الفصل الأوَّل

HTML 5

مقدمة

يناقش هذا الفصل لغة HTML5 التي تعتبر حجر الأساس في إنشاء مواقع الويب، و لأنّنا لا يمكن أن نتحدث عن HTML5 مباشرةً فسيناقش هذا الفصل لغتي HTML و XHTML أولاً، و البداية مع لغة HTML.

ما هي HTML ؟

إنَّ HTML هي اختصار abbreviation الحروف الأولى من الجملة Hyper Text Markup Language و التي تعني بترجمة غير حرفية (لغة وصف صفحات الويب)[1]، و هي ليست لغة برمجيَّة، لا، بل هي لغة وصفيَّة Markup Language، لأنَّ HTML تستخدم مجموعة من الوسوم Tags لوصف صفحة الويب.

أمَّا الوسوم Tags فهي كلمات أو أحرف محددة مسبقاً Keywords تحمل معاني خاصَّة، و تكون محصورة بين قوسين من الشكل < > مثل و <html>، تأتي الوسوم في الغالب على شكل أزواج مثل و ، يدعى الوسم الأوَّل بوسم البداية Begin Tag أو وسم الفتح Opening Tag، و يدعى الوسم الثاني بوسم النهاية End Tag أو وسم الإغلاق Closing Tag.

يتم كتابة مستندات HTML باستخدام أي محرر نصي بسيط (مثل notepad) أو متقدِّم (مثل DreamWaver أو Visual Studio)، و يتم تخزينها في ملفات تحمل امتداداً extension من الامتدادين التاليين : html. أو htm.، و بالطبع لا يوجد أي فرق بينهما.

[1] لا أعرف إن قام أحد قبلي بترجمتها بهذه الطريقة لكنّني أشعر أنها الترجمة الأنسب.

تتكون مستندات HTML بشكل أساسي من محتوى نصي عادي بالإضافة إلى مجموعة من وسوم HTML، و في الحقيقة فإن مستندات HTML تدعى بـ(صفحات الويب) Web Pages، و يتم استخدام برنامج خاص لقراءة هذه المستندات يعرف هذا البرنامج باسم مستعرض الويب Web Browser و من أمثلته Internet Explorer و Firefox و Google chrome .. إلخ.

الهدف من مستعرض الويب هو قراءة مستندات HTML و عرضها بشكل صفحات ويب، بمعنى أن مستعرض الويب لا يعرض وسوم HTML كنصوص بشكل مباشر و لكنَّه يستخدمها لإنتاج صفحة ويب بالشكل و المحتوى الموصوفين في مستند HTML.

كيف أبدأ في تعلم كتابة مستندات HTML ؟

كل ما تحتاجه لكي تبدأ في تعلم كتابة مستندات HTML هو محرر نصوص (مثل Notepad) و مستعرض ويب (مثل Internet Explorer)، بعد ذلك يمكنك قراءة الفقرات التالية و التعلم خطوة بخطوة و أؤكد لك أنَّ أفضل طريقة للتعلم –برأيي– هي كتابة الأمثلة التي ستصادفك لاحقاً بشكل مباشر باستخدام محرر النصوص و من ثمَّ اختبارها باستخدام مستعرض الويب.

المثال الأوَّل في لغة HTML

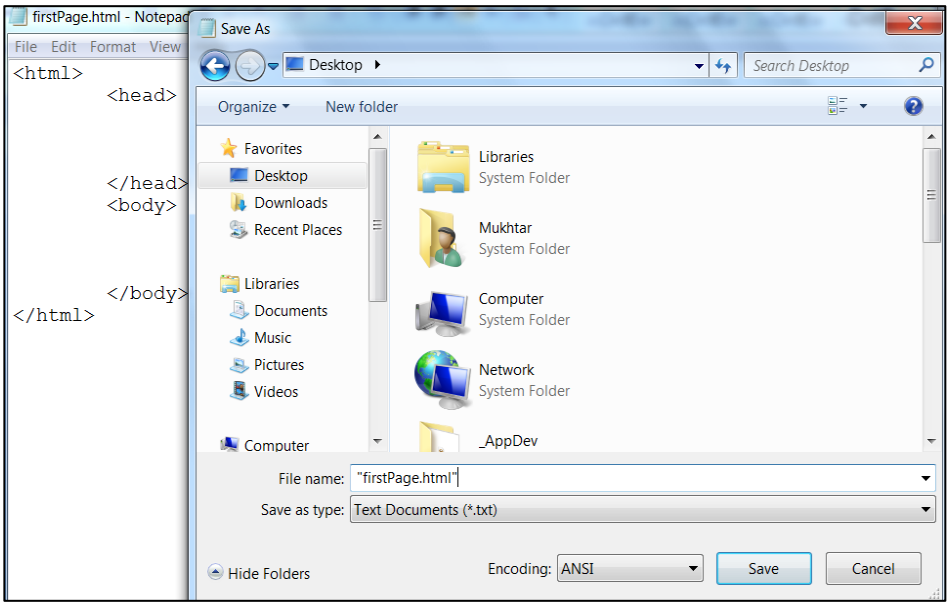
سنقوم الآن بكتابة أول صفحة HTML معاً باتباع الخطوات التالية:

نقوم أولاً بفتح أحد محررات النصوص و ليكن المفكرة Notepad على سبيل المثال و نكتب الشيفرة التالية:

```
<html>
      <head>
            <title>
                  My first HTML page
```

```
            </title>
      </head>
      <body>
            <p>
                Welcome to my first HTML page!
            </p>
      </body>
</html>
```

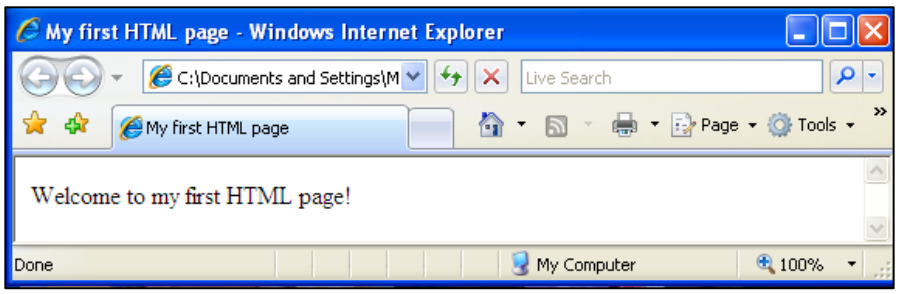
ثمَّ نقوم بحفظ الملف باسم firstPage.html



الشكل ١ : كتابة أول شيفرة HTML في برنامج المفكرة و حفظ الملف

ملحوظة : لحفظ الملف بلاحقة مختلفة عن اللاحقة txt. هناك طريقتان، الأولى أن نختار الخيار جميع الملفات *.* من القائمة المنسدلة حفظ كـ Save as type الظاهرة في الصورة أسفل اسم الملف ثم نكتب اسم الملف و لاحقته بشكل عادي، أمَّا الثانية و التي أفضِّلها شخصيَّاً فهي أن نضع اسم الملف مع لاحقته المطلوبة بين علامتي اقتباس مزدوجتين و نختار حفظ كما هو موضَّح في الصورة أعلاه.

الآن و بعد أن قمنا بحفظ الملف بالاسم firstPage.html نقوم باستعراضه باستخدام أحد مستعرضات الويب و ليكن Internet Explorer 8 على سبيل المثال لنشاهد الشكل النهائي للصفحة كمايلي:



الشكل ٢ : الشكل النهائي في المستعرض لأوّل صفحة مكتوبة باستخدام HTML

كما ترى فقد قمنا بكتابة صفحتنا الأولى باستخدام لغة HTML و هذه الصفحة تحتوي على نص ترحيب بسيط هو النص Welcome to my first HTML page!، تم توليد هذا النص باستخدام الوسم الخاص p و بتصوّري فإنَّ هذه الصفحة على بساطتها ستكون الباب الواسع للدخول إلى لغة HTML إنْ قرأت الفقرات التالية بقليل من التركيز.

عناصر HTML

تتكون كل صفحة من صفحات HTML من مجموعة من العناصر Elements يتم إنشاؤها باستخدام الوسوم Tags و الوسوم كما قلنا سابقاً عبارة عن أحرف أو كلمات تحمل معاني خاصة بالنسبة لمستعرض الويب و غالباً ما تكون أسماء الوسوم عبارة عن اختصارات لكلمات إنجليزيَّة فالوسم <p> مثلاً يستخدم لإنشاء العنصر المسؤول عن عرض مقاطع النصوص Paragraphs ضمن صفحات HTML و الذي سيتم مناقشة عمله مع عمل باقي الوسوم بشكل تفصيلي في الصفحات التالية.

تحصر الوسوم بين أقواس من الشكل < > و من أمثلتها الوسم <html> و الوسم <head> و الوسم <title> و الوسم <body> و الوسم <p>.

لكل عنصر من عناصر HTML وسم بداية Begin Tag و وسم نهاية End Tag تكون وسوم البداية محصورة بين القوسين < > أمَّا وسوم النهاية فتكون محصورة بين القوسين السابقين مضافاً إليهما رمز الخط المائل / المعروف بـ Slash بالشكل < /> و من أمثلة وسوم النهاية الوسم </html> و الوسم </head> و الوسم </title> و الوسم </body> و الوسم و يحوي كل عنصر من عناصر HTML بين وسمي بدايته و نهايته نصَّاً عاديَّاً لعرضه (كما فعل العنصر p في مثالنا السابق) و يمكن أن يحوي أي عدد آخر من عناصر HTML شرط الالتزام بترتيب متناظر لوسوم النهاية و البداية، فعند كتابة وسوم النهاية يجب أن يُراعى الترتيب فيتم كتابة وسم النهاية الخاص بوسم البداية غير المنتهي (الذي ليس له وسم نهاية) الأقرب فالأقرب، ففي حين أن الترتيب التالي يعتبر صحيحياً:

```
<html>
      <body>
            <p>
               نص عادي سيتم عرضه في المستعرض!
            </p>
      </body>
</html>
```

لأنَّ وسم النهاية الأوَّل جاء لإنهاء أقرب وسم بداية غير منتهي و هو في مثالنا، ثمَّ جاء وسم النهاية </body> لإنهاء أقرب وسم بداية غير منتهي و هو <body>، ثمَّ جاء وسم النهاية </html> لإنهاء أقرب وسم بداية غير منتهي و هو <html>.

فإنَّ الترتيب التالي يعتبر خاطئاً:

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

```
            <p>
                  نص عادي سيتم عرضه في المستعرض!
            </body>
      </html>
</p>
```

لأن وسم النهاية </body> جاء لإنهاء الوسم <body> الذي لم يكن الوسم الأقرب غير المنتهي للأسف و الحالة نفسها تكررت مع وسمي النهاية </html> و </p>.

الخلاصة: لا تنسَ كتابة وسم الإغلاق و راعِ ترتيب وسوم الإغلاق.

الفراغات White Spaces

في لغة HTML لا يوجد أي قيمة لمحارف الفراغات (مثل Space و Tab .. إلخ) عند استعراض الصفحات باستخدام مستعرض الويب و إنَّما تستخدم محارف الفراغات لغرض ترتيب الشيفرة و جعلها مقروءة بشكل أوضح فقط، فقراءة شيفرة HTML التالية على سبيل المثال:

```
<html>
      <body>
            <h1>
                  Hi!
            </h1>
      </body>
</html>
```

أسهل بكثير من قراءة شيفرة HTML التالية:

```
<html><body><h1>Hi!</h1></body></html>
```

مع أنَّهما متماثلتان تماماً عند استعراضهما باستخدام مستعرض الويب.

فائدة : من العادات البرمجيَّة الجيدة إزاحة كل المحتوى الموجود بين وسمي بداية و نهاية متماثلين بمقدار ضغطة Tab و البعض يفضِّل إزاحتها بمقدار ثلاث فراغات (ثلاث ضغطات على مفتاح Space).

الشكل العام لعناصر HTML

يتم وصف عناصر HTML باستخدام الوسوم، و فيمايلي نعرض الشكل العام لعنصر HTML:

- يبدأ عنصر HTML بوسم البداية Begin Tag.
- ينتهي عنصر HTML بوسم النهاية End Tag.
- كل ما يتم كتابته بين وسمي البداية و النهاية يدعى محتوى عنصر HTML علماً أن هناك بعض العناصر التي لا تحوي أي محتوى و التي تسمى عديمة المحتوى.
- يندمج وسما البداية و النهاية في وسم واحد في حالة العناصر عديمة المحتوى.
- يتم تمرير مجموعة من الخصائص لأغلب عناصر HTML في وسم البداية عن طريق الواصفات Attributes.

ملحوظة: دائماً و أبداً، قم بكتابة عناصر HTML بالأحرف الإنجليزيَّة بحالتها الصغيرة Lower Case.

لنشاهد الأمثلة التالية:

وسم البداية	محتوى العنصر	وسم النهاية
	Welcome to my website.	** **
	Go to index	****
<hr />		

يمثِّل كل سطر في الجدول السابق عنصراً من عناصر HTML في الصفحة بشكل تخيُّلي، فالعنصر الأوَّل يبدأ بالوسم و ينتهي بالوسم و يحوي المحتوى النصي Welcome to

my website بينما يبدأ العنصر الثاني بالوسم و ينتهي بالوسم و يحوي المحتوى النصي Go to index مع إسناد القيمة index.htm إلى واصفته href، في حين أنَّ العنصر الثالث عديم المحتوى يتكون من اتحاد وسمي البداية و النهاية في وسم واحد بالشكل <hr />.

الشكل العام لصفحات HTML

تتكون كل صفحة HTML من ثلاث مناطق:

- منطقة جسد الصفحة Body Section:

و هي المنطقة المحصورة بين وسمي <body> و </body> و هذه المنطقة هي التي تنتج الشكل النهائي للصفحة و تضم جميع العناصر التي تمثِّل المحتوى الظاهر للصفحة و الذي سيظهر في مستعرض الويب عند استعراض هذه الصفحة و بالطبع فإن الجزء الأكبر من عناصر HTML سيكون في هذه المنطقة.

- منطقة رأس الصفحة Head Section:

و هي المنطقة المحصورة بين وسمي <head> و </head> و هذه المنطقة تحوي مجموعة من عناصر HTML أغلبها لا يظهر في مستعرض الويب عند استعراض الصفحة و لكن المهمة الرئيسية لهذه المنطقة هي إعطاء معلومات عن ماهيَّة المحتوى الموجود بالصفحة إضافة لبعض المعلومات غير الظاهرة الأخرى و التي تستخدمها محركات البحث من أجل الأرشفة و البحث في الغالب.

- المنطقة الأم[٢]:

[٢] لم يرد هذا المصطلح في أي من المراجع التي قرأتها و إنَّما هو اجتهاد شخصي لتبسيط الفكرة المطروحة (برأيي).

و هي المنطقة المحصورة بين وسمي <html> و </html> و هذه المنطقة هي المنطقة التي تحدد بداية و نهاية الصفحة و هي التي تضم منطقتي الرأس و الجسد و بهذا فهي تضم كامل مستند (صفحة) HTML.

و بهذا يكون الشكل العام لأي مستند HTML كمايلي:

```
<html>
      <head>
          محتوى منطقة الرأس غير الظاهر سيكون هنا
      </head>
      <body>
          محتوى الصفحة النهائي و الظاهر فعليًّاً سيكون هنا
       </body>
</html>
```

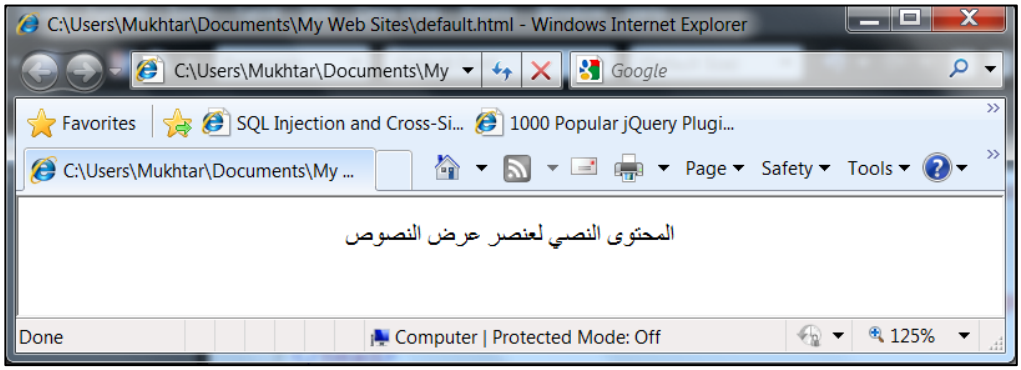
واصفات الوسوم Attributes

يتلخّص كل ما قلناه سابقاً بأنَّ لغة HTML تتيح لك تقديم المحتوى عبر إنشاء مجموعة من العناصر، مهمة كل عنصر من عناصر HTML عرض شيء محدد جداً في صفحتك فهناك عناصر لعرض النصوص و هناك عناصر لعرض الصور و هناك عناصر لعرض الروابط التشعبيَّة Links .. إلخ، يتم إنشاء كل عنصر من هذه العناصر بواسطة وسم خاص من وسوم HTML، و يكون لكل عنصر وسم بداية و وسم نهاية يحصران محتوى العنصر الظاهر بينهما.

يمكن تزويد بعض وسوم البداية بمجموعة من الخصائص الإضافيَّة و التي تخصّص سلوك عرض العنصر لمحتواه و يتم هذا عبر ما يعرف بالواصفات Attributes، انظر للشيفرة التالية على سبيل المثال:

```
<p align="center">
    المحتوى النصي لعنصر عرض النصوص
</p>
```

كما تلاحظ فقد قمنا بإضافة شيء جديد في هذه الشيفرة إلى وسم البداية الخاص بالعنصر p ألا و هو الواصفة align التي تحدد محاذاة النص الذي سيعرضه العنصر p و أعطيناها القيمة center لعرض النص في منتصف الصفحة و عند استعراضها سيظهر النص كمايلي:



الشكل ٣ : عنصر عرض النصوص <p> عند إضافة واصفة المحاذاة align

حسناً، يوجد لكل وسم بداية من وسوم HTML مجموعة معروفة سلفاً من الواصفات Attributes و يوجد لكل واصفة مجموعة معروفة سلفاً من القيم التي يمكن أن يتم إسنادها إلى الواصفة، فالواصفة align هي إحدى الواصفات المعروفة سلفاً لوسم البداية <p> على سبيل المثال و القيم التي يمكن أن تسند إليها هي left لمحاذاة النص إلى اليسار أو right لمحاذاة النص إلى اليمين أو center لمحاذاة النص إلى الوسط، و سيتم بالطبع عرض واصفات كل وسم بالتفصيل عند الحديث عنه و أظن أنه لا داعي للقول أن الشكل العام للواصفة هو "name="value حيث أنَّ name هو اسم الواصفة و value هي القيمة المُسنَدة لتلك الواصفة و التي يجب أن توضع بين علامتي اقتباس مزدوجتين " " أو علامتي اقتباس مفردتين ' '.

ملحوظة : يجب أن تتم كتابة وسوم و واصفات HTML بحروف إنجليزيَّة صغيرة Lower Case.

أظن أنَّنا نستطيع الآن أن نبدأ بالحديث عن عناصر HTML كافةً و بالتفصيل و البداية مع عناصر العناوين Headings، و لكن قبل ذلك سنعرض جدولاً بمجموعة من الواصفات المشتركة بين أغلب وسوم HTML هنا بدلاً من تكرارها مع كل وسم من تلك الوسوم، و فيمايلي الجدول:

اسم الواصفة	القيم الممكنة	الشرح
id	اي اسم فريد(غير مكرر) بالنسبة للمستند	يتم استخدام هذا الاسم للتعامل مع العنصر برمجيًّاً باستخدام لغة Java Script أو jQuery، و لها استخدامات أخرى سيتم عرض أحدها في فقرة الروابط الداخلية لاحقاً
dir	ltr rtl	لتحديد اتجاه القراءة، ltr تعني أن اتجاه القراءة من اليسار إلى اليمين أمَّا rtl فتعني أن اتجاه القراءة من اليمين إلى اليسار
align	left right center justify	لتحديد محاذاة النص
class	أي اسم فئة CSS صالح	لمنح كافة خصائص الفئة إلى العنصر و سيتم مناقشة هذا بالتفصيل في الفصل الخاص بـ CSS3
name	أي اسم فريد	تستخدم لتمييز العنصر برمجيّاً

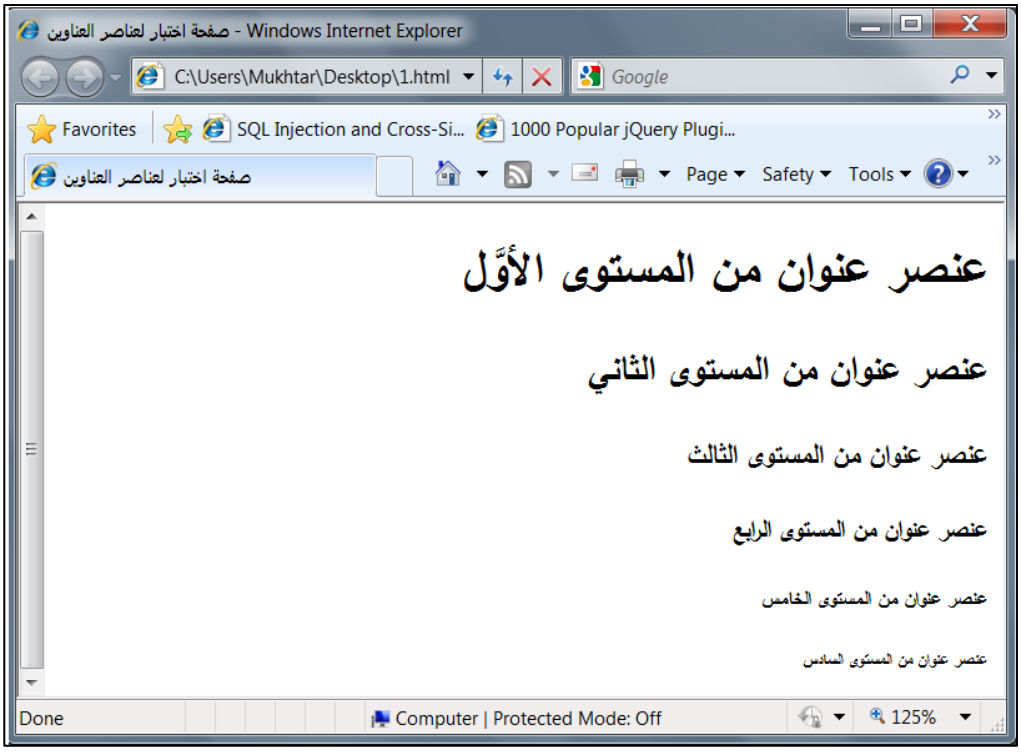
الجدول ١ : جدول الواصفات المشتركة بين أغلب عناصر HTML

عناصر العناوين Headings

توفر لغة HTML ستَّة عناصر لعرض العناوين و هي على الترتيب: h1 و h2 و h3 و h4 و h5 و h6 حيث أنَّ العنصر h1 هو أكبرها حجماً و العنصر h6 هو الأصغر و ما بينهما يتدرج في الحجم، و طبعاً حرف الـ h هنا اختصار لكلمة Heading، دعنا نجرب الشيفرة التالية على سبيل المثال:

```
<html>
     <head>
          <title>
               صفحة اختبار لعناصر العناوين
          </title>
     </head>
     <body dir='rtl'>
          <h1>عنصر عنوان من المستوى الأوَّل</h1>
          <h2>عنصر عنوان من المستوى الثاني</h2>
          <h3>عنصر عنوان من المستوى الثالث</h3>
          <h4>عنصر عنوان من المستوى الرابع</h4>
          <h5>عنصر عنوان من المستوى الخامس</h5>
          <h6>عنصر عنوان من المستوى السادس</h6>
     </body>
</html>
```

عند استعراض الصفحة السابقة في مستعرض الويب سنشاهد النتيجة التالية:



الشكل ٤ : صفحة اختبار عناصر العناوين h1 إلى h6

كما تلاحظ فقد قمنا بتزويد الوسم <body> بالواصفة dir و التي تحدد اتجاه قراءة جسد المستند و قمنا بإسناد القيمة rtl لها و ذلك لنجعل اتجاه القراءة من اليمين إلى اليسار right to left و هو ما يتناسب مع اللغة العربية بالطبع، و في حال أن هذه الواصفة لم تُعْطَ أية قيمة فإن القيمة الافتراضية لها هي ltr أي أن اتجاه القراءة الافتراضي من اليسار إلى اليمين.

ملحوظة: استخدم عناصر العناوين <h1> إلى <h6> من أجل عناوين الفقرات فقط، و لا تستخدمها لجعل النص عريضاً أو كبير الحجم فهناك وسوم خاصة لهذين الغرضين.

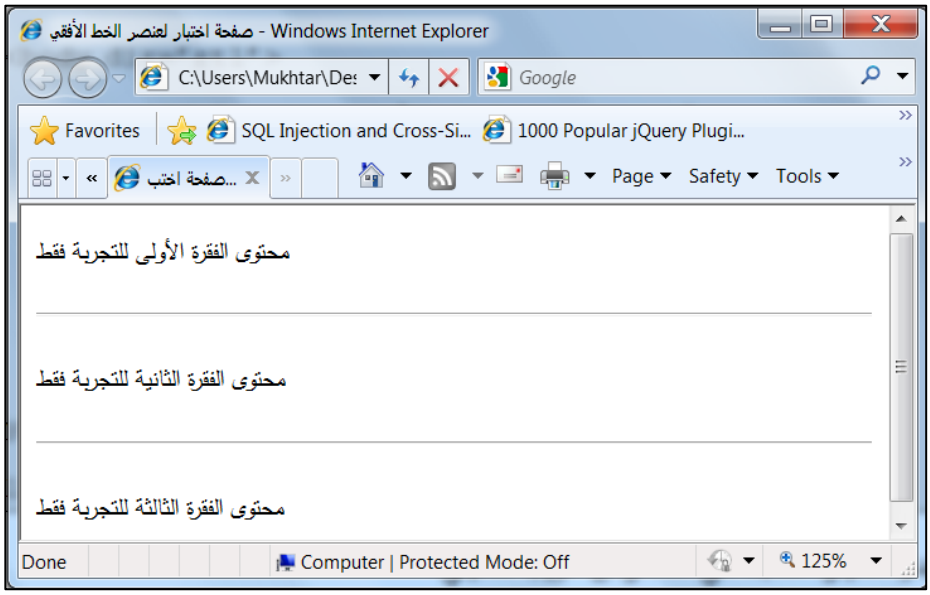
عنصر الخط الأفقي Horizontal Line

تقدِّم لغة HTML الوسم الخاص <hr /> لإنشاء الخطوط الأفقيَّة في الصفحة، انظر للمثال التالي:

```
<html>

      <head>
```

```
            <title>
                صفحة اختبار لعنصر الخط الأفقي
            </title>
        </head>
        <body dir='rtl'>
            <p>محتوى الفقرة الأولى للتجربة فقط</p>
            <hr />
            <p>محتوى الفقرة الثانية للتجربة فقط</p>
            <hr />
            <p>محتوى الفقرة الثالثة للتجربة فقط</p>
</body>
</html>
```

تبدو الشيفرة السابقة في مستعرض الويب كمايلي:



الشكل ٥ : صفحة تجربة العنصر <hr />

التعليقات Comments

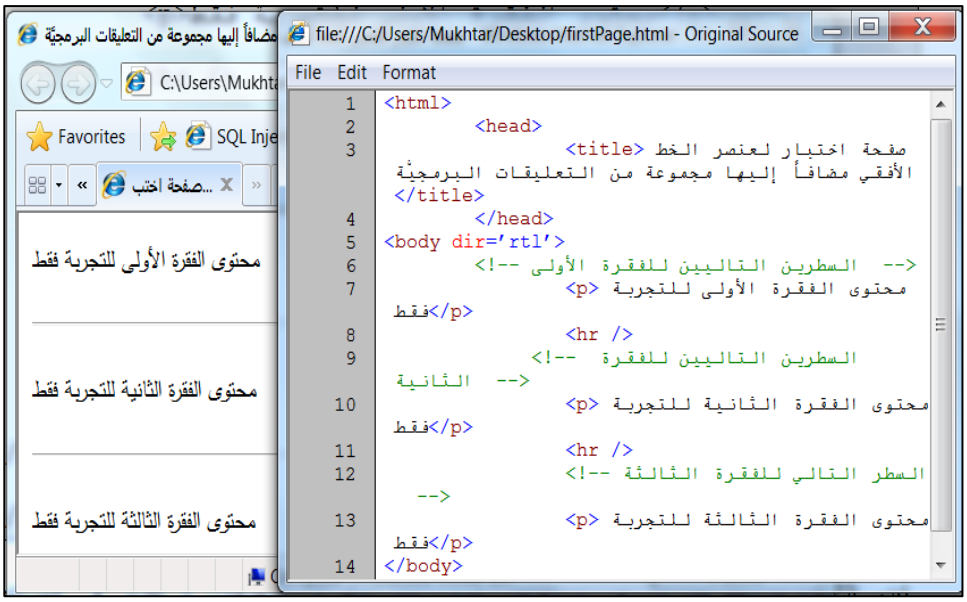
اعتاد المبرمجون في لغات البرمجة التقليديَّة على كتابة أسطر توضيحيَّة لا تُعالَج إذ أنها لا تعتبر جزءاً من الشيفرة، و إنَّما تستخدم فقط لتذكير المبرمج بأجزاء الشيفرة عندما يعود لتعديلها بعد فترة من الزمن، و هذه الأسطر تعرف بالتعليقات Comments، و مع أنَّ HTML لغة وصفية كما قلت سابقاً إلّا أنها توفر آلية لكتابة التعليقات ضمن المستندات، فتقدِّم الصيغة العامة التالية لكتابة التعليق:

```
<!-- comment -->
```

لنشاهد المثال التالي على استخدام التعليقات ضمن مستند HTML:

```
<html>
     <head>
          <title> صفحة اختبار لعنصر الخط الأفقي مضافاً إليها
مجموعة من التعليقات البرمجيَّة</title>
     </head>
     <body dir='rtl'>
          <!-- السطرين التاليين للفقرة الأولى -->
          <p>محتوى الفقرة الأولى للتجربة فقط</p> <hr />
          <!-- السطرين التاليين للفقرة الثانية -->
          <p>محتوى الفقرة الثانية للتجربة فقط</p> <hr />
          <!-- السطر التالي للفقرة الثالثة -->
          <p>محتوى الفقرة الثالثة للتجربة فقط</p>
     </body>
</html>
```

تبدو الشيفرة السابقة كمايلي في مستعرض الويب مما يؤكّد أن التعليقات لا تظهر في المستند(أثناء استعراض الصفحة جرِّب اختيار الأمر "عرض-> المصدر" أو "View->source"):



الشكل ٦ : التعليقات تظهر باللون الأخضر في نافذة عرض المصدر و لا تظهر في الصفحة

ملحوظة: لا تنسَ إشارة التعجُّب بعد القوس الأوّل في بداية وسم التعليق.

عناصر النصوص Paragraphs

تتكون صفحة الويب في الواقع من مجموعة من الفقرات النصية في الغالب، يتم إنشاء هذه الفقرات باستخدام الوسم <p> و الذي مر معنا استخدامه في الأمثلة السابقة، و يستخدم الوسم
 للنزول إلى السطر التالي و ذلك لأن محارف الفراغات بما فيها محرف المفتاح Enter ليس لها قيمة في لغة HTML كما ذكرنا سابقاً.

عناصر تنسيق النصوص Text Formatting

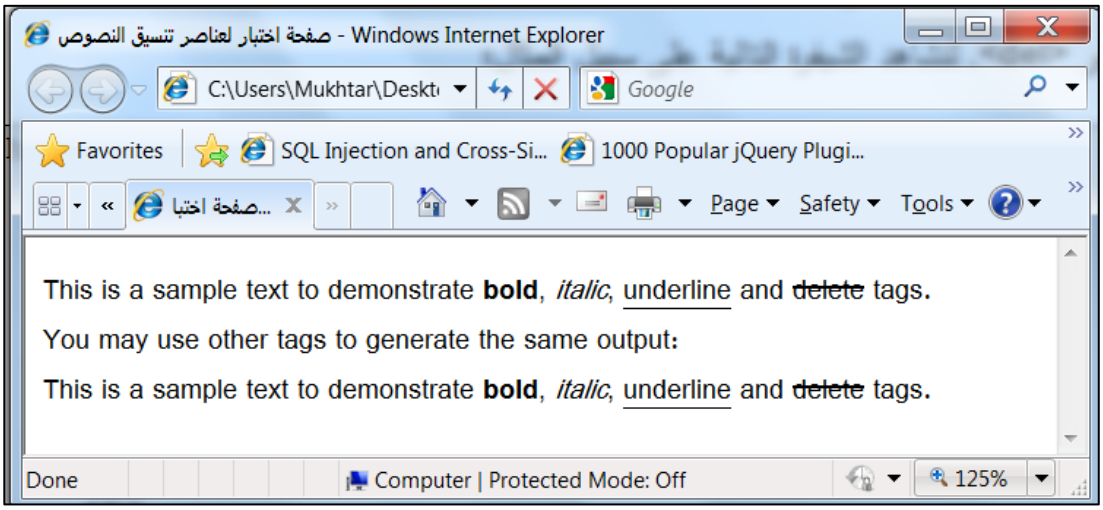
توفر لغة HTML مجموعة من العناصر لتنسيق النصوص، فلجعل النص عريضاً Bold توفر العنصر <b> (أو العنصر)، و لجعل النص مائلاً Italic توفر العنصر <i> (أو

العنصر)، و لوضع خط أسفل النص توفر العنصر <u>، و لشطب النص Delete توفر العنصر <del>، لنشاهد الشيفرة التالية على سبيل المثال:

```
<html>
  <head>
     <title>صفحة اختبار لعناصر تنسيق النصوص</title>
  </head>
  <body>
     <p>
          This is a sample text to demonstrate
          <b>bold</b>,
          <i>italic</i>,
          <u>underline</u> and
          <del>delete</del> tags.
          <!-- استخدمنا <br/> للنزول سطراً واحداً -->
          <br/>
          You may use other tags to generate the same output:
          <br/>
          This is a sample text to demonstrate
          <strong>bold</strong>,
          <em>italic</em>,
          <u>underline</u> and
          <del>delete</del> tags.
     </p>
  </body>
```

```
</html>
```

تبدو الشيفرة أعلاه في المستعرض كمايلي:



الشكل ٧ : صفحة اختبار عناصر تنسيق النصوص

ملحوظة: بالطبع هناك المزيد من عناصر تنسيق النصوص سيتم ذكرها في مرجع وسوم HTML في نهاية هذا الفصل.

المحارف و الرموز الخاصَّة Special Characters

هناك بعض المحارف و الرموز التي لا يمكن عرضها في الصفحة عن طريق كتابتها بشكل مباشر مثل الرموز المستخدمة في الرياضيات و محارف الفراغات و غيرها، توفر لغة HTML آلية خاصة لعرض هذه الرموز و ذلك باتباع الصيغة العامة التالية ;&value باستبدل القيمة value بقيمة المحرف المطلوب أن يتم عرضه في مستعرض الويب و في الجدول التالي نماذج من هذه المحارف:[٣]

[٣] للحصول على الجدول الكامل يمكن اتباع الرابط التالي:
http://www.w3schools.com/tags/ref_symbols.asp

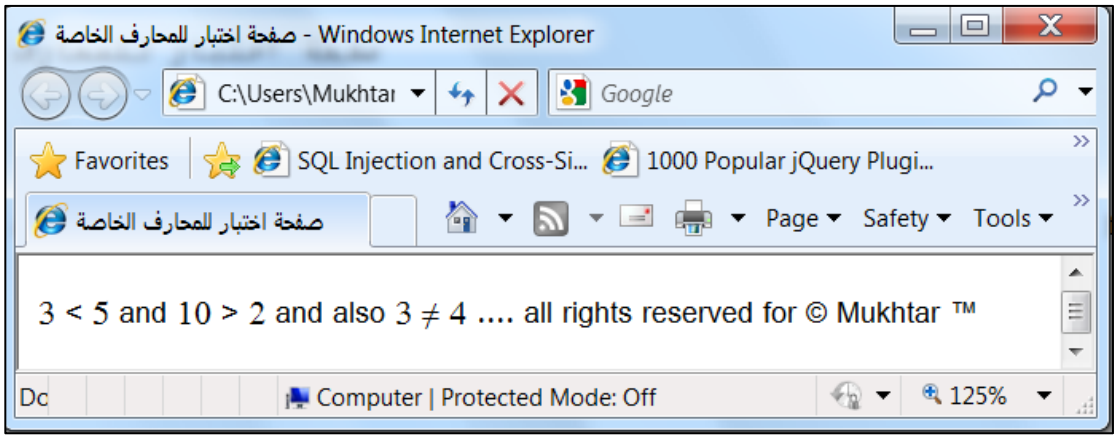
الرمز	شيفرة XHTML
<	<
>	>
≠	≠
™	™
©	©
محرف الفراغ	

الجدول ٢ : بعض المحارف الخاصة في HTML

لنشاهد المثال التالي:

```
<html>
  <head>
     <title>
          صفحة اختبار لبعض المحارف الخاصة
     </title>
  </head>
  <body>

     <p>
        3 &lt; 5 and 10 &gt; 2 and also 3 &ne; 4 ....
       all rights reserved for &copy; Mukhtar &trade;
      </p>

  </body>
</html>
```

و الذي يبدو عند عرضه في المستعرض كمايلي:



الشكل ٨ : مثال لاستخدام بعض المحارف الخاصة في الصفحة

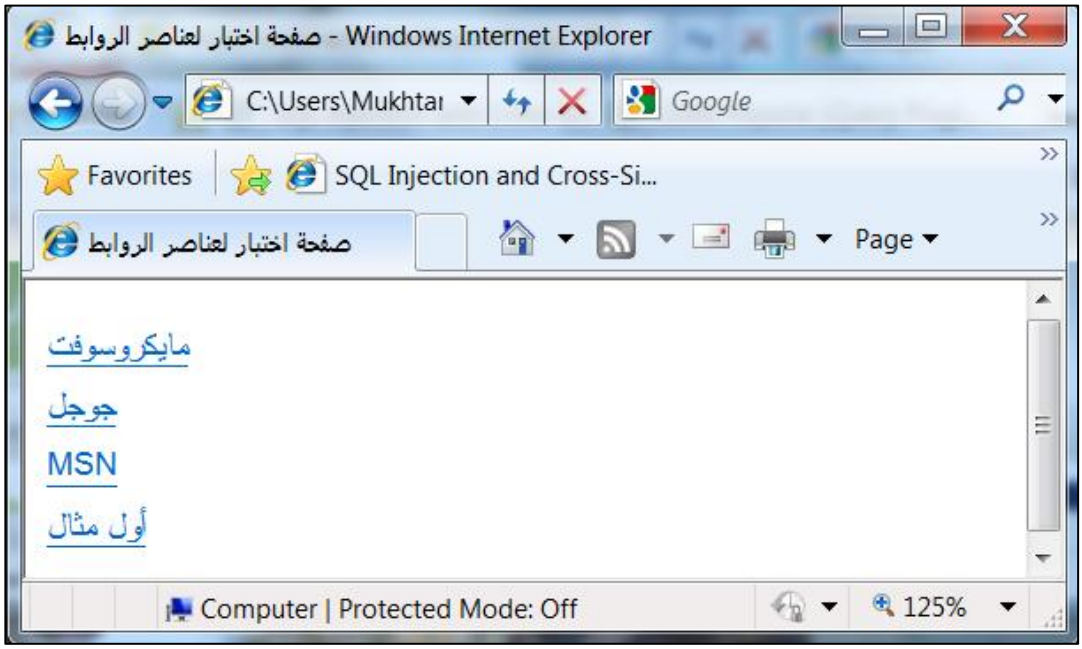
عناصر الروابط Hyper Links

توفر لغة HTML آلية للانتقال بين الصفحات المختلفة عبر عناصر الروابط Hyper Links، و الروابط عبارة عن نصوص أو صور تنقلك عند النقر عليها من الصفحة الحالية إلى إلى صفحة ويب أخرى، تقوم مستعرضات الويب بتمييز الروابط بعرض "يد صغيرة" كمؤشر للفأرة عند الإشارة إلى أحد الروابط، و يتم إنشاء الروابط بواسطة الوسم <a>، و الـ a هذه اختصار لـ Anchor ، يتم تزويد عنصر الرابط بنص يُعرض كمحتوى له أمَّا الموقع الهدف (الذي سيتم الانتقال إليه عند النقر على الرابط) فيتم تزويده للواصفة href و التي هي اختصار لـ Hyper Reference، فللنظر للمثال التالي:

```
<html>
  <head>
     <title>
          صفحة اختبار لعناصر الروابط
     </title>
  </head>
```

```
  <body>
    <a href="http://www.microsoft.com">مايكروسوفت</a><br />
    <a href="http://www.google.com">جوجل</a><br />
    <a href="http://www.msn.com" target="_blank" >MSN</a>
    <br />
    <a href="firstPage.html" target="_self">أول مثال</a>
  </body>
</html>
```

الذي يبدو عند استعراضه بمستعرض الويب كمايلي:



الشكل ٩ : صفحة اختبار عنصر الروابط

لقد قمنا بتزويد صفحتنا بمجموعة من عناصر الروابط التي تشير إلى بعض المواقع الشهيرة و قد تعمدت في الرابط الأخير ذكر اسم ملف المثال الأوّل الذي قمنا بإنشائه معاً في بداية هذا الفصل و ذلك لتوضيح أن الروابط يمكن أن تكون مطلقة Absolute (كالثلاثة الأولى) أو نسبية Relative

(كالرابط الأخير) و يقصد بالنسبية أن الرابط المذكور يكون عبارة عن مسار Path ملف الصفحة الهدف بالنسبة للصفحة الحالية و في حالتنا فقد كانت الصفحة الهدف firstPage.html في نفس مجلد الصفحة الحالية [٤].

عند النقر على أحد هذه الروابط سيتم الانتقال إلى الرابط URL المذكور في الواصفة href المرافقة لعنصر الرابط الذي تم النقر عليه، من الجدير بالذكر أنّنا قمنا بتزويد بعض الروابط بالواصفة target و التي تحدد مكان عرض الصفحة الهدف في المستعرض بمعنى أنّها تجيب على الأسئلة التالية:

هل سيتم عرض الصفحة الهدف في نفس الصفحة الحالية؟ (في هذه الحالة فإن قيمة الواصفة هي القيمة self_)، هل سيتم عرض الصفحة الهدف في نافذة مستعرض جديدة؟ (في هذه الحالة فإن قيمة الواصفة blank_)، هل سيتم عرض الصفحة الهدف في إطار محدد؟ (في هذه الحالة فإن قيمة الواصفة هي اسم الإطار و سيتم تناول الإطارات بشكل مفصّل لاحقاً).

آخر ما يجب ذكره عن الروابط حالياً أنَّه يمكن للرابط أن يشير إلى أي مورد Resource متوفر على الويب و ليس فقط الصفحات، كما يمكن أن يشير الرابط إلى بريد إلكتروني E-Mail و ذلك بأن نضع القيمة :mailto قبل البريد الإلكتروني الذي نريد أن نضع رابطاً له، انظر المثال التالي:

```
<html>
  <head>
     <title>
          روابط خاصة
     </title>
```

[٤] يمكنك تعلم المزيد عن المسارات Paths من خلال زيارة الموقع:
http://msdn.microsoft.com/en-us/library/aa365247(VS.85).aspx

```
  </head>
  <body>
    <a href="http://www.somesite.com/files/file1.zip"> انقر هنا
لتحميل الملف</a>
    <br />
    <a href ="mailto:mokhtar_ss@hotmail.com">راسلني</a>
  </body>
</html>
```

ما سيحدث عند النقر على الرابط الأوّل هو ظهور مربّع تحميل الملف Save File Dialog و ما سيحدث عند النقر على الملف الثاني هو ظهور برنامج مدير البريد الإلكتروني (مثل Outlook Express) في وضع إرسال رسالة جديدة إلى البريد الإلكتروني المذكور في الرابط.

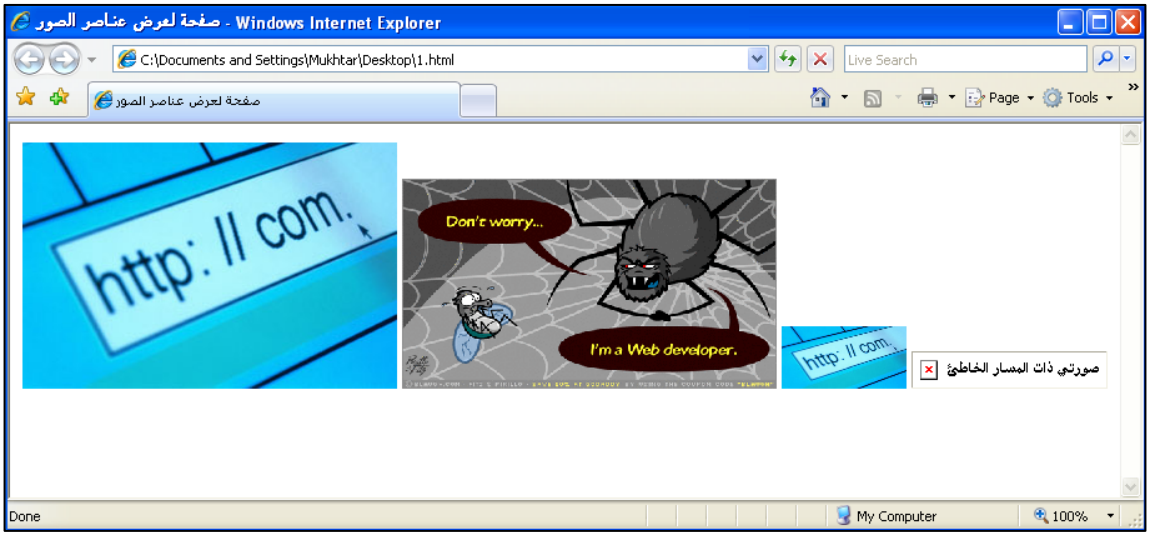
عناصر الصور Images

توفر لغة HTML عنصراً خاصاً لعرض الصور ضمن الصفحة هذا العنصر هو العنصر <img/> (اختصاراً لـ Image) الذي يقوم بعرض الصورة ضمن الصفحة اعتماداً على مسارها الذي يمرر للعنصر عبر الواصفة src (اختصاراً لـ Source)، و كما هو الحال بالنسبة للعنصر الخاص
 فإنّ العنصر <img/> لا يمتلك أي وسم نهاية و لذلك فإنّ شكله العام في الغالب يكون ، لنشاهد المثال التالي:

```
<html>
  <head>
    <title>
        صفحة لعرض عناصر الصور
    </title>
  </head>
```

```
  <body>
     <img src="myPic.jpg" />
      <img src="myPic2.gif" alt="الصورة الثانية" />
      <img src="myPic.jpg" alt="الصورة الثالثة" width="100px"
height="50px" />
      <img src="مسار خاطئ" alt="صورتي ذات المسار الخاطئ" />
  </body>
</html>
```

و الذي يبدو عند عرضه في المستعرض كمايلي:



الشكل ١٠ : صفحة اختبار لعنصر الصور

كما تلاحظ فقد قام عنصر عرض الصور بعرض الصور ذات المسارات المذكورة في كل وسم <img/>، و من واصفات هذا العنصر الواصفة alt التي تسند إليها قيمة نصيّة يتم عرضها في حال تعذر الوصول إلى الصورة المذكورة في الواصفة src و هذه الحالة حدثت معنا في الصورة الرابعة و التي زودناها بمسار خاطئ عن قصد لعرض هذه الحالة، كما يمكن عرض الصورة بمقاس محدد و ذلك عبر تزويد العنصر img بواصفتي الارتفاع height و العرض width و التي

تسند إليهما قيم رقميَّة تحدد عرض و ارتفاع الصورة المعروضة بالبكسل(هذا ما تعنيه px في القيم المسندة لهاتين الواصفتين).

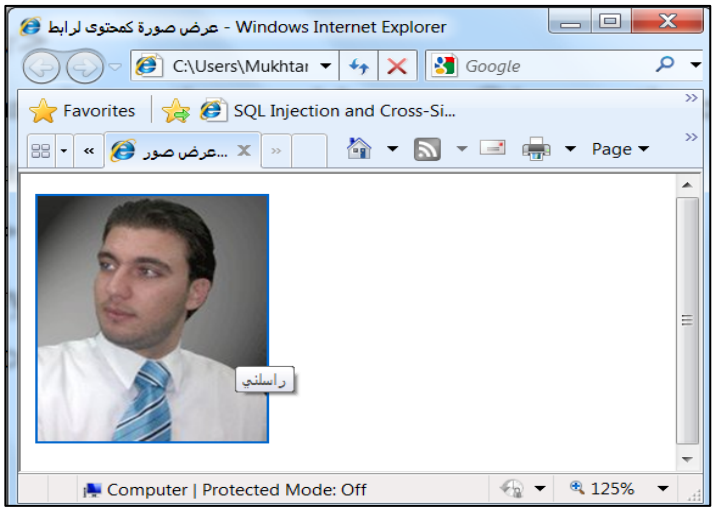
ملحوظة: على فرض أن صفحتنا تحتوي ٥ صور فإنَّ عرض الصفحة بشكل تام و صحيح سيحتاج لتحميل ٦ ملفات هي ملف الصفحة و ملفات الصور الخمسة و هذا ما قد يسبب حملاً زائداً على موقعك مستقبلاً في حال الإفراط في استخدام الصور.

الصور بدلاً من نصوص الروابط

من الشائع في مواقع الويب أن يتم استبدال نصوص محتوى عناصر الروابط <a> بالصور و ذلك لتجميل محتوى الموقع بحيث يتم الانتقال للصفحة الهدف للرابط عند النقر على الصورة بدلاً من النقر على النص و يتم ذلك بتضمين العنصر <img/> كمحتوى للعنصر <a> و فيمايلي نعرض مثالاً بسيطاً لذلك:

```
<html>
  <head>
     <title>
          عرض صورة كمحتوى لرابط
     </title>
  </head>
  <body>
     <a href = "mailto:mokhtar_ss@hotmail.com">
        <img src="myPic.jpg" alt="راسلني" />
     </a>
  </body>
</html>
```

و هذا ما يبدو في المستعرض كمايلي:



الشكل ١١ : استخدام الصور بدلاً عن النصوص كمحتوى للروابط

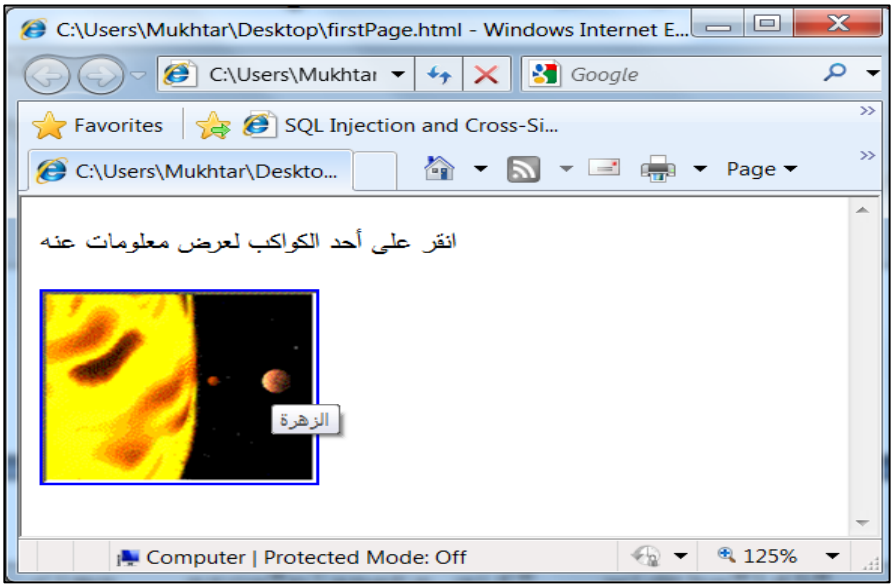
الخرائط الصوريَّة Image Maps

الخرائط الصوريَّة تعني جعل أجزاء محدّدة من صورة ما روابطاً، و يتم هذا على ثلاث مراحل، الأولى هي أن يتم وضع الصورة في الصفحة بشكل تقليدي عن طريق الوسم <img/>، أما الثانية فهي إنشاء عنصر الخريطة <map> مع إسناد قيمة الواصفة name الخاصة به كاسم للخريطة و من ثمَّ إنشاء مجموعة من عناصر <area> بإحداثيات المناطق التي ستتحول إلى روابط في الصورة، و المرحلة الثالثة تتم بربط عنصر الخريطة و عنصر الصورة و ذلك بإسناد اسم الخريطة إلى الواصفة usemap الخاصة بعنصر الصورة، لنشاهد المثال التالي:

```
<html>

<body>

      <p>انقر على أحد الكواكب لعرض معلومات عنه</p>

      <img src="planets.gif" width="145" height="126"
alt="الكواكب" usemap="#planetmap" />

      <map name="planetmap">
```

```
            <area shape="rect" coords="0,0,82,126" alt="الشمس"
href="sun.htm" />

            <area shape="circle" coords="90,58,3" alt="المريخ"
href="mercur.htm" />

            <area shape="circle" coords="124,58,8" alt="الزهرة"
href="venus.htm" />

      </map>

</body>

</html>
```

يبدو المثال التالي في المستعرض كمايلي:

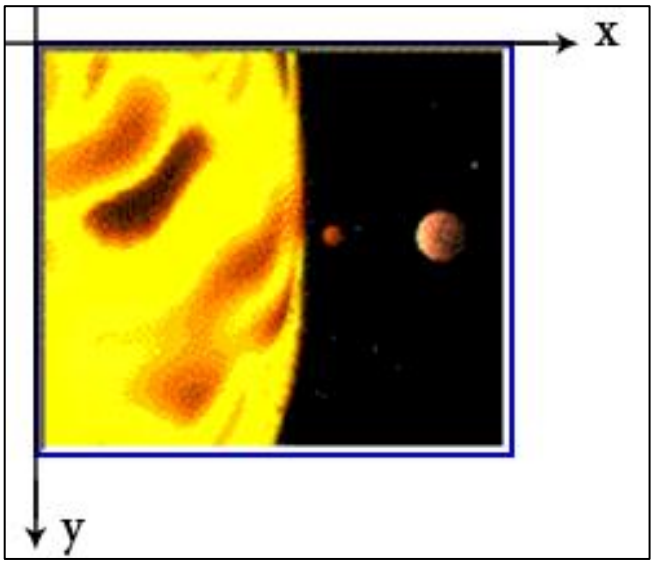


الشكل ١٢ : مثال على صنع خريطة صورية

كما تلاحظ فقد تمت إضافة الصورة بشكل طبيعي أولاً، ثمَّ بدأنا بإنشاء الخريطة و أعيطناها اسماً بالسطر:

```
      <map name="planetmap">
```

بعد ذلك قمنا بإنشاء منطقة لكل كوكب من الكواكب عبر الوسم area و الذي له واصفتان الأولى shape لتحديد شكل المنطقة (مستطيلة، دائرية .. إلخ)، أمَّا الثانية فهي واصفة الإحداثيات coords حيث تم تمرير أربعة إحداثيات في حالة المستطيل لتحديد أبعاد المستطيل يمثل الإحداثيان الأول و الثاني الركن الأيسر العلوي من المستطيل بينما يمثل الإحداثيان الثالث و الرابع الركن الأيمن السفلي من المستطيل و هذا كافٍ لتحديده، و بالطبع تم هذا على اعتبار أن مركز الجملة الديكارتيَّة للصورة هو الركن الأيسر العلوي من الصورة، و بهذا فإن النقطة (٠,٠) و النقطة (82,126) تحددان منطقة كوكب الشمس بالمستطيل المرسوم بينهما، أمَّا في حالة المنطقة الدائرية فيتم تمرير ثلاثة إحداثيات يعبر الأول و الثاني عن نقطة مركز الدائرة و يعبر الثالث عن نصف قطرها.



الشكل ١٣ : شكل تخيلي لتوضيح المبدأ الهندسي في رسم المناطق

و في النهاية قمنا بتمرير اسم الخريطة إلى الواصفة usemap الخاصة بالصورة.

ملحوظة: يتم تمرير اسم الخريطة إلى الواصفة usemap مسبوقاً بالرمز #

الروابط الداخليَّة Internal Linking

أحياناً يكون المحتوى الموجود ضمن صفحة معيّنة كبيراً إلى حدٍّ ما و في حالة مثل هذه يصبح الانتقال إلى فقرة محددة ضمن الصفحة عملية تستغرق بعض الجهد و الوقت، توفِّر لغة HTML آليَّة للانتقال إلى جزء محدد من الصفحة مباشرة عبر ما يعرف بالروابط الداخلية Internal Links

و التي يتم تحقيقها عبر استخدام وسم الرابط <a> و تمرير معرَّف العنصر المراد الانتقال إليه كقيمة للواصفة href بالشكل التالي:

```
<a href="#id"> انتقل إلى العنصر ذو المعرف id</a>
```

حيث أنّ id يمثل معرف العنصر المراد الانتقال إليه ضمن الصفحة و المثال التالي يوضِّح هذه الفكرة:

```
<html>
  <head>
     <title>
        مثال على الروابط الداخلية
     </title>
  </head>
  <body>
      <a href="#Para4">إلى الفقرة الرابعة</a>
     <h1>الفقرة الأولى</h1>
     <p>هنا محتوى الفقرة الأولى</p>
     <h1>الفقرة الثانية</h1>
     <p>هنا محتوى الفقرة الثانية</p>
     <h1>الفقرة الثالثة</h1>
     <p>هنا محتوى الفقرة الثالثة</p>
     <h1 id="Para4">الفقرة الرابعة</h1>
     <p>هنا محتوى الفقرة الرابعة</p>
  </body>
</html>
```

القوائم Lists

توفر لغة HTML نوعين من عناصر القوائم:

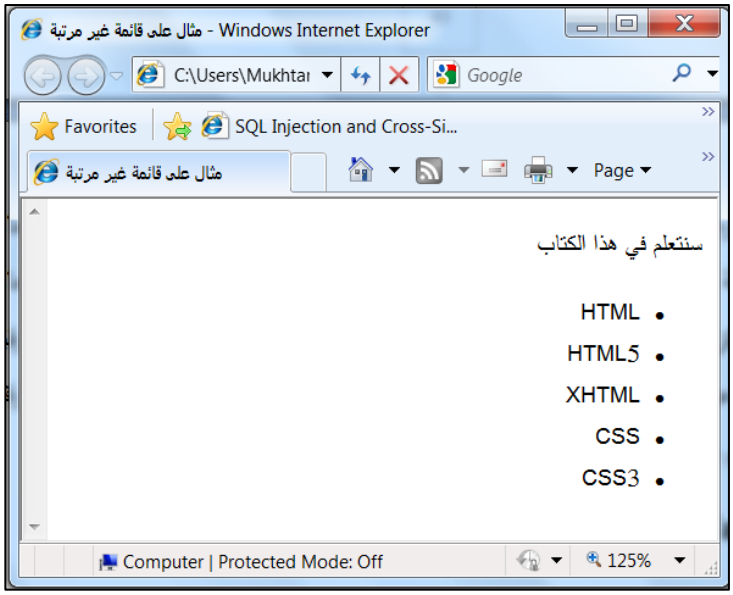
١- القوائم غير المرتبة Unordered List عبر الوسم <ul>.

٢- القوائم المرتبة Ordered List عبر الوسم <ol>.

بعد أن نقوم بتحديد نوع القائمة التي نرغب بإنشائها نضيف محتواها كمجموعة من عناصر <li> و التي تعني List Item، لنشاهد المثال التالي و الذي يعرض كيفيَّة إنشاء قائمة غير مرتبة بسيطة:

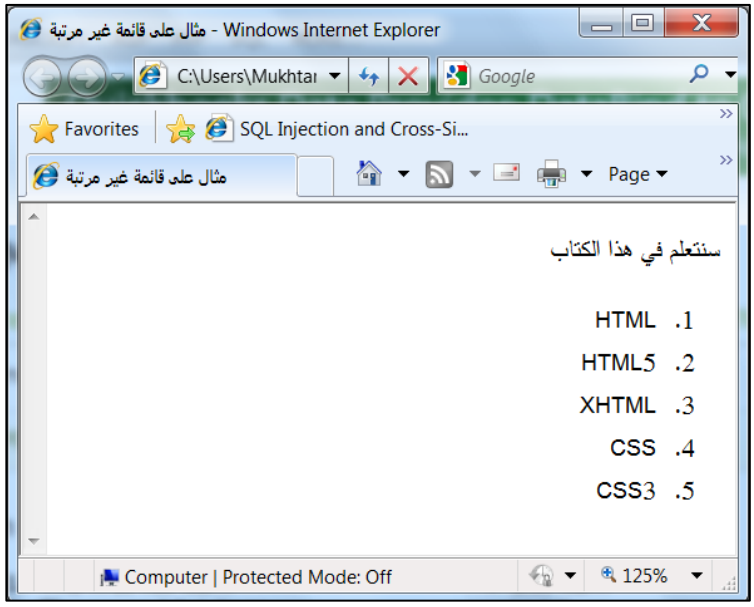
```
<html>
  <head>
     <title>مثال على قائمة غير مرتبة</title>
  </head>
  <body dir="rtl">
     <p>سنتعلم في هذا الكتاب</p>
     <ul>
          <li>HTML</li>
          <li>HTML5</li>
          <li>XHTML</li>
          <li>CSS</li>
          <li>CSS3</li>
     </ul>
  </body>
</html>
```

و الذي يبدو عند عرضه في المستعرض كمايلي:



الشكل ١٤ : مثال لقائمة غير مرتبة بسيطة

بتغيير وسمي البداية و النهاية لعنصر القائمة من <ul> و إلى <ol> و ستصبح القائمة مرتبة بالشكل التالي:



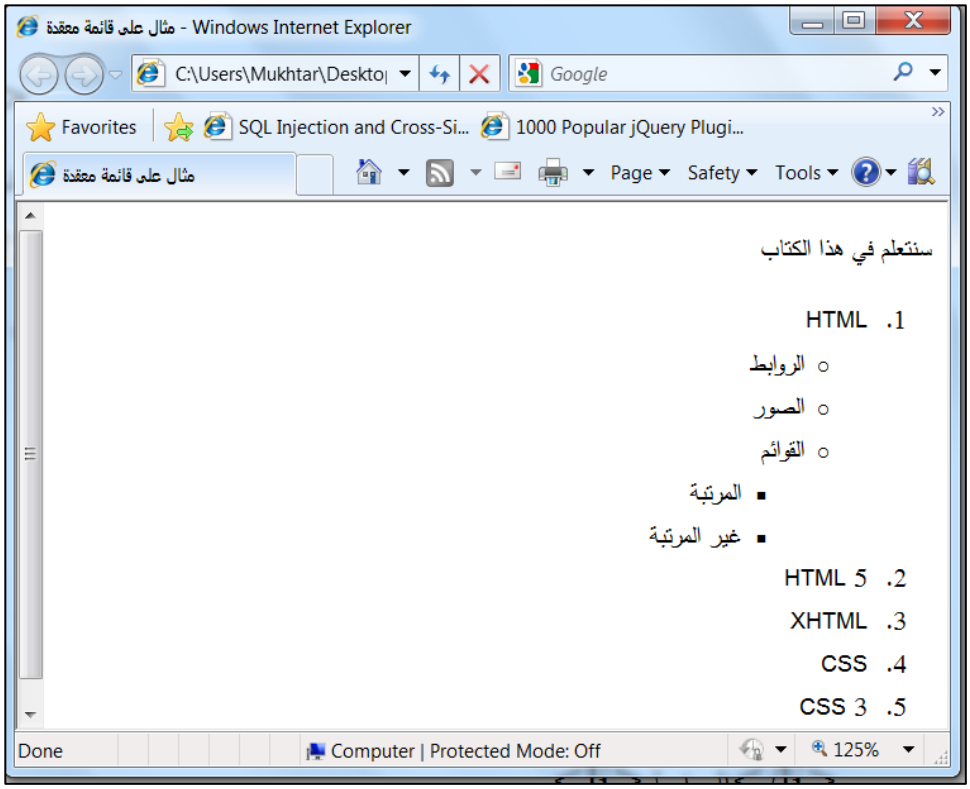
الشكل ١٥ : مثال على قائمة مرتبة بسيطة

بالطبع و كما ذكرت منذ بداية هذا الفصل فإن أي عنصر من عناصر HTML يمكن أن يحوي بين وسمي بدايته و نهايته أية مجموعة أخرى من العناصر و هذا يعني أننا نستطيع إنشاء قائمة تحوي قوائم فرعيَّة كعناصر لها، و المثال التالي يوضح هذه الفكرة:

```
<html>
  <head>
     <title>مثال على قائمة معقدة</title>
  </head>
  <body dir="rtl">
     <p>سنتعلم في هذا الكتاب</p>
     <ol>
          <li>
             HTML
             <ul>
                <li>الروابط</li>
                <li>الصور</li>
                <li>القوائم
                     <ul>
                          <li>المرتبة</li>
                          <li>غير المرتبة</li>
                     </ul>
                </li>
             </ul>
          </li>
          <li>HTML 5</li>
```

```
            <li>XHTML</li>
            <li>CSS</li>
            <li>CSS 3</li>
        </ol>
    </body>
</html>
```

يبدو المثال أعلاه في المستعرض كمايلي:



الشكل ١٦ : مثال على قائمة معقدة

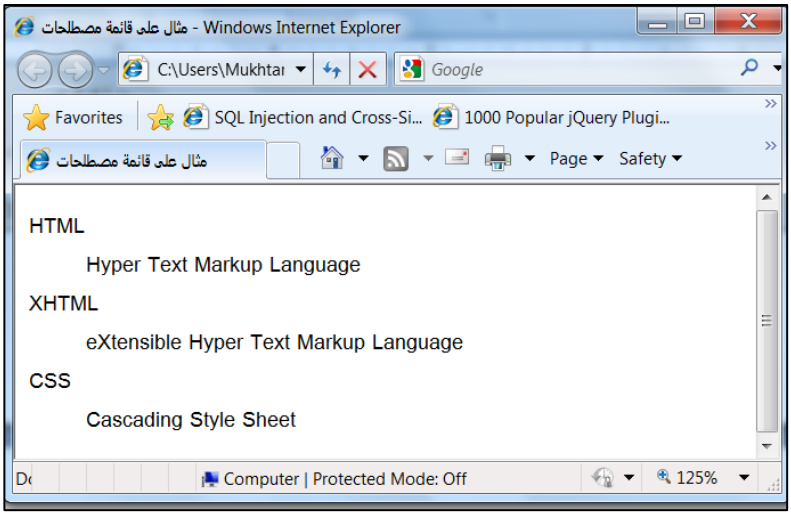
آخر ما يجب ذكره عن القوائم أنّه يمكن استخدام الواصفة type لتغيير شكل القائمة ففي حالة القوائم غير المرتبة يمكن إسناد إحدى القيم التالية إليها: square أو disc أو circle لوضع رمز المربع أو القرص أو الدائرة أمام كل عنصر على الترتيب، و في حالة القائمة المرتبة يمكن إسناد

إحدى القيم ١ أو A أو a أو I أو i إلى الواصفة type لتحويل الترقيم إلى ترقيم بأرقام عربية، ترقيم بأحرف إنجليزية كبيرة، ترقيم بأحرف إنجليزية صغيرة، ترقيم بأرقام لاتينية كبيرة، ترقيم بأرقام لاتينية صغيرة على الترتيب.

يوجد نوع خاص من القوائم يعرف بقوائم المصطلحات يتم إنشاؤه باستخدام الوسم <dl> لعنصر القائمة و الوسم <dt> لعنصر المصطلح و الوسم <dd> لعنصر تعريف المصطلح، لنشاهد المثال التالي:

```
<html>
 <head>
    <title>مثال على قائمة مصطلحات</title>
 </head>
 <body dir="rtl">
    <dl>
        <dt>HTML</dt>
        <dd>Hyper Text Markup Language</dd>
        <dt>XHTML</dt>
        <dd>eXtensible Hyper Text Markup Language</dd>
        <dt>CSS</dt>
        <dd>Cascading Style Sheet</dd>
    </dl>
 </body>
</html>
```

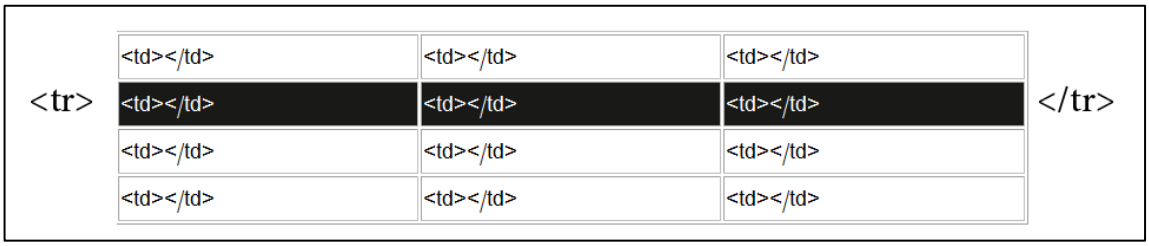
تبدو قائمة المصطلحات في المستعرض كمايلي:



الشكل ١٧ : هكذا تبدو قائمة المصطلحات في المستعرض

الجداول Tables

توفر لغة HTML آلية لإنشاء الجداول عبر العنصر <table> و يتم ذلك عبر إنشاء العنصر <table> أوّلاً و تمرير واصفتي العرض width و عرض الحدود border بالبكسل، و من ثمَّ إنشاء محتويات الجدول عبر عنصر أسطر الجدول table rows و الذي يتم تحقيقه باستخدام الوسم <tr>، ثمَّ يتم ذكر محتويات خلايا table data كل سطر من أسطر الجدول على حدى و ذلك عبر الوسم <td>، و لتسهيل حفظ الوسوم يمكن تمثيلها بشكل مرئي كمايلي:



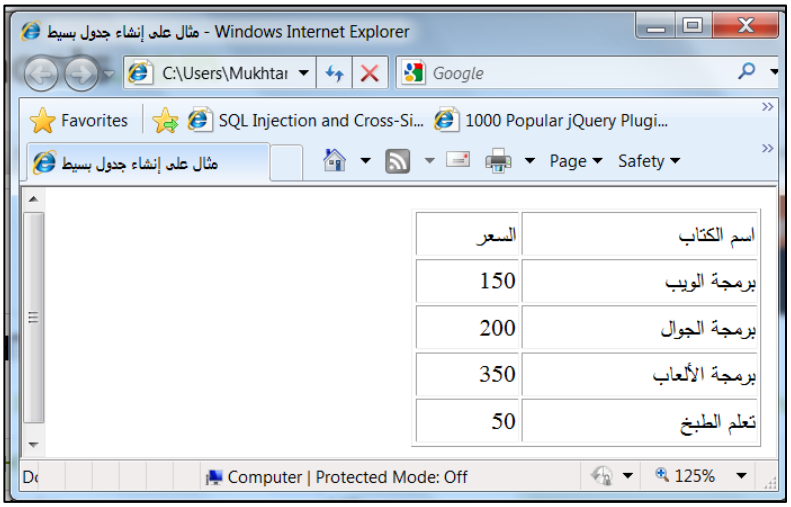
الشكل ١٨ : شكل تخيلي لتوضيح الوسوم المستعلمة لإنشاء الجدول

و لتطبيق هذا المفهوم عملياً بغية توضيحه أكثر دعنا نشاهد الشيفرة التالية التي تقوم بإنشاء جدول بسيط:

```
<html>
  <head>
     <title>مثال على إنشاء جدول بسيط</title>
  </head>
  <body dir="rtl">
     <table border="1" width="50%">
          <tr>
               <td>اسم الكتاب</td>
               <td>السعر</td>
          </tr>
          <tr>
               <td>برمجة الويب</td>
               <td>١٥٠</td>
          </tr>
          <tr>
               <td>برمجة الجوال</td>
               <td>٢٠٠</td>
          </tr>
          <tr>
               <td>برمجة الألعاب</td>
               <td>٣٥٠</td>
          </tr>
          <tr>
               <td>تعلم الطبخ</td>
               <td>٥٠</td>
```

```
            </tr>
        </table>
    </body>
</html>
```

يبدو هذا الجدول عند عرضه في مستعرض الويب كمايلي:



الشكل ١٩ : مثال على إنشاء جدول بسيط

كما رأينا فقد قمنا أولاً بإنشاء عنصر الجدول <table> بعرض width ٥٠% من العرض الكلي للصفحة (حاول تغيير حجم نافذة المستعرض لتلاحظ أن ذلك يبقى صحيحاً!)،و بعرض حدود border ١ بكسل، ثمَّ قمنا بإنشاء سطر جديد باستخدام الوسمين <tr></tr>، يحوي هذا السطر على خليتين <td></td> تحوي الأولى القيمة "اسم الكتاب" و تحوي الثانية القيمة "السعر"، ثمَّ قمنا بتكرار نفس الأسلوب مع الأسطر الأخرى التي تحوي بيانات الكتب، أظن أن الموضوع أصبح واضحاً الآن.

ملحوظة: يمكن إعطاء قيمة عرض الجدول بالبكسل أيضاً أو بأي واحدة قياس من الواحدات المدعومة في لغة HTML و التي سيتم ذكرها لاحقاً.

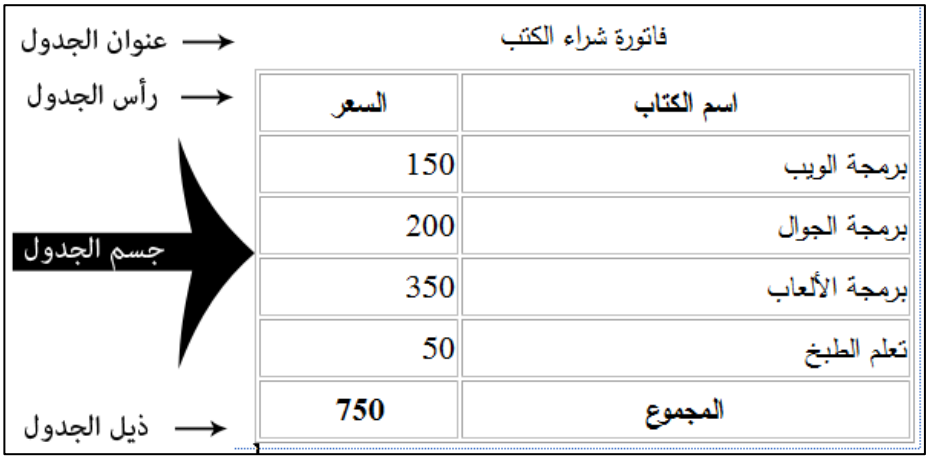
يقسم كل جدول من جداول HTML إلى أربع مناطق:

١- منطقة رأس الجدول Table Head: يُنشأ هذا القسم باستخدام الوسمين <thead> و </thead> و يحوي في الغالب على عنصر سطر <tr></tr> يحوي بين وسمي بدايته و نهايته عناوين أعمدة الجدول في خلايا من النوع <th> بدلاً من <td>، حيث تم الاصطلاح أن <td> تعني خلية بيانات table data في حين أن <th> تعني خلية عنوان table heading.

٢- منطقة جسم الجدول Table Body: يُنشأ هذا القسم باستخدام الوسمين <tbody> و </tbody> و يحوي على سطور البيانات و التي تنشأ باستخدام <tr> و بما أن خلاياها تحوي معلومات فإن الخلايا تنشأ باستخدام الوسوم <td>.

٣- منطقة ذيل الجدول Table Foot: يُنشأ هذا القسم باستخدام الوسمين <tfoot> و </tfoot> و يحوي في الغالب على عنصر سطر <tr></tr> يحوي بين وسمي بدايته و نهايته ملخصات عن أعمدة الجدول في خلايا من النوع <th>.

٤- عنوان الجدول Caption: يُضاف عنوان الجدول بكتابة نص العنوان بين وسمي <caption> و </caption>، يوضِّح الشكل التالي مناطق الجدول المختلفة:



فاتورة شراء الكتب

اسم الكتاب	السعر
برمجة الويب	150
برمجة الجوال	200
برمجة الألعاب	350
تعلم الطبخ	50
المجموع	**750**

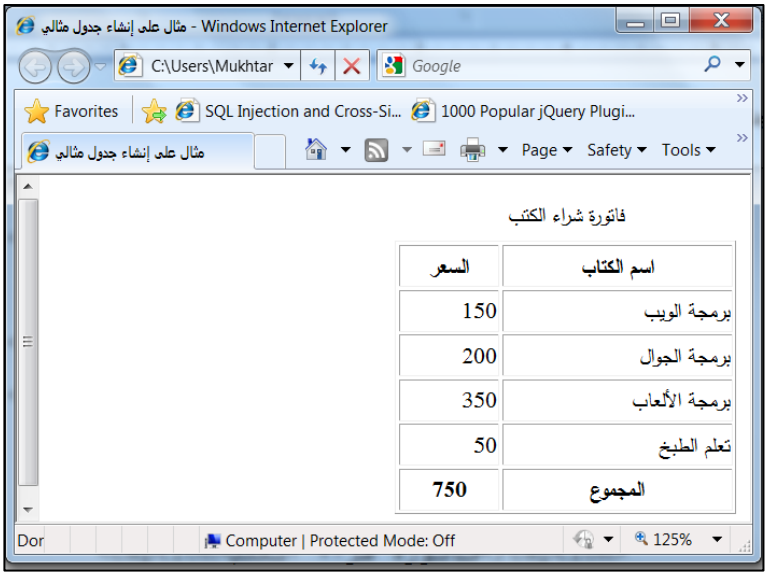
الشكل ٢٠ : رسم توضيحي لأقسام الجدول في HTML

لتحقيق الجدول المذكور في صفحتنا سنكتب شيفرة HTML التالية:

```
<html>
  <head>
     <title>مثال على إنشاء جدول مثالي</title>
  </head>
  <body dir="rtl">
     <table border="1" width="50%">

        <caption>فاتورة شراء الكتب</caption>

        <thead>
           <tr>
              <th>اسم الكتاب</th>
              <th>السعر</th>
           </tr>
        </thead>

        <tbody>
           <tr>
              <td>برمجة الويب</td>
              <td>150</td>
           </tr>
           <tr>
              <td>برمجة الجوال</td>
```

```
                    <td>200</td>
                </tr>
                <tr>
                    <td>برمجة الألعاب</td>
                    <td>350</td>
                </tr>
                <tr>
                    <td>تعلم الطبخ</td>
                    <td>50</td>
                </tr>
            </tbody>

            <tfoot>
                <tr>
                    <th>المجموع</th>
                    <th>750</th>
                </tr>
            </tfoot>

        </table>
    </body>
</html>
```

تبدو هذه الشيفرة في مستعرض الويب كمايلي:



الشكل ٢١ : مثال لإنشاء جدول HTML مثالي

كما تلاحظ فإنَّ العنوان يظهر أعلى الجدول، كما أن الأسطر الموجودة في منطقتي الرأس و الذيل تظهر بخط عريض bold.

الجداول غير البسيطة

في الحقيقة ما تم استعراضه في الفقرة الماضية كان جميلاً و بسيطاً أيضاً فبالنسبة لحالة مثل حالة فاتورة الشراء فإن جدولاً بعمودين و بمجموعة من الأسطر أمر بسيط و لكن ماذا عن الحالات الأعقد؟، ماذا عن الحالات التي لا تكون فيها الخلايا متساوية الأحجام؟، و لا الأسطر متساوية عدد الخلايا؟، تجيب HTML على هذه التساؤلات المحقّة عبر تقديم الواصفتين rowspan و colspan.

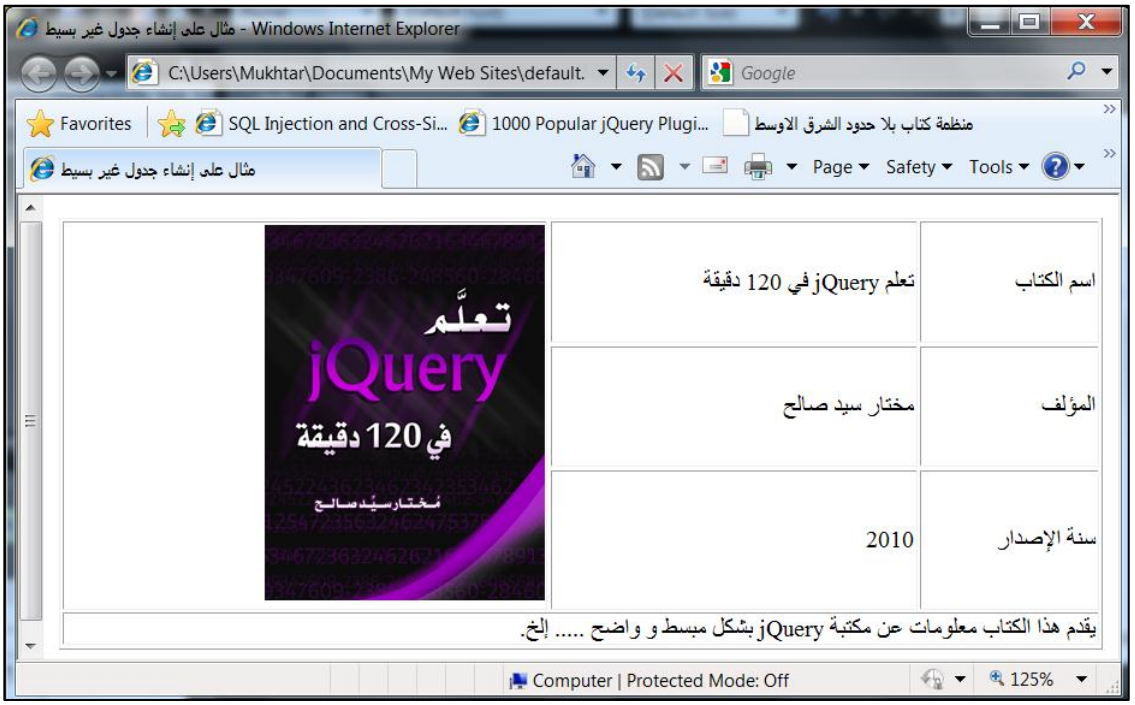
تستخدم الواصفتان rowspan و colspan مع وسم بداية الخلية <td> أو <th> و تسند إليهما قيمة رقمية صحيحة (١ أو ٢ أو ٣ .. إلخ) تغير من حجم الخلية فتجعلها بحجم خليتين أو ثلاث أو أربع .. إلخ، حيث أن rowspan تتحكم بعرض الخلية، أما colspan فتتحكم بارتفاع الخلية.

لنشاهد الشيفرة التالية على سبيل المثال:

```
<html>
  <head>
     <title>مثال على إنشاء جدول غير بسيط</title>
  </head>
  <body dir="rtl">
     <table border="1" width="100%">
        <tbody>
             <tr>
                <td>اسم الكتاب</td>
                <td>تعلم jQuery في ١٢٠ دقيقة</td>
                <td rowspan="3">
                  <img src="jQuery120.png">
                </td>
             </tr>
             <tr>
               <td>المؤلف</td>
                <td>مختار سيد صالح</td>
             </tr>
             <tr>
                <td>سنة الإصدار</td>
                <td>2010</td>
             </tr>
             <tr>
                <td colspan="3">
                     يقدم هذا الكتاب معلومات عن مكتبة jQuery
```

```
                                                بـشكل مـبسط و واضح ..... إلـخ.
                        </td>
                    </tr>
                </tbody>
            </table>
        </body>
    </html>
```

تبدو الشيفرة السابقة عند عرضها في المستعرض كمايلي:



الشكل ٢٢ : مثال على إنشاء جدول غير بسيط باستخدام الواصفتين rowspan و colspan

كما ترى فإنّ الجدول يتكون من أربعة سطور يحوي السطر الأوّل على ثلاث خلايا واحدة منها بارتفاع يعادل ارتفاع ثلاثة أسطر مجتمعةً (هذا ما تعنيه rowspan="3")، أمّا السطر الأخير من الجدول فيحوي على خلية واحدة بعرض أعمدة الجدول الثلاثة مجتمعةً (و هذا ما تعنيه

colspan="3")، أظن أن استخدام الواصفتين rowspan و colspan أصبح واضحاً الآن، و بالنسبة لأصدقائي الذين لم تتضح الصورة بالنسبة لهم فأعتذر منهم و أنصحهم أن يقوموا بتغيير القيم المذكورة في الواصفتين rowspan و colspan في المثال السابق و يلاحظوا الفرق الحاصل في أحجام الخلايا فهذا يساعد كثيراً في تشكيل تصوّر أفضل لديهم كما أظن.

ملحوظة: ليطمئن قلبي أحب أن أذكرك أنه بإمكانك وضع أي من عناصر HTML كمحتوى في خلايا الجدول.

النماذج Forms

تستخدم النماذج Forms في لغة HTML لاستقبال المدخلات Inputs من المستخدم بغية عرضها أو تخزينها أو القيام بعمليات معالجة معينة عليها، يتم إنشاء النماذج باستخدام الوسمين <form> و </form> الذان سيحويان بينهما مجموعة من عناصر الإدخال <input> مثل مربعات النصوص text boxes و عناصر الاختيار check boxes و الأزرار buttons و القوائم المنسدلة select lists.. إلخ.

يتم تزويد وسم بداية عنصر النموذج <form> بواصفتين أساسيتين الأولى هي الواصفة action و التي تحدد اسم الملف البرمجي الذي سيقوم بمعالجة البيانات المدخلة في النموذج و يكون الملف البرمجي مكتوباً بإحدى لغات البرمجة من طرف السيرفر Server Side مثل PHP أو ASP.NET أو JSP أو .. إلخ (و البرمجة من طرف السيرفر موضوع خارج نطاق هذا الكتاب)، أمّا الواصفة الثانية فهي الواصفة method و التي تحدد طريقة إرسال البيانات إلى الملف البرمجي المذكور سابقاً بإحدى طريقتين:

١ – الطريقة الأولى هي الطريقة get حيث يتم إرسال البيانات في شريط العنوان في المستعرض بعد اسم الملف البرمجي بشكل أزواج مفتاح/قيمة، انظر للرابط التالي على سبيل المثال:

http://www.site.com/file.php?name=Mukhtar&age=23&job=ComputerEngineer

كما تلاحظ فإن الملف البرمجي في هذه الحالة هو http://www.site.com/file.php أمَّا القيم المرسلة فهي:

name = Mukhtar و age = 23 و job = ComputerEngineer

٢ – الطريقة الثانية هي الطريقة post حيث يتم إرسال البيانات بشكل غير ظاهر في شريط العنوان (في الحقيقة يتم إرسالها مع ترويسة طلب HTTP).

بما أننا –في هذا الكتاب– سنهتم بكيفية إنشاء النماذج فقط و لن نناقش كيفية معالجة الطلبات فبإمكانك الآن تجاهل ما تم ذكره عن الواصفة method و التركيز على كيفية بناء النماذج.

بعد إنشاء العنصر <form> سنقوم بوضع مجموعة من عناصر الإدخال بين وسمي بدايته و نهايته باستخدام الوسم <input> و طبعاً سيتم تزويد كل عنصر إدخال باسم فريد يُسنَد كقيمة للواصفتين name و id، و من ثمَّ سيتم تحديد نوع عنصر الإدخال عبر الواصفة type و الجدول التالي يوضح القيم الممكنة و ما يمثله كل منها:

القيمة	ما تمثله
button	زر أوامر
checkbox	عنصر اختيار متعدد
file	مربع لاختيار ملف بغية رفعه إلى الموقع
hidden	عنصر إدخال مخفي
password	عنصر إدخال كلمة المرور
radio	عنصر اختيار فردي
reset	زر لإعادة ضبط القيم الافتراضية لعناصر

	الإدخال كاملةً في النموذج
submit	زر موافق (إرسال الطلب)
text	مربع إدخال نص

الجدول ٣ : القيم المختلفة للواصفة type الخاصة بعنصر <input>

لا تقلق إنْ بدت لك المعلومات كثيرة و معقدة و تابع معي الشيفرة التالية:

```
<html>
  <head>
     <title>
        مثال على إنشاء نموذج بسيط
     </title>
  </head>
  <body dir="rtl">
     <form method="get" action="file.php">
       الاسم :
       <input type="text" name="nameText" id="nameText" />
       <br/>
       كلمة المرور :
       <input type="password" name="pass" id="pass" />
       <br/>
       <input type="submit" name="okButton" id="okButton"
value="موافق" />
     </form>
  </body>
</html>
```

و التي تبدو عند عرضها في مستعرض الويب و بعد إدخال بعض البيانات كمايلي:

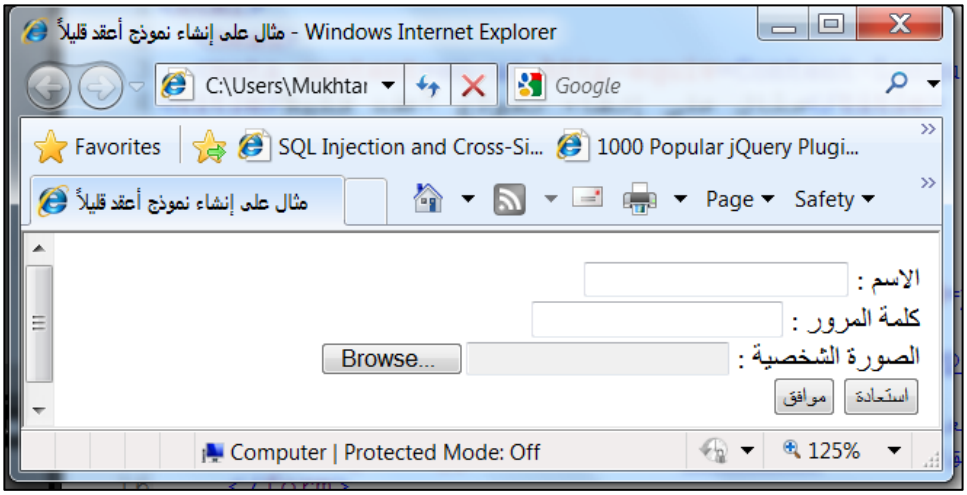
الشكل ٢٣ : مثال على إنشاء نموذج بسيط

حسناً الموضوع بسيط كما رأيت، دعنا نقوم بتطوير نموذجنا هذا ليشمل مجموعة أخرى من عناصر الإدخال على سبيل التجربة، انظر للشيفرة التالية:

```
<html>

  <head>

     <title>مثال على إنشاء نموذج أعقد قليلاً</title>

  </head>

  <body dir="rtl">

     <form method="get" action="file.php">

          الاسم: <input type="text" name="nameText"
id="nameText" />

          <br/>

          كلمة المرور: <input type="password" name="pass"
id="pass" />

          <br/>
```

```
            الصورة الشخصية : <input type="file" name="pic"
id="pic">

            <br/>

            <input type="reset" name="resBtn" id="resBtn"
value="استعادة" />

            <input type="submit" name="okButton" id="okButton"
value="موافق" />

      </form>

  </body>

</html>
```

تبدو الشيفرة السابقة في مستعرض الويب كمايلي:



الشكل ٢٤ : إنشاء نموذج أعقد بقليل

مهلاً! .. و لكن ماذا عن عناصر الاختيار radio و checkbox ؟

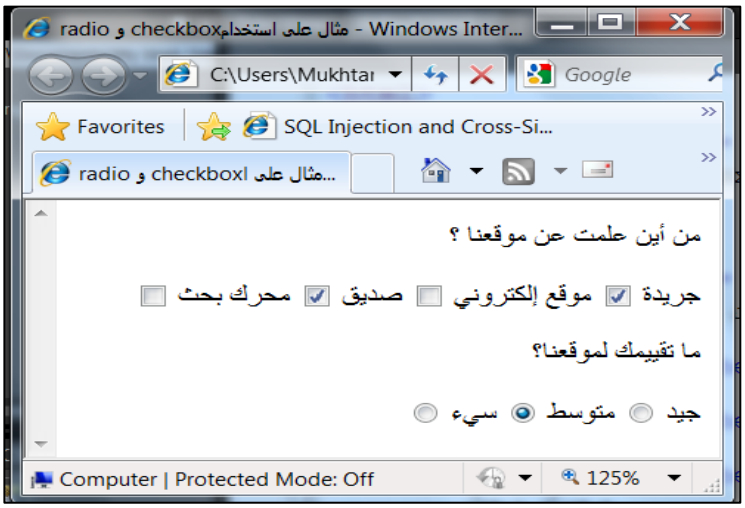
حسناً، في الواقع يتم استعمال هذه العناصر في النماذج بغية تمكين المستخدم من الإجابة على سؤال محدد و ذلك باختيار جواب واحد فقط من عدة إجابات محتملة (في حالة عنصر radio) أو الإجابة على سؤال محدد باختيار أكثر من جواب (في حالة عنصر checkbox) و يتم استخدام

هذه العناصر بنفس الطريقة السابقة مع ملحوظة إعطاء جميع العناصر التي تمثل إجابة لنفس السؤال القيمة ذاتها للواصفة name و قيماً مختلفة للواصفة value، لنشاهد المثال التالي:

```
<html>
  <head>
      <title> مثال على استخدام checkbox و radio</title>
  </head>
  <body dir="rtl">
    <form method="post" action="file.php">
      <p>من أين علمت عن موقعنا ؟</p>
      جريدة
      <input type="checkbox" name="wrKnow" value="nPaper" />
      موقع إلكتروني
      <input type="checkbox" name="wrKnow" value="wbSite" />
      صديق
      <input type="checkbox" name="wrKnow" value="frnd" />
      محرك بحث
      <input type="checkbox" name="wrKnow" value="srEng" />
      <p>ما تقييمك لموقعنا؟</p>
      جيد
      <input type="radio" name="rate" value="good" />
      متوسط
      <input type="radio" name="rate" value="mid" />
      سيء
      <input type="radio" name="rate" value="bad" />
```

```
</form>
  </body>
</html>
```

و الذي يبدو في المستعرض كمايلي:



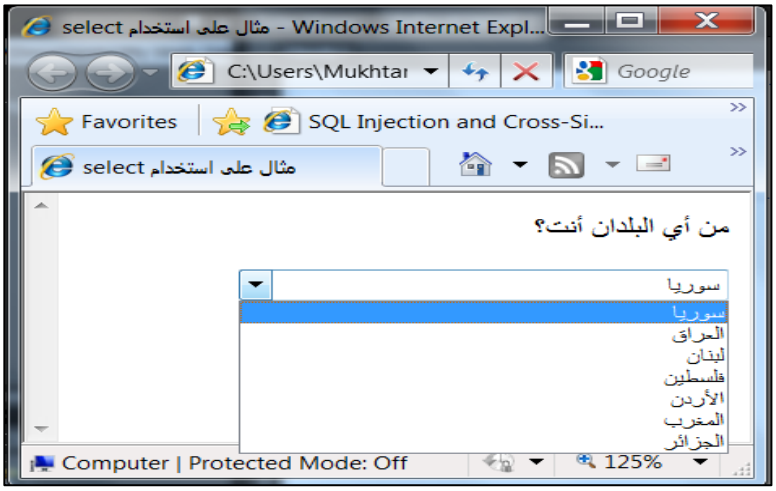
الشكل ٢٥ : مثال على استخدام checkbox و radio

هناك طريقة أخرى لتحقيق شيء مشابه لعنصر radio من حيث الهدف و لكنه يستخدم في حال وجود بدائل كثيرة (أجوبة كثيرة محتملة للسؤال) و هو عنصر القائمة المنسدلة و يتم تحقيقه عبر الوسم <select> أمَّا عناصر القائمة فيتم تحقيقها من خلال الوسم <option> و يتم تحديد أحد هذه العناصر كخيار افتراضي عن طريق تمرير الواصفة "selected="selected في وسم بدايته، لنشاهد المثال التالي:

```
<html>
  <head>
      <title>select مثال على استخدام</title>
  </head>
  <body dir="rtl">
```

```
    <form method="post" action="file.php">
        <p>من أي البلدان أنت؟</p>
        <select name="nationality">
            <option selected="selected">سوريا</option>
            <option>العراق</option>
            <option>لبنان</option>
            <option>فلسطين</option>
            <option>الأردن</option>
            <option>المغرب</option>
            <option>الجزائر</option>
        </select>
    </form>
  </body>
</html>
```

و الذي يبدو في المستعرض كمايلي:



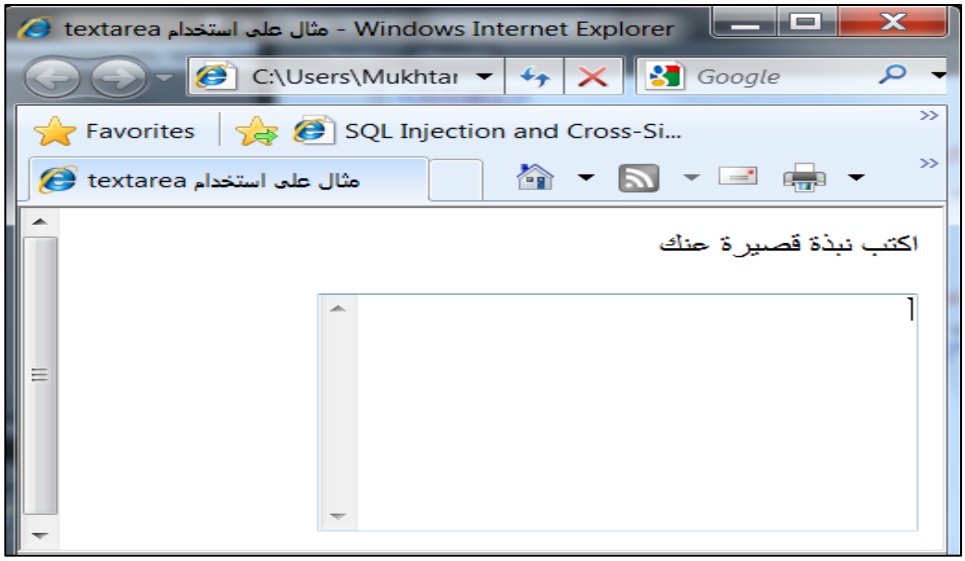
الشكل ٢٦ : مثال على استخدام عنصر select

بقي عنصر إدخال أخير لمناقشته هنا و هو العنصر <textarea> و الذي يستخدم لإدخال قيمة نصية متعددة الأسطر على عكس العنصر <input> من النوع text و الذي يستخدم لإدخال قيمة نصية وحيدة السطر.

يمرر للعنصر <textarea> قيم عددية صحيحة عبر الواصفتين cols و rows لتحديد عرضه و ارتفاعه، انظر الشيفرة التالية:

```
<html>
  <head>
      <title>
            مثال على استخدام textarea
      </title>
  </head>
  <body dir="rtl">
      <form method="post" action="file.php">
         <p>اكتب نبذة قصيرة عنك</p>

         <textarea cols="30" rows="10" name="bio">
         </textarea>

      </form>
  </body>
</html>
```

و الذي يبدو عند استعراضه في مستعرض الويب كمايلي:



الشكل ٢٧ : مثال على استخدام العنصر textarea

تقسيم عناصر الإدخال إلى مجموعات

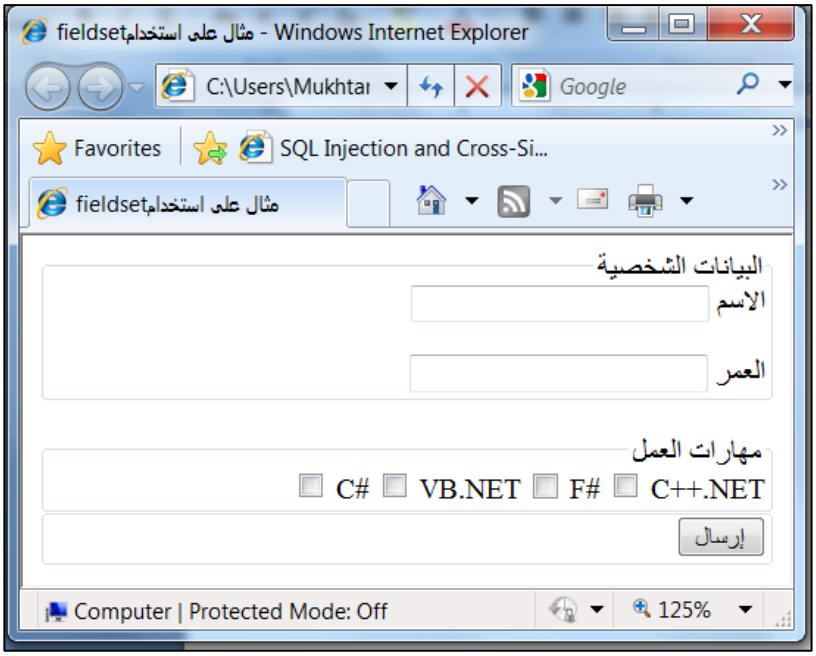
يمكن تقسيم عناصر الإدخال في النماذج الكبيرة إلى أقسام منطقية يوضع كل منها في إطار مستقل و هذا ما توفره لغة HTML باستخدام الوسمين <fieldset> و </fieldset> اللذان سيحويان بينهما مجموعة عناصر الإدخال و يتم تزويد كل مجموعة بعنوان توضيحي عبر الوسم <legend>، كما يمكن منح كل عنصر من عناصر الإدخال عنواناً يدل على ماهيَّة المعلومات المدخلة فيه عن طريق الوسم <label> بتمرير اسم عنصر الإدخال إلى واصفة for الخاصة بعنصر العنوان <label>، دعنا نشاهد المثال التالي:

```
<html>
  <head>
      <title>
            fieldset مثال على استخدام
      </title>
  </head>
```

```
  <body dir="rtl">
      <form method="get" action="">

          <fieldset>
              <legend>البيانات الشخصية</legend>
              <label for="nameTXT">الاسم</label>
              <input type="text" name="nameTXT" id="nameTXT"
maxlength="50" />
              <br/><br/>
              <label for="ageTXT">العمر</label>
              <input type="text" name="ageTXT" id="ageTXT"
maxlength="2" />
          </fieldset>

          <br/>

          <fieldset>
              <legend>مهارات العمل</legend>
              C#
              <input type="checkbox" name="prog" value="cs"/>
              VB.NET
              <input type="checkbox" name="prog" value="vb"/>
              F#
              <input type="checkbox" name="prog" value="fs"/>
              C++.NET
              <input type="checkbox" name="prog" value="cpp"/>
```

```
            </fieldset>
            <fieldset>
                  <input type="submit" value="إرسال" />
            </fieldset>

      </form>
  </body>
</html>
```

تبدو الشيفرة السابقة كمايلي في مستعرض الويب:



الشكل ٢٨ : مثال على استخدام fieldSet

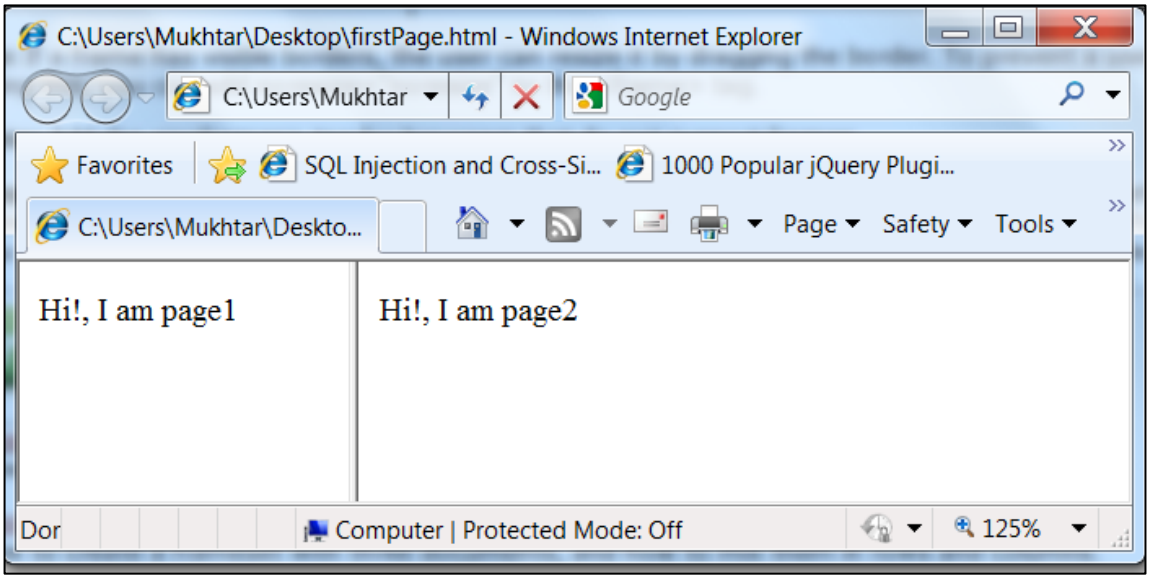
الإطارات Frames

تسمح لك لغة HTML بتضمين صفحات ويب ضمن صفحات أخرى، و ذلك عبر استخدام الإطارات Frames.

يتم استعمال الإطارات في الصفحة على مرحلتين، الأولى بإنشاء حاوية الإطارات باستخدام الوسم <frameset> و الذي يحدد بدايةً عدد الإطارات التي سيتم تضمينها، و الحجم الذي سيأخذه كل إطار من حجم الصفحة الكلي، و يتم تحديد الحجم بالبكسل أو بالنسبة المئوية، أمّا المرحلة الثانية فيتم فيها إضافة الإطارات بين وسمي البداية و النهاية للعنصر <frameset> و ذلك باستخدام الوسم <frame> و إسناد مسار الصفحة التي سيحويها الإطار للواصفة src ، لنشاهد المثال التالي لصفحة ويب تضم إطارين الأوّل بحجم قدره ٣٠% من حجم الصفحة الكلي و الثاني بالحجم الباقي من الصفحة:

```
<html>

      <frameset cols="30%,*">
            <frame src="page_1.htm" />
            <frame src="page_2.htm" />
      </frameset>

</html>
```

يبدو هذا في مستعرض الويب كمايلي:

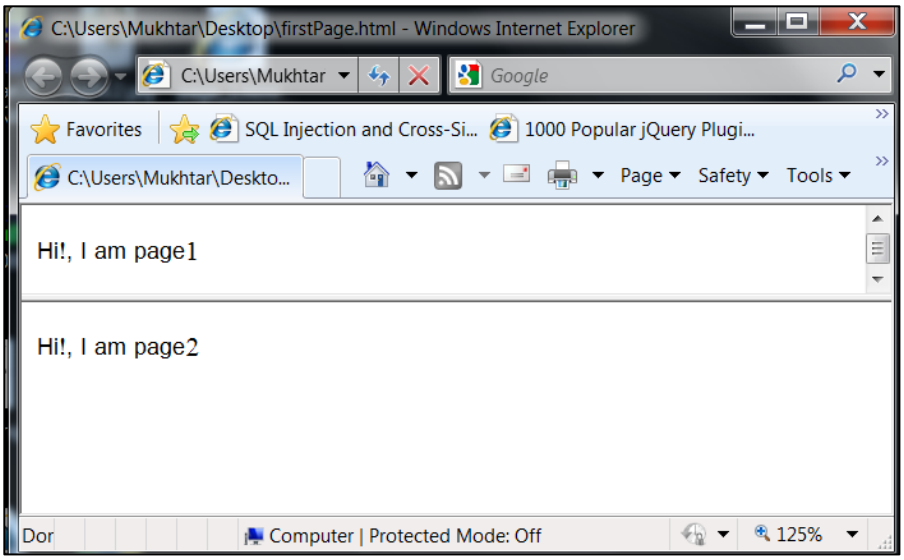


الشكل ٢٩ : استخدام الإطارات بشكل عمودي

ملحوظة: الصفحة التي تستعمل فيها الإطارات لا تحوي على وسم <body>.

يمكن أن يقوم المستخدم بتغيير حجم الإطارات عن طريق تحريك الخط الفاصل بينهما و الظاهر بوضوح في الشكل السابق (انظر الشكل ٢٩)، يمكن إلغاء هذا السلوك للإطارات عن طريق إضافة الواصفة "noresize="noresize إلى وسوم <frame>.

في المثال السابق كانت الإطارات بشكل عمودي و لهذا تم استعمال الواصفة cols مع الوسم <frameset>، و يمكن جعل الإطارات بشكل أفقي عن طريق استبدال الواصفة cols بالواصفة rows، لتصبح الصفحة كمايلي في مستعرض الويب:



الشكل ٣٠ : استخدام الإطارات بشكل أفقي

تذكرة: يمكن جعل الرابط يعرض محتواه في إطار معيَّن عن طريق إسناد اسم الإطار إلى واصفة target الخاصة بالرابط <a> (انظر فقرة الروابط التشعبيَّة Links).

في حين أن الوسمين <frameset> و <frame> يستخدمان لإنشاء صفحة أم تضم مجموعة من الصفحات فقط في إطاراتها فإن لغة HTML توفر الوسم <iframe> لتضمين صفحة ويب في

صفحة أخرى بشكل متناسق مع بقية محتوى الصفحة، و يستخدم الوسم <iframe> بالصيغة التالية:

```
<iframe src="URL"></iframe>
```

حيث أنَّ URL هو مسار الصفحة التي سيتم تضمينها، و يمكن ضبط حجم الإطار عن طريق الواصفتين height و width بالبكسل أو بالنسبة المئوية.

آخر ما يجب ذكره هنا أن هناك مستعرضات لا تدعم الإطارات، كما أنّ الإطارات ستلغى مستقبلاً من اللغة، على الرغم من أن بعض المستعرضات ما زالت تدعمها إلى تاريخ كتابة هذه الأسطر.

عناصر الـ Meta

ذكرت في بداية هذا الفصل أنَّ هناك عناصر غير مرئية تضاف في منطقة رأس الصفحة بين وسمي <head> و </head> تستخدمها محركات البحث من أجل تصنيف صفحتك و عرضها ضمن نتائج البحث، و هذه العناصر هي عناصر الـ meta، و بالطبع فمن الأهمية بمكان جعل محركات البحث تصنّف صفحتك بشكل صحيح كي لا تخسر معظم زوار موقعك القادمين إثر عملية بحث ما على أحد محركات البحث.

يتم تحقيق عناصر الـ meta من خلال الوسم meta و بتمرير اسم name و محتوى content لعنصر الـ meta كواصفتين له، و لعلَّ أهم عنصري meta في أي صفحة ويب هما العنصران اللذان يحملان الاسمين keywords و description، و اللذان يستخدمان كمايلي:

```
<html>

      <head>

            <meta name="keywords" content="xhtml website test
meta head"/>

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

```
    </head>
    <body>محتوى الصفحة النهائي فعليًّا سيكون هنا</body>
</html>
```

حيث يعرض عنصر الـ Meta ذو الاسم Keywords مجموعة من الكلمات المفتاحية التي تظن أن المستخدمين سيبحثون عن أحدها ضمن المستعرض و الموجودة في صفحتك، أما عنصر الـ Meta ذو الاسم Description فيحوي وصفاً مختصراً عن المحتوى الذي تقدمه صفحتك.

بالطبع عند عرض هذه الصفحة في مستعرض الويب لا تتوقع أن تشاهد محتوى عناصر الـ Meta ظاهراً فمهمّة هذه العناصر كما قلنا هو تقديم بيانات عن البيانات الموجودة ضمن صفحتك.

و بانتهاء حديثنا عن عناصر الـ Meta نكون قد أنهينا –بفضل الله– مناقشة لغة HTML بأغلب وسومها، و سنتابع في الصفحات التالية استعراض وسوم HTML العصريَّة و التي تعرف باسم HTML5 ، و لكن قبل ذلك سنستعرض لغة HTML الموسَّعة أو ما يُعرف بـ XHTML، أمَّا الآن فأرى أن نعرض جدولاً بوسوم HTML التقليديَّة ليكون مرجعاً سريعاً عند الحاجة.

الوسم	الشرح
<!-- ... -->	تعليق
<!DOCTYPE>	لتحديد نوع المستند (سيتم الحديث عنه عند عرض XHTML)
<a>	رابط تشعبي
<abbr>	اختصار لمصطلح إنجليزي مركب
<acronym>	اختصار لمصطلح إنجليزي
<address>	إضافة معلومات اتصال بمالك الصفحة
<applet>	تضمين تطبيق Applet ضمن الصفحة
<area />	تعريف منطقة ضمن الخريطة الصوريَّة
<b>	نص عريض
<base />	تعريف بادئة افتراضية لمسارات الروابط
<basefont />	تعريف نوع/لون/حجم خط افتراضي لنصوص للصفحة
<bdo>	تعريف اتجاه النص
<big>	نص كبير الحجم
<blockqoute>	اقتباس طويل
<body>	جسد الصفحة
 	سطر جديد
<button>	زر إرسال
<caption>	عنوان الجدول
	لتوسيط المحتوى
<cite>	اقتباس متوسط
<code>	شيفرة برمجية
<col />	تعريف واصفات مشتركة لأعمدة الجدول
<colgroup>	تعريف مجموعة من أعمدة الجدول لتطبيق واصفات مشتركة عليها

<dd>	وصف لمصطلح ضمن قائمة المصطلحات
<del>	نص مشطوب
<dfn>	تعريف مصطلح ضمن قائمة المصطلحات
<dir>	قائمة مجلدات
** **	قسم من المستند
<dl>	قائمة المصطلحات
<dt>	عنصر قائمة خاص بقائمة المصطلحات
****	نص مشدد عليه
<fieldset>	مجموعة حقول في نموذج
<font>	تعريف حجم/نوع/لون الخط لنص معين
<form>	نموذج
<frame />	إطار
<frameset>	مجموعة إطارات
<h1> إلى <h6>	عناصر العناوين
<head>	رأس المستند
<hr />	خط أفقي
<html>	عنصر بداية المستند
<i>	مائل
<iframe>	إطار كعنصر من عناصر الصفحة
<img />	صورة
<input />	عنصر إدخال
<ins>	نص تحته خط
<isindex>	إنشاء فهرس قابل للبحث يتعلق بالصفحة
<kbd>	نص بخط حاسوبي برمجي
<label>	عنوان لعنصر ما

<legeng>	عنوان لمجموعة عناصر الإدخال <fieldset>
** **	عنصر القائمة
<link />	ربط المستند بمستند آخر (سيمر معنا أحد استعمالاته في الفصل الثاني)
<map>	تعريف خريطة صورية
<menu>	تعريف قائمة
<meta />	بيانات تعريفية عن ماهيَّة محتوى المستند
<noframes>	لإنشاء بديل لمحتوى الإطار في المستعرضات التي لا تدعم الإطارات
<noscript>	لإنشار بديل لمحتوى السكربت في المستعرضات التي لا تدعم السكربت
<object>	تضمين كائن برمجي
** **	قائمة مرتبة
<optgroup>	مجموعة من الخيارات المرتبطة منطقياً ضمن خيارات القائمة المنسدلة
<option>	خيار في قائمة منسدلة
<p>	نص
<param />	وسيط للكائن البرمجي
<pre>	نص منسق مسبقاً
<q>	اقتباس قصير
<s>	نص مشطوب
<samp>	نموذج لشيفرة برمجية
<script>	سكربت برمجي (Java Script أو jQuery غالباً)
<select>	قائمة منسدلة
<small>	نص صغير الحجم
****	تعريف قسم سطري من المستند
<strike>	نص مشطوب
****	نص عريض
<style>	تعريف ورقة أنماط (يناقش في الفصل القادم بالتفصيل)

<sub>	نص سفلي
<sup>	نص علوي
<table>	جدول
<tbody>	جسد الجدول
<td>	خلية بيانات جدول
<textarea>	عنصر إدخال نص متعدد الأسطر
<tfoot>	ذيل الجدول
<th>	خلية عنوان رأس الجدول
<thead>	رأس الجدول
<title>	عنوان المستند
<tr>	سطر الجدول
<tt>	نص teletype
<u>	نص تحته خط
** **	قائمة غير مرتبة
<var>	جزء متغير من النص
<xmp>	نص بتنسيق مسبق

الجدول ٤ : دليل وسوم HTML

XHTML

إنَّ XHTML هي اختصار لـ eXtensible HTML و التي تعني HTML القابلة للتوسُّع (البعض يدعوها HTML الموسَّعة)، و هي لا تختلف عن HTML التقليديَّة من ناحية الوسوم و لا من ناحية الواصفات، و لكن ببساطة يمكن اعتبارها معايير أكثر صرامةً لكتابة مستندات HTML بشكل يتوافق و قواعد لغة XML (eXtensible Markup Language)، و هذا ما يعطي صفحاتنا توافقيَّة أكبر مع مستعرضات الويب، كما يدل الالتزام بمعايير XHTML على احترافية أكبر في العمل، و بالطبع لا أريد لتفكيرنا –كمطوِّري ويب– أن يبقى محصوراً في مستعرض الويب الخاص بالحاسوب، فلنفكِّر في مستعرضات الويب الخاصة بالهواتف الذكيَّة أو بالأجهزة الصغيرة الأخرى و لنسأل أنفسنا: هل ستبدو صفحاتنا كما نريد على كل مستعرضات الويب؟، هذا ما تحاول XHTML الإجابة عنه.

ففي حين أن شبكة الانترنت تحوي مليارات صفحات الويب فثَمَّةَ نسبة كبيرة من هذه الصفحات مكتوبة بشكل رديء – للأسف – يشبه مايلي على سبيل المثال:

```
<hTml>
      <heaD>
            <title> رديـئة HTML صفـحة</title>
      <BODY>
             <h1>Bad HTML
             <p>This is a paragraph
      </boDy>
```

مع أن الشيفرة أعلاه ستعمل على بعض المستعرضات إلا أنها باعتقادي لن تؤمِّن أدنى فرصة عمل لمن قام بكتابتها لأنها لا تعكس أدنى معرفه بكتابة HTML و أفضل ما يمكن أن يقال عنها أنها مكتوبة بشكل رديء، و قد ظهرت XHTML للحد من انتشار صفحات كالصفحة أعلاه.

قواعد XHTML

تتلخص XHTML في مجموعة من القواعد البسيطة التي ما إنْ يتم تطبيقها حين كتابة أي مستند HTML فسيُعَدُّ المستند مستند XHTML، و أستطيع تلخيص قواعد XHTML في مجموعة النصائح التالية[٥]:

- لا تنسَ وسم الإغلاق.
- اكتب وسوم و واصفات صفحتك بأحرف صغيرة Lower Case.
- راعِ أن يكون ترتيب وسوم الإغلاق متناظراً مع وسوم الفتح.
- ضع القيم المُسنَدة للواصفات بين علامات اقتباس.
- لا تستغنِ عن أي من وسوم <html> و <body> و <head>.
- ضع تعريف نوع المستند DOCTYPE في بداية صفحتك (سيعرض في الفقرة التالية).

تعريف نوع المستند <DOCTYPE!>

تنص معايير XHTML على أن كل صفحات الويب يجب أن تحوي في أول سطر منها على سطر التعريف <DOCTYPE!> و الذي يعني تعريف نوع المستند Document Type Definition أو DTD اختصاراً، و ذلك أن مستعرضات الويب تستخدم هذا السطر كمرجعيَّة قواعديَّة لصيغة الكتابة و لأنواع الوسوم في الصفحة، إضافة إلى استخدامها الواصفة xmlns الخاصة بوسم البداية <html> و التي تصف فضاء أسماء Name Space الصفحة.

تعرض الشيفرة التالية عنصر تعريف نوع المستند الذي تصادفه في أغلب صفحات XHTML و الذي يجب أن تكتبه كأول سطر من صفحتك إضافةً للواصفة xmlns، و بهذا يصبح الشكل العام لمستند XHTML كمايلي:

---

[٥] إنَّ أغلب ما سيتم ذكره من القواعد كان قد مر معنا في الصفحات السابقة على شكل ملحوظات.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

      <head>محتوى رأس الصفحة هنا</head>

      <body>محتوى جسد الصفحة هنا</body>

</html>
```

يعتبر النوع <DOCTYPE!> المعروض أعلاه أحد أنواع المستند و يسمَّى النوع Transitional و الفقرة التالية تناقش الاختلاف بينه و بين الأنواع الأخرى.

ملحوظة: <DOCTYPE!> ليس وسماً من وسوم اللغة و إنَّما هو سطر تعريف المستند.

الأنواع المختلفة للمستند

في XHTML هناك أربعة أنواع مختلفة للمستند و هذه الأنواع هي:

١. XHTML 1.0 Strict:

يحوي هذا النوع جميع وسوم HTML عدا الوسوم <applet> و <basefont> و و <dir> و <font> و <iframe> و <isindex> و <menu> و <noframes> و <s> و <strike> و <u>، و لا يسمح هذا النوع باستخدام إطارات <frameset>، و سطر تعريف المستند من هذا النوع هو:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

٢. XHTML 1.0 Tansitional:

يحوي هذا النوع جميع وسوم HTML و لكنَّه لا يسمح كسابقه باستخدام إطارات <frameset>، و سطر تعريف المستند من هذا النوع هو:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
```

٣. XHTML 1.0 Frameset:

يحوي جميع وسوم HTML و يسمح باستخدام الإطارات، و سطر تعريف المستند من هذا النوع هو:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

٤. XHTML 1.1:

يماثل النوع Strict و لكنّه يتيح لك إضافات وحدات برمجيَّة خاصة (وحدات Ruby مثلاً) على عكس المذكور، و سطر تعريف المستند من هذا النوع هو:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

التحقق من صحة صفحات XHTML

ننهي الحديث عن XHTML بالتذكير أنه بإمكان مطوِّر الويب (و غيره) أن يتحقق من مطابقة صفحة ويب ما لمعايير XHTML عبر الرابط:

http://www.w3schools.com/xhtml/default.asp

ملحوظة: يعكس وضع رابط التحقق من مطابقة الصفحة لمعايير XHTML في صفحتك نوعاً من الثقة بقدراتك البرمجية باعتقادي.

HTML5

ظهرت لغة HTML التي شرحناها في الصفحات السابقة (المعروفة في الأوساط التقنية بالإصدار HTML 4.01) في عام ١٩٩٩م، و في الحقيقة فإن تطورات كثيرة حصلت على مفهوم الويب منذ ذلك الحين إلى يومنا هذا، و أصبحنا نسمع بما يعرف بالـ Web 2.0، كما أصبح من النادر أن نجد موقع ويب (أعني المواقع الجيدة) لا يقدم خدمات مثل تشغيل الفيديو و الصوت و غير ذلك، و هذا ما استوجب تطوير إصدار جديد من لغة HTML بإضافة مجموعة جديدة من الوسوم و الواصفات إلى اللغة القديمة ليظهر الإصدار الجديد بالاسم HTML 5.

و من الميزات الجديدة التي جاءت بمجيء HTML 5 ظهور وسوم خاصة لعرض الفيديو و الصوت و وسوم خاصة لكتابة المقالات أو الأخبار و وسوم خاصة للرسم إضافةً إلى إضافة مجموعة جديدة من عناصر الإدخال الخاصة بالنماذج و غير ذلك مما سيعرض في الفقرات القادمة.

مستعرضات الويب الداعمة لـ HTML 5

لأن HTML 5 ظهرت منذ فترة قصيرة نسبياً (بدأ العمل على وضع معاييرها عام ٢٠٠٦) فإن معظم مستعرضات الويب القديمة لا تدعمها للأسف، و لكن المستعرضات الجديدة ستدعمها بكل تأكيد، فـ HTML 5 –برأيي– ستكون لغة بناء صفحات الويب المعيارية في المستقبل القريب و لهذا فقد بدأت بعض المستعرضات بدعمها فعلاً مثل الإصدارات الأخيرة من مستعرضات: Google Chrome و Firefox و Opera و Internet Explorer و Safari.[٦]

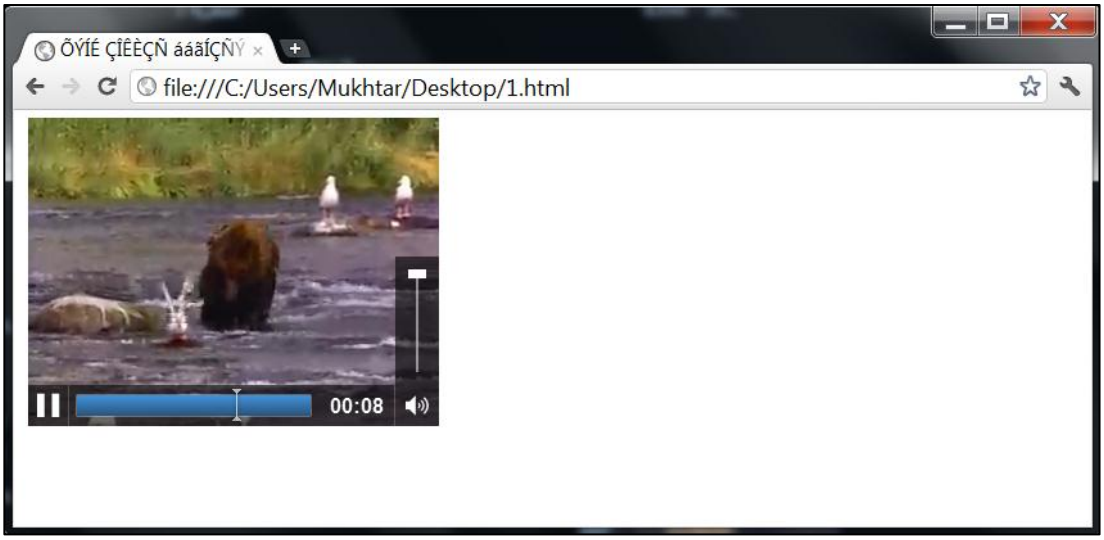
[٦] لا تعتقد أن دعم هذه المستعرضات كافٍ لجعل HTML 5 اللغة الرسمية لتطوير الصفحات فهناك عشرات مستعرضات الويب الأخرى و التي تستخدمها نسبة مرتفعة من مستخدمي الانترنت الإجماليين.

عنصر الفيديو Video

تقدم HTML 5 آلية بسيطة لعرض ملفات الفيديو ضمن صفحة الويب باستخدام العنصر الجديد <video>، و يتم ذلك بإسناد القيم المناسبة إلى واصفتي الارتفاع height و العرض width الخاصة به و من ثمَّ تمرير مسار ملف الفيديو عبر العنصر الابن <source> كقيمة لواصفته src و تمرير نوع ملف الفيديو عبر الواصفة type، لنشاهد الشيفرة التالية على سبيل المثال:

```
<html>
  <head>
        <title>
             مثال على عنصر عرض الفيديو
        </title>
  </head>
  <body>

     <video width="320" height="240" controls="controls">
        <source src="movie.mp4" type="video/mp4" />
        للأسف، المستعرض الحالي لا يدعم عنصر عرض الفيديو!
     </video>

  </body>
</html>
```

يظهر المحتوى النصي (الموجود بين وسمي البداية و النهاية للعنصر <video>) في المستعرضات التي لا تدعم HTML 5، أمَّا في المستعرضات الداعمة لها فستبدو صفحتنا كمايلي:



الشكل ٣١: عنصر عرض الفيديو الجديد كما يبدو في مستعرض جوجل كروم

ملحوظة: إلى هذه اللحظة فإن العنصر <video> يدعم تشغيل ثلاثة أنواع من صيغ الفيديو و هي الصيغ MPEG4 و Ogg و WebM.

يعرض الجدول التالي واصفات العنصر <video>:

الواصفة	القيم الممكنة	الشرح
audio	muted	كتم الصوت
autoplay	autoplay	التشغيل التلقائي لملف الفيديو
controls	controls	عرض شريط الأدوات في مشغل الفيديو
height	قيمة بالبكسل	ارتفاع مشغل الفيديو
loop	loop	إعادة تشغيل الفيديو بشكل تلقائي عند انتهائه
poster	رابط تشعبي	رابط تشعبي لصورة تعبر عن مقطع الفيديو و تظهر قبل أن يقوم المستخدم بتشغيله
src	رابط تشعبي	رابط ملف الفيديو
width	قيمة بالبكسل	عرض مشغل الفيديو

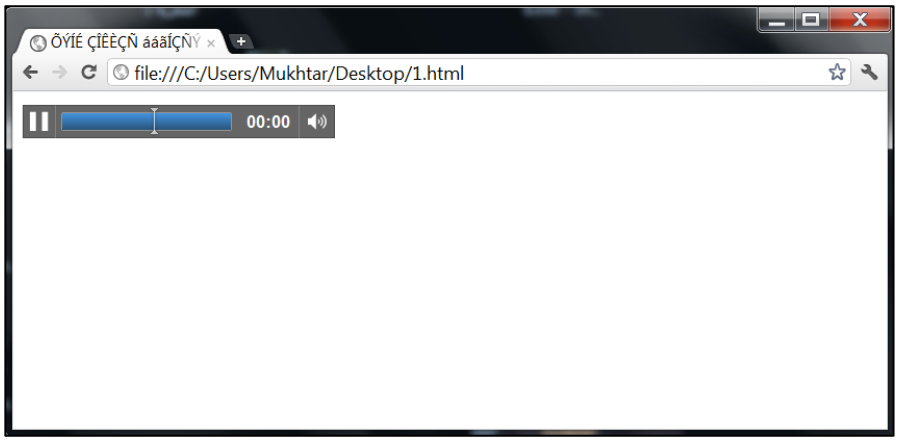
الجدول ٥ : واصفات العنصر <video>

عنصر الصوت Audio

تتيح HTML 5 عرض ملفات الصوت ضمن صفحة الويب ببساطة شديدة من خلال العنصر <audio> و الذي يستخدم بطريقة مشابهة لعنصر عرض الفيديو كمايلي:

```
<html>
  <head>
      <title>مثال على عنصر عرض الصوت</title>
  </head>
  <body>
     <audio controls="controls">
        <source src="song.mp3" type="audio/mp3" />
        للأسف، المستعرض الحالي لا يدعم عنصر عرض الصوت!
     </audio>
  </body>
</html>
```

يبدو هذا المثال في المستعرضات التي تدعم هذا العنصر كمايلي:



الشكل ٣٢ : عنصر عرض الصوت الجديد كما يبدو في مستعرض جوجل كروم

أمَّا في المستعرضات التي لا تدعم هذا العنصر فسيظهر النص المذكور بين وسمي البداية و النهاية للعنصر <audio> و الذي يحوي رسالة الاعتذار.

ملحوظة: إلى هذه اللحظة فإن العنصر <audio> يدعم تشغيل ثلاث صيغ لملفات الصوت هي MP3 و Ogg و Wav.

يعرض الجدول التالي واصفات العنصر <audio>:

الواصفة	القيم الممكنة	الشرح
autoplay	autoplay	التشغيل التلقائي لملف الصوت
controls	controls	عرض شريط الأدوات في مشغل الصوت
loop	loop	إعادة تشغيل ملف الصوت بشكل تلقائي عند انتهائه
preload	preload	تحميل الملف الصوت عند تحميل الصفحة بغض النظر عن خاصية التشغيل التقائي و عن نقر المستخدم لزر التشغيل
src	رابط تشعبي	رابط ملف الصوت

الجدول ٦ : واصفات العنصر <video>

عناصر النماذج Forms

تقدم HTML 5 مجموعة جديدة من عناصر الإدخال الخاصة بالنماذج Forms بالإضافة إلى مجموعة جديدة من أنواع عنصر الإدخال التقليدي <input>، و سنبدأ حديثنا بالحديث عن الأنواع الجديدة للعنصر <input> و التي يعرضها الجدول التالي:

قيمة الواصفة type	الشرح
email	عنصر خاص لإدخال البريد الإلكتروني
url	عنصر خاص لإدخال الروابط التشعبية
number	عنصر خاص لإدخال الأرقام

range	عنصر خاص لإدخال قيمة محصورة بمجال معيَّن سلفاً، و يظهر هذا العنصر على شكل شريط تمرير
date	عنصر لإدخال التاريخ (يوم/شهر/سنة)
month	عنصر لإدخال التاريخ (شهر/سنة)
week	عنصر لإدخال التاريخ و يستخدم لتحديد أسبوع من السنة
time	عنصر لإدخال الوقت (ساعة/دقيقة)
datetime	عنصر لإدخال التاريخ و الوقت (الساعة/اليوم/الشهر/السنة)، بالتوقيت العالمي
datetime-local	عنصر لإدخال التاريخ و الوقت، بالتوقيت المحلي
search	عنصر لعرض مربع البحث و هو شبيه بالنوع text
color	عنصر لإدخال القيم اللونيَّة

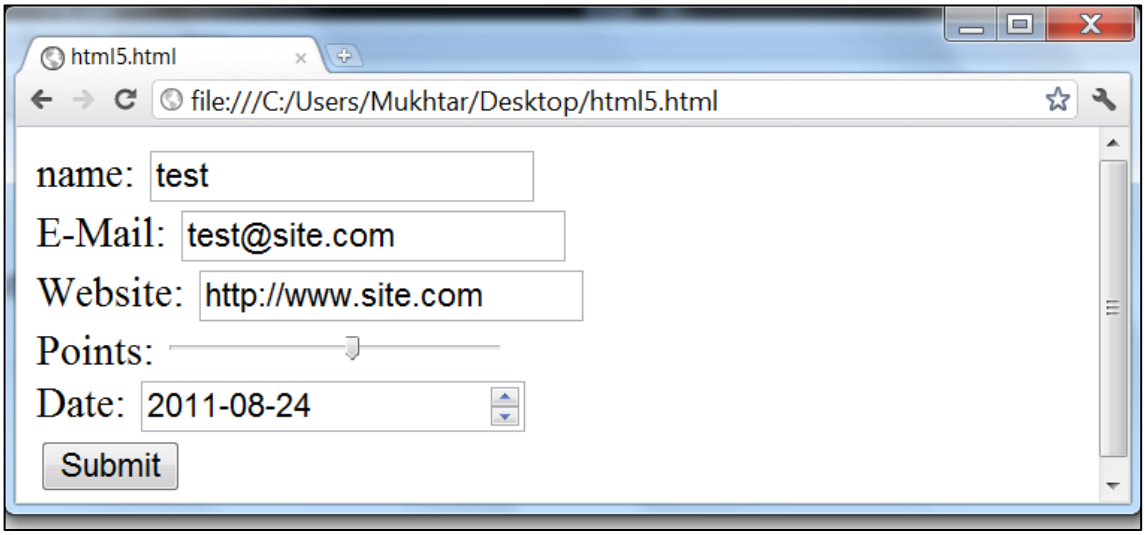
الجدول ٧ : أنواع عناصر الإدخال الجديدة في HTML 5

لنشاهد المثال التالي:

```
<!DOCTYPE HTML>
<html>
      <body>
            <form action="" method="get">
                  Name:
                  <input type="text" name="stdnt_name" /><br />
                  E-Mail:
                  <input type="email" name="stdnt_email" /><br />
                  Website:
                  <input type="url" name="stdnt_url" /><br />
                  Points:
```

```
                <input type="range" name="points" min="1"
max="10" /><br />

                Date:

                <input type="date" name="user_date" /><br />

                <input type="submit" />

            </form>

        </body>

</html>
```

يبدو المثال السابق كمايلي في مستعرض الويب:



الشكل ٣٣: عناصر الإدخال الجديدة كما تبدو في مستعرض كروم

بالإضافة لما سبق ذكره تقدم HTML5 مجموعة من وسوم النماذج الجديدة و منها الوسم <datalist> و الذي يستخدم لإنشاء القوائم المنسدلة كخيارات إدخال افتراضيَّة لأحد عناصر الإدخال الموجودة و يُنشأ محتوى القائمة بالتعاون مع وسم <option> و بطريقة مشابهة للعنصر القديم <select>، كما يتم تمرير اسم قائمة الخيارات الافتراضية (عنصر <datalist>) لعنصر الإدخال عبر الواصفة الجديدة list، لنشاهد المثال التالي:

```
<!DOCTYPE HTML>

<html>

      <body>

            <form action="" method="get">

                  name:

                  <input type="text" list="names_list"
name="stdnt_name" /><br />

                  <datalist id="names_list">

                        <option label="A" value="nameA" />

                        <option label="B" value="nameB" />

                        <option label="C" value="nameC" />

                  </datalist>

            </form>

      </body>

</html>
```

من العناصر الجديدة التي قدمتها HTML5 أيضاً العنصران <keygen> و <output>، حيث يستخدم العنصر <keygen> في عمليات تسجيل الدخول و التحقق من المستخدمين Authentication و ذلك عبر توليد زوج مفاتيح (مفتاح عمومي/مفتاح خصوصي) يرسل مع الطلب دون تدخل المستخدم، و يستخدم بالشكل:

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

أما العنصر <output> فيستخدم لعرض مخرجات Outputs من أنواع مختلفة مثل نتائج الحسابات أو القيم المُعادَة من أحد توابع Java Script كمايلي:

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

عنصر المَرسَم Canvas

تقدم HTML 5 عنصر المرسم <Canvas> و الذي يتيح لك الرسم ضمن صفحة الويب باستخدام Java Script [٧] و بذلك تستغني عن أية إضافات خارجية Plugins كإضافات الـ Flash على سبيل المثال.

لاستخدام عنصر المرسم يتم إنشاؤه أولاً بارتفاع height و عرض width محددين كمايلي:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

ثم يتم الرسم عليه باستخدام Java Script بشكل مشابه لمايلي:

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

   var c=document.getElementById("myCanvas");
   var cxt=c.getContext("2d");
   cxt.fillStyle="#FF0000";
   cxt.fillRect(0,0,150,75);

</script>
```

حيث قمنا أولاً بتحديد عنصر المرسم عن طريق الـ id الخاص به و ذلك في السطر:

```
var c=document.getElementById("myCanvas");
```

ثمَّ قمنا بإنشاء كائن Object من المرسم و هيأناه للرسم ثنائي الأبعاد 2D في السطر التالي:

```
var cxt=c.getContext("2d");
```

الآن و للقيام بالرسم سيتم استدعاء طرق Methods الكائن cxt الذي تم إنشاؤه قبل قليل بالشكل:

```
cxt.fillStyle="#FF0000";
cxt.fillRect(0,0,150,75);
```

[٧] يمكنك قراءة المزيد عن Java Script في مرجع آخر فهي خارج نطاق هذا الكتاب و لكنني سأحاول شرح استخدامها في الأمثلة.

ففي السطر الأول تم تحديد اللون الأحمر لوناً للشكل الذي سيتم رسمه (هذا ما تعنيه القيمة #FF0000)[٨]، ثمَّ تم رسم المربع بتمرير إحداثيات ركنيه الأيسر العلوي و الأيمن السفلي بشكل مشابه لما فعلناه عند تحديد مناطق الخريطة الصورية (انظر فقرة الخرائط الصورية).

يبدو المثال السابق ضمن الشيفرة الكاملة لصفحة الويب كمايلي:

```
<!DOCTYPE HTML>
<html>
      <body>

         <canvas id="myCanvas" width="200" height="100">
                لا يدعم مستعرضك عنصر المرسم
         </canvas>

         <script type="text/javascript">
             var c=document.getElementById("myCanvas");
             var cxt=c.getContext("2d");
             cxt.fillStyle="#FF0000";
             cxt.fillRect(0,0,150,75);
         </script>
      </body>
</html>
```

يعرض الجدول التالي بعض طرق Methods كائن المرسم و وظيفة كل منها:

[٨] سيتم في الفصل الثاني عرض مفهوم الألوان في HTML.

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

الجدول ٨ : بعض طرق كائن العنصر <canvas>

تخزين البيانات من طرف العميل

تتيح HTML 5 كائنين مختلفتين لتخزين البيانات من طرف العميل Client و هما:

١. الكائن localStorage: يستخدم هذا الكائن لتخزين البيانات من طرف العميل لفترة غير محدودة و يستخدم الصيغة العامة التالية للتخزين:

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

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

```
localStorage.varName;
```

٢. الكائن sessionStorage: يستخدم هذا الكائن لتخزين البيانات من طرف العميل لفترة محددة بجلسة واحدة فقط، و يستخدم الصيغة العامة التالية للتخزين:

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

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

```
sessionStorage.varName;
```

ملحوظة: هذان الكائنان يستخدمان كمخازن خاصة للبيانات أثناء كتابة شيفرات Java Script.

دليل الوسوم الجديدة في HTML 5

ختاماً لهذا الفصل نعرض فيمايلي جدولاً بالوسوم الجديدة في HTML 5:

الوسم	الشرح
<article>	مقالة
<aside>	جانب من محتويات الموقع
<audio>	لتشغيل ملفات الصوت
<canvas>	للرسم برمجياً
<command>	زر أوامر
<datalist>	قائمة خيارات افتراضية لعنصر إدخال
<details>	تفاصيل عنصر ما
<embed>	تضمين عنصر تفاعلي أو plugin خارجي
<figcaption>	عنوان عنصر <figure>
<figure>	مجموعة من عناصر الوسائط
<footer>	ذيل منطقة من مناطق الصفحة
<header>	رأس منطقة من مناطق الصفحة
<hgroup>	معلومات عن قسم من المستند
<keygen>	توليد مفاتيح التحقق في النموذج
<mark>	نص مميز

<meter>	قياسات محصورة بمدى محدد مسبقاً
<nav>	روابط تشعبية
<output>	عرض خرج من نوع ما
<progress>	شريط تقدّم لمهمة من نوع ما
<rp>	يستخدم لغة Ruby للتصريح عن ما سيتم عرضه في حالة كون المستعرض لا يدعم عنصر Ruby
<rt>	شرح لما تم التصريح عنه باستخدام عنصر Ruby
<ruby>	عنصر Ruby
<section>	قسم من المستند
<source>	لتعريف مصادر الوسائط
<summary>	يعرف ترويسة العنصر detail
<time>	التاريخ/الوقت
<video>	لعرض عناصر الفيديو
<wbr>	سطر جديد محتمل

الجدول ٩ : دليل الوسوم الجديدة في HTML 5