكتابة :-

أولاً - الوسم `<p> ... </p>` :-

وهو يحتوي على بعض الخصائص :-
أ- الخاصية `dir` وهي اتجاه النص وتأخذ القيمة `rtl` أو القيمة `ltr` ,
ب- الخاصية `align` وهي تأخذ إحدى القيم الثلاث التالية (`right , left ,center`) ,
ج- الخاصية `id` وهي تستخدم لإعطاء اسم للفقرة .
مثال:-

```
<body>
```

```
<p id="NAME" dir="rtl" align="right" > مرحبا بك </p>
```

```
</body>
```

2- الوسم `` :-

حيث يحتوي على مجموعة من الخصائص :-
أ- `color` وهي لتحديد اللون , حيث يتم كتابة اسم اللون أو رمزه السدس العشري (سيتم شرحه فيما بعد) ,
ب- `size` وهي تمثل حجم الخط وهي تحدد بقيم من 1 إلى 7 ,
ج- `face` وهي تحدد نوع الخط , واليك أسماء أشهر أنواع الخطوط المستعملة :-

- * Arial , Helvetica , sans-serif
- * Verdana , Geneva , sans-serif
- * Times New Roman , Times , serif
- * Garamond , Palation , serif

* ويمكن تحديد أكثر من نوع من الخطوط لكي إن لم يتواجد هذا النوع في متصفح الزائر سيرى النوع الثاني ومثال على ذلك :-

<body>

 النص

</body>

-3 <h> :-

وهو يكون من <h1> إلى <h6> حيث كلما كبر الرقم صغر الخط .

مثال :-

<h1> نص كبير </h1>

<h6> نص صغير </h6>

4- وهناك بعض الوسوم للتغيير على النصوص منها :-

 خط غامق

 نص غامق

<address> عنوان </address>

 تأكيد النص

<kbr> يظهر النص مثل النص المدخل من لوحة المفاتيح </kbr>

<pre> نص يحتوي على مسافات بين الكلمات </pre>

<samp> لإعطاء الأمثلة </samp>

<u> نص تحته خط </u>

<ins> خط تحت النص </ins>

 خط في وسط النص (نص مشطوب)

</s> نص مشطوب </s>

 نص مشطوب

</i> نص مائل </i>

</tt> نص بخط الآلة الكاتبة - نص مرتب- (إذا كن انجليزي) </tt>

</nobr> نص </nobr>

</small> نص صغير </small>

</big> نص كبير </big>

</blockquote> نص مزاح </blockquote>

* حيث كتبنا في داخل كل وسم وظيفته التي يؤديها وفي هذا المكان يكتب أي نص (النص المطلوب) وجميع هذه الوسوم تكتب داخل الهيكل <body>

النصوص المتحركة

لجعل النص يتحرك نستخدم الوسم <marquee> , ولهذا الوسم عدة خصائص :-

1- bgcolor لتحديد اللون .

2- width لتحديد عرض النص (الالاقفة) .

3- height لتحديد طول النص (الالاقفة) .

4- align تستعمل للمحاذاة , وتأخذ أحد القيم التالية (top , bottom , middle) .

5- hspace لتحديد المسافة التي تفصل بين الالاقفة والنص الموجود على جانبها .

6- vspace تعمل مثل الخاصية hspace لكنها تحدد المسافة الفارغة عموديا من الأعلى والأسفل .

7- behavior تحدد أسلوب النص الموجود في الالاقفة وطريقة عمله وهناك ثلاثة قيم أو بالأحرى ثلاثة أساليب لحركة النص وهي التالية :-

أ- scroll يتحرك بنفس الاتجاه من جانب إلى آخر وبصورة مستمرة , وهي القيمة الافتراضية لذلك لا يهم إن كتبت أم لا .

ب- **slide** يتحرك النص مرة واحدة من الجانب إلى الجانب الآخر ويتوقف عنده .

ج- **alternate** يتأرجح النص جيئةً وذهاباً من جانب إلى آخر .

<marquee behavior="alternate" > the text </marquee>

8- **direction** تحدد اتجاه سير النص وذلك من خلال القيم **left** أو **right** .

9- **loop** تحدد عدد المرات التي سيتحرك فيها النص داخل اللافطة , وإذا أردت أن تستمر الحركة إلى مالا نهاية فضع القيمة **1-1** أو **infinite** , وبخلاف ذلك ضع عدد المرات التي تريدها .

10- **scrollamount** هذه الخاصية تحدد المسافة بين كل لقطة وأخرى للنص , وهي تأخذ قيمةً عددية صحيحة تحدد المسافة المقطوعة بالبيكسل .

مثال :-

<marquee scrollmount="50" > the text </marquee>

11- **scrolldelay** لكي نحدد الزمن الذي يستغرقه النص في القفز بين كل خطوة وأخرى نستخدم هذه الخاصية , والقيمة المعطاة مع هذه الخاصية تمثل الزمن بالمللي ثانية (0,001 من الثانية) .

ملاحظة / إن النص الذي يوضع في الوسم **<marquee>** ليس نص ثابت ولا تستطيع أن تغير من خواصه لكن تستطيع أن تجعله وصلة تشعبيه وأن تدرج داخله صورته أيضا (سأشرح ذلك في المواضيع القادمة) .

الفصل الثاني

(الوسائط المتعددة)

إضافة الصور

يتم إضافة الصور عن طريق الوسم `` والذي هو وسم ذاتي الغلق أي ليس له وسم نهاية , حيث يحتوي على بعض الخصائص :-

- 1- الخاصية `src` والتي هي اختصار لكلمة `source` , وهي تأخذ عنوان الصورة وامتدادها .
 - 2- الخاصية `alt` ويكتب فيها التعليق الذي سيظهر في حالة عدم توفر الصورة أو في حالة حدث خطأ عند تحمل الصورة .
 - 3- الخاصية `align` وهي تستخدم لتحديد موقع النص من الصورة وتأخذ إحدى هذه القيم (`right` , `left` , `center` , `top` , `bottom`) .
 - 4- الخاصية `width` وهي تحدد عرض الصورة وتكون قيمتها إما عدد صحيح أو نسبة مئوية مثل :- 30% .
 - 5- الخاصية `height` وهي تحدد ارتفاع الصورة وتكون قيمتها إما عدد صحيح أو نسبة مئوية مثل :- 35% .
 - 6- الخاصية `vspace` لتحديد المسافة العمودية الفاصلة بين النص والحافتين العليا والسفلى للصورة .
 - 7- الخاصية `hspace` لتحديد المسافة الأفقية الفاصلة بين النص والحافتين اليمنى ويسرى للصورة .
 - 8- الخاصية `border` وهي تحدد سمك الإطار المحيط بالصورة , وتأخذ قيم من 1 إلى 7 , والقيمة الافتراضية لها صفر .
- * لا يشترط أن تكتب هذه الخواص بالترتيب , ولا يجب كتابتها كلها حيث إن الخاصية التي لا نكتبها ستأخذ القيمة الافتراضية لها .

مثال :-

```

```

لاحظ إن الخاصية **src** كتبنا فيها أولاً اسم المجلد الذي يحتوي على الصورة والذي في هذا المثال هو Image ثم / وبعدها اسم الصورة والتي في المثال اسمها tree ثم نكتب امتداد الصورة وفي المثال كان الامتداد jpg .

ولاحظ هنا إن الصورة تقع في مجلد مختلف عن المجلد الذي فيه الصفحة , لكن إن كانت الصورة والصفحة في نفس المجلد فلا نحتاج إلى ذكر اسم المجلد حيث فقط سنذكر اسم الصورة والامتداد , مثال على ذلك :-

```

```

وإن كانت الصورة على موقع في الانترنت ونريد أن نستخدمها فسوف نذكر اسم الموقع مع اسم الصورة والامتداد .

* ولاحظ أن الصورة التي نصغرها عن طريق الخاصيتين **width** و **height** أن حجمها لا يصغر وإنما فقط شكلها سيتغير لذا إن أردنا استخدام صورة في موقعنا بمساحة صغيرة فمن الأفضل تصغير حجمها أولاً , وذلك لكي لا يكون موقعنا ذو حجم كبير وبالتالي يكون تحميل الموقع بطيء من قبل الزائر .

والآن إن أردت وضع صورة كخلفية للموقع فعليك أن تكتب عنوان الصورة داخل الخاصية **background** والتي هي إحدى خصائص الوسم **<body>** . مثال :-

```
<body background="Image.jpg" >
```

ولتكون الصورة في الخلفية بشكل علامة مائية (أي أن تكون الصورة في الخلف ثابتة و الصفحة تتحرك فقط بدون أن تتحرك الصورة) يجب استخدام الخاصية **bgproperties** والتي هي إحدى خصائص الوسم **<body>** وتأخذ القيمة **fixed** .

مثال :-

```
<body bgproperties="fixed" >
```

إدراج فلاش

لإدراج فلاش نستخدم الوسم **embed** وهو وسم ذاتي الغلق أي ليس له وسم نهاية , لاحظ هذا المثال :-

```
<embed src="pop.swf" height="20%" width="15%" border="4" />
```

لا أعتقد إن هذه الخواص المستخدمة مع الوسم غير مفهومة لأنه سبق وأن شرحناها , حيث **src** أخذت اسم الفلاش وامتداده و**height** تحدد ارتفاع الفلاش و **width** تحدد عرض الفلاش أما **border** فهذه الخاصية تحدد سمك الإطار المحيط بالفلاش وهي تأخذ قيم من 1 إلى 7 , وستتعرف عليها أكثر في قسم الجداول .

صوت في الخلفية

يمكن جعل صوت في خلفية الصفحة وذلك باستخدام الوسم **<bgsound />** , وهو وسم ذاتي الغلق اي ليس له وسم نهاية . لاحظ هذا المثال :-

```
<bgsound src="pomp.mp3" loop="3" />
```

* إن الخاصية **src** معروفة حيث أخذت اسم الصوت وامتداده , أما الخاصية **loop** فهي تحدد عدد مرات التي ستتكرر فيها النغمة , وإذا جعلت قيمة هذه الخاصية **1- infinite** فهذا يعني إن الصوت سيعاد إلى مالا نهاية .

ملاحظة / هذه الطريقة لا تعمل إلا مع المتصفح إكسبلورر وبعض المتصفحات الأخرى .

عنصر الفيديو Video

هذا العنصر جديد في لغة html5 , وهو يستخدم لعرض مقطع فيديو في الصفحة وهو لا يدعم سوى ثلاث صيغ من مقاطع الفيديو وهي (webm , ogg , MPEG) , والوسم الخاص بهذه العملية هو **<video>** وهو بدوره يحتوي على وسم الابن **<source />** وهو وسم ذاتي الغلق أي ليس له وسم نهاية , وهذا الابن يحتوي على الخاصية **src** التي تسند لها قيمة تكون رابط الفيديو وتأخذ أيضا الخاصية **type** لوضع نوع الفيديو .

مثال /

```
<video width="390" height="240" controls="controls" >
<source src="movie.mp4" type="video/mp4" />
للأسف مستعرضك لا يدعم عنصر عرض الفيديو !
</video>
```

* حيث إن هذه الكتابة سوف تظهر فقط لمن لا يدعم متصفحة هذا العنصر (عنصر الفيديو) .

هذا الجدول يحتوي على خصائص الوسم <video>

الخاصية	القيمة	الشرح
audio	muted	كتم الصوت
autoplay	autoplay	التشغيل التلقائي لملف الفيديو
controls	controls	عرض شريط الأدوات في مشغل الفيديو
height	قيمة رقمية بالبيكسل	ارتفاع مشغل الفيديو
width	قيمة رقمية بالبيكسل	عرض مشغل الفيديو
loop	loop	اعادة تشغيل الفيديو بشكل تلقائي عند انتهائه
poster	رابط تشعبي	رابط تشعبي لصورة تعبر عن مقطع الفيديو وتظهر قبل ان يقوم المستخدم بتشغيله
src	رابط تشعبي	رابط ملف الفيديو
type	صيغة الفيديو	لتعريف نوع صيغة الفيديو

عنصر الصوت Audio

هذا العنصر جديد في لغة html5 , يمكن عرض مقطع صوتي في الموقع باستخدام الوسم <audio> والذي يحتوي بدوره على الوسم <source /> وهو وسم ذاتي الغلق أي ليس له وسم نهاية , والذي يحتوي على الخاصية src لذكر عنوان الصوت والخاصية type لذكر نوع الصوت , مع ملاحظة انه لا يدعم سوى ثلاث صيغ من مقاطع الصوت وهي (wav , ogg , mp3) .

مثال /

```
<audio controls="controls" >
<source src="song.mp3" type="audio/mp3" />
للأسف مستعرضك لا يدعم عنصر عرض الصوت !
</audio>
```

* حيث إن هذه الكتابة سوف تظهر فقط لمن لا يدعم متصفحة هذا العنصر (عنصر الصوت).

هذا الجدول يحتوي على عناصر الوسم <audio>

الخاصية	القيمة	الشرح
autoplay	autoplay	التشغيل التلقائي لملف الصوت
controls	controls	عرض شريط الأدوات في مشغل الصوت
loop	loop	إعادة تشغيل ملف الصوت بشكل تلقائي
preload	preload	تحميل ملف الصوت عند تحميل الصفحة بغض النظر عن خاصية التشغيل التلقائي
src	رابط تشعبي	رابط ملف الصوت

الفصل الثالث

(وسوم وخصائص)

* لتغيير لون خلفية الصفحة نستخدم الخاصية **bgcolor** والتي هي إحدى خصائص الوسم **body** , لاحظ :-

```
<body bgcolor="#dd55dd" >
```

* في كل لغات البرمجة يحتاج المبرمج لإضافة تعليقات داخل البرنامج لتسهيل العمل , وهذه التعليقات لا تظهر في البرنامج (الصفحة) , ولغة html توفر وسم خاص بالتعليقات وهو :-

```
<!-- التعليق -->
```

* ولإنهاء سطر والبدء بسطر جديد نحتاج إلى هذا الوسم الذاتي الغلق وهو :- **
**

* لتوسيط نص أو صورة أو أي شيء ببساطة نستخدم هذا الوسم :-

```
<center>
```

الشيء المراد جعله في منتصف الصفحة

```
</center>
```

* لجعل نص أو صورة ترتفع قليلاً أو تنخفض نستخدم أحد الوسمين التاليين :-

```
<sup> للأعلى </sup>
```

```
<sub> للأسفل </sub>
```

* وسم المسطرة **<hr />** , يعمل هذا الوسم على إضافة شريط أفقي يمكن استعماله كفاصل بين مختلف أجزاء صفحتك , وهو وسم ذاتي الغلق أي ليس له وسم نهاية , وهو يحتوي على الخصائص التالية :-

1- **align** وتستعمل للمحاذاة وتأخذ إحدى القيم التالية (**right , left ,center**) .

2- **size** تحدد عرض الشريط .

3- **width** تحدد طول الشريط , وتلقائياً تأخذ القيمة 100% إن لم نحدد قيمة .

4-noshade يعمل على حذف الظل من الشريط .

5-color لتحديد لون الخط (تعمل فقط مع Ms Explorer) .

```
<hr width="80%" size="3" align="center" noshade color="#006699" />
```

* button يستخدم هذا الوسم لعمل زر اعتيادي بسهولة بهذا الشكل :-

```
<button>Value</button>
```

* يستخدم هذا الوسم لطباعة فاصلة (فراغ) وهو يكتب هكذا بدون أي قيمة ,
ويكتب بعدد الفواصل المطلوبة :-

```
<body>
```

```
<p> Hi </p>
```

```
&nbsp;
```

```
&nbsp;
```

```
<p> welcome </p>
```

```
</body>
```

* الـ div هذا الوسم ينسق الصفحات كأقسام مختلفة يستخدم بصفة أساسية stylesheet لعمل الطبقات , وهو يعتبر من الوسوم المهمة , وتكتب معه بعض الخصائص منها :-

1- الخاصية align وهي تستخدم لمحاذاة الطبقة وتأخذ احد هذه القيم (left , right , center).

2- الخاصية id وهي تستخدم لإعطاء اسم لهذا القسم .

* الـ span هذا الوسم يستخدم في الارتباطات stylesheet ويعتبر مثل الـ div ولكن الـ span يؤثر على النص فقط أما الـ div فيؤثر على المقطع الأفقي كله .

* خصائص الـ body كاملة

```
<body dir="rtl" bgcolor="#ffffff" background="tree.jpg" text="#000000" link="#3399ff" vlink="#9966ff" alink="#000000" bgproperties="fixed" topmargin="0" marginheight="0" >
```

حيث تم شرح قسم من هذه الخصائص , وسنتطرق إلى الخصائص الأخرى في المواضيع القادمة .

* الوسم **<noscript>** وهو يستخدم إذا كنا نستخدم لغة برمجة من نوع سكريبت (Script) مثل JavaScript وكان مستعرض الإنترنت لا يدعم هذه اللغة فسنحتاج إلى إخبار المستخدم او عرض كود بديل لهذه اللغة التي لم تعرض هنا سنستخدم هذا الوسم حيث أنه لا يعرض ما بداخل هذا الوسم إلا إذا لم يستطع مستعرض الإنترنت من عرض لغة البرمجة من نوع Script لأي سبب كان .

مثال /

```
<html>
<head>
</head>
<body>
<script>
كود لغة سكريبت
</script>
<noscript>
الكود البديل
</noscript>
</body>
</html>
```

الفصل الرابع

الوصلات التشعبية (الروابط)

يتم الربط بين الصفحات في لغة html باستخدام الوسم **<a>** ولهذا الوسم عدة خصائص :-

- 1- **href** تحتوي هذه الخاصية على عنوان الصفحة المراد زيارتها .
- 2- **name** تستخدم هذه الخاصية لتسمية جزء معين من الصفحة لكي نتمكن من الرجوع إلى هذا الجزء عن طريق وصلة تشعبية (سأشرح هذه الخاصية بشكل مفصل بعد قليل) .
- 3- **title** تستخدم هذه الخاصية لإظهار نص معين عند الوقوف بمؤشر الفأرة على الرابط .

**** اضغط هنا ****

- 4- **target** تستخدم هذه الخاصية إذا أردنا الربط بصفحة جديدة وفتح هذه الصفحة في إطار (نافذة) جديدة وتأخذ القيمة **"_blank"** او تأخذ القيمة **"_self"** لعرض الصفحة في نفس الصفحة الحالية .

**** العبارة التي ستظهر ****

حتماً لاحظت عند تصفحك الانترنت وجود هذه الوصلات التي تكون عادةً باللون الأزرق وعند وضع مؤشر الفأرة فوقها تأخذ اللون الأحمر , يمكنك بكل سهولة تغيير هذه الألوان واستبدالها بأخرى من اختيارك , وذلك من خلال الوسم الرئيسي **body** الذي يحتوي على هذه الخصائص:-

أ- **link** يمثل لون الروابط غير النشطة (الروابط التي لم يتم النقر عليها) , تلقائياً تأخذ اللون الأزرق .

ب- **alink** يمثل لون الروابط النشطة (عند النقر على الرابط) , تلقائياً تأخذ اللون الأحمر .

ج- **vlink** يمثل لون الروابط التي تم زيارتها , تلقائياً يأخذ لون الخباز .

مثال :-

<body link="#009900" alink="#ff0000" vlink="#8866f0" >

هناك عدة أنواع من الروابط :

* يمكن وضع اسم وامتداد مقطع الصوت او الفيديو كقيمة للخاصية href ليتم تشغيله عند الضغط على الرابط .

1- ربط صفحة بصفحة أخرى , نكتب :-

 الصفحة الأولى

2- ربط صفحة بموقع , نكتب :-

 موقع البحث جوجل

3- ربط صفحة ببريد الكتروني , نكتب :-

 أرسل رسالة على البريد

4- إنشاء رابط في نفس الصفحة :

لقد لاحظت في بعض صفحات الويب عند النقر على عنوان ما في أعلى الصفحة يأخذك إلى محتوى العنوان في نفس الصفحة , ولإنشاء ذلك ما عليك إلا أتباع الخطوات التالية :-

أ- قم بتسمية الجزء المراد إظهاره بعد النقر على الرابط وذلك باستعمال الخاصية name كما يلي :-

 النص المسمى

ب- أنشئ الرابط مع وضع الرمز # قبل عنوان الجزء المراد إظهاره كما يلي :-

 العنوان 1

* في حالة ما لو أردت أن تظهر جزء معين من صفحة أخرى بالنقر على عنوان موجود في الصفحة الأولى , فعليك أن تكتب اسم الصفحة أولاً ثم الاسم هكذا :-

 العنوان 1

مثال / لإنشاء رابط للانتقال إلى جزء آخر من نفس الصفحة :-

<html><head></head><body>

 الى الفقرة الرابعة

<h1 id="para4" > الفقرة الرابعة </h1>

</body></html>

* وإذا أردت أن يكون الرابط عبارة عن صورة فالأمر في غاية البساطة فما عليك إلا وضع وسم الصورة داخل وسم الرابط بالطريقة التالية :-

```
<a href="page1.html" >  </a>
```

ستلاحظ إن الصورة محاطة بإطار , لإلغاء هذا الإطار نجعل الخاصية **border** تساوي صفر , لاحظ :-

```
<a href="page1.html" >  </a>
```

* لفتح صورة أو مقطع صوت أو فيديو عن طريق وصلة تشعبيه فما علينا إلا كتابة الاسم والامتداد , هكذا :-

```
<a href="image.jpg" > the image </a>
```

ملاحظة / يجب كتابة اسم الصفحة الرئيسية **index** وذلك لكي يتعرف عليها المتصفح ويعرضها أولاً .

الخرائط الصورية

وذلك لجعل الصورة الواحدة مقسمة إلى عدة مناطق وكل منطقة تشير إلى وصلة تشعبيه معينة , ولفعل ذلك نحتاج إلى بعض الأوسمة والخواص , وإليك هذا المثال :-

```

<map name="the name" >
<area shape="poly" coords="10,10,130,10,110,65,10,140"
href="http://www.google.com">
<area shape="rect" coords="35,335,55,76" href="http://www.yahoo.com" >
</map>
```

سنشرح الآن الكود السابق :-

الخاصية **usemap** هي إحدى خصائص الوسم **img** وهي تأخذ اسم الخاصية **name** مع ملاحظة الإشارة **#** قبل الاسم , والوسم **map** هو المسئول عن الخرائط ونعطيه اسم عن طريق الخاصية **name** بدون الإشارة **#** (يجب أن تكون الخاصيتان **usemap** و **name** تحملان نفس الاسم) , الوسم **area** لتحديد المساحة ويأخذ الخاصية **shape** التي تحدد الشكل وعن طريق الخاصية **coords** نحدد إحداثيات الشكل المخصص من الصورة (من خلال برنامج الجرافيكس يمكن تحديد إحداثيات الصورة) , حيث إن الخاصية **shape** تأخذ أحد هذه القيم :-

1- **poly** شكل عشوائي (نحدد كل إحداثيات النقط المحيطة) .

2- **circle** شكل الدائرة (نحدد إحداثيات المركز وطول نصف القطر) .

3- **rect** شكل رباعي منتظم - مستطيل أو مربع - (نحدد إحداثيات الزاوية اليمنى العليا والزاوية اليسرى السفلى) .

الخاصية **href** تحدد موقع الوصلة التشعبيه إلى أين تؤدي , ونكتب الوسم **area** بعدد الأشكال التي نحتاجها .

ملاحظة / لا يجب كتابة الإحداثيات بالتسلسل وبالترتيب .

الفصل الخامس

(الجداول والقوائم)

الجدول

يمكن إنشاء جدول عن طريق الوسم **table** ذي وسمي البداية والنهاية , الأساس هنا هو الصف , بمعنى إنه يتم تحديد عدد الصفوف أولاً ثم تحديد عدد الخلايا في كل صف ومن ذلك ينتج لنا عدد الأعمدة .

أولاً يتم تحديد الصف باستخدام الأمر **tr** بوسمي البداية والنهاية له , ثم تحدد عدد الخلايا في كل صف بواسطة الأمر **td** بوسمي البداية والنهاية حيث يتم كتابتهما بين وسمي البداية والنهاية للأمر **tr** , وبذلك نكون قد حددنا عدد الأعمدة .

مثال / لتكوين جدول به صف واحد وعمودان :-

```
<body>
```

```
<table>
```

```
<tr> <td> الخلية الأولى </td> <td> الخلية الثانية </td> </tr>
```

```
</table>
```

```
</body>
```

خصائص الوسم **table** :-

1- الخاصية **border** وهي تحدد سمك الجدول وتأخذ أعداد صحيحة , والقيمة الافتراضية لها صفر .

2- الخاصية **align** وهي تستخدم لمحاذاة الجدول , وتأخذ القيم (**center , right , left**) .

3- الخاصية **height** وهي تستعمل لتحديد ارتفاع الجدول , وتأخذ أعداد طبيعية أو نسبة مئوية.

4- الخاصية **width** وهي تستعمل لتحديد عرض الجدول , وتأخذ أعداد طبيعية أو نسبة مئوية.

5- الخاصية **cellspacing** وهي تستعمل لتحديد المسافة كل خلية من خلايا الجدول , وتأخذ أعداد طبيعية .

6- الخاصية **cellpadding** وهي تستعمل لتحديد حجم الهوامش لخلايا الجدول , وتأخذ أعداد طبيعية .

7- الخاصية **background** وهي تستعمل لوضع خلفية للجدول كصورة ما .

8- الخاصية **bgcolor** وهي تستعمل لتلوين خلفية الجدول .

9- الخاصية **bordercolor** وهي تستعمل لتلوين إطار الجدول .

مثال:-

```
<table border="1" align="center" height="30%" width="20%" cellspacing="10" cellpadding="10" background="tree.jpg" bgcolor="red" bordercolor="black" >
```

خصائص الخلايا (**tr,td**) :-

* تكتب هذه الخصائص في الأسفل مع الوسم **tr** لتحديد خصائص الصف أو مع الوسم **td** لتحديد خصائص الخلية .

1- الخاصية **align** لتحديد محاذاة النص أفقياً داخل الخلية أو الصف , وتأخذ القيم (**center** , **right** , **left**) .

2- الخاصية **valign** لتحديد المحاذاة العمودية للنص داخل الخلية أو الصف , وتأخذ القيم (**middle** , **bottom** , **top**) .

3- الخاصية **bgcolor** لتحديد اللون داخل الخلية أو الصف .

4- الخاصية **background** لتحديد الخلفية كصورة مثلاً (**للخلية فقط td**) .

5- الخاصية **width** لتحديد عرض العمود أو الصف , وتأخذ أعداداً طبيعية أو نسبة مئوية .

6- الخاصية **height** لتحديد ارتفاع العمود أو الصف , وتأخذ أعداداً طبيعية أو نسبة مئوية .

7- الخاصية **colspan** تعمل على دمج الخلية الحالية مع العدد المطلوب من الخلايا التي تليها أفقياً .

```
<td colspan="n" >
```

حيث **n** تمثل عدد الخلايا المطلوب دمجها .

8- الخاصية **rowspan** تعمل على دمج الخلية الحالية مع العدد المطلوب من الخلايا التي تليها عمودياً (أي أسفلها) .

```
<td rowspan="n" >
```

حيث **n** تمثل العدد المطلوب دمجها .

* الوسم `<caption>` وهو يستعمل لوضع عنوان رئيسي للجدول ككل وتوضع مباشرة بعد الوسم `<table>` .

`</caption>` عنوان الجدول `<caption>`

* لطباعة خلية فارغة يجب أن نضع مكان النص وسم الفراغ

`<td> </td>`

* هناك بعض الوسوم تستخدم داخل وسم الجدول `<table>` وهي :-

1- الوسم `<thead>` `</thead>` وهو يستخدم لعمل رأس للجدول ولإنشاء الخلايا في هذا القسم نستخدم الوسم `<tr>` للأعمدة والوسم `<th>` للصفوف وهو عادةً يحتوي على صف واحد يحمل أسماء عناوين الأعمدة .

2- الوسم `<tbody>` `</tbody>` وهو يستخدم لعمل جسم الجدول .

3- الوسم `<tfoot>` `</tfoot>` وهو يستخدم لعمل ذيل للجدول .

ويمكن استخدام الخاصية `align` لمحاذاة النصوص داخل كل الخلايا الموجودة بين أي من هذه الأجزاء , وكذلك يمكن استخدام الخاصية `bgcolor` لتحديد لون النصوص .

جدول داخل جدول

مثال :-

```
<table>
```

```
<tr> <td> النص </td>
```

```
<td> <table>
```

```
<tr> <td> النص </td> </tr>
```

```
<tr> <td> النص </td> </tr>
```

```
</table> </td>
```

```
</tr>
```

```
</table>
```

القوائم

يوجد عدة أنواع من القوائم :-

أولاً :- القوائم المرقمة : بإمكانك إنشاء قائمة مرقمة ترتب من خلالها ما تشاء من العناصر الموجودة في صفحتك فما عليك إلا استعمال الوسم `` مع وضع الوسم `` قبل كل عنصر ثم نغلقه بالوسم `` , ويمتلك الوسم `` الخصائص التالية :-

1- الخاصية `type` يحدد نوع الترقيم (1 , A , i) .

2- الخاصية `start` القيمة الابتدائية لترتيب القائمة (الرقم الذي ستبدأ به القائمة) .

ومن خصائص الوسم `` :-

1- الخاصية `type` التي تحدد نوع الترقيم .

2- الخاصية `value` التي تحدد القيمة الابتدائية للعنصر (رقم العنصر في القائمة) .

وإليك هذا الجدول الذي يحدد قيم الخاصية `type` :-

نوع الترقيم (القيمة)	تسلسل الترقيم في القائمة
1	1 , 2 , 3 , 4
A	A , B , C
a	A , b , c
i	I , ii , iii , iv

مثال :-

نص (عنوان القائمة) ``

`` العنصر 1 ``

`` العنصر 2 ``

`</li value="5">` العنصر 3 ``

`</li type="a">` العنصر 4 ``

`</li type="i">` العنصر 5 ``

``

ثانياً :- القوائم غير المرقمة (المنقطة) :- هذا النوع من القوائم مثل سابقه إلا أننا نستعمل هذه المرة الوسم `` الذي له نفس خصائص الوسم `` , بالإضافة إلى وضع الوسم `` قبل كل عنصر ومن ثم إغلاقه بالوسم `` .

* مع ملاحظة إن الخاصية `type` تأخذ القيم (disc , circle , square) لتحديد شكل التنقيط .

ملاحظة / يمكن إنشاء قوائم متداخلة من `` و `` لإنتاج قوائم فرعية .

ثالثاً :- قوائم الشرح أو التعريفات :- وكما يدل الاسم تستخدم عندما نريد إدراج قائمة من المصطلحات يتبع كل واحدة منها شرح أو تعليق , حيث ستحتاج إلى ثلاثة وسوم هي 1- `<dl>` لتعريف بداية القائمة ونهاية القائمة نغلقها بالوسم `</dl>` . 2- `<dt>` يوضع قبل كل مصطلح لتحديده وبعدها نغلقه بالوسم `</dt>` . 3- `<dd>` وهو وسم الشرح أو التعليق وبعدها نغلقه بالوسم `</dd>` .

مثال:-

`<dl>`

الشرح `<dd>` المصطلح `<dt>`

الشرح `<dd>` المصطلح `<dt>`

`</dl>`

الفصل السادس

(الإطارات)

لعمل إطار نحتاج إلى الوسم **frameset** وهو يوضع بدلاً من الوسم **body** , مثال :-

```
<html>
<head> <title> my page </title> </head>
<frameset>
</frameset>
</html>
```

ولهذا الوسم عدة خصائص :-

1- الخاصية **cols** وهي تعرف عدد وأحجام الإطارات العمودية وهي تأخذ عدة قيم حيث يمكن أن توضع بشكل نسبة مئوية أو "*" حيث هذا يعني بأن هذا الجزء سيأخذ الجزء المتبقي من الشاشة .

```
<frameset cols="20%,30%,*" >
```

حيث هنا قمنا بتعريف ثلاثة إطارات .

2- الخاصية **rows** هي تحدد عدد وحجم الإطارات الأفقية (صفوف) ويكون ذلك بوضع قيمة نسبية مطلقة أو "*" لأخذ الجزء المتبقي من الشاشة .

```
<frameset rows="50%,30%,*,50" >
```

* الوسم **<frame />** هو وسم ذاتي الغلق أي ليس له وسم نهاية , وهو المسئول عن تعريف محتويات الإطار , ويستخدم هذا الوسم بعدد مرات الإطارات لأنه يستخدم كل مرة لتعريف محتويات إطار .

```
<frameset cols="50%,50%">
```

```
<frame src="page1.html" />
```

```
<frame src="page2.html" />
```

```
</frameset>
```

لعمل صفحة مقسمة إلى إطارات عمودية وأفقية يجب تعريف إطار داخل إطار هكذا

```
<frameset cols="50%,50%" >  
<frame src="page1.html" />  
<frameset cols="100,*">  
<frame src="page2" />  
<frame src="page3" />  
</frameset>  
</frameset>
```

من خصائص الوسم `<frame />` ما يلي :-

- 1- الخاصية `marginheight` تحدد مقدار المسافة الفارغة المتروكة للهوامش العلوية والسفلية للإطار (ببيكسل) .
- 2- الخاصية `marginwidth` تحدد مقدار المسافة الفارغة المتروكة للهوامش اليمنى واليسرى للإطار (ببيكسل) .
- 3- الخاصية `scrolling` تحدد إمكانية ظهور أو عدم ظهور أشرطة التصفح الأفقية والعمودية على جوانب أو أسفل الإطار , وتأخذ القيمة `yes` للظهور والقيمة `no` لعدم الظهور والقيمة `auto` التي تحدد ظهور الأشرطة أو عدم ظهورها تلقائياً بحسب الحاجة إليها .
- 4- الخاصية `noresize` عند إضافة هذه الخاصية يتم منع عملية التحكم بحجم الإطار بالتصغير أو التكبير من خلال السحب والإفلات وهي تأخذ القيمة `" noresize "` .
- 5- الخاصية `src` وهي تحدد عنوان الصفحة.

```
<frame src="index.html" marginheight="40" marginwidth="30" scrolling="auto"  
noresize="noresize" />
```

* الوسم <noframes> :-

بعض المتصفحات لا تستطيع رؤية الإطارات ولكي تتيح لهم رؤية موقعك نضيف هذا الوسم في نهاية الصفحة , لاحظ هذا المثال :-

```
<html>
<head> <title> my page </title> </head>
<frameset rows="68,*" >
<frame src="page1" />
<frameset cols="150,*" >
<frame src="page2" />
<frame src="page3" name="MAIN" >
</frameset>
<noframes>
<body>
<p> ما سيظهر للمتصفحات التي لا ترى الإطارات </p>
</body>
</noframes>
</frameset>
</html>
```

وكما لاحظت في المثال السابق استخدمنا الخاصية **name** , حيث الآن نستطيع استدعاء (استخدام وصلة تشعبية) للاسم **name** (MAIN - ويمكن اختيار أي اسم آخر) من مكان آخر (صفحة أخرى) هكذا وذلك باستخدام الخاصية **target** وهي تأخذ عدة قيم منها (- **parent** , **-top** , **-self** , **-blank**)

```
<html>
<head> <title> my page </title> </head>
<frameset rows="68,*" >
<frame src="pag1.html" target="MAIN" />
<frameset cols="150,*" >
<frame src="pag2.html" target="MAIN" />
<frame src="pag3.html" target="MAIN" />
</frameset>
<noframes>
<body>
<p> ما سيظهر للمتصفحات التي لا ترى الإطارات </p>
</body>
</noframes>
</frameset>
</html>
```

ملاحظة / هناك بعض المتصفحات لا تدعم الإطارات ومن المحتمل أن يتم إلغاء الإطارات في الإصدارات الجديدة للغة البرمجة html .

الفصل السابع

(النماذج)

النماذج

وهي طريقة بسيطة جداً تسمح لزوار موقعك أن يرسلوا لك معلومات بسيطة , سواء لأغراض الرد , الاشتراك أو معلومات الشراء والعديد من الاستعمالات الأخرى .

إن الوسوم الأساسية للنماذج والتي يجب أن تدرج جميع الوسوم الأخرى المتعلقة بالنماذج داخل هذين الوسمين `</form>` `<form>` وهذا الوسم يحتوي على بعض الخصائص :-

action -1

تحدد العنوان الذي سيتم إرسال البيانات إليه لتتم معالجتها بالصورة المطلوبة , وعادةً يكون هذا عنواناً لبريد الكتروني الذي سيتم إرسال البيانات إليه , أو قد يكون عنوان لبرنامج CGI موجود على كمبيوتر الخادم server الذي تتواجد عليه صفحات الويب , حيث يستقبل هذه البيانات ويعالجها حسب التعليمات الموجودة فيه كأن يضيفها مثلاً لإحدى الصفحات (كما يحدث عادةً في دفاتر الزوار) ويتحقق من صحة بعض الحقول المدخلة ومطابقتها لمعايير معينة أو أن يقوم بالبحث عن كلمة أو عبارة ضمن صفحات الموقع كما في نماذج البحث الموجودة في موقع الويب .

```
<form action="mailto:some one@domain.com" > ..... </form>
```

```
<form action="name and address of CGI script" > ..... </form>
```

method -2

تحدد الطريقة التي سيتم بها التعامل مع العنوان المحدد في الخاصية السابقة action وهناك قيمتين لهذه الخاصية هما (get) التي تستخدم في حالة كون عملية المعالجة داخلية أي تتم داخل الخادم server نفسه , كما في استخدام نموذج البحث عن كلمة في موقع فإن عملية المعالجة (البحث) تجري مباشرة في الموقع , أما القيمة الثانية (post) وتستخدم عندما تكون عملية المعالجة خارجية كأن يتم إرسال البيانات إلى عنوان بريد الكتروني .

```
<form action="mailto:some one@domain.com" method="post" > ..... </form>
```

```
<form action="name and address of CGI script" method="get" > ..... </form>
```

type -3

هذه الخاصية تحدد طريقة الترميز التي سيتم إرسال البيانات وفقاً لها وهي تأخذ القيمتين التاليتين (يجب أن تكتب هذه القيم كما هي نصاً وحرفاً)

1- application تصل البيانات بشكل غير مرتب .

2- text تصل البيانات بشكل مرتب .

autocomplete -4

هذه الخاصية جديدة في html5 , وهذه الخاصية تستخدم لتفعيل عملية التكملة التلقائية للكلمة التي يدخلها المستخدم في حقل الإدخال حيث أننا نسمح لها القيمة (on) لتفعيلها والقيمة (off) لإلغاء تفعيلها .

novalidate -5

وهذه الخاصية تستخدم لكي لا يتم التحقق من عنصر الإدخال في صفحة المعلومات عند إرسال معلومات النموذج ونسمح لها القيمة (true) , مع ملاحظة أن هذه الخاصية لا تدعم جميع المتصفحات .

مثال/

```
<form autocomplete="on" novalidate="true" >
```

وهناك بعض الوسوم التي تستخدم بالإجراءات منها :-

أولاً :- الوسم <input />

وهو المسئول عن عملية الإدخال , وهو وسم ذاتي الغلق أي ليس له وسم نهاية , ويحتوي على مجموعة من الخصائص :-

1- type وهذه الخاصية تحدد نوع الشكل الذي تريده وهي تأخذ مجموعة قيم (text , password , hidden , radio , file , checkbox , submit , reset , button) .

2- name وتستخدم لتسمية حقل البيانات (أي اسم) وهذا الاسم يعرف الحقل في داخل النموذج نفسه , بحيث يمكن استخدامه فيما بعد للحاجات البرمجية وضرورات المعالجة إن وجدت من قبل البرامج التي تضيفها كمصمم للموقع , وحتى عندما تريد أن يرسل النموذج إليك بالبريد فأن حقله تعرف بالاسم الذي أدرجته لها من خلال هذه الخاصية .

3- **value** وهي تستخدم لكتابة نص وظهوره كمثال داخل الشكل .

مثال :-

```
<body>
```

```
<form>
```

```
Enter your email :<input type="text" name="email" value="example@yahoo.com" />
```

```
</form>
```

```
</body>
```

لتكون النتيجة هكذا :-

please enter your email :

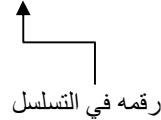
4- **size** وهي تستخدم لتحديد حجم الحقل وليس لها علاقة بكمية (حجم) الإدخال .

5- **maxlength** وهي تستخدم للتحكم بالحد الأقصى للنص المدخل .

```
<input size="40" maxlength="30" />
```

6- **tabindex** وهي تستخدم عندما يكون لديك أكثر من مربع إدخال وتريد تحديد أي من هذه المربعات ينتقل إليه المؤشر عند الضغط على مفتاح tab من لوحة المفاتيح .

```
<input type="text" name="tab1" tabindex="1" />
```



7- **accesskey** وهي تمكنك من الانتقال إلى مكان معين عند الضغط على Alt من لوحة المفاتيح وحرف يتم تحديده وذلك للاختصار .

8- **id** وهي تستخدم لإعطائه اسم حيث يمكننا الاستفادة منه فيما بعد .

الآن سوف أشرح لكم قيم الخاصية **type** :-

1- **text** ليتم إدخال قيمة نصية .

2- **password** ليتم إدخال رمز ويكون بشكل * عند الإدخال للمحافظة على سرية الإدخال .

3- **hidden** أي الحقل المخفي , وهو لن يظهر ضمن النموذج وفائدته هي لكي يعرفنا من أين هذه المعلومات (إذا كنا نستخدم عدة صفحات ومعلوماتها متشابهة) خذ هذا المثال :-

```
<input type="hidden" name="my forms" value="form1" />
```

لتصل البيانات بهذا الشكل :-

المعلومات my form = form1

المعلومات my form = form2

* من أجل إظهار النموذج بشكل مرتب ومنسق يتم وضعه داخل جدول بدون حدود (إطار) .

4- **radio** وهو يأخذ شكل ومن مسوغات استخدام هذا الشكل إن السؤال المطروح ينبغي أن يكون له إجابة واحدة (أي يستطيع الزائر اختيار إجابة واحدة فقط) .

كمثال / إن أردت سؤال الزائر عن اللون المفضل :-

الأسود الأخضر

```
<form>
```

```
<input name="color" type="radio" value="green" /> الأخضر
```

```
<input name="color" type="radio" value="black" /> الأسود
```

```
</form>
```

وستصل البيانات هكذا في حال اختيار الأسود :-

color = black

ملاحظة // يجب أن نعطي كل حقول (radio) الموجودة في نفس نموذج الإدخال نفس الاسم وذلك لكي يفهم معالج اللغة أن هذه الحقول كلها تعود لنفس السؤال وبذلك يجعل الزائر يختار خيار واحد فقط أما إذا لم نعطيها أسماء او كانت الأسماء مختلفة سيتمكن الزائر يحدد كل الخيارات الموجودة لأنه سيعتقد أنها تعود لنماذج إدخال مختلفة او لأسئلة مختلفة .

* وإذا أردنا أن يكون هنا اختيار افتراضي نضيف الخاصية **checked** إلى الوسم **<input>** وهي تأخذ القيمة " **checked** " .

5- **checkbox** وهو يكون بهذا الشكل وفرقه عن النوع radio بأن radio يمكن أن نختار منها خيار واحد أما هذه الخاصية تسمح باختيار أكثر من خيار .

مثال/ عندما تريد من المستخدم أن يختار البرامج المفضلة لديه :-

All Ac Dsee photoshop

`<input type="checkbox" name="ps" value="yes" /> photoshop`

`<input type="checkbox" name="acd" value="yes" checked="checked" /> Ac Dsee`

`<input type="checkbox" name="all" value="yes" /> All`

وتصل النتائج بهذا الشكل (عند اختيار all و photoshop)

all = yes

ps = yes

6- **submit** وهي ستقوم مباشرة بإنشاء زر سيعمل عند النقر عليه على إرسال البيانات التي تم ملئها في النموذج , ولتغيير الاسم الموجود فوق الزر نكتب الاسم الجديد في الخاصية value هكذا :-

`<input type="submit" value=" اسم الزر الجديد " />`

ولتعيين صورة للزر نستخدم هذا الكود :-

`<input type="image" src="امتدادها.موقعها" width="20" height="40" border="0"`

`alt=" النص الذي سيظهر لإرشاد المستخدم" />`

7- **reset** وهي ستقوم مباشرة بإنشاء زر سيعمل عند النقر عليه على إلغاء الأمر , ولتغيير الاسم الموجود فوق الزر نكتب الاسم الجديد في الخاصية value هكذا :-

`<input type="reset" value=" اسم الزر الجديد " />`

8- **file** وهو يقوم بإنشاء زر يمكننا من استعراض الملفات داخل الحاسوب , مثال /

`<input type="file" />`

9- **image** وهو ينشأ زر يكون عبارة عن صورة حيث سنستخدم معه الخاصية src التي ستحمل اسم وامتداد الصورة , وهذا يشبه إلى حد ما الخاصية **submit** .

`<input type="image" src=" اسم وامتداد الصورة " />`

10- **button** وهو يستخدم لملئ البيانات مع البرامج الأخرى مثل Java script .

* يوجد أيضاً وسم يسمى **<button>** حيث يعمل على إرسالنا إلى صفحة جديدة

```
<forme>
<button> اضغط هنا </button>
</form>
```

* هناك مجموعة جديدة من عناصر الإدخال في html5 (قيم للخاصية type) في الوسم `<input />` يعرضها الجدول التالي :-

الشرح	قيم الخاصية type
عنصر خاص لإدخال البريد الإلكتروني	email
عنصر خاص لإدخال الروابط التشعبية	url
عنصر خاص لإدخال الأرقام	number
عنصر خاص لإدخال قيم محصورة بمجال معين سلفاً , ويظهر هذا العنصر على شكل شريط تمرير	range
عنصر لإدخال التاريخ (يوم / شهر / سنة)	date
عنصر لإدخال التاريخ (شهر / سنة)	month
عنصر لإدخال التاريخ ويستخدم لتحديد اسبوع من السنة	week
عنصر لإدخال الوقت (ساعة / دقيقة)	time
عنصر لإدخال التاريخ والوقت (ساعة / يوم / شهر / سنة) بالتوقيت العالمي	datetime
عنصر لإدخال التاريخ والوقت بالتوقيت المحلي	datetime-locale
عنصر لعرض مربع البحث وهو شبيه بالعنصر text	search
عنصر لإدخال القيم اللونية	color

مثال /

```
<form method="get">
<input type="date" name="d" />
<input type="range" name="points" min="1" max="1" />
<input type="submit" />
</form>
```

الخصائص الجديدة في html5 للوسم <input>

ملاحظة / هذه الخصائص لا تدعمها جميع المتصفحات ما عدا هذه الخواص وهي (height , width , autocomplete) حيث تكون مدعومة من جميع المتصفحات .

1- autocomplete

وهي تستخدم لتفعيل عملية التكملة التلقائية للنص الذي يدخله المستخدم في هذا الحقل وهي تستخدم مع الحقول من النوع التالي (text , tel , color , datepickery , rang , password , email , url , search) حيث نسند لها القيمة (on) لتفعيلها والقيمة (off) لإلغاء تفعيلها .

مثال/

```
<input type="text" autocomplete="off" />
```

2- autofocus

تحدد هذه الخاصية بأن مؤشر الكتابة يعين في الحقل مباشرةً بمجرد تحميل الصفحة , وتعمل هذه الخاصية مع جميع أنواع **input** حيث نسند لها القيمة " autofocus " .

مثال/

```
<input type="text" autofocus="autofocus" />
```

3- form

هذه الخاصية تحدد النموذج الذي يتبع له عنصر الإدخال **input** ويمكن أن ينتمي عنصر الإدخال مع أكثر من نموذج وذلك بسرد قيمة الخاصية id للنموذج مع فراغ بين كل قيمة في حال أن عنصر الإدخال يمكن أن ينتمي لأكثر من نموذج , وتطبق الخاصية **form** على كل أنواع العنصر **input** .

مثال/

```
<form id="user_form">
```

```
<input type="text" />
```

```
</form>
```

```
<input type="text" form="user_form" />
```

4- خصائص **input** لتجاوز خصائص الوسم **<form>**

حيث ان هناك مجموعة من الخصائص تمكننا من تجاوز خصائص الوسم **<form>** وهذه الخصائص تعمل فقط مع العنصر **input** للنوع (image , submit) :-

أ- **formaction** :- تسمح بتجاوز الخاصية **action** للوسم **<form>** بمعنى يمكننا إنشاء زر ينزل المعلومات لصفحة معينة حيث نسند لهذه الخاصية اسم الصفحة الثانية التي سنرسل لها البيانات في حالة الضغط على هذا الزر . مثال/

```
<form action="page1.php" >
<input type="email" />
<input type="submit" />
<input type="submit" formaction="page2.php" />
</form>
```

ب- **formenctype** :- تسمح بتجاوز الخاصية للوسم **<form>**

ت- **formmethod** :- تسمح بتجاوز الخاصية **method** للوسم **<form>**

ث- **formnovalidate** :- تسمح بتجاوز الخاصية **novalidate** للوسم **<form>**

ح- **formtarget** :- تسمح بتجاوز الخاصية **target** للوسم **<form>**

حيث كل هذه الخصائص تساعدنا على إنشاء أكثر من زر لإرسال معلومات النموذج .

مثال/

```
<form method="get" id="user_form" >
<input type="email" />
<input type="submit" />
<input type="submit" formvalidate="true" />
</form>
```

width -5

تستخدم هذه الخاصية لتحديد عرض الصورة التي تستخدم عند استخدام النوع **image** للوسم **input** فقط .

height -6

تستخدم هذه الخاصية لتحديد ارتفاع الصورة التي تستخدم عند استخدام النوع **image** للوسم **input** فقط .

مثال/

```
<input type="image" src="name.gif" width="24" height="24" />
```

list -7

هذه الخاصية تحدد الوسم `<datalist>` لحقل الإدخال حيث تعمل هذه الخاصية مع أنواع الوسم `input` التالية (`datapickers` , `range` , `email` , `tel` , `url` , `search` , `text` , `color` , `number`) . لا حظ هذا المثال/

```
webpage <input type="url" list="url_list" />
<datalist id="url_list">
<option label="google" value="http://google.com" />
</datalist>
```

`min` -8

تستخدم هذه الخاصية لفرض قيود على حقل الإدخال الرقمي او حقل التاريخ وتحدد هذه الخاصية اقل حد مسموح به للإدخال , وتعمل هذه الخاصية مع أنواع الوسم `input` التالية (`datapickers` , `range` , `number`) .

`max` -9

تستخدم هذه الخاصية لتحديد اكبر قيمة مسموح بإدخالها في حقل الإدخال وتستخدم مع أنواع الوسم `input` التالية (`datapickers` , `range` , `number`) .

`step` -10

تستخدم لفرض قيود على حقول الإدخال الرقمية او قيم التاريخ وتحدد هذه الخاصية الخطوة بين القيم المسموح بإدخالها , وتعمل هذه الخاصية مع أنواع الوسم `input` التالية (`number` , `datapickers` , `range`)

مثال/

```
<input type="number" min="0" max="10" step="3" />
```

`multiple` -11

تستخدم هذه الخاصية بأنه يمكن تحديد عدد قيم لعنصر الإدخال وتعمل هذه الخاصية على الأنواع التالية للوسم `input` وهي (`email` , `file`)

`novalidate` -12

وهي تستخدم لتحديد بأن عنصر الإدخال هذا لا يجب التحقق من صحة المعلومات المدخلة فيه عندما يتم إرسالها , وهي تعمل مع الوسم `input` للأنواع التالية (`email` , `tel` , `url` , `text` , `search` , `color` , `range` , `datapickers` , `password`) .

`pattern` -13

تحدد هذه الخاصية النموذج (تعبير نظامي) التي يجب أن يطبق على مدخلات عنصر الإدخال للتحقق من صحتها وتعمل هذه الخاصية على الأنواع التالية للوسم **input** وهي (**url** , **tel** , **email** , **text** , **search** , **password**) .

مثال/

```
<input type="text" pattern="[A-z]{3}" />
```

placeholder -14

هي عبارة عن قيمه نصيه تظهر في حقل الإدخال كمساعدة للقيمة المتوقع إدخالها , وتعمل هذه الخاصية على الأنواع التالية للوسم **input** (**email** , **tel** , **url** , **password** , **text** , **search**) ويظهر هذا النص عندما يكون حقل الإدخال فارغ وتختفي عند كتابة قيمه في الحقل .

مثال/

```
<input type="text" placeholder=" الاسم " />
```

required -15

تحدد هذه الخاصية إن حقل الإدخال يجب تعبئته بقيمه قبل إرسال المعلومات في النموذج , وتعمل هذه الخاصية على الأنواع التالية للوسم **input** وهي (**number** , **datapickers** , **password** , **email** , **tel** , **url** , **text** , **search** , **file** , **radio** , **checkbox**) .

مثال/

```
<input type="text" required="required" />
```

ثانياً- الوسم <select>

وهو الوسم الثاني الذي يستخدم في النماذج **<form>** , وهو نوع من أشكال الخيارات من خيارات متعددة :- ولهذا الشكل نستخدم الوسم **<select>** بدلاً من **<input>** , ولتحديد كل خيار نستخدم الوسم **<optino>** ثم نغلقه بالوسم **</option>** .

مثال/

```
<form>
```

```
<select>
```

```
<option> example1 </option>
<option> example2 </option>
</select>
</form>
```

* من خصائص الوسم **<select>** :-

- 1- الخاصية **name** وهي تحدد اسم القائمة .
- 2- الخاصية **size** وهي تحدد ارتفاع الحقل (القائمة) وبالتالي عدد البيانات الظاهرة وهي تأخذ أي قيمة عددية صحيحة .
- 3- الخاصية **multiple** وهي تتيح اختيار أكثر من خيار واحد وهي لا تأخذ أي قيمة بل تكتب وحدها , ولأداء عدة خيارات يجب الضغط على مفتاح ctrl من لوحة المفاتيح بصورة مستمرة (هذا يخص المستخدم الزائر) .

* من خصائص الوسم **<option>** :-

- 1- الخاصية **value** لإعطاء قيمة لكل حقل بيانات من القائمة .
 - 2- الخاصية **selected** نكتبها مع الخيار الذي نريده أن يكون محدد افتراضياً , وهي تأخذ القيمة " **selected** "
- * يمكن تقسيم قائمة الخيارات إلى أقسام متعددة وذلك باستخدام الوسم **<optgroup>** ونكتب اسم هذا القسم في الخاصية **label**
- مثال/

```
<form>
<select>
<optgroup label="الدول العربية">
<option> مصر </option>
<option> العراق </option>
</optgroup>
<optgroup label="الدول الأوروبية">
<option> ألمانيا </option>
<option> فرنسا </option>
```

</optgroup>

</select>

</form>

ثالثاً- الوسم textarea

وهو يستخدم عادةً لكتابة التعليقات الحرة مع النموذج ويتم إدراجه بكتابة الوسم **<textarea>** , وأي نص يتم كتابته بين هذين الوسمين سيتم عرضه بصورة تلقائية , وهو يحتوي على مجموعة خصائص :-

1- الخاصية **name** لإعطاء اسم التعريف .

2- الخاصية **rows** تحدد العرض .

3- الخاصية **cols** تحديد الارتفاع .

4- الخاصية **wrap** تحدد طريقة التفاف النص المكتوب داخل الحقل , وهي تأخذ احد هذه القيم الثلاثة :-

أ- **virtual** تعني إن النص سيلتف على عدة أسطر عند كتابته , ولكنه سيصلك عند إرساله على شكل سطر واحد متتابع .

ب- **physical** تعني أن النص سيلتف على عدة أسطر , وسيصلك أيضاً على هذا النحو .

ج- **off** تعني أن النص لن يلتف بصورة تلقائية على عدة أسطر , لكنه على أي حال سيصلك بنفس الشكل الذي يكتب به .

مثال/

```
<textarea name="comments" cols="30" rows="6" wrap="off" >
```

النص الافتراضي

```
</textarea>
```

القوائم المنسدلة

وهي خاصية جديدة في html5 , يستخدم الوسم **<datalist>** لإنشاء القوائم المنسدلة كخيارات إدخال افتراضية لأحد عناصر الإدخال الموجودة وينشأ محتوى القائمة بالتعاون مع الوسم **<option>** مع ملاحظة انه ذاتي الغلق أي ليس له وسم نهاية , وبطريقة مشابهة للعنصر القديم

<select> كما يتم تمرير اسم قائمة الخيارات الافتراضية (عنصر <datalist>) لعنصر الإدخال عبر الخاصية الجديدة list .

مثال /

```
<form method="get">
<input type="text" list="name_list" name="std" />
<datalist id="name_list">
<option label="A" value="nameA" />
<option label="B" value="nameB" />
<option label="C" value="nameC" />
</datalist>
</form>
```

تقسيم عناصر الإدخال (النماذج) إلى مجموعات

يمكن تقسيم عناصر الإدخال في النماذج الكبيرة إلى أقسام منطقية يوضع كل منها في إطار مستقل وهذا ما توفره لغة html باستخدام الوسمين <fieldset> </fieldset> اللذان سيحويان بينهما مجموعة عناصر الإدخال , ويتم تزويد كل مجموعة بعنوان توضيحي عبر الوسم <legend> .

مثال /

```
<form method="get" >
<fieldset>
<legend> البيانات الشخصية </legend>
<input type="text" />
<input type="submit" />
</fieldset> </form>
```

* يمكن منح كل عنصر من عناصر الإدخال عنواناً يدل على ماهية المعلومات المدخلة فيه عن طريق الوسم <label> بتمرير اسم حقل عنصر الإدخال إلى الخاصية for الخاصة بعنصر

العنوان <label> ومعنى ذلك انه لو ضغطنا على هذا العنوان سينقلنا مباشرة إلى هذا الحقل ,
مثال /

```
<label for="ss"> الاسم </label>
```

```
<input type="text" name="name" id="ss" />
```

الوسمين Keygen و Output

هذان الوسمان جديان في لغة html5 , يستخدم الوسم <keygen> في عمليات تسجيل الدخول والتحقق من المستخدمين وذلك عبر توليد زوج مفاتيح (مفتاح عمومي / مفتاح خصوصي) يرسل مع الطلب بدون تدخل المستخدم .

مثال/

```
<keygen name="security" />
```

أما الوسم <output> فيستخدم لعرض مخرجات من أنواع مختلفة مثل نتائج الحسابات او القيم المعادة من احد توابع javascript .

مثال /

```
<output id="result" onforminput="rescalc()" > </output>
```

الفصل الثامن

(الرسم)

عنصر المرسم Canvas

وهو عنصر جديد في html5 , يمكن الرسم بواسطة هذا الوسم `<canvas>` وبمساعدة من لغة javascript , لاستخدام عنصر المرسم يتم إنشاؤه أولاً بارتفاع `height` وعرض `width` محددين ثم يتم الرسم عليه باستخدام javascript .

مثال/

```
<body>
<canvas id="mycanvas" width="200" height="100" >
/* تظهر فقط في المستعرضات التي لا تدعم هذا العنصر */ لا يدعم مستعرضك عنصر المرسم !
</canvas>
<script type="text/javascript" >
1- var c=document.getElementById ("mycanvas");
2- var cxt=c.getContext ("2d");
3- cxt.fillStyle="#ff0000";
4- cxt.fillRect (0,0,150,75);
</script>
</body>
```

شرح المثال بحسب الترقيم الموجود :-

- 1- قمنا في هذا السطر بتحديد عنصر المرسم عن طريق الخاصية id الخاصة به .
- 2- في هذا السطر أنشأنا كائن Object من المرسم وهيئناه للرسم ثنائي الأبعاد (2d) .
- 3- 4- في هذين السطرين اللذان هما يمثلان طرق او خصائص الكائن cxt وسيتم شرح الخصائص الأخرى لهذا الكائن في الجدول بالأسفل .
- 4- في هذا السطر وضعنا إحداثيات المستطيل وطريقة تحديدها مثل طريقة التحديد في الخرائط الصورية .

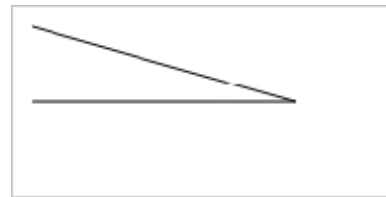
الجدول التالي يعرض بعض خصائص الكائن `cxt` (كائن الرسم)

المثال	الوصيفة	الطريقة
<code>cxt.fillStyle="#FF0000";</code>	تحديد لون التعبئة	<code>fillStyle</code>
<code>cxt.fillRect(0,0,150,75);</code>	رسم مستطيل	<code>fillRect</code>
<code>cxt.moveTo(10,10);</code>	إزاحة مبدأ الإحداثيات	<code>moveTo</code>
<code>cxt.createLinearGradient(0,0,170,50)</code>	إنشاء تدرج لوني	<code>createLinearGradient</code>
<code>cxt.addColorStop(0,"#FF0000");</code>	إضافة لون إلى التدرج اللوني	<code>addColorStop</code>
<code>cxt.drawImage(img,0,0);</code>	لرسم صورة	<code>drawImage</code>
<code>cxt.lineTo(150,50);</code>	لرسم خط مستقيم	<code>lineTo</code>

امثلة عن الرسم على canvas

مثال 1/ رسم خط وذلك بتحديد بداية ونهاية الخط

```
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px solid
#c3c3c3;">
Your browser does not support the canvas element.
</canvas>
<script type="text/javascript">
var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
cxt.moveTo(10,10);
cxt.lineTo(150,50);
cxt.lineTo(10,50);
cxt.stroke();
</script>
</body>
```



مثال 2/ رسم دائرة بتحديد حجمها لونها وموقعها ضمن العنصر canvas

```
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px solid
#c3c3c3;">
Your browser does not support the canvas element.
</canvas>
<script type="text/javascript">
var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
cxt.fillStyle="#FF0000";
cxt.beginPath();
cxt.arc(70,18,15,0,Math.PI*2,true);
cxt.closePath();
cxt.fill();
</script>
</body>
```



مثال 3/ رسم مستطيل متدرج اللون بتحديد الألوان التي ستستخدم وأبعاد هذا المستطيل

```
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px solid
#c3c3c3;">
Your browser does not support the canvas element.
</canvas>
<script type="text/javascript">
var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
var grd=cxt.createLinearGradient(0,0,175,50);
grd.addColorStop(0,"#FF0000");
grd.addColorStop(1,"#00FF00");
cxt.fillStyle=grd;
cxt.fillRect(0,0,175,50);

</script>
</body>
```

مثال 4/ وضع صورة فوق العنصر

```
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px solid
#c3c3c3;">
Your browser does not support the canvas element.
</canvas>
<script type="text/javascript">
var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
var img=new Image()
img.src="img_flwr.png"
cxt.drawImage(img,0,0);
</script>
</body>
```



الفصل التاسع

(XHTML)

xhtml :- هي لا تختلف عن html التقليدية من ناحية الوسوم ولا من ناحية الخصائص (الوصفات) , لكن ببساطة يمكن اعتبارها مواصفات أكثر صرامة لكتابة مستندات html بشكل يتوافق وقواعد xml وهذا ما يعطي توافقية أكبر للصفحة مع مستعرضات الويب .

قواعد xhtml

نستطيع وضعها بعدة نقاط :-

- 1- يجب وضع وسم الإغلاق , وترك فاصلة بين الوسم والمحرّف / في الوسوم ذاتية الغلق , مثلاً `
` .
- 2- يجب كتابة الوسوم والخصائص بحروف صغيرة .
- 3- يجب مراعاة ترتيب وسوم الإغلاق مع وسوم الفتح .
- 4- وضع القيم المسنة إلى الخصائص بين علامات اقتباس .
- 5- عدم الاستغناء عن الوسوم الرئيسية `<html>` و `<head>` و `<body>` .
- 6- وضع تعريف نوع المستند DOCTYPE في بداية الصفحة .

تعريف نوع المستند DOCTYPE

تنص معايير XHTML على أن كل صفحات الويب يجب أن تحتوي في أول سطر منها على سطر التعريف `<!DOCTYPE>` والذي يعني تعريف نوع المستند وذلك لأن مستعرضات الويب تستخدم هذا السطر كمرجعية قواعدية لصيغة الكتابة ولأنواع الوسوم في الصفحة , إضافة إلى استخدام الخاصية `xmlns` الخاصة بوسم البداية `<html>` والتي تصف فضاء أسماء الصفحة .

مثال / يمثل الشكل العام لمستند XHTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 transitional//EN"  
"http://www.w3.org/TR/xhtml1-transitional.dtd">  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
  
<head> ..... </head>  
  
<body> ..... </body>  
  
</html>
```

* إذا أردنا أن نعرف إذا كانت صفحة ما مطابقة للمعايير القياسية (XHTML) فيجب أن نذهب إلى الموقع الرسمي وهو (validator.w3.org) ونكتب داخل هذا الموقع رابط عنوان الصفحة .

الفصل العاشر

(ملاحظات عامة)

أولاً- هناك بعض المحارف والرموز الخاصة التي لا يمكن كتابتها بشكل مباشر إلا من خلال شفرة خاصة لاحظ الجدول في الأسفل :-

الرمز	شفرة xhtml	الرمز	شفرة xhtml	الرمز	شفرة xhtml
<	<	™	™	&	&
>	>	©	©	محرّف الفراغ	
≠	≠	®	®		

* هناك رموز أكثر موجودة على شكل صورة محفوظة مع مجلد الكتاب .

* في بعض الأحيان تكتب كود الرمز بشكل صحيح لكنك لا ترى الرمز يظهر في صفحتك وهذا يرجع إلى انه توجد خطوط لا تشتمل على اكواد الرموز لهذا لا بد أن تغير الخط إلى خط يشمل اكواد هذه الرموز او تستخدم الخط العام Arial

ثانياً- لتضمين صفحة ويب في صفحتنا بشكل متناسق مع بقية محتوى الصفحة نستخدم الوسم `<iframe src="عنوان الصفحة"> </iframe>` والصيغة العامة له هي :-

الوسم `<iframe>` له عدة خصائص :-

- 1- الخاصية `src` حيث نضع عنوان الصفحة مع الامتداد كقيمة لهذه الخاصية .
- 2- الخاصية `height` للتحكم بطول الإطار حيث نسند لها قيمة رقمية بالبيكسل .
- 3- الخاصية `width` للتحكم بعرض الإطار حيث نسند لها قيمة رقمية بالبيكسل .
- 4- الخاصية `frameborder` للتحكم بظهور او عدم ظهور الإطار حيث نسند لها القيمة صفر (0) لعدم ظهور الإطار والقيمة (1) لظهور الإطار .
- 5- الخاصية `scrolling` وهي تستخدم للتحكم بظهور او عدم ظهور مقبض الحركة في الإطار حيث نسند لها القيمة (no) لإخفاء مقبض الحركة في الإطار , وفي الحالة الافتراضية سيظهر مقبض الحركة في حال كانت أبعاد الإطار اقل من حجم الصفحة ويختفي إذا كانت اكبر من حجم الصفحة .

ثالثاً- الوسم <meta />

يكتب الوسم <meta /> داخل الوسم <head> , لعناصر الوسم <meta /> فوائد عديدة حيث من خلال هذا الوسم سيتم إرشاد محركات البحث إلى موقعك وإظهار الموقع في الصفحات الأولى من البحث , كما أن للوسم <meta /> أهمية في جعل النص العربي يعرض بشكل صحيح لان بعض المتصفحات لا تعرض الكتابة باللغة العربية إلا إذا عرفناها في الوسم <meta /> , وللوسم <meta /> عدة خواص من أهمها :-

1- name -2 content

حيث إن هاتان الخاصيتان مرتبطتان مع بعض 1- إذا أعطينا للخاصية name القيمة " keywords " فإن الخاصية content ستكون قيمتها الكلمات المفتاحية التي ترشد محركات البحث إلى موقعنا ونفصل بين كل كلمة وأخرى بفاصل , 2- إذا أعطينا للخاصية name القيمة " description " فإن الخاصية content ستكون قيمتها وصف للموقع وهذا الوصف يوضح تفاصيل الموقع لمحركات البحث , 3- إذا أعطينا للخاصية name القيمة " author " فإن الخاصية content ستكون قيمتها اسم مؤلف الموقع , 4- إذا أعطينا للخاصية name القيمة " copyright " فإن الخاصية content ستكون قيمتها عبارة عن سرد لحقوق النشر الخاصة بالصفحة .

مثال /

```
<head>
```

```
<meta name="keywords" content="book , read , pen" />
```

```
<meta name="description" content="this page demonstrates how to use meta elements" />
```

```
<meta name="author" content="Ahmed" />
```

```
</head>
```

* وللاستفادة من وسم <meta /> في تعريف الصفحة نستخدم الخاصية http-equiv بدلا من الخاصية name حيث سنسند لها القيمة " content-type " وبهذه الحالة ستكون قيمة الخاصية content هي :-

" text/html; charset=utf-8 " حيث إن (utf-8) تشير إلى الترميز ويعني أن الكتابة باللغة العربية او يمكن أن نضع بدلا من (utf-8) (windows-1256) وهو أيضا يشير إلى اللغة العربية

مثال /

```
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

* يمكن من خلال هذا الوسم <meta /> عمل تحديث للصفحة ومن ثم الانتقال إلى صفحة نحددها نحن خلال فترة معينة , وذلك كما في هذا المثال :-

```
<meta http-equiv="Refresh" content="1; url=pageName.php" />
```

حيث أن هنا الرقم 1 يعني ثانية واحدة ويمكن وضع أي وقت نريده مقدر بالثانية .

رابعاً- عندما نكتب في لغة html وسم بداية و وسم نهاية بالإضافة إلى المحتوى الموجود بين هذين الوسمين يسمى كل هذا بالعنصر حيث يمكننا أن نسمي أي عنصر من خلال الخاصية (id) (id="NAME") , وهذا الكلام ينطبق على كل الأوسمة .

خامساً- يمكننا أن نحدد نوع اللغة المستخدمة في الصفحة وذلك من خلال الخاصية lang التي نكتبها مع وسم البداية <html> حيث نسند لها القيمة (en) للغة الانجليزية والقيمة (ar) للغة العربية .

```
<html lang="ar">
```

سادساً- أحيانا عند كتابة بعض المصطلحات الانجليزية وبضمنها رموز خاصة قد لا تظهر هذه الرموز في مكانها المحدد مع بعض المتصفحات ولتلافي هذه المشكلة نضع الرمز المسمى (OXFD) قبل كل رمز خاص ونكتب هذا الرمز من خلال الضغط المستمر على المفتاح + Alt هذه الارقام وبالترتيب من اليسار لليمين (0253) .

سابعاً - تخزين البيانات من طرف العميل

يمكن تخزين البيانات من طرف العميل بواسطة كائنين مختلفين هما :-

1- الكائن localStorage

يستخدم هذا الكائن لتخزين البيانات من طرف العميل لفترة غير محدودة ويستخدم الصيغة العامة التالية للتخزين :-

```
localStorage.varName="value" ;
```

حيث أن varName هو اسم المتغير وvalue هي قيمة هذا المتغير ويمكن استرجاع قيمة المتغير بالصيغة العامة التالية :-

```
localStorage.varName ;
```

مثال/

```
<script type="text/javascript">  
localStorage.lastname="Smith";
```

```
document.write(localStorage.lastname);  
</script>
```

مثال / لتخزين عدد زيارات المستخدم للصفحة

```
<script type="text/javascript">  
if (localStorage.pagecount)  
{  
localStorage.pagecount=Number(localStorage.pagecount) +1;  
}  
else  
{  
localStorage.pagecount=1;  
}  
document.write("Visits "+ localStorage.pagecount + " time(s).");  
</script>
```

2- الكائن sessionStorage

يستخدم هذا الكائن لتخزين البيانات من طرف العميل لفترة محدودة بجلسة واحدة فقط ,
ويستخدم الصيغة العامة التالية للتخزين :-

```
sessionStorage.varName="value" ;
```

حيث أن varName هو اسم المتغير و value هي قيمة هذا المتغير ويمكن استرجاع قيمة
المتغير بالصيغة العامة التالية :-

```
sessionStorage.varName ;
```

مثال/

```
<script type="text/javascript">  
sessionStorage.lastname="Smith";  
document.write(sessionStorage.lastname);  
</script>
```

مثال/ لكيفية تخزين عدد زيارات المستخدم لهذه الصفحة في نفس الجلسة

```
<script type="text/javascript">  
if (sessionStorage.pagecount)  
{  
sessionStorage.pagecount=Number(sessionStorage.pagecount) +1;  
}  
}
```

```

else
{
sessionStorage.pagecount=1;
}
document.write("Visits "+sessionStorage.pagecount+" time(s) this session.");
</script>

```

ملاحظة / هذان الكائنان يستخدمان كمخازن خاصة للبيانات أثناء كتابة شيفرة javascript .

ثامناً - الخصائص الجديدة للعنصر في html5

1- **contenteditable** :- هذه الخاصية تأخذ القيمة **true** او **false** وعندما تكون قيمتها هي **true** فهذا يعني أن المستخدم يستطيع تحرير محتوى القسم القابل للتحرير , وهذا يستخدم بشكل كبير في المحررات التي يتم بناؤها للعمل على الصفحات مثل محررات المنتديات .

مثال/

```

<div contenteditable="true">
<p>
<br>
<p>
</div>

```

لاحظ بأننا أضفنا في داخل الـ **<div>** الرئيسية الوسم **<p>** وبداخله الوسم **
** وذلك لأنه عندما يضيف المستخدم سطر جديد فإن المحرر سيضيف وسم **<div>** مع كل سطر ولكن بمثالنا هذا سيضيف المحرر وسم **<p>** مع كل سطر جديد .

2- **contextmenu** :- تقبل هذه الخاصية قيمة الخاصية **id** لعنصر القائمة **<menu>** مما يحدد قائمة السياق للعنصر .

3- **data-yourvalue** :- في html5 يمكن للمبرمج تعريف الخصائص الخاصة به بشرط أن تبدأ بـ **data-** .

4- **draggable** :- هذه الخاصية يمكن أن تأخذ القيمة **auto** , **false** , **true** وهي تحدد هل يستطيع المستخدم سحب العنصر أم لا .

5- **hidden** :- تأخذ هذه الخاصية القيمة **hidden** وعندها يختفي العنصر الذي تطبق عليه هذه الخاصية وجميع محتويات هذا العنصر , لكن يبقى بالإمكان الوصول إلى العنصر من خلال javascript .

6- **item** :- هذه الخاصية إما أن تكون قيمتها سلسلة نصية فارغة **empty string** او **url** مثل **syrdev** , والهدف من هذه الخاصية هي تجميع مجموعة عناصر مع بعضها .

7- **itemprop** :- هذه الخاصية تأخذ قيمة ضمن المجموعة التي عرفتها الخاصية **item** مثال

```
<section item="syrdev.net.article" >
```

```
<h1 itemprop="syrdev.net.title">HTML5 .... بالعربي نتقدم </h1>
```

```
<h1 itemprop="syrdev.net.desc"> HTML سنتعلم في هذه المقالة المختصرة المفيدة عن </h1>
```

```
</section>
```

8- **spellcheck** :- تأخذ هذه الخاصية إحدى القيمتين **true** , **false** وفي حالة القيمة **false**

هذا يعني انه في حال كتابة المستخدم كلمة او عبارة خطأ في الحقل النصي عندها سيقترح المتصفح تصحيح هذا الخطأ وعند الضغط عليه بزر الفأرة الأيمن عندها ستظهر مجموعة من الاقتراحات لتصحيح الخطأ (اذا كانت القيمة **true** فإنه سيظهر خط اسفل الكلمات الخاطئة اما اذا كانت القيمة **false** فلن يظهر خط اسفل الكلمة الخاطئة) .

9- **subject** :- تأخذ هذه الخاصية قيمة الخاصية **id** للعنصر المراد الارتباط معه , مثال /

```
<div id="about" >
```

```
<p> المطورون السوريون هو البداية ... </p>
```

```
</div> .....
```

```
<p subject="about" > syrdev.net هو موقع المطورون السوريون </p>
```

ثامنا :- لاضافة رمز صورة في اعلى الموقع بجانب العنوان (title) يجب ان نضيف الوسم **<link>** وبهذا الشكل لاحظ

```
<link rel="shortcut icon" href="Turquoise1.cur">
```

ولاحظ أن امتداد الصورة يجب أن يكون **ico** او **cur** , ويجب أن تنتبه الى أن هذا الوسم يجب ان يضاف بين وسمي بداية ونهاية الهيد **<head>**

الخاتمة