

# CSS

## Cascading Style Sheets



برمجة مواقع الإنترنت باستخدام لغة CSS

مع تحيات

قطار التعليم والتدريب



<https://www.facebook.com/EandT.Train/?ref=hl>

معكم يحلو المسير

## المحتويات

- الدرس ١ : ما هي تقنية CSS ؟
- الدرس ٢ : كيف تعمل تقنية CSS ؟
- الدرس ٣ : الألوان والخلفيات
- الدرس ٤ : الخطوط
- الدرس ٥ : النصوص
- الدرس ٦ : الروابط
- الدرس ٧ : المطابقة والتجميع للعناصر class و id
- الدرس ٨ : تجميع العناصر باستخدام span و div
- الدرس ٩ : نموذج الصندوق
- الدرس ١٠ : الحاشية والحشو
- الدرس ١١ : نموذج الصندوق - الإطارات Borders
- الدرس ١٢ : الارتفاع والعرض
- الدرس ١٣ : تعويم العناصر (floats)
- الدرس ١٤ : وضعية العناصر
- الدرس ١٥ : طبقة فوق طبقة باستخدام z-index
- الدرس ١٦ : المعايير القياسية

## مقدمة

المصطلح CSS هو اختصار للعبارة CSS Cascading Style Sheets وهي أداة رائعة لإضافة تصميم لمواقعك، ويمكنها أن توفر عليك الكثير من الوقت باستخدامها في تصميم المواقع بطريقة جديدة تماماً، وهي لغة يجب أن تستخدم من قبل كل شخص يعمل في تصميم المواقع.

في هذا الدرس ستبدأ في تعلم هذه التقنية خلال ساعات قليلة، CSS يسهل فهمها ويمكنك أن تتعلم كل الطرق العملية للتحكم بتصميم موقعك.

تعلم CSS عملية ممتعة، وأنت تقرأ هذا الدرس تذكر أن تأخذ وقتاً لتجرب بعض التجارب وترى ماذا تعلمت من كل درس.

استخدام CSS يتطلب معرفة أساسيات HTML، إذا لم تكن تعرف HTML فيمكنك أن تبدأ [بدرس HTML](#) قبل أن تتعلم CSS.

ما هي البرامج التي أحتاجها؟

تجنب استخدام برامج مثل فرونت بيج أو دريمويفر أو مايكروسوفت وورد عند قراءة وتطبيق هذا الدرس، هذه البرامج لن تساعدك على تعلم CSS، بدلاً من ذلك ستعيقك وتؤخر تعلمك للتقنية.

كل ما تحتاجه هو محرر نصي.

مثلاً مايكروسوفت ويندوز يأتي مع برنامج يسمى (Notepad المفكرة)، وهو غالباً ما يوضع في قسم (Accessories أدوات) الذي تجده في قسم (Programs برامج)، وبإمكانك استخدام برامج مماثلة في أنظمة التشغيل الأخرى مثل Pico في لينكس و Simple text لنظام ماك.

المحرر النصي البسيط هو أداة مثالية لتعلم HTML و CSS لأنه لا يؤثر أو يغير ما كتبته من أوامر، بهذه الطريقة نجاحك وأخطائك تعتمد عليك وحدك وليس على البرنامج، وهذا يبسط عملية التعلم لأنك تستطيع كشف الأخطاء بسرعة وسهولة.

يمكنك أن تستخدم أي متصفح لهذا الدرس، ونحن نشجع على أن تستخدم أحدث المتصفحات وتقوم بتحديث متصفحك دائماً.

متصفح ومحرر نصي، هذا كل ما تحتاجه.

## درس ١ : ما هي تقنية CSS ؟

لعلك سمعت عن CSS من قبل لكن لا تعرف حقاً ما هي، في هذا الدرس سنتعلم المزيد عن CSS وماذا يمكن لهذه التقنية أن تقدم لك.

### Cascading Style Sheets. CSS هي اختصار لي.

ماذا يمكن أن أفعل بتقنية CSS ؟

CSS هي لغة تصميم تحدد شكل وثيقة HTML ، فهي تهتم بالخطوط، الألوان، الهامش، والعرض والارتفاع، صورة خلفية الموقع، وكيفية توزيع المساحات وأشياء أخرى كثيرة، انتظر فقط وسترى!

HTML يمكن أن تستخدم بشكل خاطئ لإضافة تصميم للمواقع، لكن CSS توفر المزيد من الخيارات وهي أكثر دقة وعملية، وهي مدعومة من قبل المتصفحات الرئيسية اليوم.

بعد أجزاء قليلة في هذا الدرس ستتمكن من إنشاء ملف CSS الأول لكي تعطي موقعك مظهراً رائعاً.

ما الفرق بين CSS وHTML؟

تستخدم لوضع هيكلية ونظام منطقي للمحتويات، أما CSS فهي تستخدم لإضافة تصميم لهذه المحتويات.

حسناً، قد يبدو هذا مربكاً لكن أكمل القراءة وستفهم ما قرأته قريباً.

في الماضي القريب اخترع رجل اسمه تيم بيرنرز لي شبكة الويب، في ذلك الوقت كانت HTML تستخدم فقط لإضافة هيكلية للنص، الكاتب يمكنه أن يقوم بإضافة معنى للنص بأن يقول مثلاً "هذا عنوان" أو "هذه فقرة" ويستخدم وسوم HTML مثل <h1> و.<p>

مع ازدياد شهرة شبكة الويب بدأ المصممون في البحث عن طرق لإضافة تصميم للوثائق، ولتلبية رغباتهم قام مبرمجوا المتصفحات - نيتكسب ومايكروسوفت - باختراع وسوم HTML جديدة مثل <font> الذي يختلف عن وسوم HTML الأصلية بأنه يهتم بالشكل لا بالهيكلية.

أدى ذلك إلى أن تستخدم وسوم هيكلية مثل <table> بشكل خاطئ، حيث استخدم هذا الوسم لتحديد تصميم الصفحة بدلاً من إضافة معنى للنص، والكثير من الوسوم التي اخترعت مثل <blink> كانت فقط مدعومة من قبل متصفح واحد، أصبحت جملة "ستحتاج إلى المتصفح الفلاني لرؤية هذه الصفحة" مألوفة في مواقع الشبكة.

ابتكرت CSS لعلاج هذه المشكلة لأنها توفر للمصممين طرقاً فعالة وعملية لإنشاء التصاميم وهذه التصاميم ستكون مدعومة من قبل جميع المتصفحات في نفس الوقت، والفصل بين شكل الصفحة ومحتوياتها يبسط عملية إدارة الموقع بشكل كبير.

كيف ستفيدني CSS ؟

تعتبر CSS ثورة في عالم تصميم المواقع، وفوائدها الأساسية هي:

- التحكم بالتصميم من خلال ملف واحد.
- إمكانية أكبر وأدق للتحكم بتفاصيل التصميم.
- إنشاء تصاميم خاصة لمختلف وسائل عرض الموقع مثل الشاشات والطابعات والهواتف النقالة .. إلخ
- العديد من التقنيات والأساليب المتقدمة في التحكم بالتصميم.

في [الدرس اللاحق](#) سننظر عن قرب إلى كيفية عمل CSS وكيف نبدأ باستخدامها.

## درس ٢ : كيف تعمل تقنية CSS ؟

في هذا الدرس سنتعلم كيف تقوم بإنشاء ملف التصميم الأول، سنتعلم أساسيات CSS وما هي الوسوم اللازمة لتستخدم CSS في وثيقة HTML.

الكثير من خصائص CSS تشبه تلك المستخدمة في HTML ، لذلك إذا تعلمت HTML واستخداماتها لإنشاء التصاميم فأنت في الغالب ستتمكن من تعلم CSS بسهولة، لنلقي نظرة على هذا المثال الأساسي.

القواعد الأساسية لكتابة CSS

لنقل أننا نريد اللون الأحمر ليكون خلفية للصفحة:

باستخدام HTML يمكننا أن ننجز ذلك بهذه الطريقة:

```
<body bgcolor="#FF0000">
```

مع CSS يمكن تحقيق نفس النتيجة بكتابة هذه الأوامر:

```
body {background-color: #FF0000;}
```

كما تلاحظ، أوامر CSS تتشابه كثيراً مع HTML ، والمثال أعلاه يوضح لك الأسلوب الأساسي لكتابة CSS:

```
selector {property: value;}
```

↑ المنتقي: هو وسم أو وسوم HTML التي ستطبق عليها الخصائص "مثال: body"

↑ الخاصية: يمكن أن تكون مثلاً خاصية لون الخلفية "background-color"

↖ القيمة: قيمة خاصة لون الخلفية يمكن أن تكون اسماً "red" أو رقماً مثل #FF0000

لكن أين نضع أوامر CSS ؟ هذا هو ما سنتعلمه الآن.

تفعيل CSS في صفحة HTML

هناك ثلاث طرق يمكن أن تستخدمها لتفعيل CSS في صفحة HTML ، هذه الطرق مشروحة أدناه، ونحن ننصح بأن تركز وتستخدم الطريقة الثالثة، وهي أن تضع CSS في ملف منفصل.

### الطريقة ١: ضمن وسوم HTML باستخدام خاصية style

إحدى الطرق لتفعيل CSS في HTML هي باستخدام خاصية style ، لنأخذ مثلاً على أساس المثال أعلاه الذي أردنا فيه استخدام اللون الأحمر كخلفية للصفحة، يمكن تطبيق هذا الأمر بهذا الشكل

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body style="background-color: #FF0000;">
    <p>This is a red page</p>
  </body>
</html>
```

### الطريقة ٢: ضمت ملف HTML باستخدام وسم style

هذه طريقة مختلفة بأنها تستخدم وسم <style> ، وهذا مثال لكيفية تطبيق هذه الطريقة:

```
<html>
  <head>
    <title>Example</title>
    <style type="text/css">
      body {background-color: #FF0000;}
    </style>
  </head>
  <body>
    <p>This is a red page</p>
  </body>
</html>
```

### الطريقة ٣: ملف خارجي

هذه هي الطريقة الأفضل، وهي أن تقوم بوضع رابط لملف خارجي يحوي أوامر CSS ، خلال هذا الدرس سنقوم باستخدام هذه الطريقة لجميع الأمثلة.

الملف الخارجي هو ببساطة ملف نصي يستخدم اللاحقة **css** ، ومثل الملفات الأخرى يمكنك أن تضعه في مزود موقعك أو على القرص الصلب.

مثلاً، لنقل أن ملف التصميم لديك اسمه **style.css** وهو موجود في مجلد اسمه **style** ، هذه الحالة يمكن توضيحها أكثر من خلال هذا الرسم



المهم هنا هو إنشاء رابط بين ملف HTML و ملف التصميم (style.css) ، مثل هذا الرابط يمكن إنشائه من خلال سطر واحد في HTML:

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

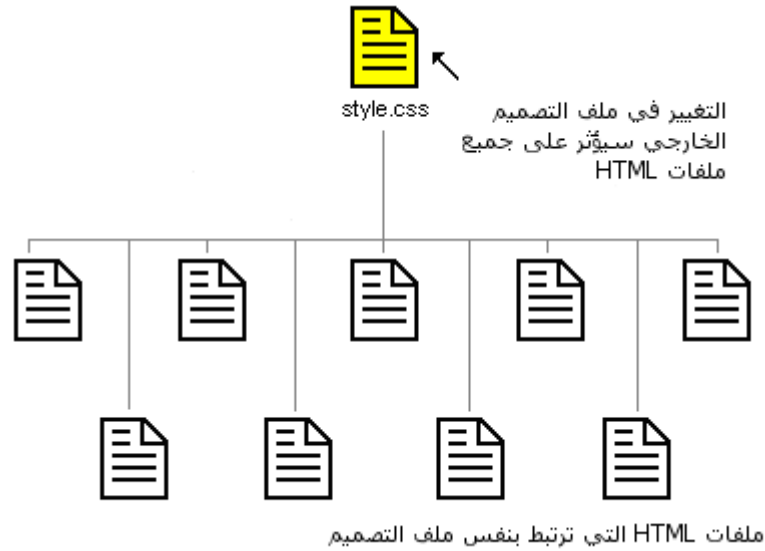
لاحظ كيف أن مسار الملف حددناه باستخدام خاصية href.

هذا الأمر يجب أن يوضع في قسم رأس الصفحة، أي بين وسمي <head> و </head> كما في المثال الآتي:

```
<html>
  <head>
    <title>My document</title>
    <link rel="stylesheet" type="text/css" href="style/style.css" />
  </head>
  <body>
    ...
```

هذا الرابط يخبر المتصفح بأن عليه استخدام التصميم من ملف CSS عندما يقوم بعرض ملف HTML. الجميل هنا أنك تستطيع ربط العديد من ملفات HTML بملف تصميم واحد، بمعنى آخر يمكن لملف تصميم واحد أن يستخدم للتحكم بتصميم العديد من ملفات HTML.





هذه الفكرة يمكنها أن توفر عليك الكثير من الوقت والجهد، إذا أردت مثلاً أن تغير لون خلفية موقع يحوي ١٠٠ صفحة فملف التصميم يمكنه أن يوفر عليك الوقت فلا تحتاج إلى تعديل ١٠٠ ملف بنفسك، باستخدام CSS يمكن تغيير ما تريد خلال ثواني بتغيير سطر واحد في ملف التصميم.

لنتدرب على ما تعلمناه حتى الآن.

جرب بنفسك

قم بتشغيل برنامج المفكرة (Notepad) أو أي محرر نصي، وقم بإنشاء ملفين، أحدهما HTML والآخر CSS وضع فيهما هذه المحتويات:

- **default.htm**

```
<html>
  <head>
    <title>My document</title>
    <link rel="stylesheet" type="text/css"
href="style.css" />
  </head>
  <body>
    <h1>My first stylesheet</h1>
  </body>
</html>
```

- **style.css**

```
body {
  background-color: #FF0000;
}
```

الآن قم بوضع الملفين في نفس المجلد، تذكر أن تحفظ الملفين باستخدام اللاحقة الصحيحة لكل ملف.  
قم بفتح ملف **default.htm** في متصفحك وانظر إلى الصفحة وهي تحوي اللون الأحمر كخلفية،  
تهانينا! لقد قمت بإنشاء ملف التصميم الأول!

أسرع واقرأ الدرس اللاحق حيث سنلقي نظرة على بعض خصائص CSS.

## الدرس ٣: الألوان والخلفيات

في هذا الدرس سنتعلم كيفية تفعيل الألوان والخلفيات لموقعك، سنقوم أيضاً بتعلم طرق متقدمة لتحديد موقع صورة الخلفية، نشرح هذه الخصائص في: CSS

- [color](#)
- [background-color](#)
- [background-image](#)
- [background-repeat](#)
- [background-attachment](#)
- [background-position](#)
- [background](#)

لون المقدمة: خاصية 'color'

خاصية color تصف لون عنصر HTML.

فمثلاً تصور أنك تريد أن تكون كل العناوين في الصفحة ملونة بلون أحمر داكن، كل العناوين رمزت باستخدام وسم <h1> ، المثال أدناه سيقوم بتوضيح كيفية تحويل كل <h1> إلى اللون الأحمر:

```
h1 {  
    color: #ff0000;  
}
```

### • [شاهد المثال](#)

الألوان يمكن أن تحدد باستخدام نظام الأرقام الست عشري كما في المثال أعلاه، أو بأن تختار اسم اللون "red" ، أو من خلال قيمة RGB والتي تعني Red و Green و Blue، مثال: (rgb(255,0,0)).

خاصية 'background-color'

خاصية background-color تحدد لون خلفية أي عنصر.

العنصر <body> يضم كل محتويات وثيقة HTML ، لذلك لتغيير خلفية الصفحة بأكملها يجب أن نفعل خاصية background-color على العنصر <body>.

يمكنك أيضاً تفعيل خاصية لون الخلفية على عناصر أخرى مثل العناوين والنصوص، في المثال أدناه قمنا باختيار ألوان خلفية لعنصري <body> و <h1>.

```
body {  
    background-color: #FFCC66;  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

### • [شاهد المثال](#)

لاحظ أننا قمنا بتفعيل خاصيتين للعنصر <h1> وقمنا بالفصل بين الخاصيتين باستخدام فاصلة منقوطة.

الصورة كخلفية "background-image"

خاصية background-image تستخدم لوضع صورة كخلفية لأي عنصر.

فمثلاً قمنا باستخدام صورة فراشة في المثال أدناه، يمكنك إنزال الصورة لتجرب بنفسك على حاسوبك، قم بالضغط على الصورة بالزر الأيمن واحفظها في جهازك، أو يمكنك استخدام أي صورة تناسبك.



لإدخال صورة الفراشة كخلفية للصفحة قم بتفعيل خاصية background-image للعنصر <body> وحدد مسار الصورة:

```

body {
  background-color: #FFCC66;
  background-image: url("butterfly.gif");
}

h1 {
  color: #990000;
  background-color: #FC9804;
}

```

### [شاهد المثال](#)

انتبه: لاحظ كيف حددنا مسار الصورة بهذا الشكل `url("butterfly.gif")` ، هذا يعني أن الصورة وضعت في نفس المجلد مع ملف التصميم، يمكنك أن تحدد مسار الصور في مجلدات أخرى باستخدام `url("../images/butterfly.gif")` أو حتى العنوان الكامل للملف `url("http://www.html.net/butterfly.gif")`.

تكرار صورة الخلفية "background-repeat"

هل لاحظت في المثال أعلاه أن صورة الفراشة تتكرر رأسياً وأفقياً لتغطي كامل الصفحة؟ الخاصية `background-repeat` تتحكم بتكرار الصورة.

في الجدول أدناه ملخص لأربع قيم يمكن أن تضعها للخاصية `background-repeat`.

مثال	الوصف	القيمة
<a href="#">شاهد المثال</a>	الصورة ستكرر أفقياً	<code>Background-repeat: repeat-x</code>
<a href="#">شاهد المثال</a>	الصورة ستكرر عمودياً	<code>background-repeat: repeat-y</code>
<a href="#">شاهد المثال</a>	الصورة ستكرر أفقياً وعمودياً	<code>background-repeat: repeat</code>
<a href="#">شاهد المثال</a>	لن تتكرر بأي شكل	<code>background-repeat: no-repeat</code>

مثلاً لتجنب تكرار صورة الخلفية يجب أن تكتب الأوامر بهذا الشكل::

```

body {
  background-color: #FFCC66;
  background-image: url("butterfly.gif");
  background-repeat: no-repeat;
}

h1 {
  color: #990000;
  background-color: #FC9804;
}

```

### • [شاهد المثال](#)

#### تثبيت صورة الخلفية "background-attachment"

الخاصية background-attachment تحدد ما إذا كانت صورة الخلفية ثابتة أو متحركة مع محتويات العنصر.

الصورة الثابتة لن تتحرك مع النص عندما يقوم القارئ بتحريك الصفحة، بينما الصورة المتحركة ستتحرك مع الصفحة بمحتوياتها.

في الجدول أدناه ملخص للقيم التي يمكنك وضعها لخاصية background-attachment ، شاهد الأمثلة ولاحظ الاختلاف بين الصورة الثابتة والمتحركة.

مثال	الوصف	القيمة
<a href="#">شاهد المثال</a>	الصورة ستتحرك مع الصفحة	Background-attachment: scroll
<a href="#">شاهد المثال</a>	الصورة ستبقى ثابتة	Background-attachment: fixed

المثال ادناه يبين كيفية اختيار القيمة المناسبة لتثبيت صورة الخلفية:

```

body {
  background-color: #FFCC66;
}

```

```

background-image: url("butterfly.gif");
background-repeat: no-repeat;
background-attachment: fixed;
}

h1 {
color: #990000;
background-color: #FC9804;
}

```

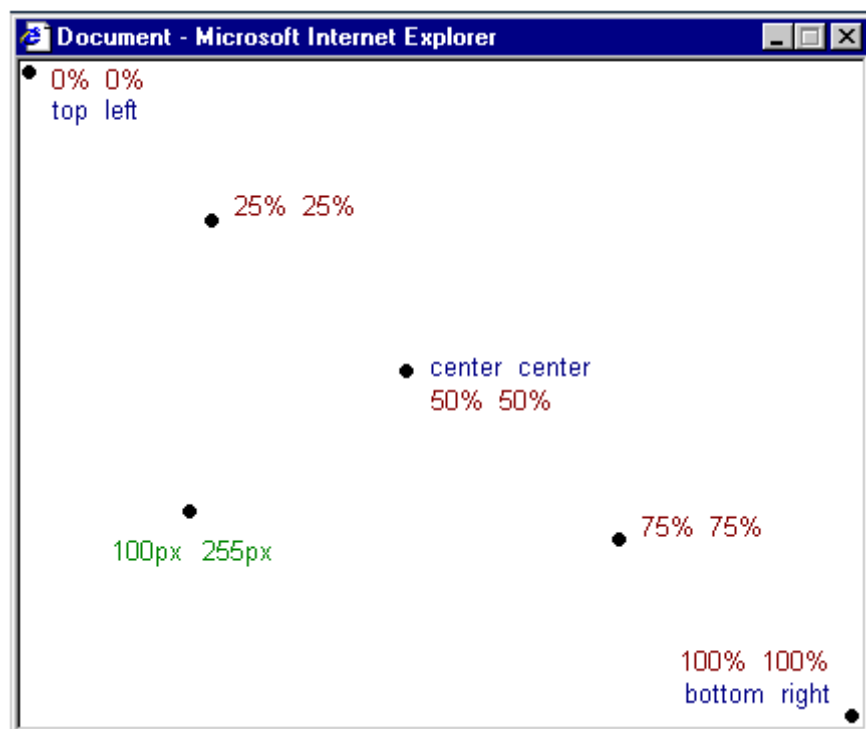
• [شاهد المثال](#)

### مكان صورة الخلفية "background-position"

تلقائياً توضع صورة الخلفية في أعلى يسار الصفحة، الخاصية background-position تسمح لك بتغيير هذه القيمة التلقائية ووضع الصورة في أي مكان تريده من الشاشة.

هناك طرق مختلفة لتحديد قيمة background-position، لكن كلها تنظم على نسق واحد، فمثلاً القيمة '100px 200px' تضع الصورة الخلفية على بعد 100 بكسل من يسار نافذة المتصفح و 200 بكسل من أعلى نافذة المتصفح.

هذا النسق يمكن تحديده أيضاً بالنسبة المئوية من عرض الشاشة وكذلك مقاييس محددة مثل البكسل والسنتيمتر، أو من خلال استخدام كلمات مثل top و bottom و center و left و right.



### الجدول أدناه يوضح بالمزيد من الأمثلة

القيمة	الوصف	مثال
background-position: 2cm 2cm	هذه الصورة وضعت على بعد ٢ سنتم من يسار الشاشة و ٢ سنتم من أعلى الشاشة	<a href="#">شاهد المثال</a>
background-position: 50% 25%	هذه الصورة وضعت في منتصف المسافة من يسار الشاشة و ربع المسافة من أعلى الشاشة	<a href="#">شاهد المثال</a>
background-position: top right	هذه الصورة وضعت في أعلى يمين الصفحة	<a href="#">شاهد المثال</a>

المثال أدناه يوضح كيفية وضع صورة الخلفية في أعلى يمين الشاشة:

```
body {
  background-color: #FFCC66;
  background-image: url("butterfly.gif");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: right bottom;
}

h1 {
  color: #990000;
  background-color: #FC9804;
}
```

• [شاهد المثال](#)

جمع كل الخصائص "background"



الخاصية background هي اختصار لكل خصائص خلفية العناصر التي قرأتها في هذا الدرس. باستخدام background يمكنك جمع عدة خصائص وبالتالي تقليل عدد الأسطر التي تكتبها في ملف التصميم وهذا يجعل الملف أسهل للقراءة. فمثلاً يمكن اختصار هذه الأسطر:

```
background-color: #FFCC66;
background-image: url("butterfly.gif");
background-repeat: no-repeat;
background-attachment: fixed;
background-position: right bottom;
```

باستخدام background يمكن تحقيق نفس النتيجة باستخدام سطر واحد فقط:

```
background: #FFCC66 url("butterfly.gif") no-repeat fixed right bottom;
```

القائمة ترتب بهذا الشكل - من اليسار إلى اليمين:

```
background-color | background-image | background-repeat | background-attachment | background-position
```

إذا لم تكتب خاصية ما سيقوم المتصفح بوضع القيمة التلقائية لهذه الخاصية، فمثلاً لم نضع الخاصية background-attachment و background-position في المثال:

```
background: #FFCC66 url("butterfly.gif") no-repeat;
```

الخاصيتين لم تحددنا وسيقوم المتصفح بوضع القيمة التلقائية لهما والقيم هي كما تعرف scroll و top left.

## ملخص

في هذا الدرس تعلمت طرقاً جديدة لا يمكنك تطبيقها باستخدام HTML فقط، المتعة تستمر في [الدرس القادم](#) والذي سنختبر فيه طيفاً واسعاً من الإمكانيات عندما نستخدم CSS لتحديد الخطوك.

## الدرس ٤ : الخطوط

في هذا الدرس سنتعلم الخطوط وكيف تفعلها باستخدام CSS ، وسنقوم أيضاً بمعالجة مشكلة كيفية عرض بعض الخطوط التي لن تظهر بشكل صحيح ما لم تكن مثبتة على الحاسوب، سنتعلم هذه الخصائص في هذا الدرس:

- [font-family](#)
- [font-style](#)
- [font-variant](#)
- [font-weight](#)
- [font-size](#)
- [font](#)

نوع الخطوط "font-family"

الخاصية font-family تستخدم لوضع قائمة خطوط تطبق على حسب الأولوية على عنصر أو صفحة ما، إذا لم يجد المتصفح الخط الأول ضمن قائمة الخطوط سيقوم باستخدام الخط الثاني في القائمة وإذا لم يجده سيحاول عرض الخط التالي، وهكذا حتى يجد الخط المناسب.

هناك نوعان من أسماء الخطوط تستخدم لتصنيفها، خطوط بأسماء محددة أو أسماء عامة، المصطلحين سنشرحهما في الفقرات اللاحقة.

### أسماء الخطوط

أمثلة لخطوط بأسماء محددة هي "Arial" و "Times New Roman" أو "Tahoma"

### أسماء عامة

الأسماء العامة للخطوط تحدد المجموعة من الخطوط التي لها شكل متماث، فمثلاً sans-serif هي مجموعة من الخطوط لا تحوي زوائد على الأحرف وهي خطوط مناسبة لعرض النص على الشاشة.

الاختلاف بين أنواع الخطوط يوضح في المثال التالي:

Times New Roman  
Garamond  
Georgia

هذه ثلاثة أمثلة لخطوط تنتمي لعائلة واحدة وهي serif، وهي تتميز بأنها تحوي زوائد على أطراف حروفها

Trebuchet  
Arial  
Verdana

هذه الخطوط الثلاثة تنتمي لعائلة واحدة وهي sans-serif، وهي لا تحوي زوائد على حروفها

Courier  
Courier New  
Andale Mono

هذه ثلاثة خطوط تنتمي لعائلة واحدة وهي monospace، وهي خطوط تتميز بأن عرض حروفها موحد

عندما تضع قائمة بالخطوط في موقعك فأنت تبدأ مع الخط المفضل لديك ثم الذي يليه، ومن الأفضل أن تنتهي القائمة باسم مجموعة عامة من الخطوط، بهذه الطريقة تضمن أن الصفحة ستعرض بنفس النوع من الخط المفضل لديك إذا لم يجد المتصفح الخط الذي قمت بتحديدته.

فمثلاً قائمة مرتبة من الخطوط مرتبة ستكون بهذا الشكل:

```
h1 {font-family: arial, verdana, sans-serif;}  
h2 {font-family: "Times New Roman", serif;}
```

### [شاهد المثال](#)

العناوين التي حددت باستخدام العنصر <h1> ستعرض باستخدام خط "Arial" ، إذا لم يكن هذا الخط مثبتاً في حاسوب المستخدم سيعرض بدلاً منه خط "Verdana" وإذا لم يكن الإثنان موجودان على حاسوب المستخدم سيتم اختيار خط من عائلة الخطوط **sans-serif** لعرض العناوين.

لاحظ كيف أننا وضعنا اسم خط "Times New Roman" بين علامتي تنصيص لأن اسمه يحوي مسافات ولذلك يجب أن يوضع بين علامتي تنصيص.

طراز الخط "font-style"

الخاصية font-style تحدد ما إذا كان الخط سيحمل القيمة **normal** أو **italic** أو **oblique** ، في المثال أدناه كل عناوين <h2> ستظهر بشكل مائل.

```
h1 {font-family: arial, verdana, sans-serif;}
h2 {font-family: "Times New Roman", serif;
font-style: italic;}
```

### • [شاهد المثال](#)

#### تباين الخط "font-variant"

الخاصية font-variant تستخدم للاختيار بين القيمتين **normal** أو **small-caps** للخط وهي متعلقة فقط باللغات الأوروبية، القيمة **small-caps** تعني أن النص سيكتب بحروف كبيرة لكنها من ناحية الحجم صغيرة، يبدو الأمر مربكاً، عليك أن تشاهد المثال الآتي لكي تفهم أكثر:

Sans Book SC	Sans Bold SC	Serif Book SC	Serif Bold SC
ABCABC	<b>ABCABC</b>	ABCABC	<b>ABCABC</b>

إذا اختيرت القيمة small-caps للخاصية **font-variant** ولم يكن هناك خط يدعم هذه الخاصية سيقوم المتصفح بعرض النص بحروف كبيرة فقط.

```
h1 {font-variant: small-caps;}
h2 {font-variant: normal;}
```

### • [شاهد المثال](#)

#### وزن الخط "font-weight"

الخاصية font-weight تصف كم ستكون سماكة أو "ثقل" الخط، يمكن للخط أن يحمل القيمة **normal** أو **bold**، وهناك متصفحات تدعم استخدام الأرقام من 100 إلى 900 لوصف ثقل الخط.

```
p {font-family: arial, verdana, sans-serif;}
td {font-family: arial, verdana, sans-serif;
font-weight: bold;}
```

## شاهد المثال

### حجم الخط "font-size"

حجم الخط يمكن تحديده بالخاصية font-size.

هناك العديد من الوحدات (مثال: بكسل، النسبة المئوية) التي يمكن استخدامها لوصف حجم الخط، في هذا المثال سنركز على الوحدات الأكثر استخداماً، والمثال يتضمن:

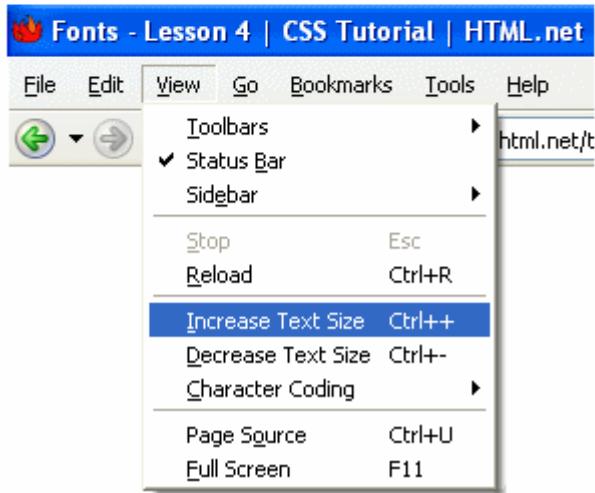
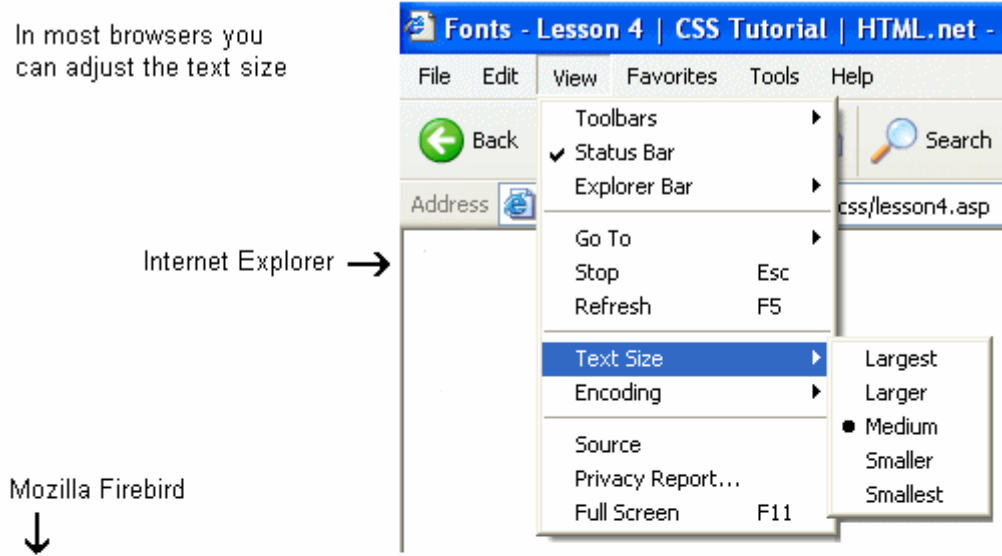
```
h1 {font-size: 30px;}
h2 {font-size: 12pt;}
h3 {font-size: 120%;}
p {font-size: 1em;}
```

## شاهد المثال

هناك فرق واحد بين وحدات القياس الأربع، وهي أن كل من 'px' و 'pt' تجعل حجم الخط محدداً بدقة وثابت بينما '%' و 'em' تسمح للمستخدم بتغيير حجم الخط إلى المستوى المناسب له، هناك العديد من المستخدمين من ذوي الاحتياجات الخاصة أو كبار السن أو من يعاني من ضعف في البصر أو يملك شاشة ذات جودة رديئة، ولكي نجعل الموقع قابلاً للوصول لهذه الفئات وللجميع لا بد من استخدام وحدات قياس يمكن تعديلها مثل '%' أو 'em'.

في المثال أدناه توضيح لكيفية تعديل حجم الخط في موزيلا فايرفوكس وإكسلورر، خاصية رائعة، ألا تظن ذلك؟

In most browsers you can adjust the text size



جمع كل خصائص الخط "font"

استخدام font الخاصية التي يمكنها أن تختصر كل خصائص الخطوط في سطر واحد.

فمثلاً تصور هذه السطور الأربعة تصف خصائص خطوط للعنصر: <p>

```
p {  
    font-style: italic;  
    font-weight: bold;  
    font-size: 30px;  
    font-family: arial, sans-serif;  
}
```

باستخدام الخاصية التي تختصر الخصائص الأخرى يمكن للسطور الأربعة أن تبسط بهذا الشكل:

```
p {  
    font: italic bold 30px arial, sans-serif;  
}
```

قائمة قيم الخاصية font ترتب بهذا الشكل:

```
font-style | font-variant | font-weight | font-size | font-family
```

## ملخص

تعلمت في هذا الدرس بعض الإمكانيات المتعلقة بالخطوط، تذكر أن أحد أهم مميزات استخدام CSS لتحديد الخطوط هو إمكانية تغيير الخط لكل صفحات الموقع خلال دقائق قليلة، CSS يمكنها أن توفر الوقت وتجعل حياتك أكثر سهولة، في [الدرس اللاحق](#) سنلقي نظرة على خصائص التحكم بالنص.

## الدرس ٥: النصوص

تنسيق وإضافة طراز إلى النصوص هي مسألة أساسية لمصممي المواقع، في هذا الدرس ستأخذ مقدمة حول الأساليب العجيبة التي تقدمها CSS لتنسيق النص، سنتحدث عن هذه الخصائص في هذا الدرس:

- [text-indent](#)
- [text-align](#)
- [text-decoration](#)
- [letter-spacing](#)
- [text-transform](#)

وضع فارغ قبل أول سطر "text-indent"

الخاصية text-indent تسمح لك بإضافة لمسة أنيقة إلى الفقرات بوضع مسافة فارغة قبل أول سطر من الفقرة، في المثال أدناه وضعنا القيمة **30px** لكل الفقرات التي تستخدم العنصر: <p>

```
p {
    text-indent: 30px;
}
```

• [شاهد المثال](#)

محاذاة النص "text-align"

خاصية text-align تشبه في HTML خاصية "align" التي كانت تستخدم في الماضي، النص يمكن محاذاته نحو اليسار "left" أو اليمين "right" أو في المنتصف "centred" وبالإضافة إلى ذلك القيمة justify ستقوم بمحاذاة النص من الجانبين كما تفعل بعض الصحف والمجلات.

في المثال أدناه النص في رأس الجدول <th> قمنا بمحاذاته نحو اليمين، بينما البيانات في الجدول <td> حاذيناها نحو المنتصف أما النص في الفقرات فقمنا بمحاذاته من الجانبين:



```
th {
    text-align: right;
}

td {
    text-align: center;
}

p {
    text-align: justify;
}
```

• [شاهد المثال](#)

زخرفة النص "text-decoration"

الخاصية text-decoration تمكنك من إضافة زخارف أو تأثيرات على النص، فمثلاً يمكنك أن تضيف سطرًا أسفل النص، أو فوقه أو عليه، في المثال الآتي كل عناصر <h1> وضعنا أسفلها خطأً أما <h2> فهي العناوين التي فوقها خط و <h3> قمنا بوضع الخط عليها.

```
h1 {
    text-decoration: underline;
}

h2 {
    text-decoration: overline;
}

h3 {
    text-decoration: line-through;
}
```

• [شاهد المثال](#)

المسافة بين الحروف "letter-spacing"

المسافة بين حروف النص يمكن تحديدها من خلال خاصية letter-spacing ، القيمة المحددة لهذه الخاصية هي ببساطة عرض المسافة التي تريدها بين كل حرف وآخر، فمثلاً إذا أردت

مسافة **3px** بين الحروف في الفقرات <p> و **6px** بين الحروف في العناوين <h1> فعليك أن تكتب الخصائص بهذا الشكل:

```
h1 {
    letter-spacing: 6px;
}

p {
    letter-spacing: 3px;
}
```

### • [شاهد المثال](#)

#### تحويل النص "text-transform"

خاصية text-transform تتحكم بحجم الخط في اللغات الغربية، يمكنك أن تختار بين القيم **capitalize** أو **uppercase** أو **lowercase** ، وبغض النظر عن كيفية ظهور النص الأصلي.

فمثلاً كلمة "headline" يمكن أن تظهر للمستخدم بهذا الشكل "HEADLINE" أو بهذا الشكل "Headline"، هناك أربع قيم يمكنك استخدامها مع القيم: text-transform:

#### **capitalize**

تقوم بتكبير الحرف الأول من كل كلمة مثال "john doe": ستصبح "John Doe"

#### **uppercase**

تجعل كل الحروف كبيرة، مثال "john doe": ستصبح "JOHN DOE"

#### **lowercase**

ستجعل كل الحروف صغيرة، مثال "JOHN DOE": ستصبح "john doe"

#### **none**

لن تقوم بعمل أي تأثير، النص سيظهر كما كتب في ملف HTML.

كمثال، سنقوم باستخدام قائمة أسماء، الأسماء كلها تستخدم العنصر (list-item) <li> ، ولنقل أننا نريد الأسماء أن يظهر حرفها الأول كبيراً أما العناوين فنريد كل حروفها كبيرة.

جرب وألقي نظرة على HTML لهذا المثال وسترى أن النص الأصلي كتب بحروف صغيرة.

```
h1 {
  text-transform: uppercase;
}

li {
  text-transform: capitalize;
}
```

• [شاهد المثال](#)

### ملخص

في الدروس الثلاثة الأخيرة تعلمنا الكثير من خصائص CSS ، لكن هناك الكثير في هذه التقنية، في [الدرس اللاحق](#) سنلقي نظرة على الروابط.

## الدرس ٦: الروابط

يمكنك أن تطبق ما تعلمته فعلاً في الدروس السابقة على الروابط (مثال: تغيير الألوان، الخطوط، وضع خط أسفل الروابط، إلخ) الجديد هنا أن CSS تسمح لك بتحديد خصائص مختلفة بحسب حالة الرابط إن لم يزر المستخدم الرابط أو زاره أو كان نشطاً أو كان مؤشر الفأرة فوق الرابط، كل هذا يمكنك من إضافة مؤثرات جميلة ومفيدة على الروابط، للتحكم بهذه التأثيرات يجب أن تستخدم ما يسمى **pseudo-classes**.

ما هي الفئة المزيفة؟

الفئة المزيفة أو **pseudo-class** يسمح لك بأن تأخذ في عين الاعتبار مختلف الحالات والأحداث عندما تقوم بتحديد خصائص لعنصر في HTML.

لنلقي نظرة على المثال، كما تعرف الروابط تحدد في HTML من خلال الوسم `<a>` لذلك يمكننا أن نستخدم `a` كمنتقي في CSS:

```
a {
    color: blue;
}
```

الرابط له عدة حالات، فمثلاً يمكن أن يزوره المستخدم أو لا يفعل، لذلك يمكنك أن تستخدم فئة مزيفة لكي تحدد شكلاً مختلفاً للرابط الذي زاره المستخدم عن الرابط الذي لم يزره بعد.

```
a:link {
    color: blue;
}

a:visited {
    color: red;
}
```

استخدم `a:link` و `a:visited` للروابط التي زارها أو لم يزرها المستخدم، الروابط يمكن أن تكون نشطة لها فئة مزيفة خاصة وهي `a:active` أما `a:hover` فهي الحالة التي يكون فيها مؤشر الفأرة فوق الرابط.

سنقوم الآن بعرض الحالات الأربع للروابط مع المزيد من الشرح والأمثلة.

الفئة المزيفة link :

الفئة المزيفة link : تستخدم للروابط التي تقود المستخدم إلى صفحات لم يزرها.  
في المثال أدناه الروابط التي لم يزرها المستخدم ستظهر بلون أزرق فاتح.

```
a:link {
    color: #6699CC;
}
```

• [شاهد المثال](#)

الفئة المزيفة visited :

الفئة المزيفة visited : تستخدم للروابط التي زارها المستخدم، المثال أدناه سيجعل كل الروابط التي زارها المستخدم بلون بنفسجي غامق:

```
a:visited {
    color: #660099;
}
```

• [شاهد المثال](#)

الفئة المزيفة active :

الفئة المزيفة active : تستخدم للروابط النشطة.

في المثال أدناه كل الروابط النشطة ستظهر بخلفية صفراء:

```
a:active {
    background-color: #FFFF00;
}
```

• [شاهد المثال](#)

الفئة المزيفة `hover` :

الفئة المزيفة `hover` : تستخدم عندما يكون مؤشر الفأرة فوق الرابط.

يمكن استخدام هذه الفئة لإنشاء مؤثرات جميلة، فمثلاً إذا أردنا أن تكون الروابط ملونة بالبرتقالي ومائلة عندما نضع مؤشر الفأرة عليها فعلينا أن نكتب `CSS` بهذا الشكل:

```
a:hover {
  color: orange;
  font-style: italic;
}
```

• [شاهد المثال](#)

المثال ١: المؤثرات تظهر عندما يوضع مؤشر الفأرة على الرابط

هذه مشهورة لإنشاء مؤثرات مختلفة عندما يوضع مؤشر الفأرة على الرابط، لذلك سنلقي نظرة على المزيد من الأمثلة المتعلقة بالفئة الزائفة `hover`:

مثال ١: وضع مسافات بين الحروف

كما نتذكر في [الدرس ٥](#) تعلمنا أن المسافة بين الحروف يمكن أن تعديلها باستخدام الخاصية `letter-spacing` ، هذا يمكن تطبيقه على الروابط لإنشاء مؤثرات خاصة:

```
a:hover {
  letter-spacing: 10px;
  font-weight:bold;
  color:red;
}
```

• [شاهد المثال](#)

مثال ١ب: الحروف الكبيرة والصغيرة

في [الدرس ٥](#) ألقينا نظرة على الخاصية `text-transform` والتي يمكنها تغيير حالة الأحرف بين الصغيرة والكبيرة، هذا يمكن استخدامه أيضاً كمؤثر على الروابط:

```
a:hover {
    text-transform: uppercase;
    font-weight:bold;
    color:blue;
    background-color:yellow;
}
```

### • شاهد المثال

في المثالين يمكن أن تأخذ فكرة عن الإمكانيات اللامتناهية لدمج بين العديد من الخصائص، يمكنك أن تقوم بإنشاء مؤثرات من ابتكارك، جرب!

مثال ٢: إزالة السطر من أسفل الرابط

أحد أكثر الأسئلة تكراراً هو كيف أزيل الخط من أسفل الرابط؟

يجب أن تفكر جيداً قبل أن تزيل الخط من أسفل الرابط، لأنه قد يجعل استخدام موقعك أكثر صعوبة، الناس اعتادوا على أن يرو الخطوط بلون أزرق وبخط أسفلها يميزها عن بقية النصوص الأخرى، ويعلمون أنهم يستطيعون الضغط عليها حتى والدتي تعلم ذلك! إذا أزلت الخط من أسفل الروابط وغيرت ألوانها فأنت تزيد من فرصة عدم فهم الزائر لموقعك وتشتته وبهذا لن يستفيدوا من محتويات موقعك.

على أي حال، إزالة الخط من أسفل الروابط سهل جداً، كما تتذكر في [الدرس ٥](#) الخاصية `text-decoration` يمكنها أن تستخدم لتحديد ما إذا كان هناك خط سيظهر أسفل النص أم لا، لإزالة الخط السفلي قم بوضع القيمة `none` للخاصية `text-decoration`.

```
a {
    text-decoration:none;
}
```

يمكنك أيضاً أن تفعل ذلك مع الحالات الأربعة للرابط:

```
a:link {
    color: blue;
    text-decoration: none;
}

a:visited {
    color: purple;
    text-decoration: none;
}

a:active {
    background-color: yellow;
    text-decoration: none;
}

a:hover {
    color: red;
    text-decoration: none;
}
```

• [شاهد المثال](#)

## ملخص

في هذا الدرس تعلمنا الفئة الزائفة، واستخدمنا بعض الخصائص من الدروس السابقة، هذا يجب أن يعطيك فكرة عن الإمكانيات التي تستطيع CSS أن تقدمها.

في [الدرس اللاحق](#) سنعلمك كيف تحدد خصائص معينة لمجموعة من العناصر أو لعنصر معين.



## الدرس ٧: المطابقة والتجميع للعناصر class و id

في بعض الأحيان تود أن تفعل خصائص ما على مجموعة أو فريق من العناصر، في هذا الدرس سنلقي نظرة على كيفية استخدام الفئات class والمعرفات id لتحديد خصائص معينة لعناصر مختارة.

كيف تختار لوناً محدداً لعنوان يختلف عن بقية العناوين في موقعك؟ كيف يمكنك أن تجمع بين روابط مختلفة وكل مجموعة تضعها في قسم مختلف وكل قسم له تصميم خاص؟ هذه أمثلة سنحجب عليها في هذا الدرس.

تجميع العناصر باستخدام الفئات "class"

لنقل بأن لدينا قائمتين من الروابط لأصناف مختلفة من الثمار، الخضار والفواكه، القائمتين في ملف HTML ستكونان بهذا الشكل:

```
<p>الخضار:</p>
<ul>
<li><a href="ab.htm">يقدونس</a></li>
<li><a href="cd.htm">جزر</a></li>
<li><a href="ef.htm">خيار</a></li>
</ul>

<p>فواكه:</p>
<ul>
<li><a href="gh.htm">تفاح</a></li>
<li><a href="ij.htm">أناس</a></li>
<li><a href="kl.htm">بطيخ</a></li>
</ul>
```

### • شاهد المثال

نريد أن تكون قائمة روابط الخضروات باللون الأصفر، وقائمة الفواكه باللون الأحمر وبقية الروابط تبقى باللون الأزرق.

لإنجاز هذا نقسم الروابط إلى مجموعتين، يمكن فعل ذلك بتحديد فئة لكل رابط باستخدام الخاصية class.

لنقم بتحديد فئات للمثال أعلاه:

```

<p>خضار:</p>
<ul>
<li><a href="ab.htm"
class="veg">بقدونس</a></li>
<li><a href="cd.htm" class="veg">جزر</a></li>
<li><a href="ef.htm" class="veg">خيار</a></li>
</ul>

<p>فواكه:</p>
<ul>
<li><a href="gh.htm" class="fru">تفاح</a></li>
<li><a href="ij.htm"
class="fru">أناناس</a></li>
<li><a href="kl.htm" class="fru">بطيخ</a></li>
</ul>

```

يمكن الآن تحديد خصائص محددة للروابط التي تتبع فئة الخضروات أو الفواكه بهذا الشكل:

```

a {
    color: blue;
}

a.veg {
    color: #FFBB00;
}

a.fru {
    color: #800000;
}

```

• [شاهد المثال](#)

كما ترى في المثال، يمكنك أن تحدد خصائص لعناصر معينة تتبع مجموعة محددة باستخدام الفئات **classname**. في ملف التصميم.

تحديد معرف خاص لعنصر معين من خلال id

بالإضافة إلى تجميع العناصر من خلال الفئات، لعلك ترغب في تحديد معرف لخاصية محددة، هذا يمكن إنجازه من خلال الخاصية `id`.

المميز في الخاصية `id` هو عدم إمكانية وجود عنصرين يستخدمانه في نفس الملف، أي لا يمكن لعنصرين أن يستخدم معرفاً `id` واحداً في نفس الصفحة، كل معرف `id` يجب أن يكون مميزاً وفريداً، أما غير ذلك فعليك استخدام الفئات `class`، دعنا نلقي نظرة على مثال محتمل لكيفية استخدام المعرف `id`:

```
<h1>١ فصل</h1>
...
<h2>١.١ فصل</h2>
...
<h2>١.٢ فصل</h2>
...
<h1>٢ فصل</h1>
...
<h2>٢.١ فصل</h2>
...
<h3>٢.١.٢ فصل</h3>
...
```

المثال أعلاه قد يكون عناوين لوثيقة ما تقسم الصفحة إلى أبواب وفصول، من الطبيعي أن نحدد لكل فصل معرفاً خاصاً `id` كما في المثال الآتي:

```
<h1 id="c1">١ فصل</h1>
...
<h2 id="c1-1">١.١ فصل</h2>
...
<h2 id="c1-2">١.٢ فصل</h2>
...
<h1 id="c2">٢ فصل</h1>
...
<h2 id="c2-1">٢.١ فصل</h2>
...
<h3 id="c2-1-2">٢.١.٢ فصل</h3>
...
```

لنقل أن عنوان الفصل ١.٢ يجب أن يكون باللون الأحمر، يمكن إنجاز ذلك في CSS بهذا الشكل:

```
#c1-2 {
    color: red;
}
```

#### • شاهد المثال

كما ترى في المثال أعلاه يمكن تحديد خصائص معينة لعنصر معين باستخدام #id في وثيقة التصميم.

ملخص

في هذا الدرس تعلمنا كيف نستخدم المنتقي من نوع class و id، يمكنك الآن أن تحدد خصائص محددة لعناصر معينة.

في الدرس اللاحق، سنلقي نظرة أقرب على عنصرين في HTML لهما يستخدمان كثيراً مع <span> CSS: و <div>.

### الدرس ٨: تجميع العناصر باستخدام span و div

العنصران <span> و <div> يستخدمان لجمع العناصر وإنشاء هيكلية للوثائق وهما يستخدمان معاً مع الخاصيتين class و id.

في هذا الدرس، سنلقي نظرة عن قريب على استخدام <span> و <div> لأنهما العنصران الأكثر أهمية عندما تتعامل مع CSS.

- التجميع بالعنصر <span>
- التجميع بالعنصر <div>

التجميع بالعنصر <span>

العنصر <span> هو ما يمكن أن تسميه العنصر المحايد والذي لا يضيف شيئاً للوثيقة نفسها، لكن مع CSS يمكن استخدامه لإنشاء مؤثرات على أجزاء محددة من النص في الوثائق.

مثال هذا هو اقتباس من جملة قالها بنجامين فرانكلين:

```
<p>Early to bed and early to rise  
makes a man healthy, wealthy and wise.</p>
```

لنقل أننا نريد أن نؤكد على ما قاله السيد فرانكلين حول فوائد عدم النوم في النهار بأن نلونها بالأحمر، لهذا الغرض يمكن وضع الكلمات بين وسمي `<span>` ، كل `span` أضفنا لها فئة `class` والتي تمكنا بعد ذلك من إضافة خصائص لها من خلال `CSS`:

```
<p>Early to bed and early to rise  
makes a man <span class="benefit">healthy</span>,  
<span class="benefit">wealthy</span>  
and <span class="benefit">wise</span>.</p>
```

أوامر `CSS` المتعلقة بالمثال:

```
span.benefit {  
    color:red;  
}
```

#### • [شاهد المثال](#)

بالطبع يمكنك استخدام المعرف `id` لإضافة الألوان على `<span>` لكن تذكر أن استخدام المعرف يجب أن يكون لمرة واحدة في الصفحة الواحدة، لذلك في المثال أعلاه عليك أن تستخدم ثلاث معرفات متميزة لكل `<span>` كما تعلمت في الدرس الماضي.

التجميع بالعنصر `<div>`

العنصر `<span>` يستخدم في العناصر كما رأيت في المثال السابق، العنصر `<div>` يستخدم لتجميع العناصر.

هذا هو الاختلاف الوحيد، تجميع العناصر باستخدام `<div>` يعمل بنفس الطريقة، لنلقي نظرة على مثال لقائمتين تحويان أسماء الرؤساء الأمريكيين مقسمة حسب الأحزاب التي ينتمون لها:

```
<div id="democrats">
<ul>
<li>فرانكلين روزفيلت</li>
<li>هاري ترومن</li>
<li>جون كينيدي</li>
<li>ليندون جونسون</li>
<li>جيمي كارتر</li>
<li>بيل كلينون</li>
</ul>
</div>
```

```
<div id="republicans">
<ul>
<li>دوايت أيزنهاور</li>
<li>ريتشارد نيكسون</li>
<li>جيرالد فورد</li>
<li>رونالد ريغان</li>
<li>جورج بوش</li>
<li>جورج دبليو بوش</li>
</ul>
</div>
```

وفي ملف CSS سنستخدم التجميع كما في المثال أعلاه بنفس الطريقة:

```
#democrats {
    background:blue;
}

#republicans {
    background:red;
}
```

• [شاهد المثال](#)

في الأمثلة السابقة استخدمنا <div> و <span> بشكل بسيط جداً، بتعديل لون الخلفية والنص، يمكن للعنصرين أن يقدموا الكثير من الإمكانيات المتقدمة، على أي حال، لن نتحدث عن هذا حالياً، سنتناول هذا الموضوع لاحقاً في.

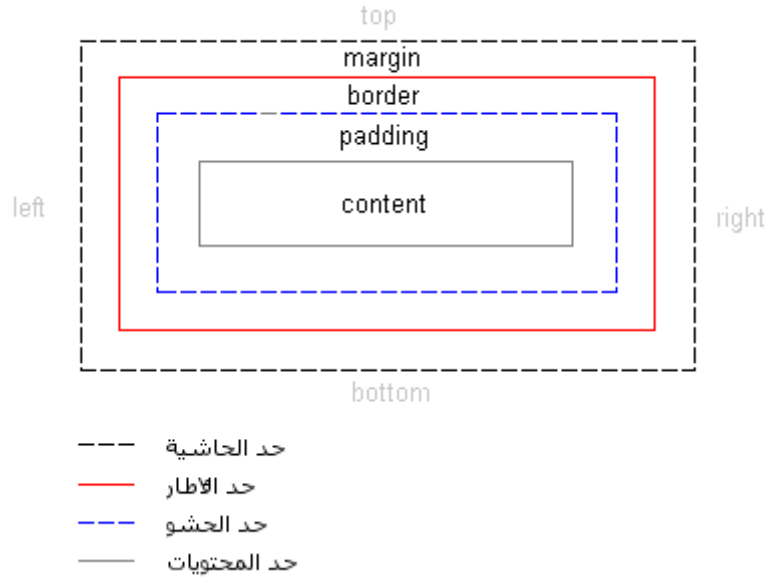
## ملخص

في الدرس ٧ و٨، تعلمت حول المنتقي id و class والعنصرين span و div. يجب أن تكون الآن قادراً على تجميع وتحديد كل عناصر ملفات HTML ، وهذه خطوة كبيرة في اتجاه إتقان CSS ، في الدرس ٩ سنتعرف على نموذج الصندوق.

## الدرس ٩: نموذج الصندوق

نموذج الصندوق في CSS يصف الصناديق التي تنشأ من خلال عناصر HTML ، نموذج الصندوق يحوي أيضاً خيارات مفصلة حول تعديل الحاشية، الإطار، الحشو والمحتويات لكل عنصر، في المثال التوضيحي أدناه سنعرض أجزاء نموذج الصندوق:

### نموذج الصندوق في CSS



الرسم أعلاه يبدو مربكاً بعض الشيء، لذلك لنقم باستخدام نموذج الصندوق في مثال فعلي باستخدام عنوان وبعض النصوص، في HTML وضعنا نصاً مقتبساً من الميثاق العالمي لحقوق الإنسان:

```
<h1>Article 1:</h1>
```

```
<p>All human beings are born free  
and equal in dignity and rights.  
They are endowed with reason and conscience  
and should act towards one another in a  
spirit of brotherhood</p>
```

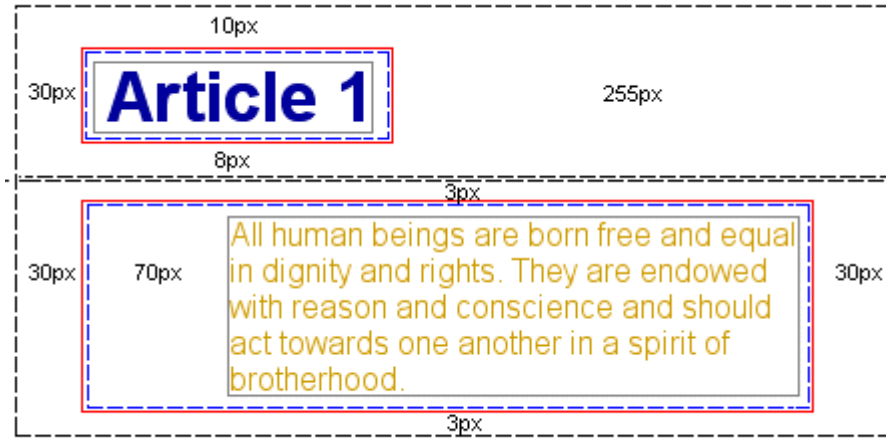
بإضافة بعض الألوان وتحديد بعض المعلومات حول الخطوط يمكن للمثال أن يصبح بهذا الشكل:



# Article 1

All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood.

المثال يحوي عنصرين `<h1>` و `<p>`، نموذج الصندوق للعنصرين سيظهر بهذا الشكل:



حتى لو بدى المثال معقداً، الرسم يوضح كل عنصر في HTML محاط بصناديق، الصناديق يمكن تعديلها من خلال CSS.

الخصائص التي تتحكم بالصناديق المختلفة هي `padding` و `margin` و `border`، في الدرسين اللاحقين سنتعامل مع هذه الخصائص الثلاثة.

- [الدرس ١٠: نموذج الصندوق `padding` و `margin`](#)
- [الدرس ١١: نموذج الصندوق `border`](#)

عندما تنتهي من الدرسين ستتمكن من التحكم بنموذج الصندوق وتقوم بتعديل ملفات HTML بشكل أفضل وأكثر دقة مقارنة مع استخدام الأسلوب القديم الذي يعتمد على الجداول في HTML.

ملخص

في هذا الدرس تعلمت نموذج الصندوق، في الدروس القادمة سنلقي نظرة أقرب على كيفية إنشاء والتحكم بنموذج الصندوق.

## الدرس ١٠ : الحاشية والحشو

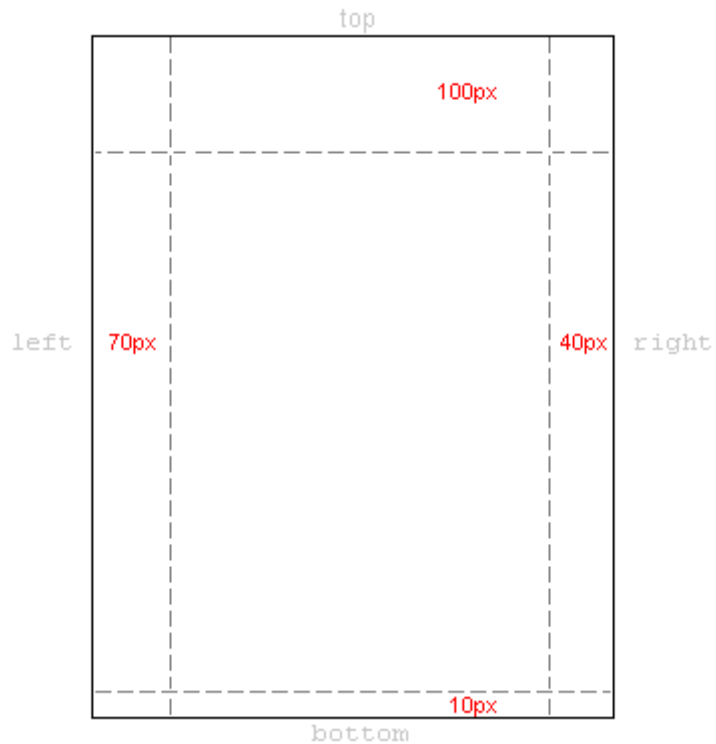
في الدرس الماضي تعرفنا على نموذج الصندوق، في هذا الدرس سنلقي نظرة على كيفية تغيير شكل العناصر بتغيير إعدادات خاصية `margin` و `padding`.

- تعديل خاصية `margin` لعنصر ما
- تعديل خاصية `padding` لعنصر ما

تعديل خاصية `margin` لعنصر ما

كل عنصر له أربع جوانب، اليمين `right` ، اليسار `left` ، الأعلى `top` ، الأسفل `bottom` ، الخاصية `margin` تتحكم ببعد كل جانب من جوانب العنصر عن العنصر المحاذي له، أو إطار الصفحة، أنظر إلى الرسم التوضيحي في [الدرس ٩](#) للمزيد من التوضيح.

في المثال الأول لنلقي نظرة على كيفية تحديد الحاشية "`margin`" للصفحة نفسها أي للعنصر `<body>` ، الرسم أدناه يوضح كيف نريد للحاشية أن تكون في الصفحة.



في ملف CSS سنكتب التالي:

```
body {
  margin-top: 100px;
  margin-right: 40px;
  margin-bottom: 10px;
  margin-left: 70px;
}
```

أو يمكننا تجميع كل هذه السطور في سطر واحد أنيق:

```
body {
  margin: 100px 40px 10px 70px;
}
```

#### • [شاهد المثال](#)

يمكنك تحديد الحاشية بنفس الطريقة لمعظم العناصر، فمثلاً يمكنك تحديد الحاشية لكل الفقرات النصية للعنصر: <p>

```
body {
  margin: 100px 40px 10px 70px;
}

p {
  margin: 5px 50px 5px 50px;
}
```

#### • [شاهد المثال](#)

تعديل خاصية padding لعنصر ما

الحشو يمكن التعبير عنه بأنه "ما في داخل العنصر نفسه" والحشو لا يؤثر في المسافة بين العنصر والآخر، بل يحدد المسافة في العنصر نفسه بين محتويات العنصر والإطار.

استخدام الحشو "padding" يمكن توضيحه بمثال بسيط حيث كل العناوين لها ألوان خلفية:

```
h1 {
    background: yellow;
}

h2 {
    background: orange;
}
```

#### • [شاهد المثال](#)

بتحديد الحشو لكل عنوان يمكن تغيير حجم العنصر الداخلي الذي يحيط بالعنوان من كل جهة:

```
h1 {
    background: yellow;
    padding: 20px 20px 20px 80px;
}

h2 {
    background: orange;
    padding-left: 120px;
}
```

#### • [شاهد المثال](#)

### ملخص

أنت في طريقك لأن تتقن استخدام نموذج الصندوق في CSS ، في الدرس التالي سنلقي نظرة أقرب على الإطارات وكيف نحدد لها ألواناً مختلفة وكيف نستخدمها لنحدد أشكال العناصر.

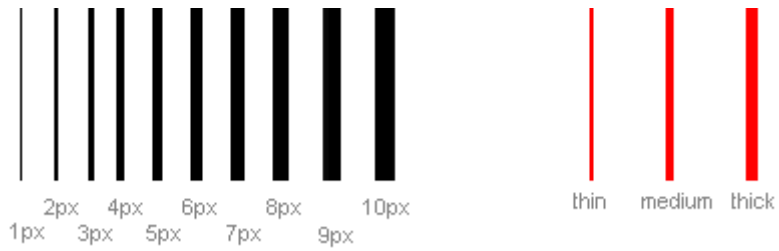
## الدرس ١١ : نموذج الصندوق - الإطارات ١١ Borders :

الجدول يمكن استخدامها لأغراض مختلفة، فقد تستخدم للزينة أو للفصل بين شيين، CSS تعطيك خيارات لا متناهية عندما تستخدم الإطارات في صفحتك.

- سماكة الإطار "border-width"
- ألوان الإطارات "border-color"
- تصميم الإطار "border-style"
- أمثلة لإطارات
- جمع كل الخصائص "border"

### سماكة الإطار "border-width"

سماكة الإطار تحدد من خلال القيمة border-width والذي يمكن تحديد قيمته من خلال الكلمات thin و medium و thick، أو من خلال قيمة رقمية تحدد سمكه بالبكسل، الرسم التوضيحي يبين هذه القيم ونتائجها:



### ألوان الإطارات "border-color"

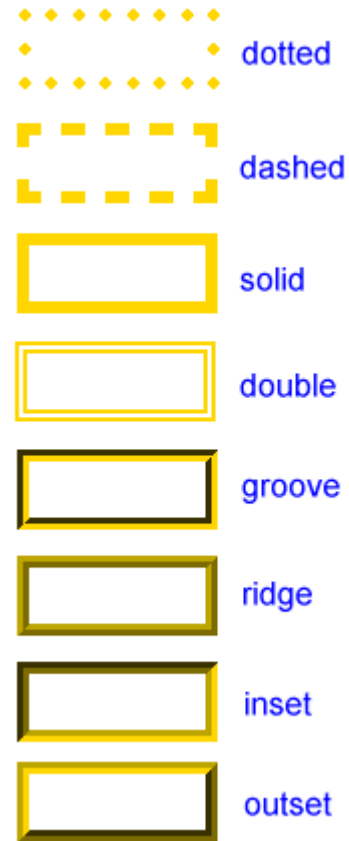


الخاصية border-color تحدد لون الإطارة، قيمة هذه الخاصية هي قيم الألوان العادية مثل "#123456" و "rgb(123,123,123)" أو أسماء الألوان مثل "yellow".

### تصميم الإطارات "border-style"

هناك أنواع مختلفة من الإطارات يمكنك أن تختار منها، في المثال التوضيحي أدناه هناك ٨ أنواع من الإطارات كما يعرضها إكسبلورر ٥.٥، كل الأمثلة المعروضة تستخدم اللون "gold" وقيمة سماكة الإطار محددة بالقيمة "thick" وتذكر أنك تستطيع عرض الإطار بألوان وسماكة مختلفة.

القيم none أو hiddem يمكن أن تستخدم عندما تريد إخفاء الإطار.



أمثلة لإطارات

الخصائص الثلاثة التي ذكرت أعلاه يمكن وضعها مع بعضها البعض لإنشاء جداول مختلفة، لتوضيح ذلك سنلقي نظرة على وثيقة قمنا بتحديد أنواع مختلفة من الإطارات للعنصر `<h1>` و `<h2>` و `<u1>` و `<p>`، النتيجة قد لا تكون جميلة لكن المثال يوضح الإمكانيات المتعددة:

```
h1 {
    border-width: thick;
    border-style: dotted;
    border-color: gold;
}

h2 {
    border-width: 20px;
    border-style: outset;
    border-color: red;
}

p {
    border-width: 1px;
```

```
border-style: dashed;
border-color: blue;
}

ul {
border-width: thin;
border-style: solid;
border-color: orange;
}
```

#### • [شاهد المثال](#)

يمكنك أيضاً تحديد خاصية تتعلق بجهة واحدة من الإطار، فنستخدم `top` للإطار العلوي، و `bottom` للإطار السفلي، و `left` للأيسر، و `right` للأيمن، وهذا مثال يوضح كيفية فعل ذلك:

```
h1 {
border-top-width: thick;
border-top-style: solid;
border-top-color: red;

border-bottom-width: thick;
border-bottom-style: solid;
border-bottom-color: blue;

border-right-width: thick;
border-right-style: solid;
border-right-color: green;

border-left-width: thick;
border-left-style: solid;
border-left-color: orange;
}
```

#### • [شاهد المثال](#)

جمع كل الخصائص "border"

كما هو الحال مع خصائص مختلفة، يمكن أن تجمع بين خصائص الجداول المختلفة في خاصية واحدة، ألقى نظرة على هذا المثال:

```
p {  
    border-width: 1px;  
    border-style: solid;  
    border-color: blue;  
}
```

يمكن اختصارها هكذا:

```
p {  
    border: 1px solid blue;  
}
```

### ملخص

في هذا الدرس تعلمت أن الإطارات في CSS لها عدد لا محدود من الخيارات. في الدرس التالي سنلقي نظرة على نموذج الصندوق وبالتحديد العرض والارتفاع.



## الدرس ١٢ : الارتفاع والعرض

حتى الآن لم نهتم كثيراً بأمر أبعاد العناصر التي قمنا بتعديل تصاميمها، في هذا الدرس سنلقي نظرة على كيفية تحديد ارتفاع وعرض العناصر.

- [width](#)
- [height](#)

تحديد العرض "width" الخاصية width تحدد عرضاً معيناً لعنصر محدد.

المثال البسيط أدناه يقدم لنا صندوقاً يمكن أن نكتب فيه نصاً:

```
div.box {
  width: 200px;
  border: 1px solid black;
  background: orange;
}
```

[شاهد المثال](#)

تحديد الارتفاع "height"

في المثال أعلاه لاحظنا أن ارتفاع الصندوق تحدده محتوياته، يمكن التأثير على ارتفاع العنصر باستخدام الخاصية height ، فمثلاً لنجرب أن نجعل ارتفاع الصندوق في المثال يبلغ ٥٠٠ بكسل:

```
div.box {
  height: 500px;
  width: 200px;
  border: 1px solid black;
  background: orange;
}
```

• [شاهد المثال](#)

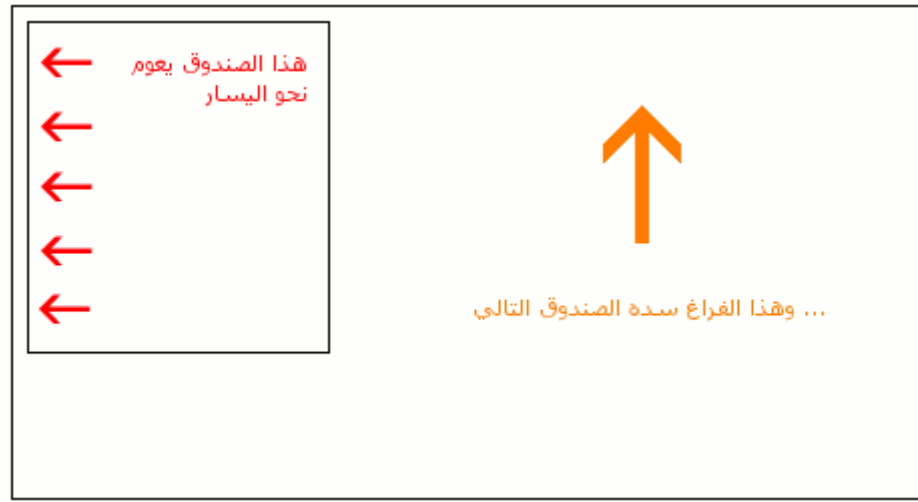
### ملخص

في الدروس 9، 10، 11 و12 أعطيناك مقدمة لنموذج الصندوق في CSS ، كما ترى نموذج الصندوق يعطيك الكثير من الخيارات الجديدة، ربما قمت باستخدام الجداول في HTML لإنشاء

التصاميم، لكن الآن ومع CSS ونموذج الصندوق يجب أن تتمكن من إنشاء تصاميم أنيقة وأكثر دقة وتتوافق أكثر مع المعايير القياسية لمنظمة W3C.

### الدرس ١٣ : تعويم العناصر (floats)

العناصر يمكن تعويمها إلى اليمين أو اليسار باستخدام الخاصية float ، هذا يعني أن الصندوق ومحتوياته ستعوم إلى اليمين أو اليسار من الصفحة، أو من ستعوم إلى إحدى الجهتين ضمن عنصر صندوق آخر، أنظر [الدرس ٩](#) للمزيد حول نموذج الصندوق، المثال التالي يوضح مبدأ تعويم العناصر:



إذا كان لدينا مثلاً نص يلتف حول صورة ستظهر النتيجة بهذا الشكل:



كيف يمكن فعل ذلك؟

ملف HTML للمثال أعلاه سيكون بهذا الشكل:

```
<div id="picture">
```

```

</div>
```

```
<p>causas naturales et antecedentes,
idcirco etiam nostrarum voluntatum...</p>
```

لكي نجعل الصورة تطفو إلى اليسار والنص يحيط بها فعلياً أولاً تحديد عرض الصندوق المحيط بالصورة ثم نعطي الخاصية float للقيمة left:

```
#picture {
    float:left;
    width: 100px;
}
```

• [شاهد المثال](#)

مثال آخر: الأعمدة

خاصية التعويم يمكن استخدامها للأعمدة في الصفحة، لإنشاء عمود عليك ببساطة أن تضع هيكلية للأعمدة في HTML باستخدام <div> كما يلي:

```
<div id="column1">
    <p>Haec disserens qua de re agatur
    et in quo causa consistat non
videt...</p>
</div>

<div id="column2">
    <p>causas naturales et antecedentes,
    idcirco etiam nostrarum voluntatum...</p>
</div>

<div id="column3">
    <p>nam nihil esset in nostra
    potestate si res ita se haberet...</p>
</div>
```

الآن عرض العمود الذي نريد هو ٣٣% ويمكننا ببساطة أن نعوم كل الأعمدة إلى اليسار باستخدام الخاصية: float

```
#column1 {
    float:left;
    width: 33%;
}

#column2 {
    float:left;
    width: 33%;
}

#column3 {
    float:left;
    width: 33%;
}
```

• [شاهد المثال](#)

float يمكن أن تحمل القيمة **left** أو **right** أو **none**.

الخاصية clear

خاصية clear تستخدم للتحكم بوضعية العناصر اللاحقة لأي عنصر الصفحة تم تعويمه.

تلقائياً العناصر اللاحقة تحرك لتغطي المساحة حول العنصر الذي تم تعويمه، أنظر إلى المثال أعلاه حيث ترى النص قد التف تلقائياً بجانب صورة بيل غيتس.

الخاصية clear يمكنها أن تحمل القيمة **left** أو **right** أو **both** أو **none** ، المبدأ هنا إذا وضعنا لخاصية clear قيمة "both" مثلاً فإن الحاشية العلوية للعنصر ستصبح أسفل الحاشية السفلية لعنصر يعلوه.

```
<div id="picture">
    
</div>

<h1>Bill Gates</h1>
```

```
<p class="floatstop">causas naturales et  
antecedentes,  
idcirco etiam nostrarum voluntatum...</p>
```

CSS: إذا لم ترغب في أن يلتف النص حول الصورة يمكنك إضافة هذه الخاصية في

```
#picture {  
    float:left;  
    width: 100px;  
}  
  
.floatstop {  
    clear:both;  
}
```

• [شاهد المثال](#)

## ملخص

التعويم مفيد في حالات كثيرة، ويستخدم غالباً مع وضعية العناصر، في [الدرس اللاحق](#) سنلقي نظرة على كيفية وضع العناصر في مكان ما بشكل نسبي أو مطلق.

## الدرس ١٤ : وضعية العناصر

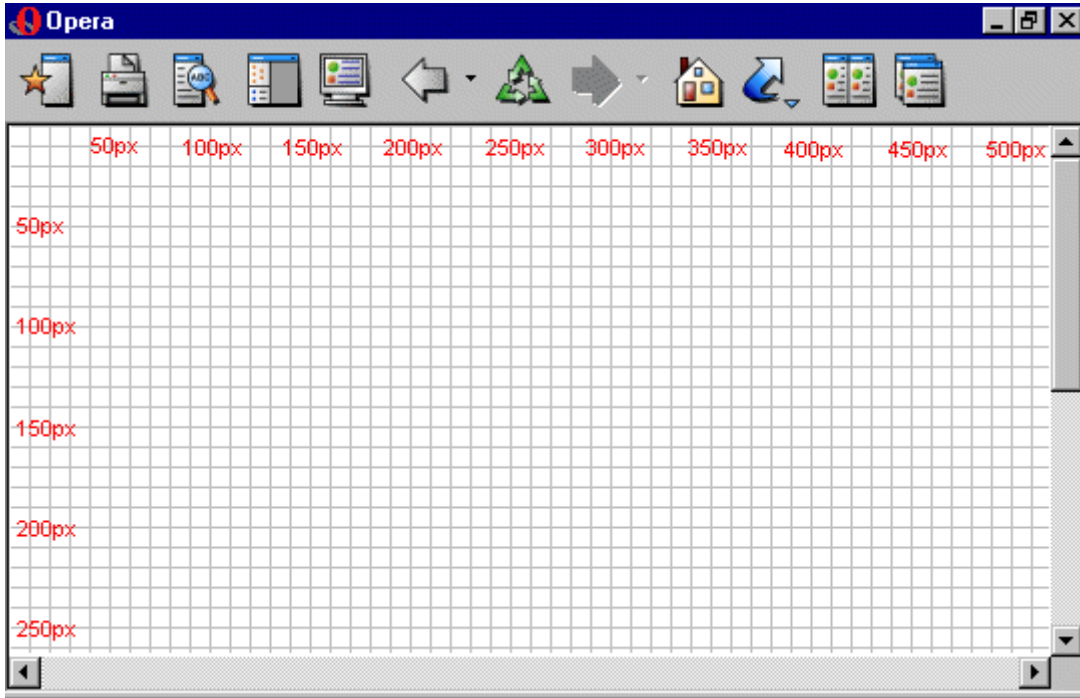
باستخدام وضعية العناصر في CSS يمكن أن تضع أي عنصر في المكان الذي تريده من الصفحة ويمكن لتعويم العناصر أن يساعدك على فعل ذلك أيضاً (اقرأ [الدرس ١٣](#))، وضعية العناصر في CSS يعطيك الكثير من الخيارات لإنشاء تصاميم متقدمة وعالية الدقة.

هذا ما سنتعلمه في الدرس:

- [مبادئ وضعية العناصر في CSS](#)
- [وضعية العناصر المطلقة](#)
- [وضعية العناصر النسبية](#)

مبادئ وضعية العناصر في CSS

تصور أن نافذة المتصفح منسقة بهذا النظام:



مبدأ وضعية العناصر في CSS هو أنك تستطيع وضع أي صندوق على أساس إحداثيات ونظام محدد.

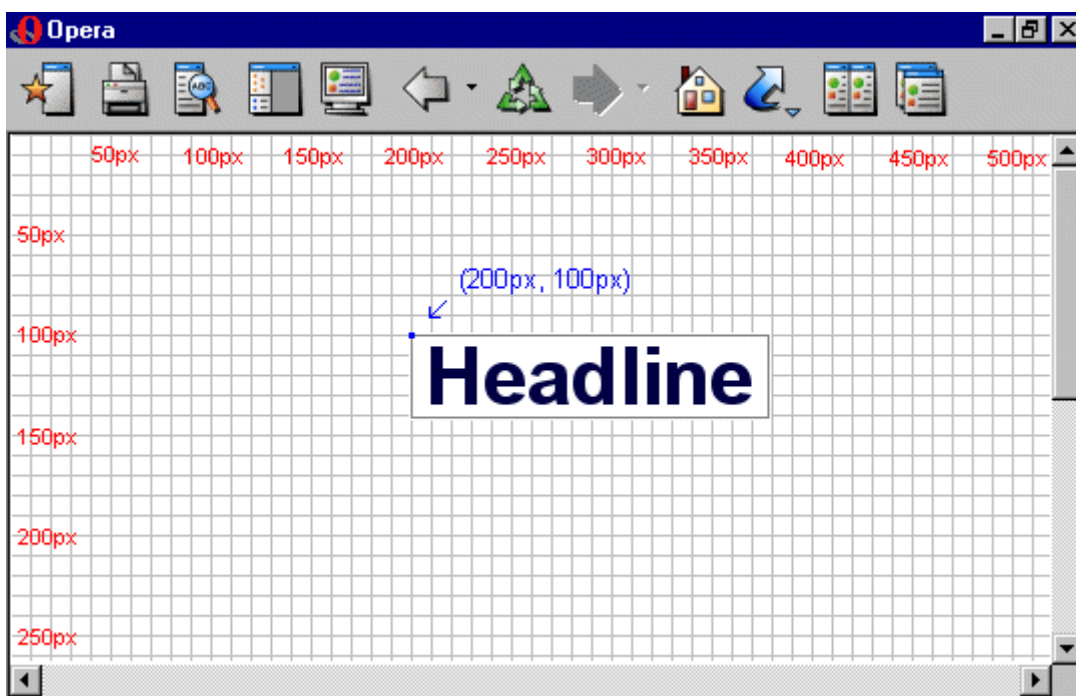
لنقل بأننا نريد أن نضع عنواناً في مكان محدد، باستخدام نموذج الصندوق - اقرأ [الدرس ٩](#) - يمكن للعنوان أن يظهر بهذا الشكل:

**Headline**

إذا أردنا للعنوان أن يكون موقعه على بعد ١٠٠ بكسل من أعلى الصفحة، و ٢٠٠ بكسل من يسار الصفحة فعلياً أن نكتب أوامر CSS بهذه الطريقة:

```
h1 {  
    position: absolute;  
    top: 100px;  
    left: 200px;  
}
```

والنتيجة ستكون بهذا الشكل:



كما ترى وضعية العناصر في CSS أسلوب دقيق جداً لوضع العناصر حيث تشاء، وهو أسهل بكثير من تجربة استخدام الجداول لفعل ذلك أو الصور الشفافة أو أي شيء آخر.

وضعية العناصر المطلقة

أي عنصر وضع بشكل مطلق لا يأخذ مساحة في الصفحة، هذا يعني أنه لا يترك فراغاً بعد أن يوضع بشكل مطلق.

لوضع أي عنصر بشكل مطلق يجب أن تعطي الخاصية position القيمة **absolute** ، ويمكن أيضاً استخدام الخصائص **left** و **right** و **top** و **bottom** لتحديد وضعية الصندوق.

كمثال للوضعية المطلقة قمنا بوضع أربع صناديق في الزوايا الأربعة للصفحة:

```
#box1 {
  position: absolute;
  top: 50px;
  left: 50px;
}

#box2 {
  position: absolute;
  top: 50px;
  right: 50px;
}

#box3 {
  position: absolute;
  bottom: 50px;
  right: 50px;
}

#box4 {
  position: absolute;
  bottom: 50px;
  left: 50px;
}
```

### • [شاهد المثال](#)

#### وضعية العناصر النسبية

لوضع العناصر بشكل نسبي عليك أن تعطي الخاصية `position` القيمة **relative** ، الاختلاف بين الوضعية النسبية والمطلقة هي كيفية التعامل مع وضع العنصر نفسه.

وضعية العنصر النسبية تحسب على أساس وضعه الأصلي في الصفحة، هذا يعني أنك إذا حركت العناصر إلى اليمين أو اليسار أو الأعلى أو الأسفل فإنه سيأخذ مساحة من النص بعد وضعه في مكانه المحدد.



كمثال على الوضعية النسبية يمكننا أن نجرب وضع ثلاث صور بشكل نسبي في الصفحة، لاحظ أن الصور تترك مساحة فارغة في أماكنها الأصلية من الصفحة:

```
#dog1 {
  position:relative;
  left: 350px;
  bottom: 150px;
}
#dog2 {
  position:relative;
  left: 150px;
  bottom: 500px;
}

#dog3 {
  position:relative;
  left: 50px;
  bottom: 700px;
}
```

• [شاهد المثال](#)

## ملخص

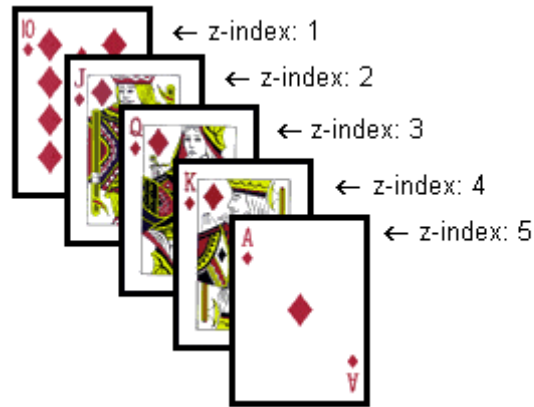
في الدرسين السابقين تعلمنا تعويم العناصر ووضعها في مكان محدد، طريقتان تعطيانك الكثير من الإمكانيات لبناء صفحات موقعك بدون الحاجة إلى استخدام الأسلوب القديم المتمثل في الصور الشفافة مع الجداول في HTML ، استخدا بدلاً من ذلك CSS فهي أكثر دقة وتعطيك الكثير من الخيارات المتقدمة، وهي أسهل في الإدارة وتوفر عليك الكثير من الجهد.

## الدرس ١٥ : طبقة فوق طبقة باستخدام z-index

يمكنها أن تتحكم بالأبعاد الثلاث، العرض والارتفاع والعمق، وقد تحدثنا بما فيه الكفاية عن العرض والارتفاع في الدروس السابقة، في هذا الدرس سنتعلم كيف نضع عناصر مختلفة في طبقات، باختصار هذا يعني ترتيب العناصر لكي نضع بعضها فوق بعض.

لهذا الغرض يمكنك أن تعطي لعنصر ما رقم (z-index) ، وهذا الخاصية تسمح للعناصر ذات الرقم الأعلى بأن تكون فوق العناصر التي ذات الرقم الأدنى.

لنقل بأننا نريد ترتيب صور للعبة الورق، يجب أن نعطي لكل ورقة رقم: z-index



في هذه الحالة الأرقام متتابة (من ١ إلى ٥) لكن يمكن تحقيق نفس النتيجة باستخدام خمسة أرقام مختلفة، المهم هو الترتيب المتصاعد.

المثال أدناه يوضح كيفية ترتيب البطاقات:

```
#ten_of_diamonds {
  position: absolute;
  left: 100px;
  top: 100px;
  z-index: 1;
}

#jack_of_diamonds {
  position: absolute;
  left: 115px;
  top: 115px;
  z-index: 2;
}

#queen_of_diamonds {
  position: absolute;
```

```
        left: 130px;
        top: 130px;
        z-index: 3;
    }

    #king_of_diamonds {
        position: absolute;
        left: 145px;
        top: 145px;
        z-index: 4;
    }

    #ace_of_diamonds {
        position: absolute;
        left: 160px;
        top: 160px;
        z-index: 5;
    }
}
```

#### • [شاهد المثال](#)

ترتيب العناصر فوق بعضها البعض سهل والإمكانيات كثيرة، يمكنك أن تضع الصور فوق النص أو النص فوق الصور أو تضع نصاً فوق نص ... إلخ.

#### ملخص

الطبقات يمكن استخدامها في الكثير من الحالات، فمثلاً جرب استخدام z-index لإنشاء مؤثرات على العناوين بدلاً من استخدام الصور، فهذا يجعل الموقع أكثر سرعة ويعطيه فرصة للظهور في محركات البحث.

## الدرس ١٦ : المعايير القياسية

منظمة W3C أو [World Wide Web Consortium](http://www.w3.org) هي منظمة مستقلة تضع المعايير التقنية لشبكة الويب (مثل HTML، CSS، XML وغيرها)، مايكروسوفت ومنظمة موزيلا وغيرها من الشركات كلها تشارك في منظمة W3C واتفقت على أن مستقبل شبكة الويب يكمن في المعايير القياسية.

إذا سبق لك أن جربت تصميم الموقع فأنت في الغالب تعرف أن هناك احتمال كبير بظهور الموقع بشكل مختلف في متصفحات مختلفة، وإنشاء موقع يظهر بشكل صحيح في كل المتصفحات الحديثة اليوم قد يستهلك الكثير من الوقت ويثير أعصابك.

الفكرة من وجود المعايير القياسية المتفق عليها هو وجود تقنيات ويب موحدة، هذا يعني أن مطور الموقع إذا قام بإنشاء موقع فهو يعلم أنه سيظهر بالشكل المطلوب في كل المتصفحات وأنظمة التشغيل، لذلك نفضل أن نقوم باستخدام المعايير القياسية ونقوم بالتأكد من أن ملفات CSS التي قمت بكتابتها تتوافق مع هذه المعايير..

### مدقق CSS

لتبسط عملية تعلم [معايير CSS](http://www.w3.org) قامت منظمة W3C بتطوير أداة [تدقيق](http://www.w3.org) تقوم بقراءة ملفات CSS وتعطيك النتائج مع الأخطاء والتحذيرات إن كان ملف CSS يحوي أي خطأ.

ولكي نيسط عليك الأمر يمكنك أن تتأكد بنفسك من صحة ملفات CSS مباشرة من هذه الصفحة، ببساطة ضع عنوان ملف CSS الخاص بموقعك واضغط على زر "تدقيق" وستنتقل إلى موقع W3C لترى النتيجة وتعرف الأخطاء إن وجدت.

<http://www.w3.org>

تدقيق

إذا لم يجد المدقق أية أخطاء ستظهر لك هذه الصورة والتي يمكنك أن تضعها في موقعك لتبين بأنك تستخدم المعايير القياسية:



المدقق يمكن إيجاده على هذا الرابط <http://jigsaw.w3.org/css-validator/> :

تم بحمد الله وشكره