

# بسم الله الرحمن الرحيم

دورة CSS3

تقديم: أنيس حكمت أبو حميد

البريد الإلكتروني: [aneeshikmat@2nees.com](mailto:aneeshikmat@2nees.com)

الموقع الإلكتروني: [www.2nees.com](http://www.2nees.com)

أَعُوذُ بِاللَّهِ مِنَ الشَّيْطَانِ الرَّجِيمِ

شَهِدَ اللَّهُ أَنَّهُ لَا إِلَهَ إِلَّا هُوَ وَالْمَلَائِكَةُ  
وَأُولُوا الْعِلْمِ قَائِمًا بِالْقِسْطِ لَا إِلَهَ إِلَّا  
هُوَ الْعَزِيزُ الْحَكِيمُ

ال عمران (18)

# الفهرس

● المقدمة (من المهم الإطلاع على المقدمة لما تحتويه من معلومات قبل البدء بهذه الدورة).

● Vendor-Prefix

● Media Query

● SELECTORS

● Web Fonts

● Text Effects and Typographic Styles

● Multiple Columns

● Background Image And Decorative Properties

# الفهرس

Border And Box Effects ●

Color & Opacity ●

Gradient ●

2D Transformations ●

Transition & Animation ●

3D Transformations ●

الخاتمة ●

# المقدمة

الحمد لله الحي القيوم، الباقي وغيره لا يدوم، رفع السماء وزينها بالنجوم، وامسك الأرض بجبال في الختوم، صور بقدرته هذه الجسوم، ثم أماتها ومحا الرسوم، ثم ينفخ في الصور فإذا الميت يقوم، ففريق إلى دار النعيم وفريق إلى نار السموم.

اللهم صل وسلم على نبيك وحبيبك محمد صل الله عليه وسلم، وعلى اله وصحبه وسلم.

إخوتي وأحبائي وأصدقائي الرائعين، أعود لكم اليوم بدورة جديدة ومهمة في عالم تصميم صفحات الإنترنت، وهي CSS3، هذه الدورة التي تعد مكمل مهم لمفهوم تنسيق الصفحات الإلكترونية باستخدام تقنيات عالمية جديدة، تسبق العصر القديم، وتجعل لصفحات الويب مدخلا جديدا، وابداعا أسمى وأعلى من العصر السابق.. فنحن الآن نتحدث عن ال CSS3 ^\_^.

# المقدمة

والآن، بما أننا سنتحدث عن ال CSS3، فينبغي أن نعلم، أن ال CSS3 هي إصدار جديد من إصدارات ال CSS .. وعندما كنا نتحدث بالسابق عن مفهوم أو مصطلح CSS، فكنا نقصد ال CSS2.1.

كما أنه أيضا إلى معلومة مهمة جدا قبل البدء، وهي أن هذه الدورة ستتحدث عن مواضيع متقدمة في ال CSS، ولهذا يجب عليك قبل البدء أن تكون قد وصلت إلى مرحلة إتقان التكويد بال HTML وال CSS على أقل تقدير.

المتطلبات السابقة:

<http://goo.gl/tt3xXI> :HTML

<http://goo.gl/0kBdCC> :Css

# المقدمة

أيضا فيما يتعلق بال css3، فإنه من المهم معرفة بعض الأمور التي تتعلق بها في الماضي والحاضر وحتى النظرة الى المستقبل!!

نعم فلا تتعجب، فما أريد إخبارك به الآن، هي أن css3 ليست مستقرة تماما، وأقصد بهذا أنها ما زالت في حركة تطوير مستمرة، وبهذا فهي غير متوافقة مع جميع المتصفحات بشكل مباشر، فإما قد تحتاج الى استخدام vendor مثل ال webkit وغيرها .. وإما أن الخاصية قد لا يدعمها متصفح معين (طبعا الكلام موجه خصوصا الى IE ابييه)، ولكن بفضل الله تعالى، الآن أغلب المتصفحات الرئيسية والحديثة أصبحت تدعم خصائص ال css3 الرئيسية، ولكن تذكر دائما أن تقوم بالتحقق والبحث هل تدعم خاصية معينة من قبل المتصفحات أم لا.. ويمكنك فحص هذه الخاصية من خلال هذا الموقع <http://caniuse.com/#feat=css-animation>.. (هذا الرابط يمثل الموقع مع مثال لمعرفة مدى تجاوبية المتصفحات مع ال css-animation)... فمثلا ستلاحظ أن ie8 & ie9 لا يدعمو هذه الميزة -\_-...

# المقدمة

والآن، بعد هذا الكلام ..ننتقل الى جزئية مهمة جدا وهي أن ال

## CSS3 ليست ال HTML5

وسبب ذكر هذه النقطة، هو أن الكثيرين قبل البدء بتعلم ال css3 يظن أنها html5 أو يظن أنهما موضوعين يندرج أحدهما تحت الآخر، وهذا غير صحيح أيضا .. (يمكنك الإطلاع [على دورة ال HTML5 من هنا](#))، ولاكن هذا لا ينفي وجود جزئيات تعمل باستخدام ال HTML5 ويكون ال css3 هو المسؤول عن اعطائها النسق المطلوب ...

ولذلك وجب التنويه لهذه النقطة



# المقدمة

## ملاحظات:

(1) إن معظم هذه الأمثلة تم تطبيقها على متصفح ال Google Chrome لذلك إذا أردت استعراض الأمثلة فيفضل استخدام متصفح ال chrome، والسبب في ذلك أنني في أغلب الأمثلة لم أراعي التوافقية الا لمتصفح ال chrome وال firefox عادة .. إختصارا لحجم الشيفرة البرمجية وتقليل الوقت المستغرق، مع العلم أنه سيتم الشرح عن كيفية معرفة التوافقية لأي خاصية من الخصائص مع المتصفحات، وهذا أمر مهم.

**Chrome: Version 36.0.19 و Firefox 37.0.1 (الإصدارات التي تم اختبارها)**

(2) إن الخصائص التي تم ذكرها هنا في مجملها هي خصائص الآن مدعومة من جميع المتصفحات الرئيسية، لذلك الخصائص التي تقرأها يمكنك تطبيقها بإذن الله تعالى في عملك، الا اذا قمت بذكر أن هذه الخاصية ما زالت غير مدعومة، بالإضافة الى ذلك يجب عليك التحقق بنفسك من التوافقية كما قلنا بالملاحظة الأولى.

# المقدمة

(3) التطبيق مهم جدا للأمثلة، ولكي تقوم بالتطبيق، فإنني أقوم بوضع صورة الناتج والشيفرة البرمجية كتسهيل للشرح وإيصال المعلومة، وجعلك تقوم بالتطبيق المباشر للمثال، وبعد الإنتهاء من فكرة معينة أو جزئية معينة، ستجد أن جميع الأمثلة التي تم ذكرها موجود في صفحة html مرقمة بالتسلسل .. ويمكنك الدخول اليها من خلال النقر على هذا



(4) هذه الدورة مجانية ومقدمة لكل شخص، وأستثني من ذلك من سيتعلم أي أمر من هذه الدورة ويستخدمها في معصية الله سبحانه وتعالى، اللهم إني قد بلغت .. اللهم فاشهد..

# ال Vendor-Prefix css

لقد تطرقنا في المقدمة الى نقطة مهمة جدا، وهي أن كثير من الخصائص المستخدمة لم تعتمد أو لم يتم مراجعتها وتطويرها للتوافق مع متصفح معين، لهذا قلنا أنه يجب أن تحذر عندما تستخدم أي من ال خصائص ال css3 .. وقد ذكرنا أيضا أنه بعض المتصفحات لتدعم خصائص معينة تحتاج الى Vendor .. وهذا هو موضوعنا .. والأهم من ذلك هو أن ال vendor والمتصفحات التي سنتحدث عنها علي المتصفحات الرئيسية وهي IE(internet explorer)، و Google Chrome و Firefox و Opera و ال Safari .. يستخدم ال vendor بالشكل التالي: -vendor-css3Prop: value;

-o- ال vender الخاص ب Opera

-webkit- :متصفحات ال webkit عموما وأشهرها ال chrome و ال safari ..

-ms- : متصفح ال IE

-moz- : Firefox

# ال Vendor-Prefix css

كما لا حظت بالشريحة السابقة .. فلكل متصفح vendor خاص به بنائاً على نوع المتصفح، لكن بالإضافة الى ما سبق .. فإننا نقوم دائماً بكتابة الصيغة القياسية بعد ال vendors .. والسبب في هذا، مستقبل ال css3 يسير ويتقدم بسرعة .. وبهذا نضمن في المستقبل عند الاستغناء عن ال vendor أننا لن نحتاج الى العودة وتعديل الكود القديم .. (شغلك يستمر بالعمل ^\_^)، وعلى أي حال فهذه الفكرة أصبحنا نشاهدها الآن وأثناء كتابتي لهذه الشرائح .. فأصبح أغلب العمل يكون على استخدام ال vendor الخاص بال webkit مع الصيغة القياسية ...

تذكر استخدام موقع can I use الذي تم ذكره في المقدمة لمعرفة المتصفحات التي تدعم كل خاصية وهل تحتاج الى vendor أم لا ^\_^

# ال Vendor-Prefix

والآن لنشاهد مثالا سريعا على الموضوع ^\_^، هذه احدى خصائص ال css3 وهي لمنع عملية ال copy ..

```
-webkit-user-select: none; /* Chrome all / Safari all */
```

```
-moz-user-select: none; /* Firefox all */
```

```
-ms-user-select: none; /* IE 10+ */
```

```
-o-user-select: none; /* opera */
```

```
user-select: none;
```

هذا المثال يشرح ويوضح فكرة استخدام ال vendor بإذن الله تعالى ..

كما أنه اذا قمت بالدخول الى موقع لفحص دعم هذه الخاصية ..ستجد أن ال ie8 و لا يدعم هذه الخاصية ..بالإضافة الى opera mini .. (وهذا حسب تاريخ كتابة هذه الشرائح 2015-1-26)

وإذا أختير الكاتبُ لرسالةٍ ما، شعرَ بقوةٍ تفرضُ نفسَها عليه؛ منها سِنَادُ رأيه،  
ومنها إقامةُ برهانه، ومنها جمالُ ما يأتي به، فيكون إنساناً لأعماله وأعمالها جميعاً،  
له بنفسه وجودٌ ولد بها وجودٌ آخر؛ ومن ثمَّ يُصبحُ عالماً بعناصره للخيرِ أو الشرِّ  
كما يُوجِّه؛ ويُلقي فيه مثلُ السرِّ الذي يُلقي في الشجرة لإخراجِ ثمرها بعملٍ طبيعيٍّ  
يُرى سهلاً كلَّ السهلِ حينَ يتمُّ، ولكنه صعبٌ أيُّ صعبٍ حينَ يبدأ.

## كتاب وحي القلم (1)

# Media Query

والآن لنبدء على بركة الله في أول مواضيع ال css3 وهو ال media query

يتواجد في عصرنا الحالي، وعلى شبكة الانترنت العديد من المواقع، هذه المواقع في السابق، كانت مصممة خصيصا لأحجام الشاشات الطبيعية مثل شاشة اللاب توب أو شاشة الكمبيوتر.. وهكذا، لكن مع تطور التكنولوجيا وظهور العديد من الأجهزة المختلفة والمتنوعة، أصبح من المهم الأهتمام بأمر هذه الأجهزة، وخصوصا بحلول الأجهزة الذكية والتابلت محل الشاشات الكبيرة لأغراض التصفح، وذلك لأسباب كثيرة كما نعلم، هذا الأمر أدى أن يجعل منظمة ال w3c من التوجه الى تطوير ال module الخاص ب media query ليصبح داعما ومتوافقا للأجهزة المختلفة، وهذه الحركة الرائعة والمخيفة في عالم الويب، أعطتنا قدرة وإمكانية على التحكم بالتصميم والتنسيق وطريقة العرض الخاصة بأي صفحة على موقع الانترنت ستعرض على أي جهاز، وظهرت هنا إحدى أهم الأفكار وهي ال Responsive Design ^\_^

# Media Query

والآن .. سنتطرق الى Syntax الخاص بهذه ال media query وهو على الشكل التالي:

```
<link href="file" rel="stylesheet" media="logic media and (expression)">
```

OR

```
@media logic media and (expression) { rules }
```

فكما تلاحظ .. من الصيغة الأولى .. فهي تشبه طريقة تضمين ملف css، لكن الجديد هو ال media ..

أما الصيغة الثانية فهي طريقة كتابة التنسيق الخاص ب css داخل الصفحة .. (وكأني قمت بعمل شرط لتنفيذ التنسيق ضمن الملف نفسه) ..

والآن السؤال الأقوى .. أنا \*\_\* مش فاهم اشي من المكتوب P: .. اذن لننتقل الى الشريحة التالية .. لنشرح مكونات هذه الصيغة ..



# Media Query

شرح مكونات الصيغة السابقة:

(1) Logic: وهي إما only وإما not للصيغة الثانية أما **الصيغة الأولى** فهي ما تم ذكره بالإضافة الى and

not: وتستخدم في حال أردنا تنفيذ rules معينة عندما لا ينطبق exp (عكس)  
only: تستخدم هذه القيمة عندما نريد اخفاء أو عدم تنفيذ ال rules على بنائاً على media معينة.

(1) أما ال Expressions فهي التعبير الذي سنقوم بكتابته ليتم تنفيذ التنسيق اذا حدث ..مثل تحديد أقل عرض 480 بكسل ...وهكذا

(2) أما ال Rules الموجودة في الصيغة الثانية فيقصد بها الشيفرة البرمجية المراد تنفيذها عند تحقق النقطة 2 (Expressions).

(3) Media: هو الجهاز المستهدف وله خيارات كثيرة سنتطرق لبعضها بإذن الله تعالى بالشرائح التالية ..

# Media Query

ملاحظة: ال media query لا تدعم ال IE8، لذلك نستخدم hack لمتصفح ال IE8 يقوم على تنفيذ التنسيق الخاص بصفحة ال Desktop مثل

```
<!--[if lt IE 9]>  
<link href="desktop.css" rel="stylesheet" media="screen">  
<![endif]-->
```

الآن كيف يمكنني كتابة ال Expressions ؟

الجواب بكل بساطة عن طريق استخدام خصائص ال css التي نعرفها مثل width وال min-width وال max-width وال height وال min-height وال max-

... height

وبالحقيقة أكثر ما نستخدمه هو ال min-max لل width لبناء تصميم ال .. responsive

```
@media screen and (max-width: 480px) {  
  P {Rules  
    color : red;  
  }  
}
```

Exp

مثال:

# Media Query

أما ال media فهي

screen, print, handheld, aural/speech, embossed, braille,  
projection, tv, tty, all

وأهم الخصائص التي نعمل عليها بالتحديد هي ال screen, all أيضا القيم  
الأخرى .. لكن هذا بالغالب ... طبعا .

ال screen: اعمل على شاشات الكومبيوتر والاب توب والتابلت والأجهزة  
الذكية ...

All: جميع الأجهزة.

Print: عند الطباعة ..

Projection: وسائل العرض مثل ال عرض slides ...

Tv: شاشات التلفزيون ..

# Media Query

والآن .. لنشاهد كيف يمكننا تضمين ملف CSS خاص لأحجام الهواتف الصغيرة .. شاهد المثال:

```
<link rel="stylesheet" media="screen and (max-width: 480px)" href="small.css" type="text/css" >
```

وهذه تعني إعمل على النافذة الصغيرة والتي يقل عرضها أو يساوي 480 بكسل <sup>^</sup> \* \_

أيضا يمكننا وضع Range أو شرط .. فمثلا أكبر من وأقل من .. شاهد المثال:

```
@media screen and (min-width: 480px) and (max-width: 800px) {  
  p {  
    color: red;  
  }  
}
```

في هذا المثال .. سيتم تنفيذ الشيفرة البرمجية إذا كانت أكبر أو يساوي 480 بكسل و أقل أو يساوي 800 بكسل

# Media Query

الآن .. سأتطرق الى ذكر نقاط مهمة جدا, ولا تقلق بخصوص الأمثلة, سيتم وضع العديد من الأمثلة بإذن الله تعالى, في نهاية هذا الشرح ..

من الخصائص أو القواعد التي يمكن استخدامها وكما ذكرناها ال height وال width داخل ال media query .. لكن هناك خصائص أخرى مثل

device-width و device-height .. وبالتأكيد يمكن استخدام ال min/max-device-w/h

بالإضافة الى وجود خاصية ال orientation .. فما هي وظيفة هذه الخصائص؟

device-width: هذه تعطينا عرض شاشة الجهاز. والآن ستقول بكل تأكيد.. ما الفرق بينها وبين ال width ؟ .. وهذا سؤال مهم ^\_^

جواب هذا السؤال هو .أن ال width تستخدم لمعرفة عرض نافذة العرض (المتصفح الذي يعرض الصفحة) .. نعم, ما تفكر به صحيح.. اذا استخدمت ال device-width فإن التنسيق سيتم تنفيذه على جهاز شاشة العرض حجمها ينطبق عليه الشرط .. وبهذا من جهاز سطح المكتب لن يعمل عليه هذا التنسيق حتى ولو قمت بتصغير المتصفح .. أما بال width .. فسيعمل التنسيق على اي شاشة بمجرد أن حجم المتصفح وصل للشرط ...

# Media Query

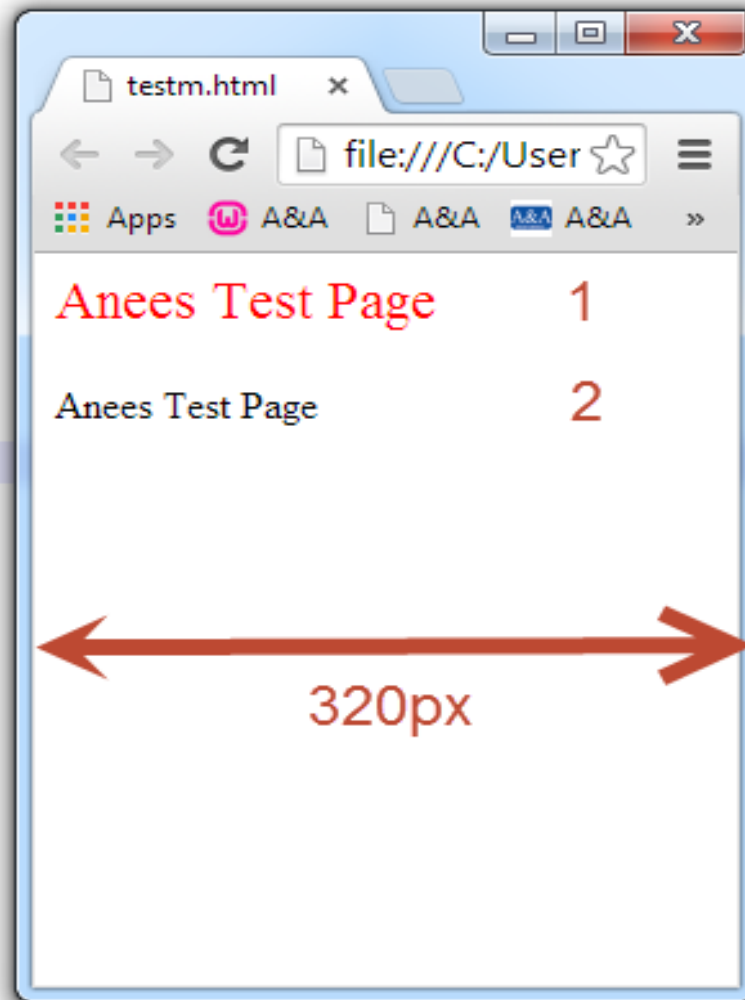
والآن لنرى مثالا على هذه النقطة:

```
<style>
  @media screen and (max-width: 320px) {
    .p1 {
      1 color: red;
      font-size: 22px;
    }
  }

  @media screen and (max-device-width: 320px) {
    .p2 {
      2 color: blue;
      font-size: 22px;
    }
  }
</style>
<head>
<body>
  <p class="p1">
    Anees Test Page
  </p>

  <p class="p2">
    Anees Test Page
  </p>
</body>
```

لاحظ أنه عند تصغير حجم المتصفح (نافذة العرض) فإنه في الحالة الأولى .. استجاب التصميم, وفي الحالة الثانية, لم يستجب التصميم لذلك



# Media Query

أما بخصوص وظيفة ال orientation فهي خاصية يمكنك استخدامها لتحديد إذا ما كان الجهاز في وضع ال Landscape أو portrait ..



portrait



Landscape

```
@media all and (max-width: 1500px) and (orientation: landscape) {  
  .p2 {  
    color: green;  
    font-size: 22px;  
  }  
}
```

هذا الشرط يقرأ بالشكل التالي, إعمل أيها التنسيق على جميع شاشات العرض, بحيث يكون حجم نافذة العرض أقل أو يساوي 1500 وبشرط أن يكون اتجاه العرض

والآن شاهد هذا المثال:

Landscape

# Media Query

ملاحظة: إذا أردت تجربة الشرط المثال السابق .. فسينفذ الشرط .. إذا كان حجم المتصفح أقل أو يساوي 1500 .. وكان نسبة عرض المتصفح أكبر من الطول الخاص به .. سيعمل شرط ال orientation على أن العرض الآن landscape .. شاهد نتيجة المثال:

```
File Edit Search View Encoding Language Settings Macro Run Plugins Window
?
information.txt x testm.html x
18      }
19
20      @media all and (max-width: 1500px) and (orient
21      .p3 {
22          color: green;
23          font-size: 22px;
24      }
25      }
26      </style>
27      </head>
28      <body>
29      <p class="p3">
30          Anees Test Page 3 (landscape)
31      </p>
32      </body>
33      </html>
34
le Ln: 28 Col: 11 Sel: 0 | 0    Dos\Windows    UTF-8 w/o BOM    INS
```

to create a whole new version of the

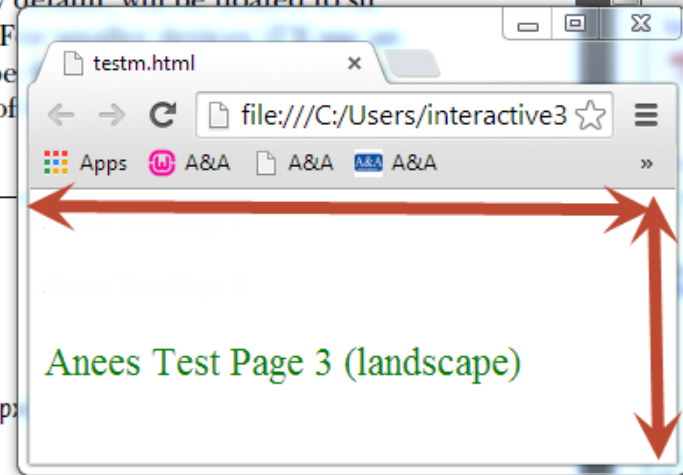
ntent for display on two different

at, by default, will be floated to sit

ane. F

not be

use of



: 320px

element, with two floated div elements



# Media Query

والآن .. نأتي الى موضوع مهم جدا جدا .. وهو ال viewport, ويجب التركيز في هذا الموضوع إن أردت أن يكون عملك صحيحا ..

لقد تكلمنا في الصفحات الماضية عن وجود خاصيتين وهما ال width وال device-width ... وقلنا أن ال width بنائاً على حجم النافذة (المتصفح) وال device-width بنائاً على الحجم الفعلي للجهاز ..

لذلك فإن المستخدم هو device-width لإعطاء التنسيق للأجهزة الصغيرة مثل الخلويات وغيرها ..

لكن هناك مشكلة نحن نحتاج أن نستخدم ال width حتى يعمل التنسيق الخاص بأي جزئية بنائاً على حجم المتصفح .. لكن المشكلة تكمن هنا بأن هناك طريقة أو آلية لكل متصفح لحساب مقدار التصغير أو التكبير الذي يحتاجه حتى يتلائم مع العرض .. وهذا يعني أن متصفح ال ie يختلف عن متصفح الأندرويد ويختلف عن ال opera .. الخ

# Media Query

الآن ..منطقة العرض الخاصة بالمتصفح هي التي تسمى ب **viewport**, وهي التي يتم محاكاتها وهي التي تختلف من جهاز الى آخر, ومن متصفح لآخر .. اذن أين المشكلة الى الآن؟

المشكلة في أنه اذا استخدمت ال `max-width:480px` على سبيل المثال فإنه سيتم المقارنة مع حجم ال `viewport` وليس مع حجم الجهاز ..ولو افترضنا أن الجهاز يستخدم متصفح الأندرويد 1 فإن قيمة ال `viewport` ستكون هي `800px` `0_0` .. اذا ما الحل `*_*`

الحل بكل بساطة وجمال .. هو استخدام هذه ال `meta tag`:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

هذه ال `metatag` ..دائماً استخدمها وضعها في ال `head` ..وهنا أجبرت المتصفح أياً كان نوعه على أن يجعل ال `width` يساوي عرض الجهاز .. وأن قيمة ال `zoom` هي 1 أي الحالة الافتراضية لحجم الشاشة ..(أي لا يوجد أي `zoom`)

# Media Query

ملاحظة 1: وهي لا تستخدم هذا ال tag الا اذا كان موقعك responsive ^\_^

ملاحظة 2: هناك خصائص أخرى يمكنك استخدامها مع هذا ال tag مثل ال user-scalable وهي لمنع عملية ال zooming مثلا .. ولها قيمتان yes, no .. وغيرها ..مثل تحديد الحد الأعلى لل zoom والحد الأدنى ..الخ

ملاحظة 3: إن ال media query بالإصل هي خصائص css2, لكن ما تم اضافته اليها من خصائص جديدة والتي ذكرناها ..فهي من سمات ال css3 لذلك ستجد أنه في ال css2 كنا نستخدم ال media query لل print كمثل \*\_^

ملاحظة 4: المتصفحات التي تدعم هذه الخصائص جميع المتصفحات الرئيسية وال ie من الإصدار التاسع (لذلك عليك استخدام ال hack الذي ذكرناه سابقا).

# Media Query

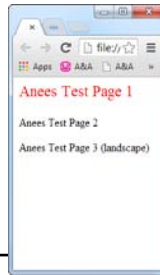
والآن لنشاهد أمثلة متنوعة على موضوع ال media query .. وللعلم .. من خلال هذا الموضوع لوحده، يمكنك أن تصنع موقع متجاوب (Responsive)، نعم ولا تتعجب من ذلك.. فهذا النوع من المواقع، ما هو الا مجموعة من صفحات تنسيق مختلفة تنطلق عند تحقق شرط معين .. نقوم على افحاء جزئية معينة أو اظهارها، تصغير جزئية أو تكبيرها .. وهكذا ..

شاهد الأمثلة من هنا:

المثال 1: (يعرض الأمثلة السابقة ويكون التنسيق الخاص بال css inline).

بخصوص لهذا المثال .. قم بتجربة جعل ارتفاع المتصفح أكبر من العرض .. وشاهد النتيجة ^\_^ ..

هذه عينة عشوائية لبعض النتائج ^\_^ \*



Anees Test Page 1

Anees Test Page 2

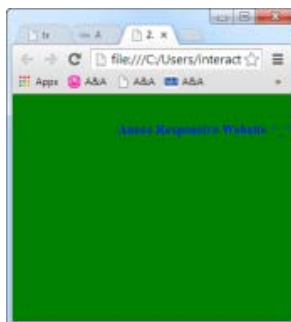
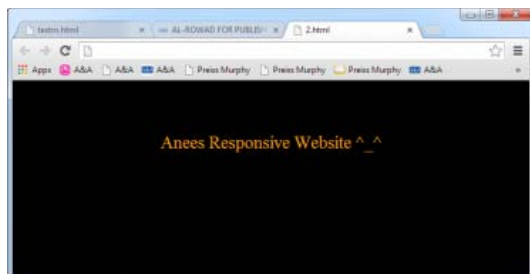
Anees Test Page 3 (landscape)

# Media Query

المثال 2: هذا المثال مهم جدا بكل تفاصيله, يرجى الإنتابه له, ويرجى الإنتباه لصفحات التنسيق التي ربط به, والتحكم بحجم المتصفح ..حتى ترى النتائج وكأنك تستخدم الهاتف المحمول..



مقتطفات من تطبيق المثال: (الفكرة أن تقوم بالتجربة والتحقق بنفسك من الشيفرة البرمجية ^\_^)



# Media Query

ملاحظة: يمكنك أيضا استخدام بعض الأدوات لعرض الموقع بأحجام شاشات مختلفة مثل استخدام أحد المواقع <http://mobiletest.me/> أو يمكنك استخدام أداة لمتصفحك مباشرة أو تجدها في بعض المتصفحات بشكل افتراضي



دع الأيام تفعل ما تشاء ..... وطب نفسا إذا حكم القضاء  
ولا تجزع لحادثة الليالي ..... فما لحوادث الدنيا بقضاء  
وكن رجلا على الأهوال جلدا ..... وشيئك السماحة والوفاء  
وإن كثرت عيوبك في البرايا ..... وسرك أن يكون لها غطاء  
تستر بالسخاء فكل عيب ..... يغطيه كما قيل السخاء  
ولا تر للأعداء قط ذلا ..... فإن شماتة الأعداء بلاء  
ولا ترج السماحة من بخيل ..... فما في النار للظمان ماء  
ورزقك ليس ينقصه التأي ..... وليس يزيد في الرزق الغناء  
ولا حزن يدوم ولا سرور ..... ولا بؤس عليك ولا رخاء  
إذا كنت ذا قلب قنوع ..... فأنت ومالك الدنيا سواء  
ومن نزلت بساحته المنايا ..... فلا أرض تقيه ولا سماء  
وأرض الله واسعة ولكن ..... إذا نزل القضاء ضائق الفضاء  
دع الأيام تغدر كل حين ..... فما يغني عن الموت الدواء

من شعر الإمام الشافعي رحمه الله تعالى

# SELECTORS

والآن بعد حديثنا الشيق عن ال media query .. ننطلق الى موضوع جديد ورائع  
أيضا, وهو ال Selectors .. !!

نعم، Selectors، لا تتعجب، فهناك كثير من ال Selectors التي تم اضافتها  
لتساعدك على تنسيق صفحات الويب، وبذات الوقت فإنها تخفف علينا من  
استخدام الجافا سكربت.. وكمثال قد تكون استخدمته كثيرا ولا تعلم أنه css3  
وهو last-child كمثل ^\_^ ... هذا ال Selectors هو أحد اضافات ال css3  
الجديدة والرائعة ^\_^

اذن .. ما هي الإضافات الأخرى؟

هذا ما سنتعلمه من هذا الموضوع بإذن الله تعالى ^\_^ \*



# SELECTORS

(1) `element1 ~ element2`: وهذه الخاصية تعني، قم بتنفيذ تنسيق معين لجميع عناصر ال `element2` في حال سبقت في أي مكان بصفحة ب `element1` وهذا يعني، أنه اذا جاء ال `element1` في جزء معين بالصفحة، فإن كل عناصر ال `element2` التي تأتي بعده ستأخذ التنسيق، **لكن يشترط لذلك أن يكونا داخل نفس ال parent..**

والآن لنشاهد المثال، فهو سيوضح الكلام السابق بإذن الله تعالى:



مهم جدا أن ننظر الى الشيفرة البرمجية المخصصة للمثال ومقارنتها بالنتائج، لديك 3 حالات وضعتها .. شاهد الفكرة من كل حالة <sup>٨</sup>\_\*

# SELECTORS

(2) `[attribute^=value]`: هذه الخاصية الجميلة تعني أعطي التنسيق لجميع العناصر التي لها attribute يبدأ بقيمة معينة أو يساويها `^_..` كمثال: `div[class^='anees']` فإنه سيقوم بتنفيذ التنسيق الذي أريده لجميع ال `div` التي لديها كلاس يبدأ ب `anees` `^_..`

لنشاهد المثال معا:

This is div start at anees class (anees) is used

This is div start at anees class (aneesClass) is used

```
<body class="ex-4">
  <div class="anees">
    This is div start at anees
  </div>
  <div class="aneesClass">
    This is div start at anees
  </div>
  <div class="Classanees">
    This Div used Classanees
  </div>
  <p class="anees">This is p s
</body>
```

```
.....
} .ex-4 div[class^='anees'] {
  background: orange;
  padding: 20px;
  margin: 10px;
  color: #fff;
  font-size: 25px;
  width: 50%;
}
- }
```

This Div used Classanees

This is p start at anees class (anees) is used

# SELECTORS

(3) `:[attribute$=value]`: هذه الخاصية الجميلة هي الخاصية المعاكسة في الوظيفة للخاصية السابقة, أي أنها تهتم أن تأتي القيمة مساوية أو في نهاية الخاصية, كمثال `div[class='anees']` فسيكون التنسيق الجديد لجميع ال `div` التي تنتهي ب `anees` أو تساوي أنيس .. لنفم بتجربة ذلك على المثال السابق ^\_^

This is div end at anees class (anees) is used

الآن: لنشاهد المثال معا \*\_^

\*\*\*\*\*  
This is div start at anees class (aneesClass) is used  
\*\*\*\*\*

This Div used (Classanees)

\*\*\*\*\*  
This is p end at anees class (anees) is used  
\*\*\*\*\*

```
<body class="ex-5">
  <div class="anees">
    This is div end at anees class (anees) is used
  </div>
  *****
  <div class="aneesClass">
    This is div start at anees class (aneesClass) is used
  </div>
  *****
  <div class="Classanees">
    This Div used (Classanees)
  </div>
  *****
  <p class="anees">This is p end at anees class (anees) is used
</body>
```

# SELECTORS

(4) `[attribute*=value]`: بعد دراستك واطلاعك على الخاصيتين السابقتين، فإنك ستسأل، ماذا لو أردت أن ينفذ التنسيق في حال وجود القيمة المطلوبة في أي جزئية (بداية - منتصف - نهاية) القيمة `^_*`؟ هل هذا ممكن؟ الإجابة هي نعم `^_*`، هذا ممكن من خلال استخدام هذه الخاصية، كمثال لو قمنا باستخدام `div[class*=anees]` في المثال السابق، فإنه سيقوم بتطبيق التنسيق لجميع ال `div` التي تحتوي كلمة `anees` وفي أي موقع `*_*`

لنشاهد المثال معا:

```
<div class="ex-6">
  <div class="anees">
    This is div end at anees c
  </div>
  *****
  <div class="aneesClass">
    This is div start at anees
  </div>
  *****
  <div class="Classanees">
    This Div used (Classanees)
  </div>
  *****
  <p class="anees">This is p end
  *****
  <div class="Class-anees-Class">
    This Div used (Class-anees
  </div>
```

```
/*Example 6*/
.ex-6 div[class*='anees'] {
  background: orange;
  padding: 20px;
  margin: 10px;
  color: #fff;
  font-size: 25px;
  width: 50%;
}
```

This is div end at anees class (anees) is used

This is div start at anees class (aneesClass) is used

This Div used (Classanees)

This is p end at anees class (anees) is used

This Div used (Class-anees-Class)

# SELECTORS

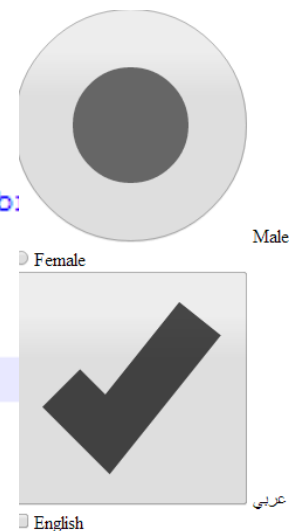
(5) **checked:**

هذه الخاصية تقوم على اعطاء التنسيق لوسائل الإدخال (radio, checkbox) بالإضافة الى option .. لكن هذه الخاصية لا تقوم على تطبيق جميع التنسيقات .. فمثلا لا يعطي color .. لكن يأخذ ال width وال height والآن، لنشاهد مثلا معا ^\* .

```
<body class="ex-7">
  <form action="#">
    <input type="radio" checked="checked" value="male" name="gender"> Male<br>
    <input type="radio" value="female" name="gender"> Female<br>
    <input type="checkbox" checked="checked" value="عربي" name="language"> عربي<br>
    <input type="checkbox" value="English" name="language"> English
  </form>
```

/\*Example 7\*/

```
.ex-7 input:checked{
  width: 200px;
  height: 200px;
  color: red; /* Will not be execute */
}
```



# SELECTORS

(6) `:disabled` و `:enabled`

هذه الخصائص تقوم على إعطاء تنسيق معين إما للعناصر التي في حالة disabled أو التي في حالة enable .. لنشاهد مثالا معا:

```
<form action="#">
  <input type="text" value="Enabled" /><br />
  <input type="text" value="Disabled" disabled="disabled" /><br />
  <input type="checkbox" value="Enabled" />Enabled<br />
  <input type="checkbox" value="Disabled" disabled="disabled"/>disabled
```

```
/*Example 8*/
```

```
.ex-8 input:disabled{
  color: red;
  background-color: black;
  margin: 10px 70px;
}
```

```
.ex-8 input:enabled{
  width: 70px;
  height: 70px;
  color: blue;
  background: orange;
}
```

Enabled

Disabled

Enabled

disabled

# SELECTORS

:empty (7)

هذه الخاصية تقوم على إعطاء تنسيق لجميع العناصر التي لا تحتوي نص بداخلها (باختصار tag فارغة من أي النص)

Anees Hikmat



2nees.com



aneeshikmat@2nees.com



Senior Front-End Engineering

لنشاهد المثال معا ^\_^

```
<body class="ex-9">
  <p>Anees Hikmat</p>
  <p></p>
  <p>2nees.com</p>
  <p></p>
  <p>aneeshikmat@2nees.com</p>
  <p></p>
  <p>Senior Front-End Engineering</p>
</body>
```

/\*Example 9\*/

```
.ex-9 p:empty {
  background-color: orange;
  width: 200px;
  height: 20px;
  margin: 10px 0;
}
```



# SELECTORS

:first-of-type (8)

وظيفة هذه الخاصية هي إعطاء تنسيق معين لأول عنصر من نوع معين  
بنائاً على ال parent الخاص به .. لنشاهد مثالا `*_^`:

```
<body class="ex-10">
  <div>Im Div Man ^^</div>
  <p>Anees Hikmat</p>
  <div>Im Div Man ^^</div>
  <p>2nees.com</p>
  <div>
    Im Div Man ^^
    <p>aneeshikmat@2nees.com 1</p>
    <p>aneeshikmat@2nees.com 2</p>
  </div>
  <p>Senior Front-End Engineering</p>
</body>
```

```
/*Example 10*/
.ex-10 p:first-of-type{
  background-color: black;
  color: orange;
  margin: 15px;
}
```

Im Div Man ^^

Anees Hikmat

Im Div Man ^^

2nees.com

Im Div Man ^^

aneeshikmat@2nees.com 1

aneeshikmat@2nees.com 2

Senior Front-End Engineering



اللهم اني اعوذ بك من الهم والحزن واعوذ بك من العجز والكسل واعوذ بك من  
الجبن والبخل واعوذ بك من غلبة الدين وقهر الرجال

دعاء

# SELECTORS

:last-child (9)

هذه الخاصية الرائعة .. قصة حب لا تنتهي مع أي موقع تحتاج تنسيقه، فهي تقوم على إعطائك ميزة رائعة في التحكم بتنسيق آخر عنصر ضمن parent معين، دون الحاجة الى سكربت، وبكل بساطة فقط `last-child`، وكثيرا ما نستخدمها مع القوائم `^_^` .. لنشاهد مثلا `^_^`

```
.ex-11 #menu-block-using-last-child ul li:last-child:hover{  
  background-color: red;  
}
```

```
.ex-11 #menu-block-using-last-child ul li:last-child{  
  border-right: 0;  
}
```

Menu Without Using Last Child



Menu Using Last Child



# SELECTORS

:last-of-type (10

تستخدم هذه الخاصية لتنسيق آخر عنصر ضمن Parent معين، هذه الخاصية هي الخاصية المقابلة لخاصية first-of-type التي ذكرناها سابقا

\*\_^ ... \*\_^



```
<ul>
  <li id="item-1">
    <a href="2nees.com" class="sf-depth-1 active">Home Page</a>
  </li>
  <li id="item-2">
    <a href="2nees.com" class="sf-depth-1 active">About Us</a>
    <ul>
      <li id="item-2-1">
        <a href="2nees.com" class="sf-depth-1 active">About first</a>
      </li>
      <li id="item-2-2">
        <a href="2nees.com" class="sf-depth-1 active">About Last</a>
      </li>
    </ul>
  </li>
  <li id="item-3">
    <a href="2nees.com" class="sf-depth-1 active">2nees.com</a>
  </li>
  <li id="item-4">
    <a href="2nees.com" class="sf-depth-1 active">Contact Us</a>
  </li>
</ul>
```

```
/*Example 12*/
].ex-12 ul li:last-of-type{
  border: 5px dotted red;
  margin: 20px;
-}
```

- [About Us](#)
- [About first](#)

- [About Last](#)

- [2nees.com](#)

- [Contact Us](#)

# SELECTORS

(11) `:not(selector)` هذه الخاصية الرائعة والمميزة، والتي اضيفت الى عالم التنسيق باستخدام ال css3 تقوم على إعطاء تنسيق معين لجميع العناصر باستثناء ما تم وضعه داخل ال `not ^_*` لنشاهد مثالا `*_*`

```
.ex-13 :not(p) {  
  color: blue;  
}
```

```
.ex-13 p:not(:last-child) {  
  color: green;  
}
```

```
<body class="ex-13">  
  <p>Anees Hikmat</p>  
  <h1>2nees.com</h1>  
  <div>Anees Hikmat</div>  
  <p>Anees Hikmat</p>  
</body>
```

لاحظ النتائج والطريقة التي كتبت بها  
not بالمثال .. وانظر النتائج ..  
فكما تلاحظ .. الخصائص التي تكلمنا  
عنها أو سنتلکم عنها .. يمكن استخدامها  
مع `^_*` not

Anees Hikmat

2nees.com

Anees Hikmat

Anees Hikmat

# SELECTORS

(12) **nth-child(n)**: هذه الخاصية الرائعة والمهمة في تنسيق الصفحات، تمكننا من خلالها تحديد child معين لإعطائه التنسيق.. مثلا nth-child(2): أي قم باعطاء التنسيق لل child الثاني .. \*\_\* .. وبهذا يمكنك التحكم في عناصر معينة بنائاً على قواعد يمكنك أن تحديدها ..

ال n يرمز الى القيمة التي سنحدد من خلالها العنصر الذي سيأخذ التنسيق، والأجمل من ذلك .. هو إمكانية وضع معادلة مثل  $3n + 3$  فهذه تعني  $3 * 0 + 3$  يساوي 3، ثم  $3 * 1 + 3 = 6$  فهذا يعني أنه كلما ازدادت أو تضاعف عدد العناصر .. يمكنك استخدامها كمعادلة تعطي التنسيق لمضاعفات ال  $3^3$  .. إنشاهد مثالا عمليا على ذلك  $^*_$

يمكنك استخدام عمليات الجميع والطرح.. كما يمكنك استخدام كلمة odd أو even لإعطاء التنسيق لصفوف فردية أو زوجية  $^*_$ ...

أو استخدام عدد صحيح مباشرة  $^*_$  .. والخاصية المقابلة لهذه الخاصية هي nth-last-child(n):... والآن لنشاهد أمثلة على ذلك  $^*_$

# SELECTORS

```
<div id="ex-14-3" class="ex-14-child">
  <div>Anees 1</div>
  <div>Anees 2</div>
  <div>Anees 3</div>
  <div>Anees 4</div>
  <div>Anees 5</div>
  <div>Anees 6</div>
  <div>Anees 7</div>
  <div>Anees 8</div>
  <div>Anees 9</div>
</div>
```

```
.ex-14 #ex-14-3 div:nth-child(4n) {
  background: red;
  padding: 10px;
}
```

Anees 1

Anees 2

Anees 3

Anees 4

Anees 5

Anees 6

Anees 7

Anees 8

Anees 9

كما تلاحظ فإن التنسيق تم تفعيله على  
العنصر 4 والثامن لأن المعادلة هي  $4n$   
وهذا يعني  $4 * 1$  و  $4 * 2$  ..  
أما  $4 * 3$  غير موجودة (12) لذلك ينتهي  
التنسيق عند Anees 8

# SELECTORS

(12) **only-of-type**: تستخدم هذه الخاصية لإعطاء تنسيق معين لعنصر يكون هو الوحيد الذي جاء ضمن ال parents من نوعه، على سبيل المثال، لو كان هذا العنصر هو ال `<h1>` فإنه ليأخذ التنسيق يجب أن يكون داخل ال `div` (على فرض أنه `parent`) فقط `<h1>` لمرة واحدة، ويمكن وجود أي `tag` آخر فهذه ليست مشكلة لهذه ال الخاصية `^_*` .. وبنفس الفكرة تتواجد خاصية **only-child** لكن هذه الخاصية تختلف عن السابقة بأنها يجب أن تكون الإبن الوحيد داخل ال `parent` ولا يوجد أي `tag` آخر أيضا معها `^_*`

والآن لنشاهد أمثلة متنوعة على الخاصيتين السابقتين `^_*`  
ملاحظة مهمة: لا تنسى تطبيق الأمثلة، وتجربة احتمالات متنوعة، كما أرجو أن تنتبه الى الشيفرة البرمجية الخاصة بكل مثال ..

# SELECTORS

تم تنفيذ التنسيق

**Anees Hikmat**

```
/*Example 15*/  
.ex-15 .ex-15-child {  
  margin: 20px;  
  padding: 20px;  
  border: 5px solid green;  
}
```

```
.ex-15 .ex-15-child h1:only-of-type {  
  color: orange;  
  font-size: 30px;  
}
```

```
.ex-15 .ex-15-child h1:only-child {  
  background: black;  
  padding: 50px;  
}
```

```
iy class="ex-15">  
<div class="ex-15-child">  
  <h1>Anees Hikmat</h1>  
</div>
```

```
<div class="ex-15-child">  
  <h1>Anees Hikmat</h1>  
  <p>Anees Hikmat</p>  
  <p>Anees Hikmat</p>  
</div>
```

```
<div class="ex-15-child">  
  <h1>Anees Hikmat</h1>  
  <h1>Anees Hikmat</h1>  
</div>
```

```
</body>
```

**Anees Hikmat**

Anees Hikmat  
Anees Hikmat

تم تنفيذ تنسيق واحد

لم يتم تنفيذ أي تنسيق

**Anees Hikmat**  
**Anees Hikmat**





# SELECTORS

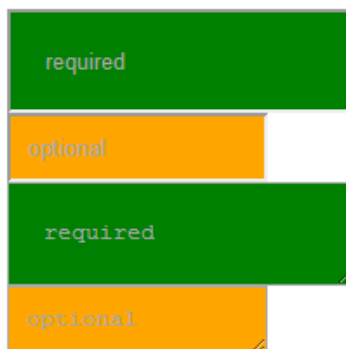
## :optional & :required (13)

خاصيتان من الإسم لوحده `^_^`، نستنتج وظيفة كل منها `^_^`، نعم، فوظيفتهما هي اعطاء تنسيق للحقل الإلزامية (required) وغير الإلزامية (optional).. لنشاهد مثالا `*_^`

```
<form>
```

```
<input type="text" placeholder="required" required /> <br />
<input type="text" placeholder="optional" /><br />
<textarea required placeholder="required"></textarea><br />
<textarea placeholder="optional"></textarea><b: /*Example 16*/
```

```
</form>
```



```
.ex-16 :required {
  padding: 20px;
  background: green;
  width: 200px;
  height: 60px;
}

.ex-16 :optional {
  padding: 10px;
  background: orange;
  width: 150px;
  height: 40px;
}
```

# SELECTORS

(14) :target

الآن، آخر خاصية سنتحدث عنها بإذن الله تعالى هي هذه الخاصية، وهي تقوم على اعطاء تنسيق معين لعنصر ما، هذا العنصر يجب أن يكون target، ولي يتم ذلك يجب أن تتوافر شروط، على سبيل المثال ..وجود عنصر له id ورابط يكون ال href يحتوي قيمة ال id للعنصر ... وبهذا يكون العنصر هو target لهذا ال anchor ..

لنشاهد مثالا معا ^ \_\_\_\_ ^ (عند تطبيق المثال انتبه لل url)



Go To P1

Go To P2

Go To P3

Go To P4

Anees Hikmat Anees Abu-hmaid

2nees.com

aneeshikmat@2nees.com

^ \_\_\_\_ ^

```
<a href="#p1">Go To P1</a>
<a href="#p2">Go To P2</a>
<a href="#p3">Go To P3</a>
<a href="#p4">Go To P4</a>
```

```
<p id="p1">Anees Hikmat Anees Abu-hmaid</p>
<p id="p2">2nees.com</p>
<p id="p3">aneeshikmat@2nees.com</p>
<p id="p4">^ ____ ^</p>
```

```
.ex-17 a{
  display: inline-block;
  text-decoration: none;
  font-size: 18px;
  margin: 20px;
  color: red;
}

.ex-17 :target {
  background: green;
  padding: 20px;
  font-size: 25px;
}
```

## أعوذ بالله من الشيطان الرجيم

لَا يُكَلِّفُ اللَّهُ نَفْسًا إِلَّا وُسْعَهَا لَهَا مَا كَسَبَتْ وَعَلَيْهَا مَا اكْتَسَبَتْ رَبَّنَا لَا تُؤَاخِذْنَا إِنْ  
نَسِينَا أَوْ أَخْطَأْنَا رَبَّنَا وَلَا تَحْمِلْ عَلَيْنَا إَصْرًا كَمَا حَمَلْتَهُ عَلَى الَّذِينَ مِنْ قَبْلِنَا رَبَّنَا وَلَا  
تَحْمِلْنَا مَا لَا طَاقَةَ لَنَا بِهِ وَاعْفُ عَنَّا وَارْحَمْنَا أَنْتَ مَوْلَانَا فَانصُرْنَا عَلَى  
الْقَوْمِ الْكَافِرِينَ

صدق الله العظيم

[البقرة/286]

# Web Fonts

في هذا الموضوع سنتحدث عن أمر رائع جدا، ومهم في عالم تطوير وتصميم المواقع ^\_\*، نعم سنتحدث عن الخطوط ^\_^...

إن ما يقوم به المصممون، هو إعطائك تصميم بأشكال والأوان معينة، بالإضافة الى خط تم استخدامه ليعطي التصميم رونقه، ويكون هذا الأمر عادة على شكل PSD File ^\_^.. وبهذا يمكنك تحديد وتفصيل واقتطاع الأجزاء التي تريد.. وأهمها بالنسبة لموضوعنا الآن.. هو الخط ^\_\*

كان في السابق يجب أن يتوفر الخط لدى جهاز العميل حتى يعمل بشكل صحيح، لكن هذا الكلام أصبح من الماضي بوجود ال css3، فمن خلالها تم تصميم module وظيفته السماح للمتصفح باستخدام خط معين دون أن يكون موجودا على جهاز العميل، وكل ما يلزمنا هو وضع ملفات الخط على server ومن ثم تضمينها في صفحات التنسيق لدينا ^\_^

# Web Fonts

الآن والسؤال الأهم ^\_^، ما هي هذه الخاصية؟

الجواب هو : **@font-face Rule**

وال rules هنا هي الخط الذي نريد استخدامه، الإمتدادات المتاحة له، والمتصفحات التي تدعم هذا الخط..بالإضافة الى موقع الملف على السيرفر  
^ ^ ... شاهد هذا المثال ^\*:

```
@font-face {  
❶ font-family: name;  
❷ src: ❸local('fontname'), ❹url('/path/filename.otf') ❺format('opentype');  
}
```

1. هنا يتم وضع اسم الخط، هذا الإسم الذي ستقوم بوضعه، هو الذي ستقوم باستخدامه لإعطاء تنسيق الخط لأي عنصر أثناء عملك في الموقع، بالإضافة الى ذلك يمكنك كتابة هذا الإسم من مقطع واحد أو من أكثر من مقطع اذا أردت ذلك، ولكن يجب عليك أن تقوم بوضع الإسم داخل “ “

# Web Fonts

```
font-family: anees;
```

```
font-family: "anees hikmat";
```

مثال بسيط على الموضوع:

(2) هذه الخاصية هي التي تخبر المتصفح عن موقع وجود الخط على السيرفر، وهي تحتوي على قيم مختلفة مثل ومتعددة مثل 3 و 4 و 5.

(3) Local وتقوم هذه المزية على أخذ الإسم الأصلي Source Name للخط، ومن ثم قحص جهاز المستخدم هل يمتلك هذا الخط أم لا \*\_^.

(4) إذا لم يكن موجودا الخط لدى المستخدم يأتي دور ال url ليقوم بتحميل الخط على المتصفح من خلال ال server (أي هنا يتم وضع مكان وجود الخط على السيرفر).

(5) Foramt: وهنا يتم تحديد ال format الخاص بالخط... وهناك العديد من ال formats مثل ال open type وال svg وغيرها...

# Web Fonts

لنشاهد مثالا على ما ذكرنا سابقا ^\_^

ملاحظات:

(1) يمكنك البحث واستخدام الخطوط عن طريق google font ..ستجد العديد من الخطوط وبإمكانك تضمينها واستخدامها بكل سهولة.

(2) اذا أردت تحميل الخط ..فيمكنك ذلك، وهذا عن طريق نفس الموقع أو أي موقع آخر (انتبه لحقوق الملكية).

(3) صيغة الخط الافتراضية عادة تكون ttf ونحن نقوم بتحويل هذه الصيغة الى webfont وذلك عن طريق مواقع الإنترنت ..قم بالبحث عن طريق جوجل عن font to webfont \*^\_\*

```
@font-face {  
  ❶ font-family: Chunk;  
  src: ❷local('ChunkFive'), ❸url('ChunkFive.ttf') ❹format('truetype');  
}  
❺ h2.webfont { font-family: ChunkFive, sans-serif; }
```

طريقة الإستخدام ←

# Web Fonts

طبعا الى الآن، لم أقم بتنفيذ أي مثال وربط نتائج، وذلك حتى يتسنى لك فهم جزئيات هذه الخاصية .. أما الشيفرة النهائية .. فستكون بالنسبة لنا عند العمل مجرد نسخ ولصق ^\_^ .. لكن أرديك أن تدرك ما هو الشيء الذي تقوم بنسخه ولصقه .. لذلك دعنا نكمل بالتدرج حتى نصل الى الصيغة التي سنعمل عليها (في العادة سنأخذها من موقع تحويل الخطوط جاهزة (-\_-

الآن سأحدث عن نقطة مهمة عما ذكر سابقا وهو Defining Different Faces

ولتعرف عن ماذا أتكلم .. أرديك أن تقوم بتحميل هذا الملف أولا ^\_\*

أدخل الى هذا <http://www.google.com/fonts#UsePlace:use/Collection:Open+Sans>

الموقع .. (google font) وقم بتحميل الخط ^\_^ (شاهد الصورة في الشريحة التالية)



# Web Fonts

## Almost done!

Verify your settings below and then copy the code for your website.


### 1. Choose the styles you want:




- Open Sans
  - Light 300
  - Light 300 Italic
  - Normal 400

### Collection (1 font family)

Open Sans Normal 400

### Download fonts

 You do not need to download the font to use it on your webpages. Instead, refer to the "Use" section.

-  Download the font families in your Collection as a [.zip file](#)
-  Sync Google fonts to your desktop using [SkyFonts](#)
-  Download all font families, including source files, at the [Google Code Project](#)

Close



# Web Fonts

الآن .. بعد القيام بتحميل الخط ..ستجد هذه الملفات:  
إذا قمت بإمعان النظر قليلا ..فستجد أن هذه الملفات بأسمائها كأنها تمثل  
font-weight ^\_^...نعم وهذا المقصد .. اذا ما هي المشكلة التي قد تظهر هنا ؟؟

هو أنني لكي أحصل على قيم متنوعة

للخط، فإنه يجب أن أقوم بتحويل جميع

الخطوط ...الى webfont ومن ثم استخدام

ال font-face لكل قيمة وكأنها خط على حدى،

وسأقوم بتعريف اسم لكل قيمة مثل

Font-family: open1 و open2 و open3 وهكذا

وهذه الطريقة صحيحة ..لكنها متعبة ..

بالإضافة الى ذلك، فيلزمني تحديد هذه القيم

عند تعريف الخط .. لذلك ظهر مفهوم ال

Defining Different Faces

OpenSans-Bold.ttf

OpenSans-BoldItalic.ttf

OpenSans-ExtraBold.ttf

OpenSans-ExtraBoldItalic.ttf

OpenSans-Italic.ttf

OpenSans-Light.ttf

OpenSans-LightItalic.ttf

OpenSans-Regular.ttf

OpenSans-Semibold.ttf

OpenSans-SemiboldItalic.ttf

# Web Fonts

والآن !\_!.. الأمور ما زالت غامضة ..نعم ..لكن لنشاهد مثالا لكي نوضح الشكل الذي وصلنا اليه^\_^

```
@font-face {  
  font-family: 'Anees Sans';  
  font-weight: bold;  
  src: local('Anees Sans'), url('opensans-bold-webfont.woff') format('opentype');  
}  
@font-face {  
  font-family: 'Anees Sans';  
  font-style: italic;  
  src: local('Anees Sans'), url('pensans-italic-webfont.woff') format('opentype');  
}
```

انتبه لكل جزئية متوفرة في الصورة..

لاحظ أنني قمت بتسمية الخط تسمية متشابهة لكل من التضمينات .. لكن الأول الهدف منه هو تضمين قيمة ال bold والثاني تضمين قيمة ال italic .. لذلك استخدمنا ال font-weight وال font-style .. أما ال url فكان يأخذني لموقع الملف الخاص بكل قيمة ^\_\*

# Web Fonts

الآن الجميل في الموضوع، هو أنك بمجرد استخدام نفس الإسم وبإعطائه قيم متنوعة .. كل ما عليك القيام به هو العمل بالطريقة التقليدية .. شاهد المثال:

```
h2 { font-family: 'Anees Sans'; font-weight: bold; font-style: italic;}  
h3 { font-family: 'Anees Sans'; font-weight: bold; }
```

هل فهمت ما أرمي اليه .. يمكنني استخدام الخصائص أو عدم استخدامها ..  
وعن طريق الإسم والقيم سيحدد الخط/ الخطوط التي سيتم جلبها أو تنفيذ  
تنسيقها ^\_^

والآن .. ماذا ^\_\* .. الآن لنعود لنلقي نظرة بسيطة عن نقطة شرحناها سابقا ..

# Web Fonts

لقد تحدثنا عن ال local في الشرائح السابقة، وقلنا أنها تفحص هل يتوفر الخط بجهاز العميل أم لا .. لكن هذه الخاصية لها سيئات عدة نذكر منها:

(1) هذه الميزة لا تدعم أي من إصدارات ال IE الأقل من 9 - -

(2) تأخذ وقت بالتنفيذ، وخصوصا لمتصفح ال safari على mac os .. والسبب

في ذلك يعود الى أن ال mac يجب أن postScript name بينما المتصفحات الأخرى تأخذ ال Full name .. هذا الكلام ماذا يعني؟! هذا يعني أنه يجب أن نقوم بتعريف local مرتين - - .. وهذه المشكلة ثانية .. مثال:

```
@font-face {  
  font-family: MyHelvetica;  
  src: local("Helvetica Neue Bold"),  
       local("HelveticaNeue-Bold"),  
       url(MgOpenModernaBold.ttf);  
  font-weight: bold;  
}
```

وهناك غيرها أيضا .. لكن نكتفي بهذا .. ^ \_ \*

والآن ماذا نفعل؟

بكل بساطة فإننا لا نستخدم ال local عادةً ^ \_ ^

ومن ذلك لا يكون البيانُ العالي أتمَّ إشراقاً إلا بتمام النفسِ البليغةِ في فضيلتها أو رذيلتها على السواء؛ فإنَّ من عجائبِ السخريةِ بهذا الإنسانِ أن يكونَ تمامُ الرذيلةِ في أثره على العملِ الفنيِّ، هو الوجهَ الآخرَ لتمامِ الفضيلةِ في أثره على هذا العملِ؛ والنقطةُ التي ينتهي فيها العلوُّ من مُحيطِ الدائرةِ هي بعينها التي يبدأ منها الانحدارُ إلى السُّفْل؛ ومن ثمَّ كانتِ الفنونُ لا تُعتبرُ بالأخلاقِ، حتى قالَ علماؤُنا: إنَّ الدينَ عن أشعرِ بمَغزِل. فالأصلُ هناك سموُّ التعبيرِ وجماله، وبلاغةُ الأداءِ ورُوَعته؛ ولا يكونُ السؤالُ الفنيُّ ما هي قيمةُ هذه النفسِ، ولكن ما طريقتُها الفنية؟ وأيُّ عَجيبٍ في ذلك؟ أليس لجهنمِ حقٌّ في كبارِ أهلِ الفنِّ، كما للجنةِ حقٌّ في نوابغِه؟ وإذا قالتِ الجنةُ: هذه فضائلي البليغة. أفلا تقولُ الجحيمُ: وهذه بلاغةُ رذائلي؟ وكيف لعمري يستطيعُ إبليسُ أن يؤديَ عملهَ الفنيِّ . . . . ويصوِّرَ بلاغتهُ العاليةَ إلا في ساقطينَ من أهلِ الفكرِ الجميلِ، وساقطاتٍ من أهلِ الجسمِ الجميلِ . . ؟

## كتاب وحي القلم (الجزء الأول)

# Web Fonts

بعد حديثنا الشيق عن موضوع ال local ^\_\* .. يجب أن ننتقل الى جزئية أخرى مهمة، وهي الإمتدادات الخاصة بكل خط؟!، نعم، فلا تتعجب، فلا يوجد تم تصميمه وإضافته عشوائيا بدون هدف ^\_\*

والآن لنبدأ:

(1) WOFF / WOFF2: هذا ال format اختصار ل Web Open Form Format، تم تصميم هذا النوع من قبل Mozilla ^\_^، بالتعاون مع مع منظمات أخرى ^\_\*، ويعد هذا النوع من أسرع الأنواع في التحميل، والسبب في ذلك يعود الى أن هذا النوع هو ملف مضغوط يجمع بين نوعين وهما TTF وال OTF ..، أيضا مما يميز هذا النوع هو إمكانية اضافة meta data و lincies داخل نفس ال font -\_- ... وهذا النوع هو الي تسير الشركات لجعله مدعوما في المتصفحات...

# Web Fonts

(2) OTF / TTF: هذان نوعان من الإمتداد وهما اختصار ل OpenType Font and TrueType Font. هذان النوعان هما النوعان الشائعان، وبسببهما تم تصميم النوع السابق (WOFF)، والسبب في ذلك يعود الى سهولة نسخ هذه الأنواع واستخدامها بشكل غير قانوني \*\_\*، بما يتميز به نوع ال open type هو دعمه لجميع الأنظمة cross-platform، أما بخصوص ال ttf فهو نوع صمم بواسطة apple & Microsoft في أواخر الثمانيات.. وكان ما يميزه هو التحكم بطريقة عرض الخط بنائاً على البكسل ^\_\*... ولذلك تجد أن أغلب الخطوط التي نقوم بتحويلها يكون من ttf، وهذا لا يعني أن الأنواع الأخرى لا تملك هذه المزية... فالخاصية السابقة جمعت بين هذا النوعين في نوع واحد.. هو التقدم التقني ^\_^... بهذا نكون ذكرنا فكرة كل امتداد من هذه من نبذة صغيرة ^\_^..



## Web Fonts

(3) EOT: هذا النوع هو اختصار ل Embedded Open Type و صمم عن طريق مايكروسفت، ويعتبرو هم المبتكرين الأصليين لموضوع ال font-face، وهذا النوع هو الوحيد الذي يدعمه متصفحات ال IE8 وأقل .. لذلك تجد أننا سنستخدم ال EOT عند تضمين الخطوط ..(سنرى الصيغة القياسية بعد الإنتهاء من النقطة 4 والأخيرة بإذن الله تعالى).

(4) SVG / SVGZ: هذا النوع هو اختصار ل Scalable Vector Graphics وهو يعتمد على ال vector لرسم أو تمثيل الخط، كما أن هذا النوع يتميز بالحجم الخفيف وهذا يعني سهولة التحميل ^\_^، لكن المشكلة أنه ال IE وال IE mobile وال Firefox غير داعم له، ويستخدم الصيغة الأولى لهم .. بالإضافة الى ذلك فإن هذا النوع تم دعمه من الإصدار 4.1 وأقل ..

# Web Fonts

والآن، بعد الحديث السابق، أصبحنا نعي بفضل الله تعالى، ماذا نعني بهذه الأنواع من الخطوط، وبهذا سيسهل علينا الآن استخدام الصيغة بإذن الله تعالى.. لنشاهد معا مثالا على طريقة تضمين الخطوط بحيث تكون مدعومة من المتصفحات الرئيسية جميعها ^\_^:

```
/*Example 18*/
```

```
@font-face {
```

```
font-family: 'MyWebFont';
```

```
src: url('webfont.eot'); /* IE9 Compat Modes */
```

```
src: url('webfont.eot?#iefix') format('embedded-opentype'), /* IE6-IE8 */
```

```
url('webfont.woff2') format('woff2'), /* Super Modern Browsers */
```

```
url('webfont.woff') format('woff'), /* Pretty Modern Browsers */
```

```
url('webfont.ttf') format('truetype'), /* Safari, Android, iOS */
```

```
url('webfont.svg#svgFontName') format('svg'); /* Legacy iOS */
```

```
}
```

# Web Fonts






ملاحظة: الصيغة السابقة تسمى Fontspring Bulletproof Syntax وهناك صيغة أخرى أيضا يمكن أن تراها وهي أيضا داعمة لجميع المتصفحات الرئيسية وهي Bulletproof Syntax وهي تشبه هذا المثال:

```
@font-face {  
  font-family: 'webfont';  
  src: url('webfont.eot');  
  src: local('@'),  
       url('webfont.woff') format('woff'),  
       url('webfont.ttf') format('truetype'),  
       url('webfont.svg#webfont') format('svg');  
- }
```

هذه الصيغة شبيهة بالصيغة السابقة، لكن يعتمد أولا على جعل قيمة ال local غير منفذة من الأجهزة (الخط غير موجود) ويتم استدعائه غالبا من الخارج external ^\_\*  
.. فإذا رأيت أي من الشكلين .. فالوضع تمام إن شاء الله ^\_^

# Web Fonts

والآن لنشاهد المتصفحات التي تدعم @font-face: (الصورة من موقع w3schools في تاريخ 2015-2-9)

Font format					
TTF/OTF fonts	9.0*	4.0	3.5	3.1	10.0
WOFF fonts	9.0	5.0	3.6	5.1	11.1
SVG fonts	Not supported	4.0	Not supported	3.2	9.0
EOT fonts	6.0	Not supported	Not supported	Not supported	Not supported

وأخر دعوانا أن الحمد لله رب العالمين ^\_^... والآن بهذا نكون انتهينا من الخطوط بإذن الله تعالى.. مع ملاحظة أنه يوجد العديد من السمات والخصائص لم يتم الحديث عنها بإمكانك البحث عنها.. قم بالبحث عن “More Font Properties” على سبيل المثال ^\_^\*.. لكن ما يهمني ما تم ذكره ^\_^

# Web Fonts

ملاحظة: بالنسبة لموقع جيد يقوم بتحويل ال font الى webfont موقع

<http://www.fontsquirrel.com/tools/webfont-generator>

إذا قمت بالبحث كما أخبرتك بالشرائح الأولى ستجده ضمن القائمة الأولى في بحث جوجل ^\_\*

+ Add Fonts

1

Open Sans Regular

TTF

938 glyphs

212 KB



BASIC

Straight conversion with minimal processing.

OPTIMAL

Recommended settings for performance and speed.

EXPERT...

You decide how best to optimize your fonts.

Agreement:

2

Yes, the fonts I'm uploading are legally eligible for web embedding.

Font Squirrel offers this service in good faith. Please honor the EULAs of your fonts.

3

DOWNLOAD YOUR KIT

كان حسن الخلق مطلباً نبويًا يدعو ربه أن يجمله به ويديمه عليه  
فكان من دعائه عليه الصلاة والسلام :  
" اللهم كما حسنت خلقي فحسن خلقي " (أحمد وصححه ابن حبان) ،

وكان يقول: " واهدني لأحسن الأخلاق لا يهدي لأحسنها إلا أنت  
واصرف عني سيئها لا يصرف عني سيئها إلا أنت " (رواه مسلم).

اللهم صل وسلم على الحبيب المصطفى عليه أفضل الصلاة والسلام

# Text Effects and Typographic Styles

هل تحب الحلوى؟

إذا كنت من محبين الحلوى فتابع معنا المثال، وإلا فيجب عليك حبها حتى  
نكمل التشبيه على قطع الحلوى هههه .. ^\_^

بالنسبة لمواقع الإنترنت المتواجدة وبكثرة، فإننا نشاهد العديد من الصفحات  
المتنوعة، لكن الصفحات التي تجذبك إليها، هي الصفحات التي تقدم خدمة جيدة  
بأفضل أسلوب تصميم، وأوضح طريقة عرض، وبهذا فإن الجمال والحركات  
والتنسيقات الرائعة للموقع ككل، وللمحتوى خصوصا، مهم جدا، وتعتبر هي  
الحلوى الخاصة بأي مشروع والمقدمة منك للزائر ^\_^ ..  
وبالتأكيد هذا الأمر يتعدى قطع الحلوى، بل ويصنف في عالم تصميم المواقع  
على أنه العامود الفقري لأي موقع ولأي مشروع ...

# Text Effects and Typographic Styles

والآن، وأول ما سأبدء به، أن نسترجع معا فكرة الإحداثيات والمحاور التي تعلمناها في دروس الرياضيات سابقا ..

أول ما أحب إخبارك به، هو أن ال css تقوم على استخدام مبدأ أو نظام الإحداثيات الخاص بديكارت، وبهذا وبمعرفة نظام ديكارت، فإنه يتوفر لدي محورين، المحور السيني -الأفقي- (x-axis) والمحور الصادي -العامودي- (y-axis) .. هذان المحوران، يتقاطعان في نقطة المحور والتي

تسمى ب (origin) .. وليتضح الشكل الذي

أصبح في أذهاننا .. لنشاهد الصورة:

(انتبه الى (-) أن يكون الى اعلى الصورة و

(+) يكون الى أسفل الصورة)

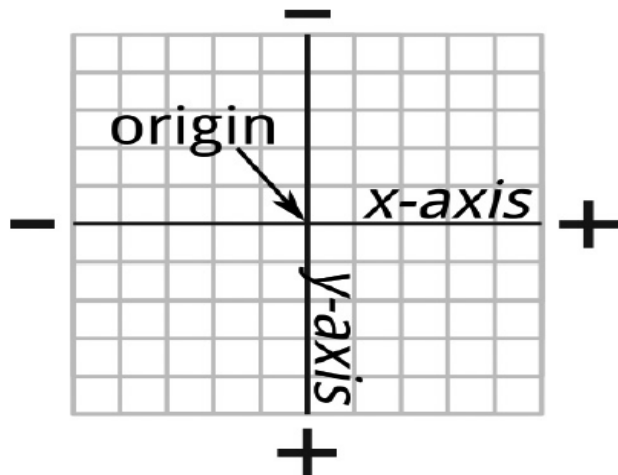


Figure 6-1: The x and y axes and the point of origin



# Text Effects and Typographic Styles

الآن بنائاً على الشكل السابق، فإن كل مربع يظهر، هو على الشاشة يعادل 1 بكسل - تخيل ذلك -

الآن إشارة السالب والموجب الموجودة بالرسم هي التي تتمثل مقدار البعد والقرب عن نقطة المحور (origin)، بنائاً على اتجاه المحور بكل تأكيد،

الآن، وبعد شرح هذه المفاهيم البسيطة لو قمنا

بأخذ الصورة التالية ومن قمنا بتحليلها فما هي

النتائج التي سنخرج بها؟ لنجرب ذلك معاً  $\wedge$   $\_$   $\wedge$

(\* انتبه لقيمة وموقع كل قيمة من القيمتين

والمكتوبة على الصورة  $\_$   $\wedge$  \*)

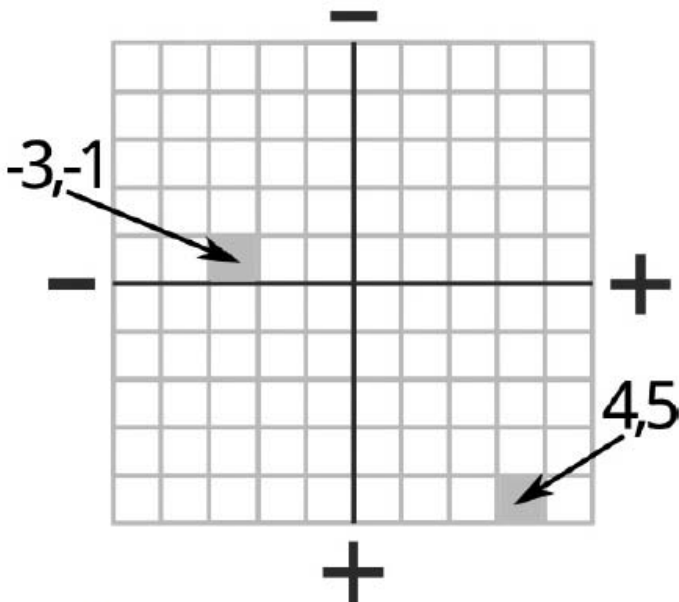


Figure 6-2: Two sets of coordinates

# Text Effects and Typographic Styles

إذا لاحظت القيم السابقة فسنستج التالي:

(1) النقطة 4,5 تعني أننا نبعد بمقدار 4 بكسل عن نقطة المحور أفقياً باتجاه الخط الموجب وال 5 تعني اتجه لأسفل بمقدار 5 بكسل ..

(2) النقطة -1, -3 تعني ابتعد عن نقطة المحور بمقدار 3 بكسل الى اليسار في الصورة، واصعد الى الأعلى بمقدار 1 بكسل..

(3) إذا لاحظت في الإستنتاجات السابقة، فإني تطرقت لذكر كلمات مثل ابتعد، اتجاه، أسفل، اصعد .. الخ، هذا الكلام صيغ بهذا الشكل حتى تكون الصورة أقرب اليك، لأنني أريدك أن تتخير أي تأثير مثلاً ال shadow هو عملية تظليل بكسل مع عملية سحب لإحدى الإتجاهات ..

# Text Effects and Typographic Styles

والآن، لنبدأ بتطبيق أول أمر css3 يختص بال text ... وهو ال text-shadow \* \_ ^  
وهذا الموضوع فعليا هة تطبيق عملي وممتاز على ما شرحناه سابقا  
بخصوص المحاور \* \_ ^ ...  
الصيغة العامة:

```
text-shadow: x-axis y-axis blur-radius color|none|initial|inherit;
```

شرح الصيغة العامة: (ال none وال initial وال inherit... اختياريات ووظائفهم معروفة)

(1) x-axis: هي القيمة الإبتعاد عن نقطة المحور بالنسبة للمحور الأفقي (الزامي)

(2) Y-axis: // // // // // // // // العامودي (الزامي)

(3) blur-radius: وهي مقدار الضبابية لل shadow \* \_ ^. اختياري والإفتراضي 0

(4) Color: وهي قيمة اختيارية يمكنك تحديد لون ال shadow من خلالها ..

# Text Effects and Typographic Styles

والآن .. لنشاهد مثالا عمليا عن الموضوع ^\_^:

Anees Hikmat Anees Abu-hmaid  
Anees Hikmat Anees Abu-hmaid

```
.ex-18 #p1 {  
    text-shadow: 20px 15px red;  
}
```

لاحظ أن اللون الأحمر هو اللون الذي إختارته لل shadow .. ودقق النظر بمقدار الإزاحة للون الأحمر.. (تذكر المحور السيني والصادي) .. القيم موجبة .. قتم السحب لليساار بمقدار 20 بكسل و 15 بكسل لإسفل .. لذلك شرحنا الموضوع السابق .. ^\_\* ...

# Text Effects and Typographic Styles

مثال 2: والآن .. لنقوم بتجربة لإستخدام الضبابية ^\_^ ..

2nees.com Website for Web Developer

```
].ex-18 #p2{  
  text-shadow: 5px 16px 7px blue;  
-}
```

القيمة 7 تمثل مقدار الضبابية لل shadow .. وكلما زاد الرقم .. زادت كمية

التلاشي ^\_^

# Text Effects and Typographic Styles

مثال 3: من الخصائص والأشياء الجميلة هي إعطاء أكثر من shadow لنفس النص، وهذا أمر رائع جداً، فمن خلاله يمكننا تصميم أنواع من ال shadow بحيث تغيير الشكل النص بشكل كلي .. ولكي نقوم بهذه الحركة، كل ما علينا فعله هو استخدام الفاصلة بين ال shadow values ...

```
.ex-18 #p3{  
    text-shadow: 257px 0px 3px red,  
                0px 17px 2px blue,  
                -1px -17px 3px green;  
}
```

# Text Effects and Typographic Styles

مثال 4:

Anees Hikmat Like This Smile ^ \_ ^

ما رأيك في النص الموجود في الصورة .. لاحظ كيف يبدو؟ وكأنه محفور داخل الكرتون .. هذا التأثير أو التنسيق عن طريق ال CSS فقط لا غير .. وعن طريق استخدام ال text-shadow ^ \_ ^ لنشاهد الشيفرة البرمجية معا

```
.ex-18 #p4{  
  background-color: #565656;  
  color: #333;  
  text-shadow: 0 1px 0 #777, 0 -1px 0 #000;  
  font-size: 40px;  
}
```

^ \_ ^



”ان خشيتي من الاختلاف والفرقة  
ستظل تقلقني حتى وأنا في القبر  
إن اتحدنا فصوله الدوله تستطيع دفع الأعداء  
فإن لم تتحد الأمة فلا راحة لي“

رحم الله آل عثمان

السلطان العثماني سليم الاول 1470-1520



# Text Effects and Typographic Styles

أيضا، من الخصائص الرائعة في تنسيق النصوص، والتي نخدمنا كمصممين لمواقع الويب، هي **خاصية text-overflow** <sup>^\_^</sup>، هذه الخاصية الجميلة تقوم على إعطائنا إمكانية بالتحكم في النص الزائد عن حجم الإطار الذي يحتويه، ويكون هذا عن طريق عمل Trim للنص .. <sup>^\_^</sup> مثل **ذهب الطالب ل...** لاحظ أن تكملة النص تم استبدالها بنقاط .. هذا الخيار اسمه ellipsis <sup>^\_^</sup> أما الخيار الافتراضي لل text-overflow هو clip

والآن لنشاهد مثلا <sup>^\_^</sup>:

```
text-overflow: clip;  
text-overflow: ellipsis;  
...
```

# Text Effects and Typographic Styles

الآن شاهد هذه النتيجة: (انتبه للتنسيق والنتيجة ثم انتقل للشريحة التالية)

Anees Hikmat A...

2nees.com Website  
for Web Developer

aneeshikmat@2nees.com

Anees Hikmat Like Th

```
.ex-19 #p1{  
  width: 200px;  
  text-overflow: ellipsis;  
  white-space: nowrap;  
  overflow: hidden;  
}
```

```
.ex-19 #p2{  
  width: 200px;  
  text-overflow: ellipsis;  
  overflow: hidden;  
}
```

```
.ex-19 #p3{  
  width: 200px;  
  text-overflow: ellipsis;  
  white-space: nowrap;  
}
```

```
.ex-19 #p4{  
  width: 200px;  
  text-overflow: clip;  
  white-space: nowrap;  
  overflow: hidden;  
}
```

# Text Effects and Typographic Styles

إذا لاحظت في المثال السابق، فإنني قمت باستخدام مجموعة من الخصائص، و إعطاء تنسيقات مختلفة .. لكن الشكل الذي عمل بالشكل الذي نريده .. كان الشكل الأول .. وهذا يعني أنه لتعمل هذه الخاصية يجب علينا تحديد أو إعطاء مجموعة من الخصائص أيضا منها خاصية:

ال overflow وال white-space ... ال overflow معروفة .. وخاصية hidden تقوم على إخفاء أي شيء سيخرج خارج نطاق الحاوي ..

أما ال white-space: فهي من خصائص ال css1 لكن خاصية ال nowrap هذه تعني أن يقوم باعتبار المساحات مساحة واحدة، ويستمر النص على نفس السطر، أي ولا يمكن أن ينزل النص على سطر جديد، الا بوجود br ..

قد يكون من أجمل استخدامات هذه الخاصية عند ال responsive ... اذا كنت أريد التأكد بأن النص لن يخرج خارج الحاوي، وبذات الوقت دون تغيير التنسيق أو تصغير

حجم الخط ^\_^

شاهد تطبيق المثال:



# Text Effects and Typographic Styles

أيضا من الخصائص التي يمكن استخدامها مع النصوص والمضافة في ال css3 هي خاصية ال Resize وهي أيضا من الخصائص التي نستخدمها في حالة كان النص أكبر من حجم الحاوي... (إذا لاحظت في الخاصية السابقة .. كنا نقوم بجعل النص على شكل ...) أما الآن .. فإن هذا الخيار يتيح لنا تغيير حجم الحاوي إما أفقيا أو عاموديا أو كلاهما معا ^\_^ ..

On a certain afternoon, in  
the late springtime, the bell

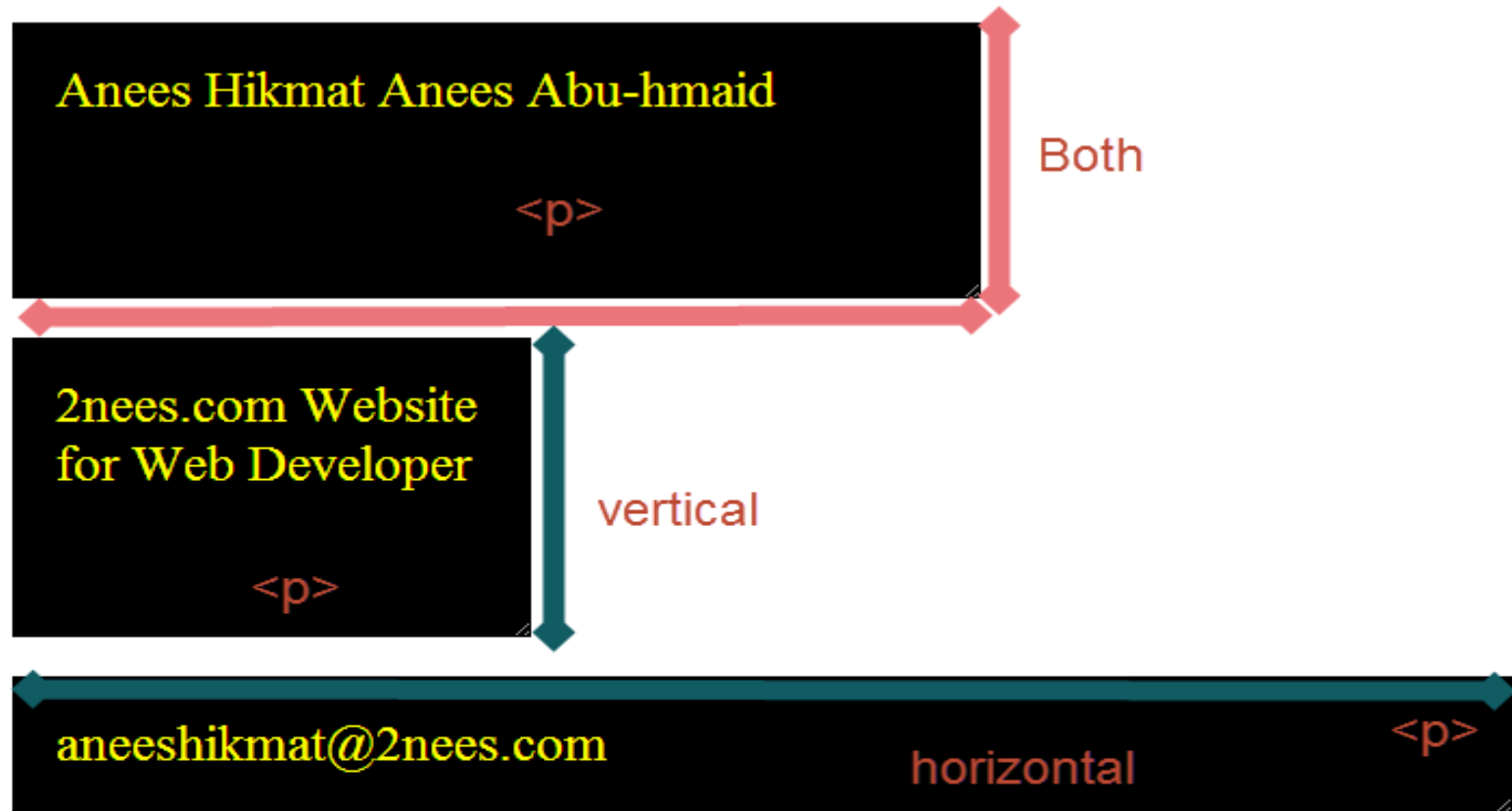
On a certain afternoon, in the  
late springtime, the bell upon  
Tunstall Moat House was heard  
ringing at an unaccustomed

شاهد الصيغة العامة:

```
resize: none|both|horizontal|vertical|inherit;
```

والآن .. لنشاهد مثلا عمليا على هذه الخاصية الجميلة ^\_^

# Text Effects and Typographic Styles



```
Hikmat  
Like This  
Smile
```

TextArea  
Resize: none

# Text Effects and Typographic Styles

```
/*Example 20*/
```

```
.ex-20 p, .ex-20 #txtArea{  
  overflow: auto;  
  background: black;  
  padding: 20px;  
  margin: 20px;  
  font-size: 25px;  
  color: yellow;  
  width: 200px;  
}
```

```
.ex-20 #p1{  
  resize: both;  
}
```

```
.ex-20 #p2{  
  resize: vertical;  
}
```

```
.ex-20 #p3{  
  resize: horizontal;  
}
```

```
.ex-20 #txtArea{  
  resize: none;  
}
```

```
<body class="ex-20">
```

```
<p id="p1">Anees Hikmat Anees Abu-hmaid</p>
```

```
<p id="p2">2nees.com Website for Web Developer</p>
```

```
<p id="p3">aneeshikmat@2nees.com</p>
```

```
<textarea id="txtArea">Anees Hikmat Like This Smile ^ ____ ^</textarea>
```

```
</body>
```



مثال:

تذكر أن تتأكد من توافقية كل خاصية نتحدث عنها  
بنفسك كما تعلمنا \* ^ \_

# Text Effects and Typographic Styles

## خاصية ال word-wrap

هذه الخاصية أيضا من خصائص ال css3 والتي تهتم بالنصوص، هذه الخاصية تقوم بحل مشكلة مزعجة في كثير من الأحيان، وهي في حالة وجود كلمة شكلها مثل هذا `asdasdqwejlkdjflskjdfkjdfljsldfkjlsdkfjcvb,cvn,mndf,we` على سبيل المثال .. هذا الشكل من الكلمة اذا جاء في حاوي حجمه صغير، فسيخرج النص خارج الحاوي.. شاهد الشكل التالي:

The condition of silicosis is sometimes factitiously referred to as Pneumonoultramicroscopicsilicovolcanoconiosis.

الحالة الإفتراضية

والآن لحل هذه المشكلة بطريقة نفصل

بها الكلمة لتنزل على سطر جديد وكأنها كلمة تحتوي على فراغات

سنقوم باستخدام `^_^` word-wrap

# Text Effects and Typographic Styles

الصيغة العامة:

```
word-wrap: normal|break-word;
```

يحدث تكون ال break-word هي القيمة التي تسمح بتجزئة الكلمة .. أما normal فهي الحالة الافتراضية...

```
/*Example 21*/  
.ex-21 p{  
  background: black;  
  padding: 20px;  
  margin: 20px;  
  font-size: 25px;  
  color: blue;  
  width: 200px;  
}
```

```
.ex-21 #p1{  
  word-wrap: break-word;  
}
```

```
.ex-21 #p2{  
  overflow: auto;  
}
```

شاهد المثال: (الشريحة التالية الناتج)  
ملاحظة: يمكننا استخدام طرق أخرى لحل  
هذه المشكلة .. كما تعلمنا سابقا أو بأي  
طريقة أخرى <sup>^</sup>\_\*



# Text Effects and Typographic Styles

Anees Hikmat  
Anees Abu-  
hmaiddddddddddd  
dddddddddddddd  
dddddddddddddd  
dddddd

باستخدام  
word-wrap: break-word;

2nees.com Website  
for Web  
Developerrrrrrrrrr

حل للمشكلة باستخدام  
overflow: auto;

aneeshikmat@2nees.commmmmmmmmmmmmmmmm

الحالة الافتراضية

لا تنسى تطبيق ما تتعلمه وما تشاهده

يقول الرافي رحمة الله:  
"عَلِمَ اللهُ! ما فَتَنَ المغرورين من شبابنا إلا ما أخذهم من هذه الحضارة،  
فإنَّ لها في زينتها ورونقها أخذةً كالسحر،  
فلا يميزون بين خيرها وشرها،  
ولا يفرّقون بين مبادئها وعواقبها،  
ثم لا يقتنون منها إلا ما يدعوهم إلى ما يميت  
ويصدّهم عما يحيي وما يحول بينهم وبين قلوبهم،  
فليس إلا المتابعة والتقليد".

الرافي: تحت راية القرآن

# Multiple Columns

بعد حديثنا عن التنسيقات الخاصة بالنصوص، فإننا سننتقل الى موضوع مهم بالنسبة لأي موقع، وهو الإهتمام بأن يقرأ زائر الموقع المحتوى بكل سهولة، مما يشجعه دائما على الزيارة المستمرة للموقع، بالإضافة الى ضمان أكثر فائدة يمكن تحقيقها لزائر هذا الموقع، مما يعود بالفائدة الإيجابية على الموقع والمؤسسة الراعية لهذا الموقع، لذلك تجد إهتمام كبير بطريقة عرض الفقرات، ومن أهم الأساليب المتواجدة حاليا لعرض النصوص هو تقسيم النصوص - المحتوى الخاص بأي موقع على شكل أعمدة Columns... وخصوصا بسبب اختلاف أحجام الشاشات، وظهور شاشات wide بمساحات كبيرة فعلا، وهذا يعني لو أن النص سيعرض بدون محددات، فسيكون ممتدا على حجم كبير وهو شكل مزعج لعين القارئ .. لذلك نجد أن المواقع الإخبارية والمجلات تعتمد هذا الأسلوب ^\_^ ...

# Multiple Columns

الآن، باستخدام ال css3 أصبح من الممكن التحكم بشكل سهل بهذا المحتوى،  
وتقسيمه على شكل أعمدة...

وهناك طريقتين لذلك:

الأولى: تسمى هذه الطريقة ب Prescriptive Columns وهي طريقة يكون فيها عدد  
الأعمدة محدد.. (column-count)

الثانية: تسمى هذه الطريقة Dynamic Columns وهي طريقة يكون فيها حساب عدد  
الأعمدة بشكل تلقائي...ولهذا أفضل ما يستخدم هذا الشكل في حالة وجود

Flex Layout (column-width) .. ^\_^

# Multiple Columns

لنبدأ الآن على بركة الله تعالى بال Prescriptive Columns:

هذه الطريقة تعد الطريقة البسيطة أو السهلة لتقسيم محتوى معين على شكل

أعمدة، والصيغة العامة لها هي: `E { column-count: columns; }`

ال E يرمز الى ال wrapper الذي يحتوي المحتوى المراد تقسيمه، وال columns

تمثل عدد الأعمدة المراد تقسيم المحتوى على عددها ...

وهذا يعني على سبيل المثال `div {column-count: 5}` أي قم بتقسيم المحتوى

المتواجد داخل ال div الى 5 أعمدة `^_^....`

شاهد المثال في الشريحة التالية :

(يمكنك انشاء نصوص عشوائية عن [طريق هذا الموقع](#))

# Multiple Columns

لاحظ الشكل التالي، والذي يحتوى على جزئين.. الجزء الأول هو الشكل أو الناتج الافتراضي للمحتوى، أما الشكل الثاني فهو باستخدام ال columns ^\_^ ( الشيفرة البرمجية في الشريحة التالية):

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean quis convallis erat, id sollicitudin ligula. Nulla nec risus eget mauris sollicitudin hendrerit sed cursus libero. Suspendisse potenti. Curabitur nec porttitor nibh, quis scelerisque neque. Praesent dictum feugiat ultricies. Proin eleifend lacus sit amet justo vestibulum commodo. Phasellus blandit lectus eget magna viverra convallis. Vivamus vitae nisi urna. Donec sodales ut metus porta sodales. Cras at lorem sit amet risus placerat finibus. Integer ornare risus ante, sit amet facilisis justo maximus quis. In hac habitasse platea dictumst. Phasellus rutrum posuere nulla, id ultricies justo sodales eget. Aenean volutpat vehicula sem quis tempor. Curabitur placerat mi ut leo tristique bibendum. Nunc commodo pellentesque erat vel molestie. Nam lacinia vel orci ut luctus. Morbi dignissim mauris nec arcu pulvinar, a gravida dui pretium. In at lectus elementum, auctor ex quis, dapibus magna. Quisque maximus lorem at molestie dignissim. Pellentesque imperdiet elit nec lacus congue, ac placerat mauris interdum. Nulla pulvinar ornare tincidunt. Etiam purus sem, mollis vitae nisl ut, fermentum euismod purus. Curabitur sollicitudin, urna sit amet euismod posuere, lectus orci placerat arcu, sit amet imperdiet magna nulla eget sapien. Fusce auctor diam nec libero lobortis rhoncus. Etiam tristique nisl ut magna sagittis, vel dapibus ligula lobortis. Cras facilisis eu magna eu blandit. Donec nec nisl vel tellus congue blandit ac non ex. Maecenas auctor molestie urna eget faucibus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean quis convallis erat, id sollicitudin ligula. Nulla nec risus eget mauris sollicitudin hendrerit sed cursus libero. Suspendisse potenti. Curabitur nec porttitor nibh, quis scelerisque neque. Praesent dictum feugiat ultricies. Proin eleifend lacus sit amet justo vestibulum commodo. Phasellus blandit lectus eget magna viverra convallis. Vivamus vitae nisi urna. Donec sodales ut metus porta sodales. Cras at lorem sit amet risus placerat finibus. Integer ornare risus ante, sit amet

facilisis justo maximus quis. In hac habitasse platea dictumst. Phasellus rutrum posuere nulla, id ultricies justo sodales eget. Aenean volutpat vehicula sem quis tempor. Curabitur placerat mi ut leo tristique bibendum. Nunc commodo pellentesque erat vel molestie. Nam lacinia vel orci ut luctus. Morbi dignissim mauris nec arcu pulvinar, a gravida dui pretium. In at lectus elementum, auctor ex quis, dapibus magna. Quisque maximus lorem at molestie dignissim. Pellentesque imperdiet elit nec lacus congue, ac placerat

mauris interdum. Nulla pulvinar ornare tincidunt. Etiam purus sem, mollis vitae nisl ut, fermentum euismod purus. Curabitur sollicitudin, urna sit amet euismod posuere, lectus orci placerat arcu, sit amet imperdiet magna nulla eget sapien. Fusce auctor diam nec libero lobortis rhoncus. Etiam tristique nisl ut magna sagittis, vel dapibus ligula lobortis. Cras facilisis eu magna eu blandit. Donec nec nisl vel tellus congue blandit ac non ex. Maecenas auctor molestie urna eget faucibus.

# Multiple Columns

أما بخصوص الشيفرة البرمجية التي استخدمت في المثال السابق فهي:

```
<body class="ex-22">
  <div class="defStyle">
    Lorem ipsum dolor sit amet,
    Curabitur placerat mi ut le
  </div>

  <div class="usingColumn">
    Lorem ipsum dolor sit amet,
    Curabitur placerat mi ut le
  </div>
</body>
```

```
.ex-22 .usingColumn {
  -webkit-column-count: 3;
  -moz-column-count: 3;
  column-count: 3;
}
```

أهم ما يجب ملاحظته هنا، هو استخدام ال vendor لكي تدعم أو تعمل هذه الخاصية على المتصفحات المختلفة .. (-webkit-, -moz-) <--- تابع

# Multiple Columns

وطبعا، ولأهمية الموضوع سأكررها مجددا، لا تنسى استخدام موقع مثل caniuse حتى تتحقق من دعم المتصفحات لخاصية معينة.. ولكن كيف علمت بأنها تحتاج الى ال vendor؟! فذلك أيضا من خلال الموقع.. شاهد:

← → ↻ ↶ ⌚ ☆ http://caniuse.com/#search=column

#

## CSS3 Multiple column layout - CR

Method of flowing information in multiple columns

Current aligned

Usage relative

Show all

IE	Firefox	Chrome	Safari	Opera
		31		
		36		
		37		
8		38		
9		39		
10	35	40		
11	36	41		
TP	37	42		28
	38	43		29
	39	44		

Chrome 40  
Partial support with prefix: **-webkit-**  
Browser usage  
Global: 27.38%



# Multiple Columns

أيضا، من الحركات الجميلة، ك Front end Developer أو كمطور مواقع هو انشاء آلية للتعامل مع هذه الخصائص المختلفة، دون الحاجة الى تكرارها دوما، ويكون ذلك من خلال استغلال ما تعلمناه سابقا عن طريق ال selectors .. فمثلا، إن أكثر ما أقوم باستخدامه لتقسيم المحتوى الى أعمدة هو أن تكون بعدد 2 أو 3 أو 4 أعمدة .. لهذا أقوم بإنشاء الشكل بالتنسيق:

```
.ex-22 .content-column-2 {  
  -webkit-column-count: 2;  
  -moz-column-count: 2;  
  column-count: 2;  
}
```

بهذا الشكل نقوم باستخدام  
كلاس بإسم محدد لتنفيذ  
هذه العملية ...

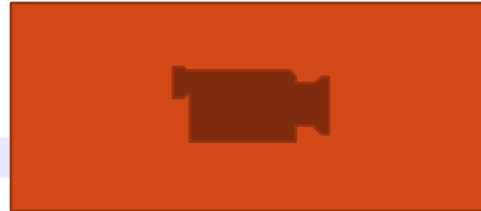
```
/* OR */  
.ex-22 div[class*='column-3'] {  
  -webkit-column-count: 3;  
  -moz-column-count: 3;  
  column-count: 3;  
}
```

هنا قمنا بتحديد أن أي  
كلاس يحتوي على  
column-3

```
/* OR */  
.ex-22 div[class*='-4'] {  
  -webkit-column-count: 4;  
  -moz-column-count: 4;  
  column-count: 4;  
}
```

أما هنا فأخبرناه أينما  
وجدت -4 فقم بتنفيذ  
هذه القيم ..

هذا الشكل يمثل نوعا للإستخدام الجميل لما تعلمناه من خصائص دائما فكر بأن يكون عمالك مرنا ولكل شكل من الذي ذكر حسنات وسلبيات، وهناك أشكال كثيرة أخرى، لكن أحببت لفت نظركم الى هذا الموضوع



# Multiple Columns

الطريقة الثانية column-width: Dynamic Columns:

هذه الطريقة الجميلة تستخدم لتقسيم المحتوى الى أعمدة بنائاً على ال width الذي سنقوم بتحديدده لكل عامود، وليس عن طريق تحديد عدد الأعمدة، وبهذا فإن هذه الطريقة فعالة جداً لأي تصميم Flex.. بالإضافة الى ذلك، فإن عدد الأعمدة يتم تحديده بنائاً على ال width الذي اعطي للأعمدة معتمداً على حكم ال parent .. (يقوم المتصفح بعمل fill للأعمدة بنائاً على ال width)

الصيغة العامة وهي سهلة جداً:  $\wedge\_ \wedge$

```
E { column-width: length; }  
div { column-width: 100px; }
```

# Multiple Columns

```
/*Example 23*/
```

```
.ex-23 div{
```

```
background: black;
```

```
padding: 20px;
```

```
margin: 20px;
```

```
font-size: 25px;
```

```
color: blue;
```

```
-webkit-column-width: 150px;
```

```
-moz-column-width: 150px;
```

```
column-width: 150px;
```

```
}
```

والآن .. لنشاهد مثالا ^\_^

إن ما تم ذكره بخصوص دعم المتصفحات

للطريقة الأولى ينطبق هنا.. لكن لاحظ

كيف أنه تم تكرار 7 أعمدة .. لأن ال

Width الموجود هو 150

مثال 1

Lorem ipsum neque. amet risus quis tempor. elementum, sem, mollis tristique nisl ut  
dolor sit amet, Praesent placerat Curabitur auctor ex quis, vitae nisl ut, magna sagittis,  
consectetur dictum feugiat finibus. Integer placerat mi ut dapibus fermentum vel dapibus  
adipiscing elit. ultricies. Proin ornare risus leo tristique magna. euismod ligula lobortis.  
Aenean quis eleifend lacus ante, sit amet bibendum. Quisque purus. Cras facilisis  
convallis erat, sit amet justo facilisis justo Nunc maximus Curabitur eu magna eu  
id sollicitudin vestibulum maximus quis. commodo lorem at sollicitudin, blandit. Donec  
ligula. Nulla commodo In hac pellentesque molestie urna sit amet nec nisl vel

# Multiple Columns

الآن لنشاهد مثالاً آخر.. وانتبه للقيم الآن ثم انتقل للنتائج (انظر لعدد الأعمدة):

Lorem ipsum dolor amet, consectetur adipiscing elit. Aenean quis conyallis nec porttitor nibh, quis scelerisque neque. Praesent dictum feugiat ultricies. Proin vitae nisi urna. Donec posuere sodales ut metus porta sodales. Cras at lorem sit amet rictus Phasellus rutrum posuere nulla, id ultricies justo sodales eget. Aenean volutpat lacinia vel orci ut luctus. Morbi dignissim mauris nec arcu pulvinar, a gravida dui imperdiet elit nec lacus congue, ac placerat mauris interdum. Nulla pulvinar nulla eget ligula lobortis. Cras facilisis eu magna eu blandit. Donec nec nisl vel tellus

```
.ex-23 .ex-23-2 {  
  width: 1000px;  
  -webkit-column-width: 100px;  
  -moz-column-width: 100px;  
  column-width: 100px;  
}
```

مثال 2

# Multiple Columns

في المثال السابق، كان ال width لل parent يساوي 1000 بكسل، وكان حجم  
العمود يساوي 100 بكسل، لكن عدد الأعمدة الذي ظهر كان يساوي 8 أي  $8 * 100 = 800$ ، إذن أين المسافة الباقية؟!

أولا، يوجد هناك gap بمقدار 16 بكسل تقريبا كمساحة افتراضية بين الأعمدة  
المنشأة، وبهذا يكون يصبح لدينا  $8 - 1 * 16 = 112$ .. (عدد الفراغات بين  
الأعمدة يساوي عدد الأعمدة - 1) ..

وبهذا يصبح المجموع لدينا  $800 + 112 = 912$ ..

حقيقة هذه الخوارزمية، تعد من الخوارزميات الذكية، لذلك فهي تقوم على تغيير

حكم الأعمدة بما يناسب ال parent تلقائيا  $^_^$ ، وبهذا فإن حجم الأعمدة  
بالإضافة الى حجم ال gabs سيختلف بناءا على ما يكون مناسباً للعرض، ويكون

الحد الأدنى لل width هنا هو 100 بكسل (min-width)  $^_*$

أما القياسات الفعلية (حساب تقريبي وليس فعلي) لمثالنا فقد أصبحت تقريبا  
تساوي  $8 * 109$  و ال  $gap * 18.2 * 7$

# Multiple Columns

ملاحظة مهمة أيضا:

عندما نقوم بإنشاء أعمدة متعددة، وتحتوي هذه الأعمدة على فقرات مثلا، فإن التنسيق الأصلي لهذه الفقرة لا يذهب، بل يبقى موجودا .. ولذلك يمكنك التحكم بالمسافة بين الفقرات والهامش وغيرها .. شاهد المثال:  
انبته لمواقع الفراغات والتي تمثل المساحة الافتراضية للمساحات بين الفقرات ..

الشفيرة البرمجية: (النتائج في الشريحة التالية)

```
<div class="ex-23-3">
```

```
<p>Lorem ipsum dolor sit amet, consecteturrrrrrrr </p><p>adipiscing eli  
Curabitur placerat mi ut leo tristisque bibendum. </p><p>Nunc commodo pe
```

```
</div>
```

# Multiple Columns

Lorem ipsum dolor  
sit amet,  
consecteturrrrrrr

فراغ الفقرة  
margin

adipiscing elit.  
Aenean quis  
convallis erat, id  
sollicitudin ligula.  
Nulla nec risus eget  
mauris sollicitudin  
hendrerit sed cursus  
libero. Suspendisse  
potenti. Curabitur  
nec porttitor nibh,  
quis scelerisque  
neque. Praesent  
dictum feugiat

lectus eget magna  
viverra convallis.  
Vivamus vitae nisi  
urna. Donec sodales  
ut metus porta  
sodales. Cras at  
lorem sit amet risus  
placemat finibus.  
Integer ornare risus  
ante, sit amet  
facilisis justo  
maximus quis. In  
hac habitasse platea  
dictumst. Phasellus  
rutrum posuere  
nulla,

فراغ الفقرة  
margin

id ultricies justo

فراغ الفقرة  
margin

Nunc commodo  
pellentesque erat vel  
molestie. Nam  
lacinia vel orci ut  
luctus. Morbi  
dignissim mauris  
nec arcu pulvinar, a  
gravida dui pretium.  
In at lectus  
elementum, auctor  
ex quis, dapibus  
magna. Quisque  
maximus lorem at  
molestie dignissim.  
Pellentesque  
imperdiet elit nec  
lacus congue, ac  
placemat mauris

purus. Curabitur  
sollicitudin, urna sit  
amet euismod  
posuere, lectus orci  
placemat arcu, sit  
amet imperdiet  
magna nulla eget  
sapien. Fusce auctor  
diam nec libero  
lobortis rhoneus.  
Etiam tristique nisl  
ut magna sagittis,  
vel dapibus ligula  
lobortis. Cras  
facilisis eu magna eu  
blandit. Donec nec  
nisl vel tellus congue  
blandit ac non ex.  
Maecenas auctor

# Multiple Columns

إذا وكما لاحظنا من المثال السابق .. فإن الخصائص لا تضيع مع ال columns لذلك نقوم نحن بتصفيرها إن أردنا ذلك `*_^*`  
شاهد المثال: (لاحظ كيف قمنا بتصفير القيم، واعطاء هامش `*_*`)

○ Lorem ipsum dolor sit amet, consecteturrrrrr

○ adipiscing elit.

Aenean quis convallis erat, id sollicitudin ligula. Nulla nec risus eget mauris sollicitudin hendrerit sed cursus libero. Suspendisse potenti. Curabitur nec porttitor nibh, quis scelerisque neque. Praesent dictum feugiat ultricies.

Proin eleifend lacus sit amet

Vivamus vitae nisi urna.

Donec sodales ut metus porta sodales. Cras at lorem sit amet risus placerat finibus. Integer ornare risus ante, sit amet facilisis justo maximus quis. In hac habitasse platea dictumst. Phasellus rutrum posuere nulla,

○ id ultricies justo sodales eget. Aenean

volutpat vehicula sem quis

○ Nunc commodo pellentesque erat vel

molestie. Nam lacinia vel

```
.ex-23 .ex-23-4{  
  -webkit-column-width: 250px;  
  -moz-column-width: 250px;  
  column-width: 250px;  
}
```

```
.ex-23 .ex-23-4 p{  
  margin: 0;  
  padding: 0;  
  text-indent: 70px;  
}
```

ut, f

pur

urna

or

di

pu

pr

el

da

m

di

in

co



# Multiple Columns

أيضا من الملاحظات الجميلة على هذا الموضوع ^\_^ ..  
ماذا سيحدث لو قمنا بدمج الطريقة الأولى والثانية معا !!! \*\_^\*

```
.ex-23 .ex-23-5 {
```

```
width: 1000px;
```

```
-webkit-column-count: 4;
```

```
-moz-column-count: 4;
```

```
column-count: 4;
```

```
-webkit-column-width: 100px;
```

```
-moz-column-width: 100px;
```

```
column-width: 100px;
```

```
}
```

شاهد الشكل التالي:

إن أول ما نفكر به عند رؤيتنا لهذا

الشكل هو Error \*\_^\* .. لكن في

خوارزمتنا الرائعة هذه .. فهذا لن

يحدث ^\_^

والذي سيحدث فعلا ^\_^ .. هو عملية دمج بين الطريقتين، بحيث يكون الحد الأعلى لعدد الأعمدة هو 4، والحجم الافتراضي (أو أقل حجم) هو 100px



# Multiple Columns

وبنأنا على ما تحدثنا به الآن ^\_^، ماذا تتوقع أن تكون النتيجة ^\_\* ..  
شاهد النتيجة ^\_\* : (لا تنسى مشاهدة النتائج – جميعها مهمة وبها فكرة)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean quis convallis erat, id sollicitudin ligula. Nulla nec risus eget mauris sollicitudin hendrerit sed cursus libero. Suspendisse potenti. Curabitur nec porttitor nibh, quis scelerisque neque. Praesent dictum feugiat ultricies. Proin eleifend lacus sit amet justo vestibulum

convallis. Vivamus vitae nisi urna. Donec sodales ut metus porta sodales. Cras at lorem sit amet risus placerat finibus. Integer ornare risus ante, sit amet facilisis justo maximus quis. In hac habitasse platea dictumst. Phasellus rutrum posuere nulla, id ultricies justo sodales eget. Aenean volutpat vehicula sem quis tempor. Curabitur placerat mi ut leo

molestie. Nam lacinia vel orci ut luctus. Morbi dignissim mauris nec arcu pulvinar, a gravida dui pretium. In at lectus elementum, auctor ex quis, dapibus magna. Quisque maximus lorem at molestie dignissim. Pellentesque imperdiet elit nec lacus congue, ac placerat mauris interdum. Nulla pulvinar ornare tincidunt. Etiam purus

Curabitur sollicitudin, urna sit amet euismod posuere, lectus orci placerat arcu, sit amet imperdiet magna nulla eget sapien. Fusce auctor diam nec libero lobortis rhoncus. Etiam tristique nisl ut magna sagittis, vel dapibus ligula lobortis. Cras facilisis eu magna eu blandit. Donec nec nisl vel tellus congue blandit ac non ex. Maecenas auctor molestie urna eget

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

{لَا يَسْتَوِي الْقَاعِدُونَ مِنَ الْمُؤْمِنِينَ غَيْرُ أُولِي الضَّرَرِ وَالْمُجَاهِدُونَ فِي سَبِيلِ اللَّهِ بِأَمْوَالِهِمْ وَأَنْفُسِهِمْ فَضَّلَ اللَّهُ الْمُجَاهِدِينَ بِأَمْوَالِهِمْ وَأَنْفُسِهِمْ عَلَى الْقَاعِدِينَ دَرَجَةً وَكُلًّا وَعَدَّ اللَّهُ الْحُسْنَى وَفَضَّلَ اللَّهُ الْمُجَاهِدِينَ عَلَى الْقَاعِدِينَ أَجْرًا عَظِيمًا (95) دَرَجَاتٍ مِنْهُ وَمَغْفِرَةً وَرَحْمَةً وَكَانَ اللَّهُ غَفُورًا رَحِيمًا (96)}

صدق الله العظيم

سورة النساء (95 – 96)

# Multiple Columns

والآن ..نتابع موضوعنا أيضا وبجزئة جميلة جدا ^\_^ وهي ال

## Column Gaps and Rules

لقد تحدثنا في إحدى الشرائح السابقة عن وجود Gap بين الأعمدة، هذا ال Gap حجمه الافتراضي هو 1em .. (أي 16 بكسل)

الآن يمكنني تغيير حجم هذه المساحة من 1em الى المساحة التي أحتاجها، ولا يمكن أن تكون المساحة التي يتم وضعها أقل من 1em \* ..

الصيغة العامة هي:

```
E { column-gap: length; }
```

```
div { column-gap: 4em; }
```

# Multiple Columns

شاهد المثال: (لاحظ المسافة بين الأعمدة \*^\_\*)

```
.ex-24 .ex-24-1 {  
  -webkit-column-gap: 4em;  
  -moz-column-gap: 4em;  
  column-gap: 4em;  
  -webkit-column-count: 4;  
  -moz-column-count: 4;  
  column-count: 4;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean quis convallis erat, id sollicitudin ligula.

vitae nisi urna. Donec sodales ut metus porta sodales. Cras at lorem sit amet risus placerat finibus. Integer ornare

vel orci ut luctus. Morbi dignissim mauris nec arcu pulvinar, a gravida dui pretium. In at lectus

urna sit amet euismod posuere, lectus orci placerat arcu, sit amet imperdiet magna nulla eget sapien. Fusce

# Multiple Columns

الآن بعدما شاهدنا هذه الخاصية الجميلة .. نكون انتهينا من الشق الأول من العنوان فقط ^^، وهناك شق آخر وهو ال `^_^ Rules`

ال `column-rule` هي خاصية تتيح لنا انشاء خط فاصل بين الأعمدة (شبيهة بال `border` من حيث المبدأ)، لكن هذه الخاصية تستخدم مع الأعمدة، ولا يمكننا هنا استخدام ال `border *_^` .. لأنك إن كنت تذكر، فهذه الأعمدة تنشأ من قبل الخوارزمية الذكية، وتنظم المحتوى بناءً على حساباتها `^_^` .. شاهد الصيغة العامة لل `column-rule ^_^`

لاحظ أن الصيغة الخاصة بها تشبه

ال `border`، ومكنني اختصارها

بسطر واحد فقط ^^

```
E {  
  column-rule-width: length;  
  column-rule-style: border-style;  
  column-rule-color: color;  
  column-rule: length border-style color;  
}
```

الإختصار

```
column-rule-width: length;  
column-rule-style: border-style;  
column-rule-color: color;
```

```
column-rule: length border-style color;
```

# Multiple Columns

مثال:

لاحظ أنني قمت باستخدام الاختصار  
^\_\*، ولا تنسى المتصفحات الأخرى



```
.ex-24 .ex-24-2 {  
  -webkit-column-gap: 6em;  
  -moz-column-gap: 6em;  
  column-gap: 6em;  
  -webkit-column-rule: 3px solid red;  
  -moz-column-rule: 3px solid red;  
  column-rule: 3px solid red;  
  -webkit-column-width: 100px;  
  -moz-column-width: 100px;  
  column-width: 100px;  
}
```

Lorem  
ipsum dolor  
sit amet,  
consecteturrrrrr  
adipiscing  
elit. Aenean  
quis  
convallis

commodo.  
Phasellus  
blandit  
lectus eget  
magna  
viverra  
convallis.  
Vivamus

ultrices  
justo  
sodales  
eget.  
Aenean  
volutpat  
vehicula  
sem quis

dapibus  
magna  
Quisque  
maxim  
lorem a  
molest  
digniss  
Dolore

# Multiple Columns

وأخيرا .. سنتحدث بإذن الله تعالى عن موضوع مهم وآخر موضوع في هذه

الجزئية وهو Containing Elements within Columns

الآن .. سنتحدث عن جزئية مهمة جدا، وهي مبدأ الإحتواء، ولكي يصل

المراد من هذه الكلمة، أريد أن أسألك سوألا.. هل لاحظت أمرا في كل

الأمثلة التي تطرقنا اليه في هذا الموضوع ؟

هناك أمر مشترك في جميع هذه الأمثلة!

جميع الأمثلة التي ذكرناها، كانت عبارة عن نصوص فقط، ولم تحتوي

على عناصر أخرى مثل الصور .. هنا ستحدث مشكلة وسيقوم بحلها كل

متصفح بطريقة مختلفة عن الآخر .. والآن وقبل البدء لنرى كيف ستظهر

الصورة إن قمنا بوضعها داخل column بالشكل الافتراضي ..



# Multiple Columns



## Google Chrome



# Multiple Columns

كما تلاحظ، فإن متصفح ال Firefox قام بعرض الصورة فوق النص، أما ال Chrome فقام بعمل crop للصورة ..، طبعاً ما ينطبق على الصور ممكن أن ينطبق على عناصر ال html الأخرى مثل ال headers مثلاً .. لذلك من قام بتطوير هذا ال module في ال css3 راع هذه الإحتمالات ^\_^، لكن الأمر المحزن أن الحلول المقدمة ليست مدعومة بعد من أغلب المتصفحات .. لذلك سنقوم بذكرها لكي يبقى ذلك في مخيلتك حتى يتم اعتمادها رسمياً في المتصفحات الرئيسية .. مثل خاصية column-span

خاصية ال column-span هي خاصية تحدد الحجم الذي يجب العنصر أخذه أو حجز مساحته، في الحالة الافتراضية العنصر يعرض في صف واحد فقط.. لكن هذا الخيار يمكن تعديله بواسطة هذه الخاصية الرائعة ...

# Multiple Columns

`column-span: 1|all`

الصيغة العامة:

القيمة 1 هي القيمة الافتراضية، و all تعني اعرض العنصر على جميع الأعمدة..

هذه الخاصية حتى تاريخ كتابة هذه الشرائح 1-4-2015 لم تكن مدعومة من ال Firefox.. لذلك اذا أردت مشاهدة أو تجربة هذه الخاصية يجب عليك استخدام ال Chrome 4.0+ مع خاصية ال -webkit-

كما أنها غير داعمة لجميع العناصر، لذلك سنقوم بتجربة الخاصية مع ال

\*\_ ^ H1 tag

# Multiple Columns

Google Chrome

www.2nees.com

Lorem ipsum sit amet, consectetur	nec porttitor nibh, quis scelerisque neque.	Vivamus vitae nisi urna. Donec sodales ut	dictumst. Phasellus rutrum posuere	mole Nam vel c luctu
-----------------------------------	---	---	------------------------------------	----------------------

```
.ex-25 .ex-25-2, .ex-25 .ex-25-1{
```

```
-webkit-column-count: 8;
```

```
-moz-column-count: 8;
```

```
column-count: 8;
```

```
}
```

```
.ex-25 .ex-25-2 h1{
```

```
-webkit-column-span: all;
```

```
column-span: all;
```

```
}
```

FireFox

www.2nees.com

potenti, viverra quis.	Crabitur convallis. habit
>Lorem ipsum sit amet, consectetur	Vivamus platea
ipsum dolor nibh, quis vitae nisi	dictu



# Multiple Columns

والآن صديقي .. بعد مشاهدتك للخاصية السابقة، هناك أيضا خصائص أخرى مثل break-after, break-before, and break-inside يمكنك البحث عنها والإطلاع عليها لأنها ليست مدعومة حتى تاريخ كتابة هذه الكلمات \*\_\*\_، فهي من الخصائص التي اضفيت الى هذا ال module \*\_^.

والآن قد تتسائل يا صديقي، لماذا قمنا بطرح هذا الموضوع وذكرنا احدى الخصائص، وتطرقنا كذلك لذكر لمجموعة خصائص أخرى، علما أن هذه الخصائص غير مدعومة، والإجابة بكل بساطة، هي أن هذا العلم يتطور بشكل سريع، والخصائص الغير مدعومة اليوم، سيتم دعمها غدا، وهي خصائص لم توضع لمجرد التسلية، بل لها وظيفة رائعة ومهمة ستقوم بها .. لذلك يجب عليك أن تعلم عنها أو بوجودها على أقل تقدير، أو المشاكل التي يمكنك حلها من خلال استخدام تلك الخصائص \*\_^...

ثلاث يعزّ الصبر عند حلولها  
ويذهل عنها عقل كل لبيب  
خروج اضطرار من بلاد يحبّها  
وفرقة إخوان وفقد حبيب

زهير بن أبي سلمى

# Background Image And Decorative Properties

هذا الموضوع قد يكون من أجمل المواضيع التي أضفيت الى خصائص تنسيق صفحات الويب، وإن سألت عن السبب وأنت تقول ال background Image هي خاصية متواجدة منذ ال css فسأقول لك نعم، لكن في ال css3 أصبحت هناك ميزة رائعة جدا، وتستخدم لإعطاء المصمم أو المطورة قدرة رائعة على التحكم بالتصميم، وهي إمكانية اضافة أكثر من background-image (أي أكثر من صورة كخلفية لنفس العنصر ^\_^) وهذه الميزة لم تكن متوفرة من قبل، والميزة الثانية والخطيرة جدا، وهي امكانية التحكم بحجم الخلفية وذلك عن طريق شيفرة ال css3 وبكل سهولة وبشكل منسجم مع التنسيق... ^\_^، وبإمكاننا القول أن هاتان الميزتان هما أجمل ما أضيف تحت هذا الموضوع و [يمكنك الإطلاع على المزيد من هنا](#)

# Background Image And Decorative Properties

والآن أول جزئية في هذا الموضوع وهي ال

^\_^ Multiple Background Images

أولا ما أحب إخبارك به، هو أن هذه المزية الجديدة في ال Backgrounds and Borders module هي ليست خاصية جديدة، وإنما مزية، فكما نعلم، فإن خاصية ال background وخصائصها موجودة من قبل ال css3 لكن المزية التي اضيفت بالسماح لوضع أكثر من صورة هي الجديدة، وبكل تأكيد ترافق مع السماح بتعدد الخلفيات الى تعدد أيضا الخصائص التي ترتبط بكل خاصية ..^\_^ مثل ال repeat وال ..pos \. والآن لنشاهد الصيغة العامة ومثالا عليها \*\_^



# Background Image And Decorative Properties

الصيغة العامة: `E { background-image: value, value; }`

كما تلاحظ من الصيغة العامة، فإن المهم هنا أن القيم المتعددة يفصل بينها فاصلة... مثال 1:

لاحظ كيف تم وضع صورتين كخلفية وتم الفصل بينهم بفواصل، كما أنه تم تحديد موقع كل صورة بناءً على الترتيب التي وضعت به الصور، 95% 85% موقع الصورة 2nees.png.. وهكذا، كما أنه يمكننا استخدام خاصية واحدة لكل الخلفيات وذلك عن طريق وضع قيمة واحدة فقط مثل `no-repeat` ...

```
.ex-26-1 {  
  background-image: url("2nees.png"), url("web.png");  
  background-position: 95% 85%, 50% 50%;  
  background-repeat: no-repeat;  
}
```

# Background Image And Decorative Properties



# Background Image And Decorative Properties

مثال 2: أيضا يمكننا استخدام الطريقة المختصرة `^_*` شاهد المثال:

```
.ex-26 .ex-26-2 {  
    background: url("2nees.png") no-repeat 40% 50%,  
                url("web.png") no-repeat 80% 0%;  
}
```

2NEES.COM

دورة CSS

وباللغة العربية

وبأكثر من 200 صفحة `^_*`

## Background Image And Decorative Properties

ملاحظة: بما أن هذه الخاصية من خصائص ال css3 فإنه الإصدارات القديمة من المتصفحات الرئيسية أو ال IE9- لا تدعم هذه الخاصية .. لذلك يجب عليك ضمان أن تظهر صورة واحدة أو خلفية واحدة على الأقل، لذلك نقوم بكتابة الصيغة التقليدية قبل اضافة الميزة الجديدة ..

شاهد المثال:

(\* بالنسبة لمتصفح ال IE الغير داعمة لهذه الخاصية، فإنها ستقوم باختزال التعدد في صورة واحدة، أما المتصفحات الأخرى الغير داعمة فستقوم بتنفيذ السطر الأول (وضع خلفية واحدة فقط)، هذا في حالة استخدام الطريقة المختصرة، لكن اذا استخدمت الطريقة الطويلة \*background- فإن ال IE لن يقوم بإظهار أي صورة .. لذلك وجب التنويه ..اييه

شاهد الشريحة التالية ^\_^

# Background Image And Decorative Properties

هذا الشكل هو الذي يجب عليك كتابة شيفرتك البرمجية به، ففي حال دعم المتصفحات لمزية التعدد سيتم التنفيذ بدون أي مشاكل، والا سيتم تنفيذ الخاصية لوضع صورة واحد فقط ...

```
.ex-26 .ex-26-3{
```

```
background: url("web.png") no-repeat 80% 0%;/*For Non Supporting Browser*/
```

```
background: url("2nees.png") repeat-y 50% 50%,  
            url("web.png") no-repeat 80% 0%;
```

```
-}
```

طبعا هذا الكلام كما ذكرنا بالشريحة السابقة لا ينطبق على ال IE .. شاهد الشريحة التالية لترى كيف ستعرض النتائج على ال IE ...

# Background Image And Decorative Properties

Anees Hikmat Anees Abu-hamid 2nees.com

IE8 Browser

لا يوجد صورة لأن التنسيق مكتوب على شكل  
**background-image: url();**

```
.ex-26 .ex-26-1{  
    background-image: url("2nees.png"), url("web.png");  
    background-position: 95% 85%, 50% 50%;  
    background-repeat: no-repeat;  
}  
  
.ex-26 .ex-26-2{  
    background: url("2nees.png") no-repeat 40% 50%,  
                url("web.png") no-repeat 80% 0%;  
}  
  
.ex-26 .ex-26-3{  
    background: url("web.png") no-repeat 80% 0%; /*For Non Supportin  
                url("2nees.png") repeat-y 50% 50%,  
                url("web.png") no-repeat 80% 0%;
```

Anees Hikmat Anees Abu-hamid 2nees.com

2NEES.COM

قام بتنفيذ الصورة الأولى

هذا الشكل يظهر في  
**Non Support  
Browser for IE  
9 & under**

Anees Hikmat Anees Abu-hamid 2nees.com

شاهد المثال:

2NEES.COM

2NEES.COM

2NEES.COM

قام بتنفيذ الصورة الأولى

اللَّهُمَّ أَنْتَ رَبِّي لَا إِلَهَ إِلَّا أَنْتَ ، خَلَقْتَنِي وَأَنَا عَبْدُكَ ، وَأَنَا عَلَى عَهْدِكَ  
وَوَعْدِكَ مَا اسْتَطَعْتُ ، أَعُوذُكَ مِنْ شَرِّ مَا صَنَعْتَ ، أَبوءُ لَكَ بِنِعْمَتِكَ  
عَلَيَّ وَأَبوءُ بِذَنْبِي فَاغْفِرْ لِي فَإِنَّهُ لَا يَغْفِرُ الذُّنُوبَ إِلَّا أَنْتَ

دعاء

# Background Image And Decorative Properties

الآن، وبعد حديثنا الجميع عن استخدام أكثر من صورة كخلفية ننتقل الى  
المزية الثانية والرائعة وهي مزية ال التحكم بحجم الخلفية والتي تسمى ب  
^\_^ Background Size

هذه المزية الرائعة تسمح بتغيير حجم الصورة التي وضعت كخلفية بشكل  
متناسق ^\_^، وحقيقة هذه المزية من أروع المزايا التي أستخدمها في بناء  
المواقع \*\_\*.. لأنها تعطي رونقا رائعا للتصميم، بالإضافة الى طريقة  
التغيير التي يمكنني اتباعها ...  
الصيغة العامة:

```
E { background-size: value; }
```

```
E { background-size: width height; }
```



# Background Image And Decorative Properties

في الصيغة العامة قمت بوضع شكلين، ال value أقصد بها كلمات مثل cover, contain ... لذلك قمت بوضعها على شكل value  
أما السطر الثاني فهو يمثل الشكل بالقيم أو الأرقام مثل auto أو 10% أو 50px .. وهكذا

طبعا هذا فقط للتوضيح ^\_\* ..

شاهد مثلا على الصيغة العامة .. ^\_^:

```
background-size: 100% 70%;  
background-size: auto 200px;  
background-size: cover;
```

# Background Image And Decorative Properties

الآن لنقوم بذكر وشرح القيم والحالات الخاصة بال  $^{\wedge}_{\wedge}$  size :

(1) ال width: وهو عرض الصورة.

(2) ال height: ارتفاع الصورة.

(3) اذا جائت قيمة واحدة مثل 15% فهذا يعني أن ال width = 15% وال height = 15% أيضا.

(4) اذا جائت auto لوحدها فهذا يعني أن ال width وال height = auto وهي الحالة الافتراضية لل background-size

(5) اذا جائت auto مع قيمة مثل 50px فإن الخلفية سيصبح حجمها 50 بكسل ارتفاعا، ويصبح ال width حجمه بشكل تلقائي صغيرا بما يتناسب مع الإرتفاع ..  $^{\wedge}_{\wedge}$

## Background Image And Decorative Properties

(6) Cover وتعني اجعل حجم الخلفية يبقى أكبر ما يمكن بنائاً على المساحة المعطاة لها ويسمح له بتجاوز حجمها.. ويمكن في بعض الحالات أن تختفي اجزاء من الصورة في حالة وجودة pos .. (هذه السمة نستخدمها كثيراً وخصوصاً في التلاعب البصري أثناء ال (parallax scrolling

(7) Contain: وتعني اجعل حجم الخلفية يبقى أكبر ما يمكن بنائاً على المساحة المعطاة لها دون أن يسمح للخلفية بتجاوزها ... وهذا يعني لو أن حجم الصورة 640 \* 1366 وكان حجم الحاوي للخلفية 300 \* 1000 فإن حجم الصورة سيتوقف عندما يصبح ارتفاعها 300 ومن ثم يقوم بعمل resize لل width بنائاً على ال height ..

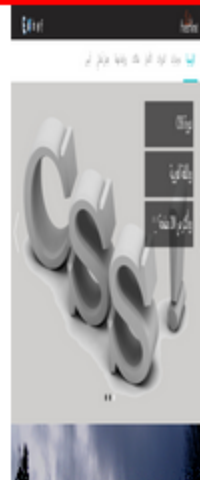
والآن لنشاهد مجموعة من الأمثلة المتنوعة (وانتبه لكل مثال):

# Background Image And Decorative Properties

والآن لنرى أمثلة على استخدام ال `background-size` \* \_ ^ (1)

```
.ex-27 .ex-27-1{  
    background: url("web.png") no-repeat center top;  
    background-size: 100px 70%;  
}
```

Abu-hamid 2nees.com



```
...  
.ex-27 div{  
    padding: 20px;  
    margin: 20px;  
    font-size: 25px;  
    color: brown;  
    width: 80%;  
    height: 300px;  
    border: 4px solid red;  
}
```

# Background Image And Decorative Properties

والآن لنرى أمثلة على استخدام ال `background-size` \* \_ ^ (2)

```
.ex-27 .ex-27-2 {  
  background: url("web.png") no-repeat center top;  
  background-size: 75% auto;  
}
```

Anees Hikmat Anees Abu-hamid 2nees.com الرئيسية مرحبا بك الدورات الأخبار مقالات روابط مهمة بعض أعمالنا



# Background Image And Decorative Properties

والآن لنرى أمثلة على استخدام ال `background-size` \* \_ ^ (3)

```
.ex-27 .ex-27-3 {  
    background: url("web.png") no-repeat center top;  
    background-size: cover;  
-}
```

دورة CSS

وباللغة العربية

وبأكثر من 200 صفحة ^ \_ ^

# Background Image And Decorative Properties

والآن لنرى أمثلة على استخدام ال `background-size` \* \_ ^ (4)

```
.ex-27 .ex-27-4 {  
    background: url("web.png") no-repeat center top;  
    background-size: contain;  
}
```



# Background Image And Decorative Properties

والآن لنرى أمثلة على استخدام ال background-size \* \_ ^ (5)

```
.ex-27 .ex-27-4 {  
    background: url("web.png") no-repeat;  
  
    background: url("2nees.png") no-repeat 2% 15%,  
                url("2nees.png") no-repeat 80% 80%,  
                url("2nees.png") no-repeat 2% 80%,  
                url("2nees.png") no-repeat 80% 2%,  
                url("2nees.png") no-repeat center,  
                url("web.png") no-repeat;  
    background-size: 200px auto, 300px, 15%, 100px 75px, contain, cover;  
}
```





# Background Image And Decorative Properties

بهذا نكون انتهينا من هذه الجزئية أيضا بإذن الله تعالى، وسننطلق الى جزئية جديدة بنفس الموضوع، لكن قبل ذلك، أرجوا أن تكون قد نفذت جميع الأمثلة السابقة، فهي أمثلة مهمة لفهم كل تطبيق ..

والآن يمكنك رؤية النتائج والشيفرة البرمجية من هنا:



في الصحيحين عن أبي موسى الأشعري رضي الله عنه عن سيدنا محمد صل الله عليه وسلم قال: [المؤمن للمؤمن كالبنيان يشد بعضه بعضا وشبك بين أصابعه].

وهذا تشبيه لبيان تعاضد المؤمنين ببعضهم وتقويهم ببعض كما أن البنيان الممسك بعضه ببعض يشد بعضه بعضا، لأن أقواهم ركن، وضعيفهم مستند لذلك الركن القوي، فإذا والاه قوي به" (فيض القدير 6/252).

حديث شريف

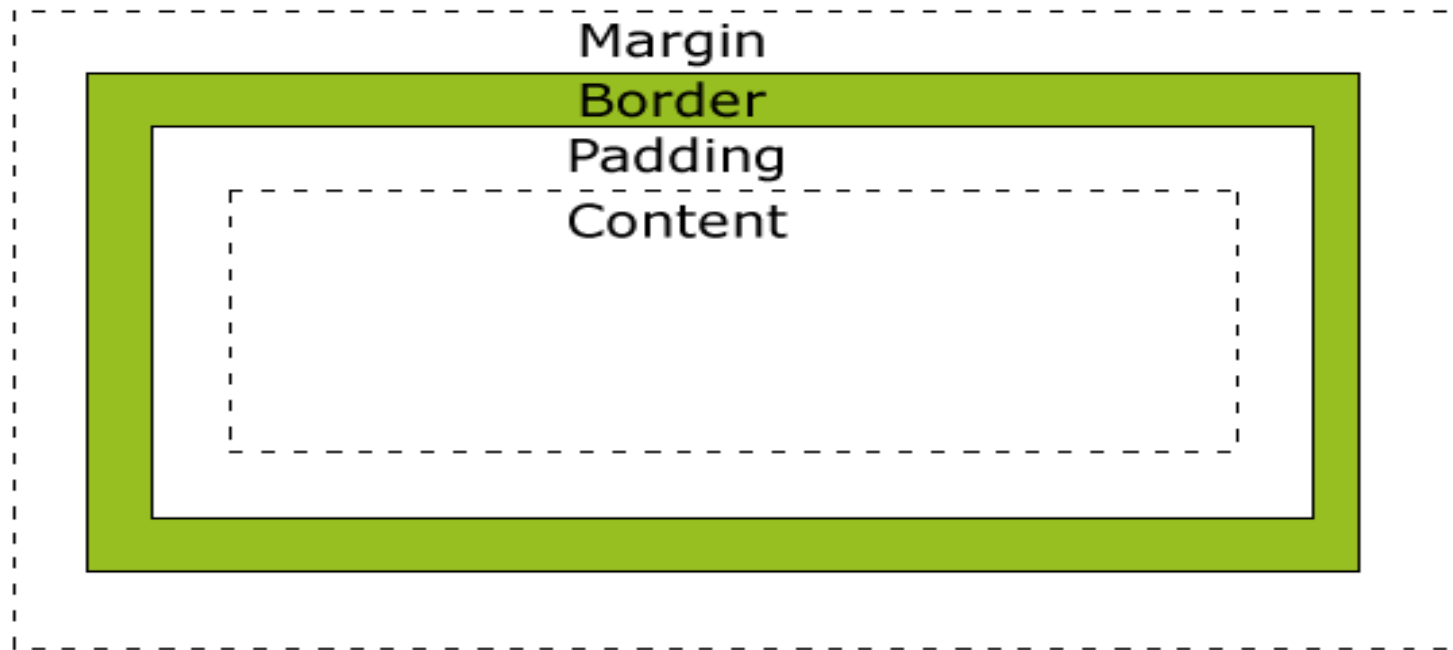
# Background Image And Decorative Properties

والآن بعد حديثنا الممتع عن المزيّتان السابقتين، لنأخذ بعض الإضافات الزخرفية الجديدة في عالم التنسيق والتي ترادفت مع ظهور ال `css3` `^_^` والموضوع هو `Background Clip and Origin` `^_^`

في الإصدار الثاني من ال `css` كان موقع الخلفية يحدد بشكل `relative` بناءً على العنصر الحاوي ووجود ال `padding` .. لكن في ال `css3` قامت بتقديم خاصيتين جديدتين للتحكم بموقع الصورة ونقطة البدء، والإعتبرات المحيطة مثل ال `padding` وال `border` وغيرها من الخصائص ...

# Background Image And Decorative Properties

الخاصية الأولى التي سنتحدث عنها في هذه الجزئية هي ال background-clip، ومن خلال هذه الخاصية وقيمها يمكننا تحديد الجزء الذي سيسمح للخلفية بالوصول اليه كحد أقصى، وذلك بنائاً على ال Box-model .. شاهد ال box-model ثم انتقل للشريحة التالية:



# Background Image And Decorative Properties

كما تلاحظ فإن التسلسل هو margin ثم border ثم padding ثم content ، في الحالة الافتراضية في ال css3 فإن الخلفية (لون أو صورة) تبدأ من ال Border (يعني ال padding وال content وال border سيتم شملهم بالخلفية) ، وهذه القيمة تسمى ب border-box .. والآن، لنشاهد الصيغة الافتراضية لخاصية background-clip:

```
background-clip: border-box|padding-box|content-box;
```

ال border-box: يشمل ال border وال padding وال content وهو الحالة الافتراضية.

ال padding: يشمل ال padding وال content.

ال content: يشمل ال content فقط .. (كل هذه حسب ال box-model)

# Background Image And Decorative Properties

مثال (1)

```
ex-28 .ex-28-1 {  
    background-clip: border-box;  
}
```

# Background Image And Decorative Properties

(مثال 2)

```
ex-28 .ex-28-2 {  
    background-clip:padding-box;  
}
```

# Background Image And Decorative Properties

مثال (3)

```
.ex-28 .ex-28-3 {  
    background-clip: content-box;  
}
```

Anees Hikmat Anees Abu-hamid 2nees.com



# Background Image And Decorative Properties

أما الخاصية الثانية في هذه الجزئية فهي ال `background-origin` من خلال هذه الخاصية يمكنني التحكم بشكل أكبر في الموقع الذي أرغب بدء الحساب به للخلفية (الصورة).. والصيغة العامة والقيم مشابهة للخاصية السابقة؟! شاهد الصيغة العامة:

```
background-origin: border-box | padding-box | content-box ;
```

لكن القيمة الافتراضية هنا هي ال `padding-box`...  
لكن السؤال المهم الآن؟! ..ما الفرق بين الخاصيتين؟!!!!!  
الجواب سيكون بعد استعراض تطبيق للأمثلة على هذه الخاصية ثم  
المقارنة بإذن الله تعالى.. \*\_^ (انتبه جيدا للأمثلة، والفروقات)

# Background Image And Decorative Properties

مثال 1:

```
].ex-28 .ex-28-4{  
    background: url("web.png") no-repeat;  
    background-origin: border-box;  
}
```



Anees Hikmat Anees Abu-hamid 2nees.com

الدورات الأخبار مقالات روابط مهمة بعض أعمالني أنيس



صفحة 2

# Background Image And Decorative Properties

مثال 2:

```
.ex-28 .ex-28-5{  
    background: url("web.png") no-repeat;  
    background-origin: padding-box;  
}
```

# Background Image And Decorative Properties

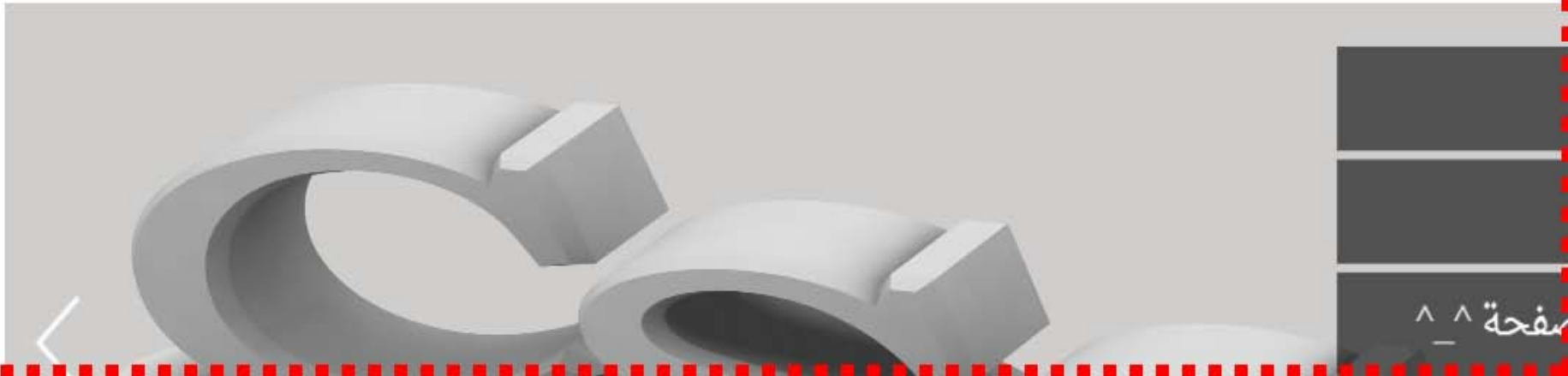
مثال 3:

```
.ex-28 .ex-28-6{  
    background: url("web.png") no-repeat;  
    background-origin: content-box;  
}
```

Anees Hikmat Anees Abu-hamid 2nees.com



الدورات الأخبار مقالات روابط مهمة بعض أعمالني أنيس



صفحة ^ ^

# Background Image And Decorative Properties

والآن يا صديقي هل أمعنت النظر الى الأمثلة .. إذن أرجوا منك الآن الذهاب الى أمثلة ال background-clip مرة أخرى وتلقي نظرة من جديد ثم عد الى هنا ^\_\*.

الفرق الجوهرى هو أن ال origin يقوم على الإهتمام بنقطة البدء للخلفية، وبنائاً على ذلك يقوم بوضع نقطة البدء ويكمل ..

بينما ال clip يهتم بنقطة النهاية مع مراعاة أنه يقوم بعمل اقتطاع للصورة اذا كانت أكبر من النطاق ...

الآن لنستعرض الفروقات بمثال سريع يوضح الفكرة، صورة بالف كلمة

P:

# Background Image And Decorative Properties



**border-box**



**padding-box**



**content-box**



الصادق جريء :

والصدق يدعو صاحبه للجرأة والشجاعة ؛ لأنه ثابت لا يتلون ، ولأنه واثق لا يتردد ، ولذلك جاء في أحد تعريفات الصدق : القول بالحق في مواطن الهلكة ، وعبر عن ذلك الجنيد بقوله : حقيقة الصدق : أن تصدق في موطن لا ينجيك منه إلا الكذب .

عن الصدق

## Border And Box Effects

منذ إصدار ال css1، لم يكن هناك تغييرات كثيرة على موضوع ال Borders، لأن الفكرة التي وجد من أجلها نفذت ولم يكن يحتاج المطورين لأكثر من ذلك، لكن في السنوات الأخيرة، أصبحت هناك حاجة لتطوير هذه الخاصية، فتنسيق صفحات الويب أصبح أكثر احترافية، ورغبة الزبائن أصبحت أكثر الحاحا في طلب التفاصيل، ومع ذلك عندما تشاهد بعض الأفكار البسيطة تكتشف أنها ستحتاج الى عمل برمجي معقد فعلا.. لذلك تم اضافة خصائص جديدة الى ال CSS3 بتواكب متطلبات العصر وبشكل سهل، منها ال shadow وال corner... الخ

والآن لنبدأ على بركة الله تعالى ^\_^



# Border And Box Effects

أولاً: Border radius

قبل إصدار ال css3 كنا إذا أردنا صنع curve فكنا نقوم بإنشاء صورة باستخدام ال photoshop مثلا، ونستخدم هذه الصورة في الموقع، لكن المواقع الجديدة أصبحت flex.. بحيث يجب أن يكون هناك وسيلة سهلة للتجاوب، فلا أستطيع القيام بهذا العمل لكل صورة على حدا في الموقع حتى أحافظ على النسق.. لذلك في ال css3 فقط بخاصية ال border-radius سينتهي كل الأمر وبشكل أجمل وأكثر احترافية \*\_ \*\_...

الصيغة العامة:

```
E { border-v-h-radius: x y; }
```

## Border And Box Effects

ال v: ترمز الإتجاه العامودي (فوق – تحت) أي (top – bottom).  
ال h: ترمز الى الإتجاه الأفقي (يمين – يسار) أي (left – right).  
أما ال x و ال y فهما الطول اللزام لعملية القطع والذي يعتمد لحساب ال  
curve .. بالإضافة الى أنه يمكنك مباشرة تحديد ال curve للإتجاهات  
الأربعة مثل border-radius وكما يمكنك استخدام أو وضع قيمة واحدة لل  
x وال y \* \_ ^ ...

والآن لنشاهد معا أمثلة متنوعة على الموضوع ^ \_ ^

ملاحظة: أكبر حجم يمكن أن يكون لل radius هو 50% من حجم العنصر

# Border And Box Effects

مثال 1

```
.ex-29 .ex-29-1{  
border-top-left-radius: 20px;  
border-top-right-radius: 20px;  
border-bottom-right-radius: 20px;  
border-bottom-left-radius: 20px;  
}
```



```
.ex-29 .ex-29-2{  
border-top-left-radius: 150px 20px;  
border-top-right-radius: 150px 50px;  
border-bottom-right-radius: 150px 50px;  
border-bottom-left-radius: 150px 20px;  
}
```



```
.ex-29 .ex-29-3{  
border-radius: 20px;  
}
```



# Border And Box Effects

مثال 2:

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

```
.ex-29 .ex-29-4{  
    border-radius: 200px 100px;  
}  
  
.ex-29 .ex-29-5{  
    border-radius: 50%;  
}  
  
.ex-29 .ex-29-6{  
    border-radius: 100px 50px 10px;  
}top-left & bottom-right top-right & bottom-left  
  
.ex-29 .ex-29-7{  
    border-radius: 30px 50px 70px 90px;  
}top-left & top-right & bottom-right & bottom-left;
```



# Border And Box Effects

ثانياً: Border Image

من الخصائص الجديدة أيضاً، والتي تمكننا من استخدام صورة كإطار بدلاً من الشكل التقليدي، هي خاصية ال Border Image... واضحة وسهلة ^\_^

الصيغة العامة: `E { border-image: source slice repeat; }`

ال source هو رابط الصورة المراد وضعها كإطار لعنصر معين.  
ال slice هي قيمة أو مجموعة من القيم (مثل ال border أو margin يعني ممكن أن نضع رقم واحد أو رقمين أو ثلاثة أو أربعة حسب المراد) ووظيفتها هو تحديد البعد أو المسافة لكل زاوية من زوايا الإطار.  
ال repeat وهي التي تستخدم لتكرار الصورة، وهنا يمكن وضع قيمة أخرى وهي القيمة الافتراضية stretch وتعني اجعل الصورة تساوي حجم الإطار..

# Border And Box Effects

```
.ex-30 .ex-30-1{  
  border-image: url('border.png') 30 stretch;  
  border-width: 30px;  
}
```

```
.ex-30 .ex-30-2{  
  border-image: url('border.png') 30 10 50 39 stretch;  
  border-width: 30px;  
}
```

```
.ex-30 .ex-30-3{  
  border-image: url('border.png') 28 repeat;  
  border-width: 50px;  
}
```

شاهد المثال: ثم إنتقل

للشريحة التالية ^\_\*:



Anees Hikmat Anees Abu-hamid 2nees.com



Anees Hikmat Anees Abu-hamid 2nees.com



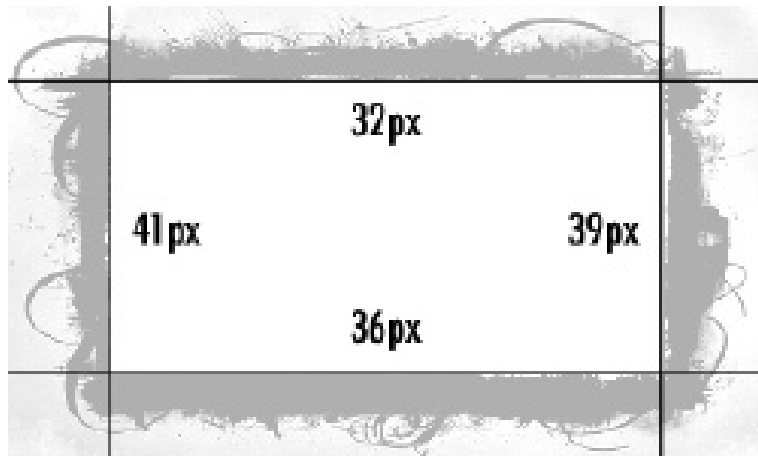
Anees Hikmat Anees Abu-hamid 2nees.com

# Border And Box Effects

إذا أمعنت النظر في المثال السابق ستجد الإطار بأشكال متعددة، هذا الإطار عبارة عن صورة، وصورة واحدة فقط  $^{\wedge}$   $^{\wedge}$ ، ويمكنني صناعة المزيد من الأشكال بناءً على هذه الصورة.. (قمت بأخذ الصورة من موقع w3schools) .. لكن ما أريد شرحه الآن .. هو لماذا اختلف الشكل بوجود ال slice حتى تتضح الفكرة .. أما ال width فقامت باستخدامها لجعل الإطار أكبر حجماً  $^{\wedge}$   $^{\wedge}$  ..

```
border-image: url('border.png') 32 39 36 41 stretch;
```

لاحظ أن ال 32 هي ال top .. وبهذا فإن كل رقم منهم يمثل مسافة بين الإطار والجهة المختارة  $^{\wedge}$  \* .. فقط هذا الأمر طبق المثل وشاهد النتائج بنفسك لتتضح الفكرة  $^{\wedge}$  \*



# Border And Box Effects

والآن سأكلفك بوظيفة بسيطة جدا .. ^\_\*

```
E {  
border-image-source: image;  
border-image-slice: length; /* Up to four values allowed */  
border-image-width: length; /* Up to four values allowed */  
border-image-outset: length; /* Up to four values allowed */  
border-image-repeat: keyword; /* Up to two values allowed */  
}
```

هذا الشكل يمثل جميع خصائص ال border-image والتي يمكننا استخدامها، ويوضع بال comment العدد الأقصى للقيم التي يمكن استخدامها .. (طبعا هنا لم نستخدم الطريقة المختصرة.. وهناك بعض المزايا الجديدة .. قم بالبحث عن المزايا التي لا تعرفها ثم طبق هذا المثال &\_&



# Border And Box Effects

ملاحظات مهمة قبل المتابعة \*\_\$\_:

(1) في بعض إصدارات ال firefox ظهرت مشكلة عند التحديثات وهي عدم ظهور ال border-image في التنسيق على متصفحات ال firefox وكان حل هذه المشكلة باستخدام ال border-style: solid; مع خاصية ال border-image

(2) أيضا ال firefox عندما يقوم بصناعة ال curve ل dotted أو dashed فإنه يقوم بتحويله الى solid في بعض المناطق .. وهذه bug مسجلة في ال firefox ولم تعالج حتى تاريخ كتابة هذه الكلمات ^\_^.

## Border And Box Effects

ثالثا وآخر جزئية بهذا الموضوع ^\_^ وهي خاصية ال box-shadow ^^ عندما تحدثنا سابقا عن النصوص وخصائصها، ذكرنا خاصية جميلة جدا وقتها، وهي خاصية ال text-shadow .. والآن على غرار هذه الخاصية، يوجد لدينا بال css3 خاصية ال box-shadow, هذه الخاصية تقوم على اعطاء ظل على مستوى عنصر معين ^\_^ (box-element) .

الصيغة العامة:

```
E { box-shadow: inset horizontal vertical blur spread color; }
```

إذا لاحظت فهي شبيهة بخصائص ال text-shadow \*\_\* ومع ذلك لنشرح الخصائص من جديد ونوضح الاختلافات في الشريحة التالية ^\_^

## Border And Box Effects

- (1) inset: وهو كلمة اختيارية لتحديد على الظل سيكون خارج ال box أم داخل ال box. (القيم إما inset أو outset وهي الحالة الافتراضية)
- (2) vertical & Horizontal وهما القيمتان التان تحددان المسافة بين ال box والظل ^\_\*.
- (3) Blur: وهي الخاصية المسؤولة عن مقدار التشتت أو الضبابية للظل ..
- (4) spread: وهي كزية تسمح بتغير حجم الظل، كما أن القيم السالبة مسموحة هنا، وتعني أصغر حجما من الحجم الأصلي للظل ^\_\*.
- (5) Color: وهي خاصية تحديد لون الظل، هذه الخاصية الزامية في ال webkit- واختيارية في ال firefox ويكون الافتراضي أسود، لكن أنت دائما قم بتحديد اللون لتعمل على جميع المتصفحات ^\_\*

# Border And Box Effects

والآن لنشاهد مثالا على هذه الخاصية (الشيفرة البرمجية في الشريحة التالية)

Anees Hikmat Anees Abu-hamid 2nees.com

Anees Hikmat Anees Abu-hamid 2nees.com

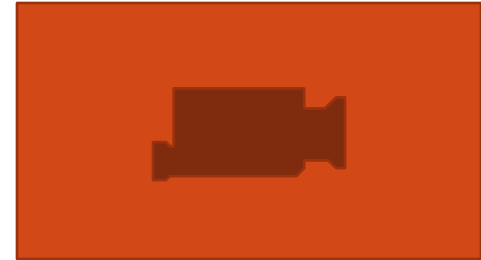
Anees Hikmat Anees Abu-hamid 2nees.com

Anees Hikmat Anees Abu-hamid 2nees.com

# Border And Box Effects

وهذه هي الشيفرة البرمجية:  
والآن شاهد المثال، وقم بالتجربة العملية بنفسك .. (لا تنسى أن التطبيق هو المهم)

```
/*box-shadow: inset(optional) horizontal vertical blur(optional) spread(optional) color; */
.ex-31 .ex-31-1{
  box-shadow: 50px 20px red;
}
.ex-31 .ex-31-2{
  box-shadow: 50px 20px 30px red;
}
.ex-31 .ex-31-3{
  box-shadow: inset 0px 0px 53px -1px green;
}
.ex-31 .ex-31-4{
  box-shadow: inset 0px 0px 53px -1px green, 18px 17px 46px red, -18px -17px 6px blue;
}
```



لاحظ هنا أننا قمنا بوضع أكثر من ظل  
مرة واحدة ^\_^



”إن الدين ليس بديلاً من العلم والحضارة. ولا عدواً للعلم والحضارة. إنما هو إطار للعلم والحضارة، ومحور للعلم والحضارة، ومنهج للعلم والحضارة في حدود إطاره ومحوره الذي يحكم كل شئون الحياة.“

سيد قطب رحمه الله

# Color And Opacity

إن الألوان بحد ذاتها، بحر من العلم يصعب حصره، ففيه من الجمال والإبداع والرونق ما يعطي جمالا لكل مخلوق بهذه الحياة والحمد لله، وبالعالمنا عالم الويب، فإن تناسق الألوان ودقتها لها الأثر الكبير في جذب المستخدمين، وكم يشعر المستخدم بالراحة أثناء نظره الى موقع، كم أنت مبدع، بالإضافة الى الألوان فإن ال opacity لأي عنصر قد تلعب دورا جماليا للموقع، كما تلعب دورا مهما الآن في عالم animation الخاص بالويب ^\_\*، لهذا سنتطرق بهذا الفصل بإذن الله تعالى الى موضوعين وهما:

الألوان وال opacity ...

# Color And Opacity

أولاً: ال Opacity

تمكننا هذه الخاصية الرائعة من التحكم بمقدار ظهور عنصر معين على شاشة العرض، وتكون القيم الخاصة به بين 0 وال 1 بحيث يكون عند النقطة 0 غير مرئي ويزداد ظهوراً كلما اقتربنا من ال 1

الصيغة العامة:

```
E { opacity: number; }
```

ال number: هو أي رقم بين 0.0 وال 1.0



# Color And Opacity

مثال 1: لاحظ نتائج تطبيق هذا المثال ..

المربع الأول والرابع 0.1 نسبة الوضوح لهم

لذلك تجد أنها شبه معدومة الرؤية .. ولاحظ أنه

كلما اقرب العدد من 1، فإن المربع يصبح أوضح

تدرجيا ^\_^

```
.ex-32 .ex-32-1{  
  opacity: 0.1;  
}
```

```
.ex-32 .ex-32-2{  
  opacity: 0.4;  
}
```

```
.ex-32 .ex-32-3{  
  opacity: 0.7;  
}
```

```
.ex-32 .ex-32-4{  
  opacity: 0.1;  
  background: url('2nees.png');
```

```
.ex-32 .ex-32-5{  
  opacity: 0.4;  
  background: url('2nees.png');
```

```
.ex-32 .ex-32-6{  
  opacity: 0.7;  
  background: url('2nees.png');
```

```
.ex-32 .ex-32-7{  
  width: 80%;  
  background: url('web.png');
```



Anees Hikmat Anees Abu-  
hamid 2nees.com



Anees Hikmat Anees Abu-  
hamid 2nees.com



Anees Hikmat Anees Abu-  
hamid 2nees.com



Anees Hikmat Anees Abu-  
hamid 2nees.com



Anees Hikmat Anees Abu-  
hamid 2nees.com



Anees Hikmat Anees Abu-  
hamid 2nees.com

# Color And Opacity

مثال 2: لاحظ هذا المثال الجميل، وانظر كيف أنني قمت بوضع خلفية وبقها خلفية، وبوجود ال opacity ظهرت وكأنها توقيع رقمي على الصورة .. ^\_^ \* .. إدماج دائما تعلمناه مع ما سنتعلمه بإذن الله ^\_^ ..  
لكن هنا في ملاحظة أريدك أن تبقىها بذهتك هو أن هذه الخاصية تؤثر على جميع ما بداخلها .. لذلك النص أيضا لن يكون شديد الوضوح ..

لاحظ هنا كيف قمت باستخدام هذه الخاصية مع ما تعلمناه سابقا ^\_^



# Color And Opacity

مم، ستقول اذا النص كان لا يقرأ فماذا سنستفيد، والجواب هو، الخاصية هذه لها استخدامتها، وليس بالضرورة في كل الأوقات أن يكون النص ظاهرا بنسبة 100% مباشرة .. فقد ينتج هذا عن طريق حدث معين أو animation معين .. وهكذا ..

كما أنه يمكننا التحايل على هذه المشكلة (جدلا) بعدة طرق ^\_^..منها استخدام ال position أو وضع النص في حاوي ومن ثم وضع الخلفية في حاوي بداخله واعطاء هذا الحاوي الداخلي المخصص للخلفية ال opacity.. واذا كانت الخلفية لون يمكن استخدام ال rgba أو استخدام صورة مفرغة وغيرها الكثير من الطرق التي يصعب احصائها ولكل منها حالة يأتي زمان علينا لنقوم باستخدامها ^\_^...والآن شاهد المثال: ^\_^



# Color And Opacity

ثانيا الألوآن .. في ال css3 بالإضافة لما هو موجود بال css2 فإن نظام الألوآن تطور بسبب استخدام(استحداث) ال Alpha وهي مقياس لمدى شفافية لون معين .. وبهذا نحن سنستغني عن ال opacity ونستخدم ال Alpha color .. لكن السؤال الآن .. هل الإسغناء كامل ؟

الجواب طبعا لا ، لأن هذه الخاصية سنستخدمها مع الألوآن فقط، أما ال opacity فهي تستخدم على نطاق حاوي .. لذلك انتبه أن تخط .. وستتضح الأمور مع الأمثلة بإذن الله تعالى .. الموضوع سهل جدا..

الآن هل تذكر بالشريحة السابقة قلنا عن ال rgba ؟!

ال (red, green, blue) نعرفها ... لكن ما هذه ال a في rgba ؟!

# Color And Opacity

هذه الـ  $a$  هي اختصار لـ  $\alpha$  والتي من خلالها نستطيع تحديد نسبة الوضوح للون معين  $^{\wedge}_{\wedge}$ ، ويجب أن تكون القيم الخاصة بها من 0 الى 1  $^{\wedge}_{\wedge}$ .

شاهد الصيغة العامة:

```
E { color: rgba(red, green, blue, alpha); }
```

مثال على القيم:

```
color: rgba(255, 0, 0, 0.5);
```

# Color And Opacity

مثال 1: هذا المثال يوضح الفكرة، لكن أثناء تطبيقي للمثال، خطرت ببالي فكرة لم يكن بخاطري الحديث عنها .. شاهد المثال ونسبة الوضوح،

ولاحظ أنه في حال أعطيت لون

الخط خاصية alpha باستخدام ال

Rgba فإن ال border يتأثر أيضا

بنفس المعيار، الا اذا قمت بتحديد

اللون لل border ...

```
.ex-33 .ex-33-1 {  
  color: rgba(255,0,0,0.5);  
}
```

```
.ex-33 .ex-33-2 {  
  color: rgba(255,255,0,0.3);  
  border-color: green;  
}
```

# Color And Opacity

مثال 2: هذا المثال يقارن بين استخدام ال

Opacity واستخدام ال rgba

```
.ex-33 .ex-33-3{  
  color: rgba(0,0,0,0.5);  
  border-color: green;  
  background: url('2nees.png');  
}
```

```
.ex-33 .ex-33-4{  
  opacity: 0.5;  
  color: rgb(0,0,0);  
  border-color: green;  
  background: url('2nees.png');  
}
```

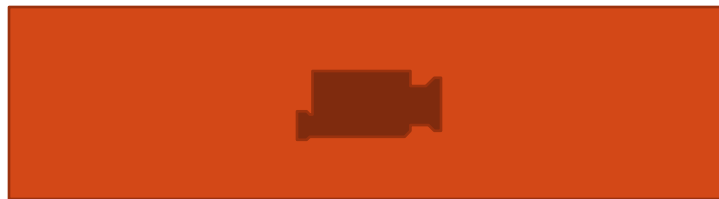


# Color And Opacity

مثال 3: هذا المثال يوضح كيف يمكن دمج ما تعلمناه سابقا واستخدام خاصية اللون الجديدة `rgba` .. `^_^` طبعاً سأترك لك المثال بدون تطبيق لتقوم أنت بتنفيذه `^_^`

```
.shadow .text { box-shadow: 10px 10px 4px rgba(0,0,0,0.7); }  
.border .text { border: 10px solid rgba(0,0,0,0.5); }  
.text-semi p { color: rgba(0,0,0,0.6); }  
.text-shadow p { text-shadow: 5px 5px 1px rgba(0,0,0,0.6); }
```

والآن لنشاهد تطبيق الأمثلة السابقة (من غير هذا المثال (P):





# Color And Opacity

ملاحظة قبل مغادرة هذا الفصل ^\_^..

هناك خاصية جديدة بال css3 تدعى ب appearance هذه الخاصية تقوم على اعطاء عنصر معين تنسيق الشكل الافتراضي لعنصر ما كما يقرأ من عند المتصفح .. طبعا هذه الخاصية غير مدعومة من ال IE وال opera mini... وتعمل على المتصفحات الرئيسية مع -webkit- وال -moz-

```
<div class="ex-34-1">  
  Anees Hikmat Anees Abu-hamid  
  2nees.com  
</div>
```

Anees Hikmat Anees  
Abu-hamid 2nees.com



شاهد المثال:

```
].ex-34 .ex-34-1 {  
  -webkit-appearance: button;  
  -webkit-appearance: button;  
  appearance: button;  
- }
```

”إن الدين ليس بديلاً من العلم والحضارة. ولا عدواً للعلم والحضارة. إنما هو إطار للعلم والحضارة، ومحور للعلم والحضارة، ومنهج للعلم والحضارة في حدود إطاره ومحوره الذي يحكم كل شئون الحياة.“

سيد قطب رحمه الله

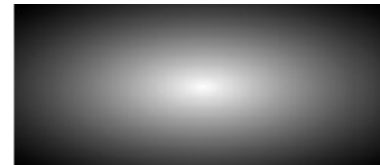
# Gradient

ال Gradient تعني التدرج، وبال css هي انتقال تدريجي بين مجموعة من الألوان (على الأقل لونين إثنين) ويمكن تصنيفهما الى نوعين:

(1) Liner Gradient: ويقصد به التدرج الذي يتكون بين نقطتين اثنتين، ويكون مستوى التدرج مختلفا اعتمادا على المسافة بين النقطتين بحيث يبقى التدرج متناسبا على طول الخط الذي تم تشكيله..



(2) Radial Gradients: وهو يمثل تدرج على بين لونين اعتمادا على نقطة تمثل المركز لهذا التدرج، وبأبسط صورته، يكون هذا التدرج على شكل دائري أو بيضوي.. ويسمى أيضا هذا النوع بالتدرج الشعاشي P: ..



# Gradient

ملاحظة: في هذه الخاصية يجب عليك مراعاة المتصفحات التي تدعم هذه الخاصية بالإضافة الى طريقة دعمها ..شاهد الجدول قبل البدء:  
هذه الصورة أخذت من موقع w3schools في تاريخ 05-04-2015 \*  
وسأعتمد بالشرح الأسلوب الخاص بهم أيضا لهذا الموضوع فهو سهل

^ ^  
\_

Property					
linear-gradient	10.0	26.0 10.0 -webkit-	16.0 3.6 -moz-	6.1 5.1 -webkit-	12.1 11.1 -o-
radial-gradient	10.0	26.0 10.0 -webkit-	16.0 3.6 -moz-	6.1 5.1 -webkit-	12.1 11.6 -o-
repeating-linear-gradient	10.0	26.0 10.0 -webkit-	16.0 3.6 -moz-	6.1 5.1 -webkit-	12.1 11.1 -o-
repeating-radial-gradient	10.0	26.0 10.0 -webkit-	16.0 3.6 -moz-	6.1 5.1 -webkit-	12.1 11.6 -o-

# Gradient

والآن.. لنبدأ بالحديث عن أول نوع وهو التدرج الخطي  $\wedge\_ \wedge$  (Liner Gradient): يمكن تحديد الإتجاه للتدرج لهذا النوع بأحد الآتي  $\* \wedge$  down/up/left/right/diagonally

الصيغة العامة:

```
background: linear-gradient(direction, color-stop1, color-stop2, ...);
```

ال direction هو الإتجاه الذي سيسر به التدرج ..

ال color-stop1 هو اللون الأول ..

ال color-stop2 هو اللون الثاني ...

وهما أقل عدد من الألوان ممكن ..ويمكنك اضافة أكثر من ذلك  $\wedge\_ \wedge$

# Gradient

مثال 1: لاحظ المثال، لم يتم هنا تحديد الإتجاه، والسبب أن الإتجاه

الإفتراضي هو من Top to Bottom

```
.ex-35 .ex-35-1{  
  background: -webkit-linear-gradient(black, white); /* For Safari 5.1 to 6.0 */  
  background: -o-linear-gradient(black, white); /* For Opera 11.1 to 12.0 */  
  background: -moz-linear-gradient(black, white); /* For Firefox 3.6 to 15 */  
  background: linear-gradient(black, white); /* Standard syntax */  
}
```

النتائج:

Anees Hikmat Anees Abu-hamid 2nees.com

# Gradient

مثال 2: لاحظ جيدا هذا المثال، في هذا المثال نريد أن نبدأ التدرج من اليسار، لكن هناك ملاحظات مهمة وهي:

أ) ال webkit ونستخدمه لل safari خصوصا فإنه يتم تحديد البداية ب left -حسب هذا المثال- بينما يتم تحديد المتصفحات الأخرى أو الطريقة القياسية بالنقطة التي سنذهب لها ..فتكون right... (من اليسار اذن الإتجاه سيكون لليمين)

ب) أيضا فإن الصيغة القياسية يضاف له ..to (to right).

```
.ex-35-2 {
```

```
background: -webkit-linear-gradient(left, black, white); /* For Safari 5.1 to 6.0 */
background: -o-linear-gradient(right, black, white); /* For Opera 11.1 to 12.0 */
background: -moz-linear-gradient(right, black, white); /* For Firefox 3.6 to 15.0 */
background: linear-gradient(to right, black, white); /* Standard syntax */
```

# Gradient

مثال 3: في هذا المثال سنستعرض كيف يمكننا التدرج بين نقطتان تمثلان المحور السيني (أفقي) والمحور الصادي (عامودي) ..وكما انطبق على المثال السابق ينطبق هنا .. top عكسها bottom ^\* \_  
هذا المثال سيكون لإستعراض التدرج من أعلى اليسار (top left) وهذا يعني (Bottom right) للمتصفحات ما عدى ال webkit ^\* \_

```
.ex-35 .ex-35-3 {  
  background: -webkit-linear-gradient(left top, black , white); /* For Saf  
  background: -o-linear-gradient(bottom right, black, white); /* For Opera  
  background: -moz-linear-gradient(bottom right, black, white); /* For Fir  
  background: linear-gradient(to bottom right, black , white); /* Standard  
- }
```



# Gradient

مثال 4: هنا في هذا المثال سنقوم بوضع الزاوية (angle)، وتكون هذه الزاوية بين نقطتين، الخط الأفقي مع خد التدرج  $^*_$  وبهذا تتشكل لنا زاوية، ولتبسيط الأمر فإن الزاوية  $0deg$  تمثل ال bottom to top وال  $90deg$  تمثل ال left to right وال  $180deg$  تمثل top to bottom وال  $90deg$  تمثل ال right to left  $^*_$ ..ومشاهدة المثال خير من الف كلمة

:^ \_ ^

```
.ex-35 .ex-35-6 {  
  background: -webkit-linear-gradient(90deg, black , white); /* I  
  background: -o-linear-gradient(90deg, black, white); /* For Ope  
  background: -moz-linear-gradient(90deg, black, white); /* For I  
  background: linear-gradient(90deg, black , white); /* Standard  
- }
```

# Gradient

مثال 5: والآن لنشاهد مثالا على كيفية انشاء تدرج بأكثر من لون `^_^`، والطريقة بكل بساطة تكون عن طريق اضافة فاصلة ومن ثم لون جديد، ويمكن طبعا استخدام ال `rgba` مثلا... أو صيغ اللون المختلفة كقيم `*_^`

```
.ex-35 .ex-35-10{  
background: -webkit-linear-gradient(45deg, red , green, blue, orange, yellow);  
background: -o-linear-gradient(45deg, red , green, blue, orange, yellow); /* B  
background: -moz-linear-gradient(45deg, red , green, blue, orange, yellow); /*  
background: linear-gradient(45deg, red , green, blue, orange, yellow); /* Stan  
}
```



Anees Hikmat Anees Abu-  
hamid 2nees.com

# Gradient

مثال 6: الآن ما رأيك بأن نتلاعب بالخصائص التي تعلمناها معا `^_*`، لنقوم الآن بعمل تدرج، لكن هذا التدرج سيكون لنفس اللون من الدرجة الفاتحة الى الأعمق `^_*`.. طبعا في هذا المثال سنقوم باستخدام ال `rgba` حتى نحصل على مقدار وضوح بين `0 - 1`، وكما تلاحظ فإن الشفافية تتدرج من اللون الأبيض الى اللون المطلوب `^_^`

```
.ex-35 .ex-35-12 {
```

```
background: -webkit-linear-gradient(90deg, rgba(255,0,0,0), rgba(255,0,0,1));  
background: -o-linear-gradient(90deg, rgba(255,0,0,0), rgba(255,0,0,1)); /* For  
background: -moz-linear-gradient(90deg, rgba(255,0,0,0), rgba(255,0,0,1)); /* B  
background: linear-gradient(90deg, rgba(255,0,0,0), rgba(255,0,0,1)); /* Stand
```

Anees Hikmat Anees Abu-  
hamid 2nees.com

# Gradient

مثال 7: لاحظ في هذا المثال التدرج لأكثر من لون، وبالتحكم أيضا بمقدار الوضوح الخاص بكل لون أثناء التدرج  $^*_$

```
.ex-35 .ex-35-13{
```

```
background: -webkit-linear-gradient(90deg, rgba(255,0,0,0), rgba(170,55,100,0.4), rgba(255,255,0,0.7), rgba(255,0,0,1));  
background: -o-linear-gradient(90deg, rgba(255,0,0,0), rgba(170,55,100,0.4), rgba(255,255,0,0.7), rgba(255,0,0,1)); /* F  
background: -moz-linear-gradient(90deg, rgba(255,0,0,0), rgba(170,55,100,0.4), rgba(255,255,0,0.7), rgba(255,0,0,1)); /*  
background: linear-gradient(90deg, rgba(255,0,0,0), rgba(170,55,100,0.4), rgba(255,255,0,0.7), rgba(255,0,0,1)); /* Stan
```

Anees Hikmat Anees Abu-  
hamid 2nees.com

# Gradient

مثال 8: الآن، وآخر مثال ونقطة للجزء الأول من هذه الخاصية وهو التكرار، هل خطر ببالك أن تقوم بتكرار التدرج عند نقطة معينة؟! طبعا يمكنك فعل ذلك باستخدام الأمر repeating-linear-gradient أي بإضافة كلمة repeating قبل ال linear-gradient ..^\* شاهد المثال:

```
.ex-35 .ex-35-14{  
  /* Safari 5.1 to 6.0 */  
  background: -webkit-repeating-linear-gradient(143deg, red 32px, yellow 44px, green 51px);  
  /* Opera 11.1 to 12.0 */  
  background: -o-repeating-linear-gradient(143deg, red 32px, yellow 44px, green 51px);  
  /* Firefox 3.6 to 15 */  
  background: -moz-repeating-linear-gradient(143deg, red 32px, yellow 44px, green 51px);  
  /* Standard syntax */  
  background: repeating-linear-gradient(143deg, red 32px, yellow 44px, green 51px);  
-}
```



Anees Hikmat Anees Abu-  
hamid 2nees.com

# Gradient

شاهد نتائج تطبيق الأمثلة وغيرها ^\_\* ثم انتقل الى الشريحة التالية:

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com 0deg

Anees Hikmat Anees Abu-  
hamid 2nees.com 45deg

Anees Hikmat Anees Abu-  
hamid 2nees.com 90deg

Anees Hikmat Anees Abu-  
hamid 2nees.com -90deg

Anees Hikmat Anees Abu-  
hamid 2nees.com 180deg

Anees Hikmat Anees Abu-  
hamid 2nees.com -135deg

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

# Gradient

الآن وقبل أن تذهب لمشاهدة الشيفرة البرمجية للأمثلة السابقة والأمثلة التي لم نذكرها .. أريد أن أخبرك بأمر جميل، يمكنك من خلال بعض مواقع الإنترنت انشاء التدرج الذي تريده بشكل سهل، كل ما عليك القيام به هو التحكم بالواجهات التي أمامك، لكن هذا الشرح مهم جدا، ولم نعتمد على هذه الأداة في الشرح، وذلك لسببان، الأول كيف يمكنك أن تستخدم شيفرة برمجية لا تستطيع التعديل عليها أو معرفة القيم والإحتمالات الممكنة أو الصيغة الخاصة بها، والثاني هو لتوضيح أنه يمكنك بناء التدرج الذي تريده فقط بمعرفتك هذه الأمور وبالتطوير من مهاراتك مع قليل من التأمل .. فقد تحتاج في المستقبل وإن استخدمت الأدوات الموجودة الى تعديل، قد لا تستطيع صنعه مع هذه الأدوات، وهذا قد يحصل في كثير من المشاريع .. لذلك وجب التنبيه، والآن من هنا يمكنك مشاهدة احدى المواقع ومشاهدة المثال



قد يجمع المالَ شخصاً ثم يحرمه ... عما قليل فيلقى الذل والحرب  
العلم كنز وذخر لا فناء له ... نعم القرين إذا ما صاحب الصحبا



# Gradient

والآن سنتحدث بإذن الله تعالى عن النوع الثاني وهو Radial Gradients

لنشاهد أولاً الصيغة العامة:

```
background: radial-gradient(shape size at position, start-color, ..., last-color);
```

لاحظ هنا أنني أستخدم ال radial-gradient بدلاً من ال liner .. ^\_\*  
ال shape هو الشكل الذي سيستخدم لتركيب التدرج هل هو بيضوي (الحالة  
الإفتراضية ellipse) أم دائري (circle).

ال size هو الحجم الخاص بالتدرج ويمكن أن تستخدم مع 4 كلمات  
محجوزة وهي closest-side، farthest-side، closest-corner، ....

# Gradient

farthest-corner وهي الخاصية الافتراضية للحجم...  
ال at position هي الموقع الذي سيحتوي نقطة المركز الخاصة بالتدرج،  
وبالحالة الافتراضية هي center .. ^\*  
ال start color هو أول للون .. (أي اللون الأقرب لنقطة المركز، وكلما  
ابتعد اللون بالترتيب، ابتعد اللون عن مركز التدرج .. ^\* شاهد المثال 1:

```
.ex-36 .ex-36-1{  
background: -webkit-radial-gradient(red, green, blue); /* Safari 5.1 to 6.0 */  
background: -o-radial-gradient(red, green, blue); /* Opera 11.1 to 11.5 */  
background: -moz-radial-gradient(red, green, blue); /* Firefox 3.6 to 15 */  
background: radial-gradient(red, green, blue); /* Standard */  
}
```

Anees Hikmat Anees Abu-  
hamid 2nees.com

هذا الشكل القياسي لهذا النوع من التدرج.. لنشاهد المزيد ^\*

# Gradient

مثال 2: يمكننا التلاعب بالأبعاد الخاص بكل اشعاع (لون) وذلك عن طريق اعطاء قيمة بعد اللون <sup>^</sup>\_\*

```
.ex-36 .ex-36-2 {  
  background: -webkit-radial-gradient(red 30%, green, blue);  
  background: -o-radial-gradient(red 30%, green, blue); /* Fc  
  background: -moz-radial-gradient(red 30%, green, blue); /*  
  background: radial-gradient(red 30%, green, blue); /* Stand  
}
```



Anees Hikmat Anees Abu-  
hamid 2nees.com

# Gradient

مثال 3: والآن لنقم بتغيير الشكل الافتراضي من البيضوي الى الشكل الدائري ^\*:

```
.ex-36 .ex-36-3{  
  background: -webkit-radial-gradient(circle, red, green, blue); /*  
  background: -o-radial-gradient(circle, red, green, blue); /* For C  
  background: -moz-radial-gradient(circle, red, green, blue); /* For  
  background: radial-gradient(circle, red, green, blue); /* Standard  
}
```

Anees Hikmat Anees Abu-  
hamid 2nees.com

# Gradient

مثال 4: والآن شاهد كيف يمكننا التلاعب بالموقع والحجم الخاص بهذا

النوع من التدرج

```
.ex-36 .ex-36-4{  
  background: -webkit-radial-gradient(closest-side at 80% 30%, red, green, blue); /*  
  background: -o-radial-gradient(closest-side at 80% 30%, red, green, blue); /* For C  
  background: -moz-radial-gradient(closest-side at 80% 30%, red, green, blue); /* For  
  background: radial-gradient(closest-side at 80% 30%, red, green, blue); /* Standard  
}
```



```
.ex-36 .ex-36-5{  
  background: -webkit-radial-gradient(farthest-side at 80% 30%, red, green, blue); /*  
  background: -o-radial-gradient(farthest-side at 80% 30%, red, green, blue); /* For  
  background: -moz-radial-gradient(farthest-side at 80% 30%, red, green, blue); /* Fc  
  background: radial-gradient(farthest-side at 80% 30%, red, green, blue); /* Standar  
}
```

```
.ex-36 .ex-36-6{  
  background: -webkit-radial-gradient(closest-corner at 80% 30%, red, green, blue); /*  
  background: -o-radial-gradient(closest-corner at 80% 30%, red, green, blue); /* For  
  background: -moz-radial-gradient(closest-corner at 80% 30%, red, green, blue); /* F  
  background: radial-gradient(closest-corner at 80% 30%, red, green, blue); /* Standa  
}
```

```
.ex-36 .ex-36-7{  
  background: -webkit-radial-gradient(farthest-corner at 80% 30%, red, green, blue);  
  background: -o-radial-gradient(farthest-corner at 80% 30%, red, green, blue); /* Fc  
  background: -moz-radial-gradient(farthest-corner at 80% 30%, red, green, blue); /*  
  background: radial-gradient(farthest-corner at 80% 30%, red, green, blue); /* Stand  
}
```

# Gradient

مثال 4: نتيجة المثال:

Anees Hikmat Anees Abu-hamid  
2nees.com

Anees Hikmat Anees Abu-hamid  
2nees.com

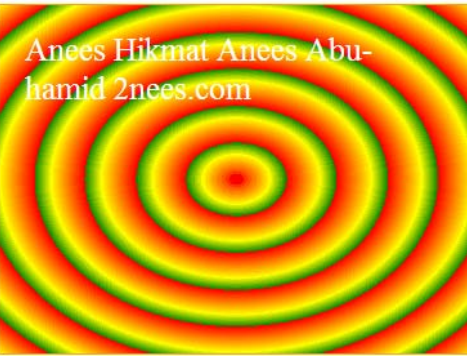
Anees Hikmat Anees Abu-hamid  
2nees.com

Anees Hikmat Anees Abu-hamid  
2nees.com

# Gradient

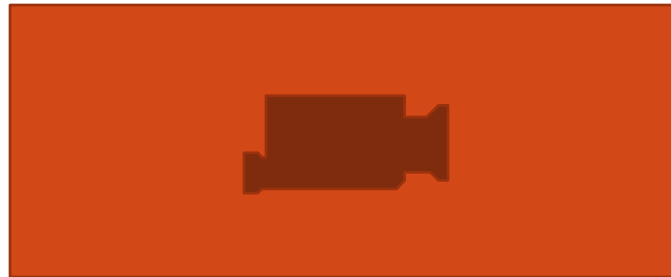
مثال 5: والآن آخر جزئية بهذا الموضوع وهي الخاصة بالتكرار `*_^`،  
وكما أضفنا خاصية التكرار على `lienr` سنقوم بإضافتها هنا وذلك أيضا  
وبكل بساطة من خلال `repeating-radial-gradient*_^...`  
لنشاهد المثال ال 5 والأخير بإذن الله تعالى لهذا الموضوع `*_^`:

```
.ex-36 .ex-36-9{  
  /* For Safari 5.1 to 6.0 */  
  background: -webkit-repeating-radial-gradient(red, yellow 10%, green 15%);  
  /* For Opera 11.6 to 12.0 */  
  background: -o-repeating-radial-gradient(red, yellow 10%, green 15%);  
  /* For Firefox 3.6 to 15 */  
  background: -moz-repeating-radial-gradient(red, yellow 10%, green 15%);  
  /* Standard syntax */  
  background: repeating-radial-gradient(red, yellow 10%, green 15%);  
}
```



# Gradient

والآن بحمد الله تعالى، نكون انتهينا من هذا الفصل والجزء المثير ^\_^،  
والآن لننطلق لموضوع أكثر تشويقاً وإثارة من هذا الفصل، الفصل الذي  
يحببه المصممين والمطورين والأشخاص الذي سيقومون بالإطلاع على  
موقعك .. ^\_^، إن آثار فضولك ما هو هذا الموضوع ^\_^، سأخبرك به  
بإذن الله تعالى بعد أن تطلع على الأمثلة التي ذكرناها مع الشيفرة البرمجية،  
وانظر الى الأمثلة التي نذكرها ونتائجها ^\_^\* ...





مما جاء في الثناء قبل الدعاء قول النبي صل الله عليه وسلم:

اللهم ربنا لك الحمد أنت قيم السموات والأرض ومن فيهن، ولك الحمد أنت رب السموات والأرض ومن فيهن، أنت الحق وقولك الحق ووعدك الحق ولقاؤك الحق

متفق عليه

# 2D Transformations

في عصر الويب القديم  $\_ \wedge$ ، كان ال HTML يمثل مجموعة من الخطوط العمودية والأفقية، والتي تشكل مساحات معينة وتحجزها على شكل block أو boxes .. وكان الأسلوب الذي يمكن أن نخرج عن هذا النمط أو بالأحرى نحسن أو نلعب بترتيب هذا الشكل عن طريق استخدام الصور، لكن هذا الكلام لم يعد كافيا أيضا، نحتاج الى ميزات جديدة تمكننا من التعامل مع هذه ال blocks المختلفة، وفعلا ذلك ما كان  $\_ \wedge$ ، فنذ عام 2008 أطلقت ال webkit مودبول جديد يسمح باستخدام ميزات جديدة ورائعة مثل rotate وال resized .. الخ، وهذه الخصائص هي ما سنتحدث عنه بإذن الله تعالى في هذا الفصل ..

بالنسبة Transformations فهي تعني تغيير الهيئة من شكل الى آخر  $\_ \wedge$  \* أيضا .. هذه الخاصية مدعومة جميع المتصفحات الرئيسية  $\_ \wedge$  \* باستثناء ال IE8

# 2D Transformations

والآن .. لنبدأ على بركة الله تعالى بهذا الموضوع الجديد، والذي يعد من أجمل الإضافات الجديدة على عالم الويب  $^_^$  ( طلعت كل الخصائص أجمل من بعض P: )

ال Transformations يمكنك من عمل تغيير على حجم العنصر أو/ و تغيير شكله أو/ و موقعه .. (وطبعا هناك نوعان من هذا التأثير إما 3D وإما 2D والذي سنتحدث عنه الآن هو 2D).

E {

```
-moz-transform: function(value); /* Firefox */  
-ms-transform: function(value); /* Internet Explorer */  
-o-transform: function(value); /* Opera */  
-webkit-transform: function(value); /* WebKit */  
transform: function(value);
```

الصيغة العامة:

}

# 2D Transformations

كما تلاحظ من الصيغة العامة، فإن ال vendor الخاصة بالمتصفحات تم استخدامها مثل ال webkit وال ms وال o... الخ، هذه ال vendor تختلف من إصدار الى إصدار، فمثلا في الإصدارات الجديدة للمتصفحات لم نعد بحاجة الى استخدام هذه ال vendor.. لكننا نبقى عليها للمستخدمين الذين لم يقوموا بتحديث متصفحاتهم ..-\_- ولا تنسى دائما استخدام caniuse.com

شرح الصيغة العامة:

transform هي الخاصية التي سنستخدمها للقيام بالتأثير على عنصر معين وهي التي تمثل موضوع هذا الجزء من الدورة ^\_\*

Function(value): وتمثل هذه اسم الدالة التي سنستخدمها مع القيم التي

نرغب بإرسالها ^\_\*

# 2D Transformations

والآن لنبدأ على بركة الله تعالى في أول دالة لهذه الخاصية وهي ال rotate (1): هذه الدالة الرائعة تمكنني من تغيير زاوية العرض، أي تغيير طريقة العرض الأفقية أو العمودية فقط .. فأصبح بإمكاننا أن نجعل العنصر يعرض بشكل مائل 45 درجة أو مقلوب 180 درجة .. الخ نستخدم ال deg لتحديد الدرجة ^\_\* .. والآن لنشاهد مثالا ^\_\*

Anees Hikmat Anees Abu-  
hamid 2nees.com

```
.ex-37 .ex-37-1{  
  opacity: 0.5;  
}
```

ملاحظة: ال div الذي يحتوي الكلاس ex-37-1 يمثل الشكل الإعتيادي لل div .. فقط لتوضيح المثال ^\_\*

```
.ex-37 .ex-37-1-1{  
  -ms-transform: rotate(45deg); /* IE 9 */  
  -webkit-transform: rotate(45deg); /* webkit */  
  -moz-transform: rotate(45deg); /* Firefox */  
  transform: rotate(45deg);  
}
```

Anees Hikmat Anees Abu-  
hamid 2nees.com

# 2D Transformations

يمكنك طبعا استخدام الإشارة السالبة أيضا، كما تحدثنا سابقا في موضوع التدرج (Gradient) P: ...

الآن سأحدث عن جزئية مهمة يجب أن تعرفها وأن تأخذها بعين الاعتبار أثناء عملك على ال transform.. وهو الموقع الخاص بهذا ال transform بالنسبة للعناصر الأخرى في الملف.. والسبب في ذلك يعود الى إهمال هذه الخاصية لل page layout.. وما تقوم به هذه الخاصية فعلا.. هو إنشاء layer جديدة.. لهذا ستلاحظ أن بعض العناصر يمكن أن تتأثر من جراء استخدام هذه الخاصية، فتقوم هذه ال layer على اخفاء عنصر موجود.. شاهد المثال في الشريحة التالية ٨\_\*

# 2D Transformations

```
.ex-37 .ex-37-2{  
  -ms-transform: rotate(-90deg); /* IE 9 */  
  -webkit-transform: rotate(-90deg); /* webkit */  
  -moz-transform: rotate(-90deg); /* Firefox */  
  transform: rotate(-90deg);  
}
```

مثال 2:

حقيقة في خاصية كهذه  
فإنها تقوم على جعل  
العنصر ذو موقع  
Relative .. والنقطة أو  
الموقع الأصلي يحتفظ به  
حسب تسلسل العنصر في  
الملف .. لذلك لو قمت بإطفاء  
هذه الخاصية .. فإنه سيكون  
في موقعه الطبيعي ^\_\*  
شاهد الصورة في الشريحة  
التالية ..

Anees Hikmat Anees Abu-  
hamid 2nees.com

aneeshikmat@

\* I Like a css ^\_\*

# 2D Transformations

مثال 3:

Anees Hikmat Anees Abu-  
hamid 2nees.com

aneeshikmat@2nees.com \* I Like a css ^\_\*

```
Q Elements Network Sources Timeline Profiles Resources Audits Console
Styles Computed Event Listeners DOM Breakpoints Properties
element.style {
}
.ex-37 .ex-37-2 {
  -ms transform: rotate(-90deg);
  webkit transform: rotate(-90deg);
  moz transform: rotate(-90deg);
  transform: rotate(-90deg);
}
.ex-37 div {
  padding: 20px;
  margin: 20px 20px 80px;
  font-size: 25px;
  color: blue;
  width: 300px;
  height: 50px;
  background: black;
```



# 2D Transformations

ولحل هذه المشكلة .. يمكنك اتخاذ أكثر من أسلوب، لكن الأسلوب الذي يعالج هذه المشكلة بطريقة توافق هذه الخاصية، هو عن طريق تغيير النقطة التي تمثل المركز لهذا العنصر.. أي سنستخدم الخاصية transform-origin .. في هذه الخاصية يمكنني تحديد الموقع بإحدى الكلمات التالية left, right, top, bottom, center أو يمكنني استخدام ال px, em, % .. ويمكن كتابة قيمة واحدة أو قيمتين لهذه الخاصية .. القيمة الواحدة .. تعني اجعل النقطة للخط الأفقي والعامودي يساوي هذه القيمة .. أما اذا قمت باستخدام قيمتين .. فالأولى تكون للخط الأفقي .. والثانية للعامودي ..

والآن لنشاهد مثالا نرى به كيف قمنا بحل المشكلة باستخدام هذه الخاصية

# 2D Transformations

مثال 4: لاحظ الآن كيف أنني قمت بالتحكم في موقع الخاص transform بتحديد top .. وهذه العملية .. حلت المشكلة ^\_\* ويمكن الإستفادة منها أيضا أكثر من ذلك مع الخصائص الأخرى ..

```
.ex-37 .ex-37-3 {  
  -ms-transform: rotate(-90deg); /* IE  
  -webkit-transform: rotate(-90deg);  
  -moz-transform: rotate(-90deg); /*  
  transform: rotate(-90deg);
```

```
-ms-transform-origin: top;  
-webkit-transform-origin: top;  
-moz-transform-origin: top;  
transform-origin: top;
```

Anees Hikmat Anees Abu-  
hamid 2nees.com

# 2D Transformations

مثال 5: (1 هي الحالة الافتراضية أما 2 و 3:

```
:x-37 .ex-37-5-2{  
  -ms-transform-origin: left top;  
  -webkit-transform-origin: left top; الحالة 2  
  -moz-transform-origin: left top;  
  transform-origin: left top;
```

```
:x-37 .ex-37-5-3{  
  -ms-transform-origin: 270% 20em;  
  -webkit-transform-origin: 270% 20em;  
  -moz-transform-origin: 270% 20em; الحالة 3  
  transform-origin: 270% 20em;
```

Come With Me  
To Learn CSS3

Come With Me  
To Learn CSS3

Come With Me  
To Learn CSS3

الحالة 2

Come With Me  
To Learn CSS3

Come  
To Learn

Come With Me  
To Learn CSS3

الحالة 3



أَعُوذُ بِاللَّهِ مِنَ الشَّيْطَانِ الرَّجِيمِ

فَبِمَا رَحْمَةٍ مِّنَ اللَّهِ لِنْتَ لَهُمْ وَلَوْ كُنْتَ فَظًّا غَلِيظَ الْقَلْبِ لَانفَضُّوا مِنْ حَوْلِكَ فَاعْفُ عَنْهُمْ وَاسْتَغْفِرْ لَهُمْ وَشَاوِرْهُمْ فِي الْأَمْرِ فَإِذَا عَزَمْتَ فَتَوَكَّلْ عَلَى اللَّهِ إِنَّ اللَّهَ يُحِبُّ الْمُتَوَكِّلِينَ (159)

سورة ال عمران – الآية 159

# 2D Transformations

الدالة الثانية  $^{\wedge}_{\wedge}$ :

translate (2): تقوم هذه الخاصية الجميلة على تغيير موقع العنصر من إحداثي إلى إحداثي آخر... (هنا نتحدث عن position وليس نقطة origin)

حقيقة هذه الدالة يمكن تقسيمها إلى أجزاء وهي

translate و translateY و translateX  $^{\wedge}_{\wedge}$  بحيث x المحور الأفقي و

y المحور العامودي.. وال translate هي للمحور الأفقي والعامودي...

الصيغة العامة:

```
]E {  
    transform: translateX(value);  
    transform: translateY(value);  
-}  
]E {  
    transform: translate(translateX, translateY);  
-}
```

# 2D Transformations

مثال 1:



Anees Hikmat Anees Abu-  
hamid 2nees.com

```
.ex-38 .ex-38-1{  
  opacity: 0.5;  
}  
  
.ex-38 .ex-38-1-1{  
  -ms-transform: translate(35px,-70px); /* IE 9 */  
  -webkit-transform: translate(35px,-70px); /* webkit */  
  -moz-transform: translate(35px,-70px); /* Firefox */  
  transform: translate(35px,-70px);  
}
```

# 2D Transformations

مثال 2:



Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

**المحور السيني**  
:x-38 .ex-38-2 {  
-ms-transform: translateX(35px); /\*  
-webkit-transform: translateX(35px);  
-moz-transform: translateX(35px); /\*  
transform: translateX(35px);

**المحور الصادي**  
:x-38 .ex-38-3 {  
-ms-transform: translateY(-70px); /\*  
-webkit-transform: translateY(-70px);  
-moz-transform: translateY(-70px); /\*  
transform: translateY(-70px);

:x-38 .ex-38-4 {  
-ms-transform: translate(50%); /\* IE  
-webkit-transform: translate(50%); /\*  
-moz-transform: translate(50%); /\* Fx  
transform: translate(50%);

يعتبر المحور ص = 0 في هذه الحالة

# 2D Transformations

والآن الى الدالة الثالثة وهي ال skew -\_\*:

skew(3) : تسمح لك هذه الخاصية الغريبة P: من تغيير زاوية النظر (الإنحراف) لعنصر معين على مستوى المحور السيني أو الصادي أو كلاهما معا .. وبما أننا نتعامل هنا مع زاوية .. فيجب عليك استخدام ال deg الصيغة العامة: (تشبه الصيغة الخاصة بال translate ^\_^):

```
E {  
  transform: skewX(value) ;  
  transform: skewY(value) ;  
}  
E { transform: skew(skewX, skewY) ; }
```



# 2D Transformations

مثال 1:

Anees Hikmat Anees Abu-  
hamid 2nees.com

لاحظ هنا مقدار انحراف الزاوية  
للعنصر على مستوى المحور س  
والمحور ص  $\hat{\_} \hat{\_}$  ..

شعرت بالدوار -\_- من هذا المثال  
:P :P :P :P :P

Anees Hikmat Anees Abu-  
hamid 2nees.com

```
.ex-39-1{  
  -ms-transform: skew(201deg,10deg); /* I  
  -webkit-transform: skew(201deg,10deg);  
  -moz-transform: skew(201deg,10deg); /*  
  transform: skew(201deg,10deg);  
-}
```

# 2D Transformations

مثال 2:

Anees Hikmat Anees  
Abu-hamid 2nees.com

Anees Hikmat Anees  
Abu-hamid 2nees.com

Anees Hikmat Anees  
Abu-hamid 2nees.com

Anees Hikmat Anees  
Abu-hamid 2nees.com

Anees Hikmat Anees  
Abu-hamid 2nees.com

Anees Hikmat Anees  
Abu-hamid 2nees.com

```
.ex-39 .ex-39-2{  
  -ms-transform: skewX(201deg); /* IE 9 */  
  -webkit-transform: skewX(201deg); /* webkit */  
  -moz-transform: skewX(201deg); /* Firefox */  
  transform: skewX(201deg);  
}
```

```
.ex-39 .ex-39-3{  
  -ms-transform: skewY(10deg); /* IE 9 */  
  -webkit-transform: skewY(10deg); /* webkit */  
  -moz-transform: skewY(10deg); /* Firefox */  
  transform: skewY(10deg);  
}
```

```
.ex-39 .ex-39-4{  
  -ms-transform: skew(-210deg); /* IE 9 */  
  -webkit-transform: skew(-210deg); /* webkit */  
  -moz-transform: skew(-210deg); /* Firefox */  
  transform: skew(-210deg);  
}
```

# 2D Transformations

ملاحظة ظريفة على موضوع ال skew ^\_^ وهي أن هذه الخاصية في مرحلة من المراحل قد تشبه عملية ال rotate ^\_^.. ويكون هذا عندما نقوم بوضع قيمة الزاوية على المحور السيني .. والمحور الصادي يحتوي ال inverse للقيمة x..

مثال: اذا كانت  $x = 15$  فإن  $x = -15$  inverse و عليه اذا استخدمت هذا الشكل skew(15, -15) سيظهر عندك rotate بزاوية -15 مثل rotate(-15deg) ... شاهد الصورة ^\_^\*

```
.ex-39 .ex-39-5{  
  margin-top: 150px;  
  
  -ms-transform: skew(15deg, -15deg); /*  
  -webkit-transform: skew(15deg, -15deg);  
  -moz-transform: skew(15deg, -15deg); /*  
  transform: skew(15deg, -15deg);  
}
```

```
.ex-39 .ex-39-6{  
  -ms-transform: rotate(-15deg); /* IE 9  
  -webkit-transform: rotate(-15deg); /* W  
  -moz-transform: rotate(-15deg); /* Fire  
  transform: rotate(-15deg);  
}
```

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com



# 2D Transformations

والآن  $^x \ ^y$  الى الدالة الرابعة لهذا اليوم:  $P$  وهي ال  $scale \ ^x \ ^y$ :  
(4)  $scale$ : هذه الدالة الرائعة والأكثر من رائعة تمكنك من تكبير أو تصغير  
حجم العنصر حتى وإن كان هذا التغيير بالحجم على مستوى الأفقي أو  
العامودي أو كلاهما معا ...  $^x \ ^y$ \*

الصيغة العامة: وكما تلاحظ، فهي تشبه أيضا الدوال السابقة من حيث  
الشكل، لكن يجب عليك الإنتباه  
هنا الى نقطة وهي أن القيمة  
تكون رقم فقط هذا الرقم مثل 2  
أو 0.5 .. ف 2 تعني ضاعف  
الحجم، 0.5 صغر الحجم بمقدار  
النصف ... وقس على ذلك ..

```
E {  
  transform: scaleX(value) ;  
  transform: scaleY(value) ;  
}  
E {  
  transform: scale(scaleX, scaleY) ;  
}
```

# 2D Transformations

```
.ex-40 .ex-40-1 {  
  -ms-transform: scale(2,3); /* IE 9 */  
  -webkit-transform: scale(2,3); /* webkit */  
  -moz-transform: scale(2,3); /* Firefox */  
  transform: scale(2,3);  
}
```

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

مثال 1: لاحظ أن الحجم  
تضاعف مرتين على المحور  
س و 3 مرات على المحور  
ص...<sup>^</sup><sub>^</sub>

الحجم الأصلي

الحجم بعد التكبير

# 2D Transformations

مثال 2:

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

تصغير حجم المحور السيني الى النصف

تصغير الحجم الصادي بمقدار 0.2  
فيصبح الحجم 0.8

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

```
.ex-40 .ex-40-2{  
  -ms-transform: scaleX(0.5); /*  
  -webkit-transform: scaleX(0.5);  
  -moz-transform: scaleX(0.5); /*  
  transform: scaleX(0.5);  
}
```

```
.ex-40 .ex-40-3{  
  -ms-transform: scaleY(0.8); /*  
  -webkit-transform: scaleY(0.8);  
  -moz-transform: scaleY(0.8); /*  
  transform: scaleY(0.8);  
}
```

# 2D Transformations

مثال 3:

```
ex-40 .ex-40-4 {  
  -ms-transform: scale(1.5); /* I  
  -webkit-transform: scale(1.5);  
  -moz-transform: scale(1.5); /*  
  transform: scale(1.5);  
}
```

لاحظ هنا في حال قمنا  
بوضع قيمة واحدة فقط  
داخل ال scale فإن الحجم  
سيتضاعف على مستوى  
المحور السيني والصادي  
معاً

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com





# 2D Transformations

مثال 4: لاحظ في هذا المثال أنني قمت بعمل flipping بالإتجاه العامودي ومضاعفة حجمه أيضا ....

```
.ex-40 .ex-40-5 {
```

```
-ms-transform: scale(1.4, -1.4); /* I  
-webkit-transform: scale(1.4, -1.4);  
-moz-transform: scale(1.4, -1.4); /*  
transform: scale(1.4, -1.4);
```

كما أنني قمت بمضاعفة الحجم الأفقي، لكن دون عمل flipping

^ ^

---

Anees Hikmat Anees Abu-  
hamid 2nees.com

hamid 2nees.com

Anees Hikmat Anees Abu-



# 2D Transformations

والآن .. لننتقل لنقطة جديدة  $^{\wedge}_{\wedge}$  ومهمة .. هل يمكن استخدام أكثر من تأثير بذات الوقت، مثلا rotate وتغيير الحجم !؟

والجواب بكل روعة هو نعم .. يمكنك ذلك من خلال استخدام الصيغة التالية:

```
E { transform: function(value) function(value); }
```

بكل تأكيد لن أعيد شرح الصيغة .. لكن الأمر الجديد هو فقط بأنك ستقوم بترك فراغ بين كل دالة .. وبهذا يمكنك استخدام هذا الدوال التي تعلمناها مع  $^{\wedge}_{\wedge}$  ... لنشاهد مثالا  $*_{*}$

# 2D Transformations

مثال:

```
.ex-41 .ex-41-1{  
  -ms-transform: scale(1.5) rotate(10deg) translateY(30px); /*  
  -webkit-transform: scale(1.5) rotate(10deg) translateY(30px);  
  -moz-transform: scale(1.5) rotate(10deg) translateY(30px); /*  
  transform: scale(1.5) rotate(10deg) translateY(30px);  
}
```

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

# 2D Transformations

والآن سنتحدث عن آخر موضوع في هذه الجزئية بإذن الله تعالى، وهي دالة ال matrix (المصفوفة)...

5 matrix: هذه الدالة لوحدها تجميع بين جميع الدوال التي تم ذكرها في موضوع ال Transformations .. وهي تأخذ 6 قيم لتنفيذ الدوال المختلفة التي تحدثنا عنها (يمكنك من خلال هذه المصفوفة تنفيذ واحدة أو أكثر) لكن عليك أن تعرف طريقة كتابة هذه المصفوفة ... وسنحاول بإذن الله تعالى شرح الفكرة بأبسط أسلوب ممكن ..ويمكنك التطوير من مهارات في حال أردت المزيد ^\_\* ..

# 2D Transformations

والآن، لنشاهد الصيغة العامة:

```
E { transform: matrix(a,b,c,d,X,Y) ; }
```

المتغيرات من  $a - Y$  جميعها ذات قيمة افتراضية 0 .. لكن سلوكها مختلف بناءً على الموقع أو نوع التغيير الذي ستتركه ..

والآن، ولتبسيط المسألة قدر الإمكان .. ما رأيكم في تجربة لعمل scale باستخدام ال matrix ؟  $^_ ^$  .. شاهد المثال الأول: (ثم انتقل للشريحة التالية):

Anees Hikmat Anees Abu-  
hamid 2nees.com

```
transform: matrix(2,0,0,2,0,0) ;
```

نتيجة المصفوفة  $^_ ^$

Anees Hikmat Anees Abu-  
hamid 2nees.com

# 2D Transformations

أول نقطة لعمل scale يجب عليك تحديد قيمة ال scale  
ومن ثم عليك تعويض هذه القيم مكان المتغيرين a و d بحيث يمثل a  
المحور السيني و d المحور الصادي  $x$  و  $y$  وبهذا يتضح المثال الأول \*  
بإذن الله تعالى  $x$  و  $y$  ...

هذا الكلام يتعلق بال scale .. إذن .. فماذا عن ال translate  $x$  و  $y$ ، بنفس  
الأسلوب نحدد القيم ونقوم بتعويض هذه القيم مكان المتغيرين X و Y  $x$  و  $y$   
بحيث X المحور السيني و Y المحور الصادي  $x$  و  $y$  .. شاهد المثال:

ثم انتقل للشريحة التالية  $x$  و  $y$

Anees Hikmat Anees Abu-  
hamid 2nees.com

Anees Hikmat Anees Abu-  
hamid 2nees.com

# 2D Transformations

الشفيرة البرمجية الخاصة بالشكل السابق هي:

```
-ms-transform: matrix(2,0,0,2,-171,-59); /* IE  
-webkit-transform: matrix(2,0,0,2,-171,-59); /*  
-moz-transform: matrix(2,0,0,2,-171px,-59px);  
transform: matrix(2,0,0,2,-171,-59);
```

لاحظ كيف تم تحديد القيم مكان المتغيرين X & Y لكن هناك ملاحظات مهمة وهي:

- (1) الإزاحة الناتجة من خلال هذه الدالة ستكون بال px.
- (2) جميع المتصفحات نكتب بها قيم الإزاحة رقم فقط، ويستثنى من ذلك الإصدارات القديمة من ال moz بحيث يجب عليك وضع px بجانب الرقم &\_&..

# 2D Transformations

بقي لدينا أمران لم نتحدث عنهم في matrix وهما ال skew وال rotate  
^\_^، هذه الدوال إن كنت تذكر، فهي تأخذ قيم مثلثية (زاوية deg) لهذا فإن  
القيم التي ستعوض مكان b و c هي قيمة زوايا مثلثية ..  
والآن سيخطر ببالك سؤال مباشرة .. كيف c و b للخاصيتين rotate وال  
!! skew

والإجابة على هذا السؤال فإننا نتعامل مع زوايا مثلثية، وبتلاعبك بالقيم  
الخاصة بالزوايا المثلثية .. ستستطيع بناء هذا الشكل .. ^\_^

شاهد المثال: (طبعا هذه الامثلة تجمع أكثر من خاصية لكن أهمها هو  
موضوعنا ال skew وال rotate ) ... إلى الشريحة التالية...



# 2D Transformations

مثال أ:

```
.ex-41 .ex-41-4{  
  -ms-transform: matrix(2,0.30,0.60,2,0,0); /*  
  -webkit-transform: matrix(2,0.30,0.60,2,0,0);  
  -moz-transform: matrix(2,0.30,0.60,2,0,0);  
  transform: matrix(2,0.30,0.60,2,0,0);  
  skew  
}
```

Anees Hikmat Anees Abu-  
hamid 2nees.com

# 2D Transformations

```
-ms-transform: matrix(-0.014,1.166,1.387,0.009,103.344,-168.730); /* IE  
-webkit-transform: matrix(-0.014,1.166,1.387,0.009,103.344,-168.730);  
-moz-transform: matrix(-0.014,1.166,1.387,0.009,103.344px,-168.730px);  
transform: matrix(-0.014,1.166,1.387,0.009,103.344,-168.730); rotate
```

aneeshikmat@2nees.com  
أنيس حكمت أبو حميد

مثال ب:

# 2D Transformations

بهذا والحمد لله نكون انتهينا من هذا الموضوع الشيق والأكثر من شيق..

ملاحظات:

(1) بخصوص دالة ال matrix فهذه الدالة لها عالم، ولم ننتهي منه بعد.. لكن هذه الأمثلة والشرح هو مدخل لهذه الدالة، ولكن إن أردت المزيد يمكنك القراءة والبحث حول الموضوع  $\wedge$   $\wedge$ ..

(2) هذا الموقع من خلاله يمكنك من تشكيل مصفوفة  $\wedge$   $\wedge$ ..

والآن.. شاهد الأمثلة السابقة  $\wedge$  \*

إنما تنجح الفكرة إذا قوى الإيمان بها، وتوفّر الأخلص فى سبيلها، وازدادت الحماسة لها، ووجد الأستعداد ...

حسن البنا رحمه الله

# Transition & Animation

قديمًا، وقبل دخولنا عالم ال CSS3 كنا نقوم بتقسيم صفحة الويب الى 3 أجزاء أو طبقات رئيسية وهي المحتوى والذي يمثل عن طريق ال HTML، والعرض والذي يمثل من خلال CSS، والسلوك والذي يمثل من خلال ال JS ^\_^... وكان بشكل عام، هناك فهم بأن كل وسيلة من هذه الوسائل يجب أن تبقى منفصلة نوعًا ما عن الأخرى.. بحيث يبقى لكل طبقة استخدامها التي صممت له، لكن هذا الفهم تلاشى تدريجياً مع تطور عالم الويب، وأصبح الآن من خلال ال CSS3 أن نقوم بعملية دمج بين هذه الطبقات بأسلوب رائع وسهل.. بحيث أمكننا دمج العرض والسلوك معا لتغيير شكل المحتوى من خلال ال Transition & Animation ^\_^..  
والآن ما هو ال Animation وما هو ال Transition \*\_\*

# Transition & Animation

- ال Transition هي وسيلة تسمح لك بتغيير قيمة معينة لعنصر معين بشكل تدريجي (بشكل سلس ورائع للعين ^\_^) من قيمة الى أخرى، وضمن فترة زمنية معينة ^\_^.
- ال Animation هي وسيلة تسمح لك بتغيير سلوك العنصر وإثراء الحركات عليه دون الحاجة الى استخدام ال js أو flash ^\_^ \*

وأول ما سيخطر ببالنا الآن ما هو الفرق بين ال Transition وال

Animation ؟ ^\_^

# Transition & Animation

الفرق بين ال Animation وال Transition قد يكون دقيق نوعا ما، لأن ال Transition هو صورة مصغرة عن ال Animation إن صح التعبير، بحيث يكون ال Transition محددًا بين نقطتين أي بين حالتين، Start to End ضمن فترة زمنية معينة، مثل عملية تصغير ال width وإعادة تكبيره، لكن ال animation بالإضافة الى ماسبق يمكن التحكم به بشكل أكثر، وتقطيعه الى فترات زمنية أكثر، وتختلف بنوع التغيير أو الحدث الذي يحصل بكل جزئية .. وهذا يعني أننا نعمل مع `^*_` freams ، بالإضافة الى أنه يمكنني تحديد التكرار هنا ولا استطيع تحديده بال `^*_` transition ، وفي الغالب فإننا نستخدم ال transition في حالات مهمة وسريعة مثل تغيير لون – حجم – تأثير بدأ ونهاية والتي تنطلق عند ال hover, active, focus .. الخ (وذلك لأن ال Transition تحتاج الى شيفرة برمجية أقل لتنفيذها عادة `^_` ) .. ولكل خاصية مكانها `^*_` ..والآن لنبدأ معا `^^`

# Transition & Animation

أولا ال Transition:

الآن لتعريف أو للقيام بأي Transition علينا القيام بخطوتين رئيسيتين وهما:

(1) تعريف الخاصية التي سنقوم بعمل ال effect لها  $_*\_^*$

(2) مدة التنفيذ الخاصة بال Effect  $_*\_^*$

والآن لنشاهد مثلا يوضح الفكرة، قبل شرح الخصائص  $_*\_^*$

هنا سيتم تغيير الخلفية من اللون البرتقالي

الى اللون الفضي بزمن مقداره 2 ثانية

$_*\_^*$  عند حدوث hover .. رائع  $_*\_^*$

```
.ex-42 .ex-42-1{  
  background: orange;  
  -webkit-transition: background-color 2s;  
  transition: background-color 2s;  
}
```

```
.ex-42 .ex-42-1:hover{  
  background-color: silver;  
}
```



# Transition & Animation

والآن لننتحدث عن خصائص ال Transition وهي:

(1) transition-property: وهنا نقوم بوضع اسم خاصية ال css المراد تنفيذ ال effect لها والقيمة الافتراضية لها هي all أي لجميع الخصائص ^\_^، ويمكن أن تحتوي خاصية واحدة أو أكثر يفصل بينها فواصل عادية ...

```
transition-property: width, background-color;
```

طبعاً يجدر هنا ملاحظة ليس كل الخصائص تدعم ال Transition بال css ... والذي يمكن أن يتأثر من الخصائص هو عادة من يملك نقطتين كما ذكرنا في تعريف ال Transition.. شاهد أشهر الخصائص \*\_^ في الشريحة التالية:

# Transition & Animation

- background-color
- border-width
- clip
- font-size
- left
- margin
- min-height
- outline-color
- padding
- text-shadow
- visibility
- z-index
- background-position
- border-spacing
- color
- font-weight
- letter-spacing
- max-height
- min-width
- outline-offset
- right
- top
- width
- border-color
- bottom
- crop
- height
- line-height
- max-width
- opacity
- outline-width
- text-indent
- vertical-align
- word-spacing

# Transition & Animation

(2) transition-duration: هذه الخاصية هي المسؤولة عن تحديد الوقت اللازم لتنفيذ Transition معين  $^*_$ ، والوقت الافتراضي هنا صفر، أي أنه يجب عليك تحديد الوقت حتى يصبح ال Transition فعال، ووحدة الزمن هنا هي الثانية (s) أو أجزاء الثانية (ms) بحيث تمثل كل 1000ms واحد ثانية ... لذلك 3s تعني 3 ثواني و 3250ms تعني 3 ثواني و 250 جزء من الثانية  $^*_$ ، وفي حالة تحديد هذه الخاصية فإن جميع العناصر تأخذ نفس الوقت، وإذا استخدمنا في ال transition-property أكثر من قيمة ونريد إعطاء كل قيمة وقت مختلف فيمكننا هنا أيضا استخدام الفاصلة، وبنفس الترتيب الذي وضع يكون هو ترتيب الزمن عند التنفيذ .. شاهد المثال  $^*_$ :

```
transition-property: width, background-color;
```

```
transition-duration: 2s; كل القيم يلزمها 2 ثانية
```

```
transition-duration: 2s, 10s; هنا 2 ثانية و 10 ثانية
```

# Transition & Animation

مثال:

```
.ex-42 .ex-42-2 {  
  transition-property: width, background-color;  
  transition-duration: 2s;  
}
```

```
.ex-42 .ex-42-2: hover {  
  background-color: red;  
  width: 600px;  
}
```

```
.ex-42 .ex-42-3 {  
  /*transition-property: all; Default */  
  transition-duration: 2s;  
}
```

```
.ex-42 .ex-42-3: hover {  
  background-color: silver;  
  width: 600px;  
  font-size: 12px;  
  height: 100px;  
}
```

```
.ex-42 div {  
  padding: 20px;  
  font-size: 25px;  
  color: green;  
  width: 300px;  
  height: 50px;  
  background: orange;  
  border: 1px solid blue;  
  margin: 50px auto 0;  
}
```

Anees Hikmat Anees Abu-hamid 2nees.com 2

ما ضُرب عبد بعقوبة أعظم من قسوة قلب، وما غضب الله عزَّ وجلَّ على قوم  
إلا نزع منهم الرحمة

مالك بن دينار رحمه الله

# Transition & Animation

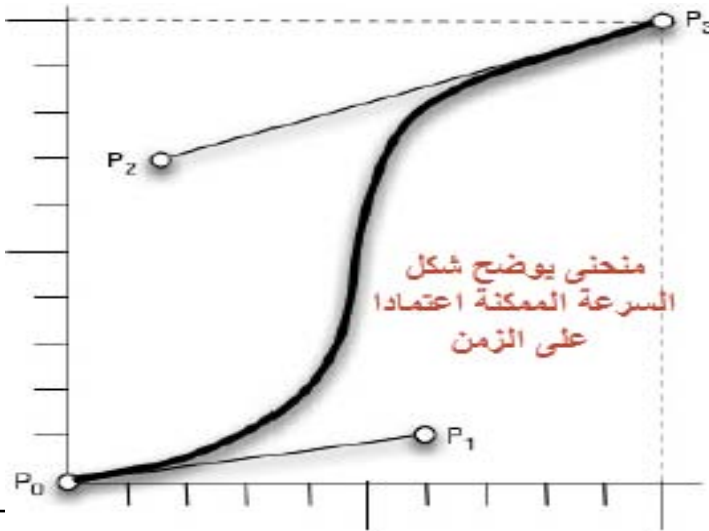
(3) :transition-timing-function

ولشرح هذه الخاصية، يجب شرح ال Time Function أولاً \*\_، وهذا الشرح مهم لأنه سيكون لل animation أيضا (أي لن أعيد شرحه في موضوع ال animation).. لذلك انتبه جيدا \*\_ \* نعني باستخدام ال time function هو التحكم بالسرعة اعتمادا على منحنى الوقت .. أي أن السرعة لتنفيذ حركة أو حدث معين قد تزيد أو تنقص في فترة زمنية معينة بما يتم تحديده من قبل المطور، ولهذا

الشكل أسلوبان لكتابته وهما

keyword أو cubic-bezier ...

والآن انتقل الى الشريحة التالية:



# Transition & Animation

القيم الخاصة بال keyword هي:

- ease: وهذه تعني بداية بطيئة ثم سرعة ثم نهاية بطيئة من جديد، وهي الحالة الافتراضية  $^*_$ .
- linear: وهذه تعني أن السرعة متساوية من البداية ووصولاً إلى نهاية الحركة  $^*_$ .
- ease-in: وهذه تعني بداية بطيئة ومن ثم يسرع حتى ينتهي  $^*_$ .
- ease-out: وهذه تعني نهاية بطيئة، أي يبدأ بسرعة وعند النهاية ينتهي ببطء  $^*_$ .
- ease-in-out: وهذه تعني دمج بين الخاصيتين السابقتين، بحيث أنها تبدأ ببطء ثم تتسارع تدريجياً حتى تصل نقطة معينة قبل النهاية لتتباطئ مجددًا. ولفهم الفرق بين هذه الخاصية وال ease شاهد الشريحة التالية  $^*_$ .

# Transition & Animation

صورة تمثل الفارق بالسرعة اعتمادا على ال keyword  $^*_$ :



ease



linear



ease-in



ease-out



ease-in-out

اما الأسلوب الثاني لتحديد الوقت فهو cubic-bezier، وفعليا إن الخصائص القيم التي تم ذكرها سابقا في ال keyword هي مسميات لقيم تم استخدامها وتعويضها من ال cubic-bezier...

ال cubic-bezier هو منحنى تم تصميمه من بيزييه الفرنسي، ويستخدم بشكل كبير في الرسوم الحاسوبية، والصيغة العامة لهذا المنحنى هي  $cubic-bezier(n,n,n,n)$ ، بحيث تمثل كل  $n$  قيمة بين  $0 - 1$ ، وإذا كانت القيمة سالبة، فهذا يعني أنها خارج النطاق (رجوع عكس الحركة أيضا)..



# Transition & Animation

إذا قمنا باستبدال  $n$  بالمحاور فيصبح لدينا الشكل التالي:

`cubic-bezier(x1, y1, x2, y2);`

يمكنك مشاهدة العديد من الأمثلة وإنشائها وتطبيق قيم مختلفة من خلال هذا

[الموقع الجميل](#) <sup>\*</sup>

شاهد ناتج على تطبيق مثال من الموقع <sup>\*</sup>

```
"Anees": "1, 1.89, 1, -0.4"
```

والآن لنتابع موضوعنا الأصلي من جديد <sup>\*</sup>  
ولنتحدث عن transition-timing-function



Anees

# Transition & Animation

إن transition-timing-function تمكننا من التحكم بسرعة تطبيق الانتقال الخاصة بالحركة التي قمنا بتحديدتها، والسرعة تختلف بناءً على ال keyword أو cubic-bezier التي تم تحديدها  $^*_$ .. كما يمكنك استخدام الفاصلة للفصل بين القيم في حال وجود أكثر من خاصية ..

والآن لنرى طريقة التطبيق  $^*_$

```
transition-timing-function: ease-in-out;
```

```
transition-timing-function: cubic-bezier(1,1.89,1,-0.4);
```

```
transition-timing-function: ease-in-out, cubic-bezier(.44,1.89,0,-0.65);
```

# Transition & Animation

مثال: (عند تنفيذك لهذا المثال.. انتبه لمعدل سرعة الحركة واتجاهها)

```
.ex-42 .ex-42-5{  
  transition-duration: 4s;  
  transition-timing-function: ease-in-out;  
}  
  
.ex-42 .ex-42-5:hover{  
  background-color: brown;  
  width: 600px;  
}  
  
.ex-42 .ex-42-6{  
  transition-duration: 4s;  
  transition-timing-function: cubic-bezier(1,1.89,1,-0.4);  
}  
  
.ex-42 .ex-42-6:hover{  
  background-color: brown;  
  width: 600px;  
}  
  
.ex-42 div{  
  padding: 20px;  
  font-size: 25px;  
  color: green;  
  width: 300px;  
  height: 50px;  
  background: orange;  
  border: 1px solid blue;  
  margin: 50px auto 0;  
}
```

# Transition & Animation

(4) transition-delay: تستخدم هذه الخاصية لتحديد وقت لتأخير عملية انطلاق ال Transition، وتكون القيمة الخاصة بها زمنيا إما بالثواني أو بأجزاء الثانية، كما أن الفاصلة هي التي تفصل بين القيم اذا وجدت أكثر من قيمة -\_-، كما أن هذه الخاصية يمكنها أن تحتوي على وقت بالسالب \*\_\*، وهذا يعني قم بطرح أو أخفاض الزمن اللازم للحركة بمقدار يساوي ال delay ويكون هذا مباشرة لحظة البدء بالحدث ..، ستتضح الصورة اذا قمت بتطبيق مثال على هذه العملية \*\_\*\_...

مثال على الصيغة العامة: `transition-delay: 2s;`

مثال على القيمة السالبة: لاحظ هنا أن زمن التأخر هو -2 وهذا يعني

4-2=2، فعند البدء سنختصر مباشرة 2 ثانية ^\_^

`transition-duration: 4s;`

`transition-timing-function: ease-in-out;`

`transition-delay: -2s;`

# Transition & Animation

مثال:

```
.ex-42 .ex-42-8{  
  transition-duration: 4s;  
  transition-timing-function: ease-in-out;  
  transition-delay: 2s;  
}
```

```
.ex-42 .ex-42-8:hover{  
  background-color: yellow;  
  width: 600px;  
}
```

```
.ex-42 .ex-42-9{  
  transition-duration: 4s;  
  transition-timing-function: ease-in-out;  
  transition-delay: -2s;  
}
```

```
.ex-42 .ex-42-9:hover{  
  background-color: yellow;  
  width: 600px;  
}
```

عند تطبيقك لهذا المثال  
لاحظ المدة الزمنية المستغرقة  
حتى ينطلق كل transition  
ولاحظ ما هو أثر القيمة  
السالبة على التأخير بالنسبة  
لل duration ..

# Transition & Animation

وأخيراً، وليس آخراً ^\_^.. سنتحدث عن آخر جزئية بإذن الله تعالى  
بموضوع ال transition وهو ال shorthand (الطريقة المختصرة) لكتابة  
ال transition ...

شاهد الصيغة:

```
]E {  
  transition: property duration timing-function delay;  
-}
```

لاحظ بهذه الطريقة نختصر الطريقة التقليدية.. وكل ما عليك فعله هو  
وضع القيم مكانها بهذا الترتيب.. (الترتيب مهم) ومن يتم الفصل بينهم  
بفراغات ^\_^ \*.. تابع الشريحة التالية-->

```
transition: all 4s ease-in-out 2s;
```

# Transition & Animation

لكن هناك ملاحظة مهمة يجب أن تنتبه لها، وهي في حالة وجود أكثر من Transition في ال short hand `^*_*` ولم نقم باستخدام ال `all`، في هذه الحالة يجب عليك كتابة الصيغة بالشكل التالي:

```
E {  
  transition:property1 duration1 timing-function1 delay1,  
            property2 duration2 timing-function2 delay2,  
            property3 duration3 timing-function3 delay3;  
}
```

شاهد المثال:

لا تنسى أن هذه الخصائص الزامية

```
/*example 11*/  
.ex-42 .ex-42-11{  
  transition:background-color 2s ease 1s,  
            width 4s ease,  
            font-size 6s ease-in-out;  
}
```

# Transition & Animation

الآن ..خطرت في نفسي فكرة جميلة لأضعها هنا قبل أن أنهي هذه الجزئية بإذن الله تعالى ^\_^..لا أدري إن خطرت ببالك بعد ^\_^، وهي هل يمكن استخدام ال transition وال transform معا ؟

هذا الجواب مهم ..وهو نعم ^\_\_\_\_^، وبهذه الطريقة نعطي رونقا وجمالا لما تعلمناه في الموضوع السابق..

```
.ex-42 .ex-42-12 {  
  transition: all 4s;  
}
```

لا تنسى أن تحدد اللون والحجم  
ولون الخلفية كما تريد ..

قم بتجربة هذا المثال:

```
.ex-42 .ex-42-12: hover {  
  background-color: yellow;  
  color: blue;  
  width: 600px;  
  -webkit-transform: rotate(180deg) scale(1.2);  
  transform: rotate(180deg) scale(1.2);  
}
```

Anes Hikmat Anes Abu-hamid 2nees.com 12



# Transition & Animation

وبهذا، نكون انتهينا بإذن الله تعالى من أول جزئية بهذا الموضوع وهو ال Transition لننتقل الى الجزئية الثانية بعد ذلك وهي ال animation \*\_\*

لكن أرجوا أن تكون قمت بتطبيق والمحاولة في جميع الأمثلة والنقاط التي تم ذكرها، بدون تطبيق .. لن تستطيع كتابة أي شيفرة مستقبلا .. انتبه لذلك

كما يمكنك مشاهدة جميع الأمثلة السابقة من هنا، لكن إنتبه أنني لم أراعي استخدام ال vendor لأجل السرعة .. (الأمثلة تعمل على أحدث إصدار من firefox و chorme بدون مشكلة بإذن الله تعالى .. (^\_^)



التطوع.. يعني التكافل والتعاون والمشاركة والعطاء والثقة والتضحية والإيثار  
وهي تنمية للإبداع وروح للتجديد

حكم عن العمل التطوعي

# Transition & Animation

تحدثنا في بداية هذا الموضوع عن ال Transition وال animation والآن،  
لنتحدث بتخصيص أكثر حول موضوع ال Animation، والذي يعد من  
أجمل المواضيع في عالم تنسيق صفحات الويب..

إن ال Animation تسمح لك بتغيير تنسيق معين من شكل لآخر، بحيث  
يمكنك تغيير خصائص ال css بالعدد الذي تريده، وبالوقت الذي تريده  
وعلى أكثر من مرة أو فترة زمنية، وللقيام بهذا الأمر، يجب علينا أن  
نعرف عن keyframes...

ال keyframes هي أول خطوة لإنشاء أي animation، ويحتوي على نقطة  
إنطلاق وانتهاء التأثير أو الحركة، ويكون اختلاف التنسيق بناءً على الزمن  
الذي أقوم بتحديد له داخل ال frame والحد الأدنى لل frame هو 2 أي  
وجود نقطة بداية ونهاية..ويمكن تفصيلها أكثر من ذلك كما نريد \*\_\*

# Transition & Animation

والآن لنبدأ بتفصيل الموضوع بشكل أكبر نوعاً ما، وذلك بتقسيم ال

Animation إلى:

(1) @keyframes:

(2) خصائص ال animation وهـي:

animation-name (أ) animation-duration (ب)

animation-timing-function (ج) animation-delay (د)

animation-iteration-count (ن) animation-direction (هـ)

animation-play-state (و) animation-fill-mode (ي)

وبعد ذلك سنتحدث بإذن الله تعالى عن ال shorthand لل animation وال

Multiple Animations ^ \_\_\_\_ ^

# Transition & Animation

```
@keyframes 'name' {  
  keyframe {  
    property : value;  
  }  
}
```

أولا ال `@keyframes`

الصيغة العامة:

أولا يتم كتابة `@keyframes` ثم يتم كتابة

اسم مميز لهذا ال `keyframes` وفضل أن

يكون الاسم ذو علاقة بال `animtion` مثل `moveCar`، وهذا الإسم يجب أن

يكون فريد، أي لا يجب أن يتكرر ل `animation` آخر بكل تأكيد `^_^`

أما ال `keyframe` فهي إما أن تكون `from/to` أو تكون بالنسبة المئوية مثل

`0% - 10% - 50% - 100%`

أما ال `property: value` فهي خاصية/خصائص ال `css` المراد عمل

`animation` لها `^_^`... لنشاهد مثال <-->

# Transition & Animation

انتبه  
للمثال:

```
.ex-43 .ex-43-1{  
  -webkit-animation: animate-example-1 5s linear 100ms infinite alternate;  
  animation: animate-example-1 5s linear 100ms infinite alternate;  
}
```

سيتم شرحها فيما بعد بإذن الله تعالى .. الآن فقط قم بتطبيق المثال كما تشاهده .

```
@-webkit-keyframes animate-example-1 {  
  from {  
    background-color: orange;  
    color: green;  
  }  
  to {  
    background-color: blue;  
    color: red;  
  }  
}
```

هنا لقد قمنا بتعريف ال  
**@keyframes to animation**  
لكن لاحظ نقطة مهمة جدا، وهي استخدام ال  
**webkit**  
ليعمل ال  
**animation**  
على المتصفحات الأخرى دون مشاكل ..  
لاحظ كيف تمت كتابة الصيغة القياسية ...

```
@keyframes animate-example-1 {  
  from {  
    background-color: orange;  
    color: green;  
  }  
  to {  
    background-color: blue;  
    color: red;  
  }  
}
```

ال  
from === 0%  
to === 100%

# Transition & Animation

مثال 2:

```
/*example 2*/
.ex-43 .ex-43-2{
  -webkit-animation: animate-example-2 5s linear 100ms infinite alternate;
  animation: animate-example-2 5s linear 100ms infinite alternate;
}

@-webkit-keyframes animate-example-2 {
  0% {
    background-color: orange;
  }
  25% {
    color: green;
  }
  50%{
    font-size: 25px;
    color: white;
  }
  75%{
    font-size: 20px;
  }
  100% {
    background-color: blue;
    color: red;
    font-size: 30px;
  }
}
```

هنا نفس المثال السابق..  
لكن باستخدام أكثر من خاصية وأكثر من  
**keyframe**  
لاحظ هنا أن البداية سيبدأ بتغيير لون الخلفية  
ثم عندما يصل لل 25% يبدأ بتغيير لون الخط  
بالإضافة الى لون الخلفية فهو لم يتوقف ..  
ثم عند 50% أخبرته بأن يبدأ بتغيير حجم الخط  
وأخبرته أيضا بأن اللون سيتدرج من الآخر الى  
الأبيض حين وصوله الى ال 50%  
بعد ذلك يبدأ بتغيير لونه من الأبيض مجددا  
وعند وصوله الى 75% يصبح حجم الخط 20  
بكسل، بعد ذلك عندما نصل لل 100% سيكون  
لون الخلفية أصبح أزرق بشكل كامل بالإضافة  
الى أن لون الخط أصبح أحمر والحجم 30  
بكسل ...

هذا شرح المثال أو التدرج به ^\_^

شاهد تطبيق الأمثلة السابقة



# Transition & Animation

والآن الى خصائص ال animation وقبل البدء أرغب بإيضاح نقطة، وهي أنه بتشغيل والتحكم بال animation يجب أن يتواجد هناك control وهنا ال control هي مجموعة الخصائص التي تستدعي وتتحكم بال keyframes.. لذلك دائما بعد كتابتك لل keyframes يجب عليك أن تستدعيه وهذا ما وجدته في المثالين السابقين .. والآن سبدأ بإذن الله تعالى في شرح أول خاصية من هذه الخصائص وهي ال animation-name (1

ال animation-name تستخدم للإشارة على ال keyframe المراد استدعائه وبهذا فهي تحمل نفس الاسم ...

مثال: animation-name: keyframe-example .. انتقل للشريحة التالية



# Transition & Animation

أيضا يجب عليك أن تراعي الدعم للمتصفحات الأخرى ..وكما استخدمنا ال Vendor لل keyframe نستخدمه للخصائص ..شاهد المثال:

```
.ex-44 .ex-44-1 {  
  -webkit-animation-name: animate-example-44-1;  
  animation-name: animate-example-44-1;  
}
```

```
@-webkit-keyframes animate-example-44-1 {  
  from {  
    .....  
  }  
  to {  
    .....  
  }  
}
```

```
@keyframes animate-example-44-1 {  
  from {  
    .....  
  }  
  to {  
    .....  
  }  
}
```

لاحظ أن الإسم هو نفسه  
اسم ال keyframe وأن  
ال vendor أيضا موجود  
كما هو موجود لل  
keyframe

# Transition & Animation

animation-duration(2) : وهي الخاصية المسؤولة عن تحديد الزمن الكلي لل animation ... وهي تشبه ال transition-duration ^\*\_ ، والقيم الخاصة بها أيضا هي بالثواني أو أجزاء الثواني .. والقيم السالبة تعد صفرا شاهد مثال:

```
animation-duration: 6s;
```

مثال 2:

```
.ex-44 .ex-44-1 {  
  -webkit-animation-name: animate-example-44-1;  
  -webkit-animation-duration: 6s;  
  animation-name: animate-example-44-1;  
  animation-duration: 6s;  
}
```

# Transition & Animation

3 animation-timing-function: ^\_^ هذه الخاصية تم شرحها سابقا أيضا ..راجع درس ال timing function الموجود بال transition اذا كنت لا تذكرها \*\_^ الصيغة العامة:

```
E { animation-timing-function: keyword OR cubic-bezier(x1, y1, x2, y2); }
```

مثال:

```
.ex-44 .ex-44-2 {  
  -webkit-animation-name: animate-example-44-1;  
  -webkit-animation-duration: 6s;  
  -webkit-animation-timing-function: ease-in;  
  animation-name: animate-example-44-1;  
  animation-duration: 6s;  
  animation-timing-function: ease-in;  
}
```

# Transition & Animation

4 animation-delay : هذه الخاصية المسؤولة عن وضع فترة زمنية قبل تنفيذ ال animation .. (تأخير أو انتظار) وهي تشبه أيضا خاصية ال transition-delay .. ^\_^ .. صفر يعني لا يوجد delay وهي الحالة الافتراضية، ويمكن استخدام الثواني وأجزاء الثواني .. كما أن القيم السالبة تعني اختصر أو اعمل skip لجزء من الوقت المخصص لل duration .. وهذه الأمور تم شرحها سابقا .. \*\_^

```
.ex-44 .ex-44-3{
```

```
-webkit-animation-name: animate-example-44-1;  
-webkit-animation-duration: 4s;  
-webkit-animation-timing-function: ease-in;  
-webkit-animation-delay: 2s;  
animation-name: animate-example-44-1;  
animation-duration: 4s;  
animation-timing-function: ease-in;  
animation-delay: 2s;
```

مثال:

```
-}
```

# Transition & Animation

الآن لنفيذ الأمثلة التي ستأتي بالشرائح التالية ..

قم بكتابة ال keyframes التالي ولا تنسى اضافة أيضا ال -webkit- :

```
@keyframes animate-example-44-4 {  
  from {  
    background-color: orange;  
    margin-left: 0px;  
  }  
  50% {  
    background-color: black;  
  }  
  to {  
    background-color: yellow;  
    margin-left: 800px;  
  }  
}
```

```
.ex-44 div {  
  padding: 20px;  
  font-size: 25px;  
  color: green;  
  width: 300px;  
  height: 50px;  
  background: orange;  
  border: 1px solid blue;  
  margin: 50px auto 0;  
}
```

# Transition & Animation

5 animation-iteration-count: ^\_^ هذه الخاصية جديدة، ولم تكن موجودة بال Transition وهي إحدى مميزات ال animation التي ذكرناها فمن خلال هذه الخاصية يمكننا تكرار ال animation عدد معين من المرات .. هذا العدد إما أن يكون لا نهائي باستخدام ال infinite أو عدد معين من المرات مثل 5 أو 7 أو 145 .. الخ، والحالة الافتراضية له هي 1، أي يقوم بعرض ال animation من البدء وحتى النهاية .. ثم يتوقف .. وإذا تم استخدام 0.. فهذا يعني توقف ال animation...لن يتم التنفيذ \*\_^

الصيغة العامة:

```
E { animation-iteration-count: count; }
```

# Transition & Animation

مثال:

```
.ex-44 .ex-44-4{
  margin-left: 0px;
  -webkit-animation-name: animate-example-44-4;
  -webkit-animation-duration: 4s;
  -webkit-animation-iteration-count: 3;
  animation-name: animate-example-44-4;
  animation-duration: 4s;
  animation-iteration-count: 3;
}

.ex-44 .ex-44-5{
  -webkit-animation-name: animate-example-44-4;
  -webkit-animation-duration: 4s;
  -webkit-animation-iteration-count: infinite;
  animation-name: animate-example-44-4;
  animation-duration: 4s;
  animation-iteration-count: infinite;
}
```

# Transition & Animation

(6 animation-direction: أيضا هذه الخاصية لم نتحدث عنها وهي من مميزات ال animation الرائعة ^ \_ \* ، تقوم هذه الخاصية على تحديد طريقة العرض أو الانتقال من البداية الى النهاية وبالعكس .. يعني وبأسلوب آخر ، في الطريقة التقليدية لل animation فإن الحالة الافتراضية هي أن ينطلق ال animation من البداية وصولا الى النهاية .. ومن ثم تجد أنه عاد الى نقطة البداية فجأة دون تدرج بالعودة مثل ال Transition .. لكن هذه الخاصية يمكن تغيير قيمتها باستخدام ال animation-direction وذلك من خلال استبدال ال normal (القيمة الافتراضية) ب alternate أو Reverse ... والآن لنشاهد الصيغة العامة . ثم نعود لشرح هذه القيم ^ \_ ^ الصيغة العامة:

```
E { animation-direction: keyword; }
```



# Transition & Animation

شرح القيم الممكنة لل animation-direction:

- (1) normal: وهي القيمة الافتراضية، وتعني أن يبدأ ال animation ومن ثم ينتهي ويعود مباشرة الى نقطة البداية دون تدرج أو أي تأثير..
- (2) reverse: وهذه القيمة تعني أن يبدأ ال animation بشكل عكسي، من نقطة الإنتهاء رجوعا الى البداية .. ^\_^
- (3) alternate: هذه القيمة تعني أن ال animation سيعمل بالشكل الإعتيادي عند الأعداد الفردية مثل 1 – 3 – 5 .. الخ وسيعمل reverse عند الأعداد الزوجية مثل 2 – 4 – 6.. الخ (الأرقام ترمز الى التكرار ^\_\*).. وهذه الخاصية رائعة ومهمة جدا.. لأن شكل الحركة هنا يصبح على شكل ذهاب وإياب ^\_\*...
- (4) alternate-reverse: هذه القيمة هي عكس للقيمة ال alternate ..

# Transition & Animation

مثال:

لا تنسى تطبيق المثال  
فالتطبيق مهم جدا جدا  
هنا .. وكتوضيح أخير  
لفكرة ال alternate  
فهي عند البداية هنا  
وحسب ال keyframe  
الذي نستخدمه .. سيبدأ من  
اليسار الى اليمين (1) ثم  
من اليمين الى اليسار (2)  
ثم من اليسار الى اليمين  
(3) ثم من اليمين لليسار  
(4) .. وهكذا ينتهي ^\_^

```
.ex-44 .ex-44-6{  
  -webkit-animation-name: animate-example-44-4;  
  -webkit-animation-duration: 4s;  
  -webkit-animation-iteration-count: 3;  
  -webkit-animation-direction: reverse;  
  animation-name: animate-example-44-4;  
  animation-duration: 4s;  
  animation-iteration-count: 3;  
  animation-direction: reverse;  
}
```

```
.ex-44 .ex-44-7{  
  -webkit-animation-name: animate-example-44-4;  
  -webkit-animation-duration: 4s;  
  -webkit-animation-iteration-count: 4;  
  -webkit-animation-direction: alternate;  
  animation-name: animate-example-44-4;  
  animation-duration: 4s;  
  animation-iteration-count: 4;  
  animation-direction: alternate;  
}
```

كان السلف الصالح إذا دخلوا بيوتهم أو رأوا ما يعجبهم من أموالهم قالوا: ما شاء الله لا قوة إلا بالله، يتأولون قول الله تعالى: وَلَوْلَا إِذْ دَخَلْتَ جَنَّتَكَ قُلْتَ مَا شَاءَ اللَّهُ لَا قُوَّةَ إِلَّا بِاللَّهِ {الكهف:39}

نقل ذلك عن الإمام مالك، وشيخه ابن شهاب، وعروة. رحمهم الله جميعا

# Transition & Animation

animation-play-state (7) تستخدم هذه الخاصية الرائعة لإيقاف ال animation في مرحلة معينة، وهي تملك قيمتان paused و running وهي الحالة الافتراضية ..

شاهد الصيغة العامة: `E { animation-play-state: keyword; }`

```
.ex-44 .ex-44-8 {  
  -webkit-animation-name: animate-example-44-4;  
  -webkit-animation-duration: 4s;  
  -webkit-animation-iteration-count: infinite;  
  -webkit-animation-direction: alternate;  
  animation-name: animate-example-44-4;  
  animation-duration: 4s;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}
```

لاحظ هنا عند مرور مؤشر الفأرة فوق العنصر ..سيتوقف ال

```
animation  
  الخاص به ^_^  
.ex-44 .ex-44-8: hover {  
  -webkit-animation-play-state: paused;  
  animation-play-state: paused;  
}
```

مثال:

Anees Hikmat Anees Abu-  
hamid 2nees.com 8



# Transition & Animation

(7 animation-fill-mode: وأخر خاصية بإذن الله تعالى في هذا الموضوع الشيق، خاصية ال animation-fill-mode...

هذه الخاصية تستخدم لتحديد التنسيق الخاص بعنصر ما، بعد انتهاء تنفيذه أو في حالة وجود delay time حسب القيمة المراد وضعها .. ويمكن أن تحتوي هذه الخاصية على القيم forwards و backwards و none .. بحيث none هي الحالة الافتراضية و both .. هذه القيم تعني:

(1 forwards: سيتم أخذ التنسيق الذي تم انتهاء ال animation عنده وإبقائه على العنصر.. (وهذه هي أهم قيمة ^\_^)

(2 backwards: سيتم أخذ التنسيق لأول keyframe سيتم تنفيذه لل Animation...

# Transition & Animation

(3) Both: هذه القيمة تقوم على الدمج أو تنفيذ خاصية ال forwards وال  
^\_^... backwards

(4) none: هنا لن يتم جلب أو تنفيذ أي تنسيق قبل أو بعد انتهاء ال  
animation وهي الحالة الافتراضية \*\_^

ملاحظة مهمة: هنا ال delay أو directions لها تأثير على وضوح فائدة  
هذه الخصائص .. فمثلا تتضح جليا ال backwards مع ال delay.. كما أن  
ال directions وال iterations يؤثر على النتيجة النهائية للخاصية بنائا  
على القيم المستخدمة...والآن لنشاهد مثلا في الشريحة التالية:

# Transition & Animation

مثال على forwards :

```
.ex-44 .ex-44-9{  
  -webkit-animation-name: animate-example-44-9;  
  -webkit-animation-duration: 4s;  
  -webkit-animation-iteration-count: 2;  
  -webkit-animation-fill-mode: forwards;  
  animation-name: animate-example-44-9;  
  animation-duration: 4s;  
  animation-iteration-count: 2;  
  animation-fill-mode: forwards;  
}
```

الآن هذا المثال لل forwards ..^\_\*.. لكن تطبيق ال backwards لتشاهد الفرق دقيق نوعا ما .. لذلك سأدعها لك كوظيفة لتشاهد الفرق .. يمكنك البحث عن أمثلة عن طريق اسم الخاصية .. وللمساعدة هذه مواقع تحتوي أمثلة رائعة [الموقع الأول](#) [الموقع الثاني](#) (هذه الروابط قمت بجلبها من عملية البحث التي أخبرتك بها من جوجل .. مباشرة ^\_\*)

# Transition & Animation

والآن وبعد حديثنا عن الخصائص .. هناك نقطة مهمة، وهي أنني اذا احتجت الى كتابة هذه الخصائص جميعها .. فهذا يعني أنني أحتاج لكتابة أوامر ال animattion 8 مرات (عدد الخصائص) واذا أردنا أن تدعم المتصفحات مثل ال -webkit- مع الصيغة القياسية فسيصبح لدينا  $8 * 2 = 16$  سطر .. لاحظ الشريحة السابقة .. قمنا بكتابة 8 أسطر .. هذا الأمر مزعج حقيقة .. لذلك وجدت طريقة جميلة وسهلة تختصر علينا هذا العناء وهي باستخدام ال Shorthand Animation ... هل تذكر أول مثال تطرقنا له عند حديثنا عن ال keyframe وقلنا سنشرح السطر فيما بعد .. وللتذكير فقط .. سأضعها هنا .. شاهد:

```
.ex-43 .ex-43-1{
```

```
-webkit-animation: animate-example-1 5s linear 100ms infinite alternate;  
animation: animate-example-1 5s linear 100ms infinite alternate;
```

```
}
```



سيتم شرحها فيما بعد بإذن الله تعالى .. الآن فقط قم بتطبيق المثال كما تشاهده ..



# Transition & Animation

هذه الطريقة المختصرة كما تلاحظ ... لا نستطيع التطرق اليها أو شرحها الى بعد شرح الخصائص كل واحدة على حدة، وهذا مهم جدا ..

والآن ما هي الصيغة العامة للطريقة المختصرة: (الترتيب مهم جدا)

```
E {  
  animation: name duration timing-function delay iteration-count direction fill-mode play-state;  
}
```

القيم الافتراضية للصيغة السابقة هي (طبعاً الاسم والمدة اجباري عدا ذلك فهو اختياري)... شاهد القيم الافتراضية بالترتيب..

```
E {  
  animation: none 0 ease 0 1 normal none running;  
}
```

# Transition & Animation

مثال 1:

```
.ex-44 .ex-44-11{  
  -webkit-animation: animate-example-44-9 4s ease 0s 2 alternate forwards running;  
  animation: animate-example-44-9 4s ease 0s 2 alternate forwards running;  
}
```

الصيغة القياسية .. بكامل الخصائص والقيم الافتراضية ..

هنا قمت باختصار القيم التي لا أحتاجها مثل ال

**ease & running**

لأنها الخصائص الافتراضية

```
.ex-44 .ex-44-12{  
  -webkit-animation: animate-example-44-9 4s 0s 2 alternate forwards;  
  animation: animate-example-44-9 4s 0s 2 alternate forwards;  
}
```

طبعاً هنا لم استطع اختزال ال 0 لأنني سأستخدم تكرار. هذه العملية حتى يميز بين التكرار وزمن التأخير

ناتج تطبيق هذا المثال هو ذاته هذا المثال:

```
.ex-44 .ex-44-10{  
  -webkit-animation-name: animate-example-44-9;  
  -webkit-animation-duration: 4s;  
  -webkit-animation-iteration-count: 2;  
  -webkit-animation-direction: alternate;  
  -webkit-animation-fill-mode: forwards;  
  animation-name: animate-example-44-9;  
  animation-duration: 4s;  
  animation-iteration-count: 2;  
  animation-direction: alternate;  
  animation-fill-mode: forwards;  
}
```

# Transition & Animation

مثال 2:

```
.ex-44 .ex-44-13{  
  -webkit-animation: animate-example-44-9 2s forwards;  
  animation: animate-example-44-9 2s forwards;  
}
```

لاحظ هذا المثال.. فإني اخترت الصيغة بشكل سهل، وبما أنني لا أحتاج إلى ال delay ولا يوجد iteration لم أقم بوضع 0 أو 1 كقيم افتراضية لهم، لذلك أحببت أن أنه إلى هذه الفكرة ..

والآن لننتقل بإذن الله تعالى إلى فكرة جديدة وهي ... ^\_^

# Transition & Animation

ال Multiple Animations .. ^\_^

نعم يا صديقي العزيز ^\_^، فال Animation يسمح لك باستخدام أكثر من animation لنفس العنصر، وهذه من الخصائص الرائعة فعلا ^\_^\*..  
ولكتابة أكثر من animation فقط كل عا عليك القيام به هو السير على هذه الصيغة العامة: ^\_^\* ( تشبه ال Transition )

في حال كانت كل خاصية على حدى

```
E {  
  animation-name: first-anim, second-anim;  
  animation-duration: 6s, 1250ms;  
  animation-delay: 0, 750ms;  
}
```

في حال استخدام الطريقة المختصرة

```
E { animation: first-anim 6s, second-anim 1250ms 750ms; }
```

# Transition & Animation

مثال:

```
].ex-44 .ex-44-14 {  
  -webkit-animation: animate-example-44-9 4s 0s 5 alternate,  
  animate-example-44-14 4s forwards;  
  animation: animate-example-44-9 4s 0s 5 alternate,  
  animate-example-44-14 4s forwards;  
}
```

```
@-webkit-keyframes animate-example-44-9 {  
  from {  
    background-color: gray;  
    margin-left: 0px;  
  }  
  50% {  
    background-color: black;  
  }  
  to {  
    background-color: yellow;  
    margin-left: 800px;  
  }  
}
```

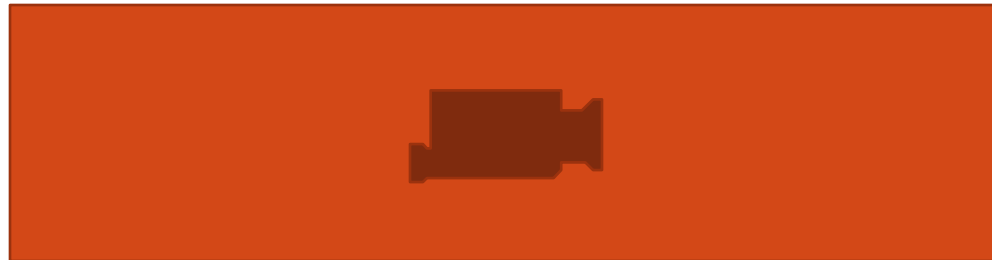
```
@-webkit-keyframes animate-example-44-14 {  
  from {  
    color: brown;  
    border: 3px solid green;  
  }  
  to {  
    color: red;  
    border: 8px solid red;  
  }  
}
```

# Transition & Animation

هكذا بحمد الله تعالى .. نكون انتهينا من موضوع ال Animation، وبفضل الله تعالى، نكون قد تطرقنا لموضوع قد يكون من أكثر المواضيع أهمية ..

والآن جميع الأمثلة السابقة يمكنك مشاهدتها مباشرة من هنا، لكن أحيطك علما لكثرتها وبسبب تواجدها بنفس الصفحة .. عليك أن تسير على الأمثلة جزء جزء .. بعد ذلك إما أن تحذف الأجزاء التي تنتهي من دراستها لتشاهد المثال الذي بعدها بوضوح .. أو أن تقوم بعمل refresh بعد قيامك بال

..\*\_^ scroll



حاسبوا أنفسكم قبل أن تحاسبوا ولا يدع قوم الجهاد في سبيل الله إلا ضربهم  
الله بالفقر و لا ظهرت الفحشاء في قوم إلا عمهم الله بالبلاء

أبو بكر الصديق رضي الله عنه

# 3D Transformations

بعد حديثنا الجميل عن موضوع ال 2D Transformations فإننا سننتقل معا بإذن الله تعالى الى مفهوم جديد وهو ال 3D Transformations ..

كنا سابقا وفي أي أمر أو خاصية تطرقنا اليه من خصائص ال css نتطرق الى المحور x والمحور y فقط.. وعندما نتحدث عن العناصر كنا نكتفي فقط بذكر الطول والعرض .. لكن هذا لم يعد كافيا .. والتكنولوجيا تطورت أيضا .. فأصبح لدينا محور جديد وهو المحور z ولم نعد نكتفي فقط بالطول والعرض .. وإنما أيضا أصبحنا نهتم بالعمق (depth) ...  
هذه الإضافات الجديدة .. هي بالأصل ما يعني 3D .. أي ثلاثي الأبعاد \*\_ \*\_



# 3D Transformations

طبعا بما أننا نتحدث عن تقنية جديدة ومطورة حديثا في عالم الويب.. وبما أنها تتعامل على مستوى ثلاثي الأبعاد.. لإغن هذا يحتاج الى القيام بعمليات معالجة كبيرة من قبل الحاسوب، بالإضافة الى الحاجة الى وجود متصفح حديث ويدعم هذه الخصائص...

الذين قاموا بتقديم توصية لل w3c لإعتماد ال 3D Transformation هم جماعة ال webkit ^\_\*.. وبنائا على هذه التوصية والفائدة التي يمكن أن تقدمها مستقبلا تم اعتماد هذه التوصية ضمن ال W3C \*\_\* والبدء بالعمل وإن لم يكن على كافة القواعد التي وضعوها... والآن لنبدأ بعد هذه المقدمة البسيطة بهذا الموضوع الجميل على بركة الله تعالى... \*\_\*

# 3D Transformations

الآن .. لتتخيل كيف يمكن أن يكون  
شكل ال 3D أو العنصر اذا  
كان 3D عليك بتأمل الصورة  
التالي، والنظر الى المحاور  
ال  $3^*$

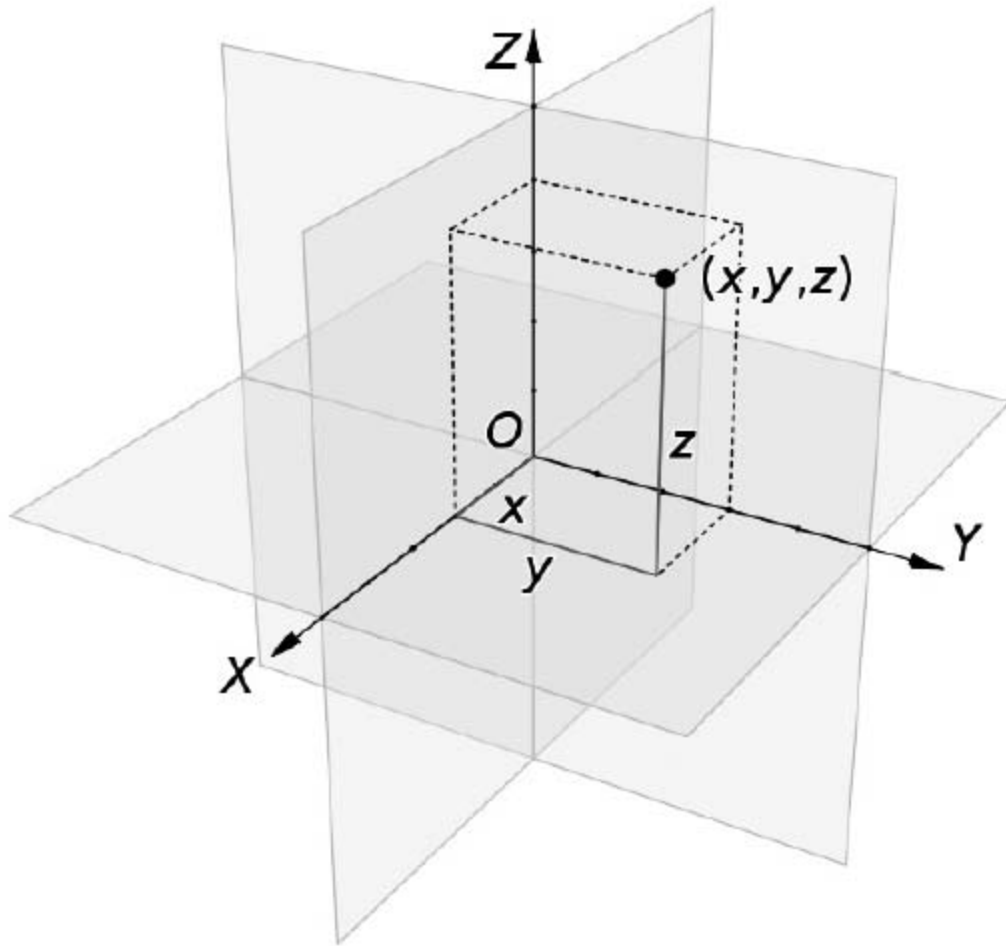


Figure 14-1: The Cartesian coordinate system, with the axes  $x$ ,  $y$ , and  $z^1$

# 3D Transformations

أول خاصية سنتحدث عنها بإذن الله تعالى هي ال Transform Style، هذه الخاصية رغم أنها بسيطة جدا، إلا أنك إن لم تقم بتعديل قيمتها فلن يعمل معك ال 3D Transformations أو لن يظهر التداخل الخاص بال 3D بشكل صحيح للعناصر ال Child (للأبناء) ^\_\*، أي سيعمل على parent فقط بالحالة الافتراضية ..

```
E { transform-style: keyword; }
```

الصيغة العامة:

وهناك قيمتان لهذه الصيغة العامة:

(1) flat: وهي الصيغة الافتراضية، وتعني أن الأبناء لن يتم تطبيق ال 3D عليهم

(2) preserve-3d: هذه القيمة تعني أن الأبناء سوف تحافظ على نمط ال 3D .. أي سيتم تطبيق ال 3D لهم ترادفا مع ال parent

# 3D Transformations

مثال 1: لاحظ... هذا الشكل يوضح الفرق بين الشكلين من خلال استخدام العنصر ال transform-style ... لاحظ التفاصيل بدقة ...



Figure 14-3: Comparing the effects of different values for the transform-style property

بعد مشاهدتك لهذا الشكل .. لنذهب لنشاهد مثال عملي على الموضوع في الشريحة التالية، وانتبه كيف لم يظهر تأخير للمربع الأول والثاني، واختلف الكلام لل 3 والرابع ... (لا تنسى النظر للشيفرة البرمجية)

# 3D Transformations

Anees Hikmat Anees Abu-hamid 2nees.com 1

Anees Hikmat Anees Abu-hamid 2nees.com 2

Anees Hikmat Anees Abu-hamid 2nees.com 3

Anees Hikmat Anees Abu-hamid 2nees.com 4

125deg



طبق المثال ^\_^:

Anees Hikmat Anees Abu-hamid 2nees.com 1

Anees Hikmat Anees Abu-hamid 2nees.com 2

Anees Hikmat Anees Abu-hamid 2nees.com 3

Anees Hikmat Anees Abu-hamid 2nees.com 4

-27deg

# 3D Transformations

والآن Transformation Functions.. طبعاً تحدثنا بموضوع ال 2D عن الدوال التي يمكن استخدامها مثل ال rotate وال scale.. الخ، لكن الآن سنعود للحديث عن الدوال لكن بوجود خصائص ال 3D ^\* \_

وأول خاصية هي ال 3D Rotate ^\* \_:  
الصيغة العامة:

```
E {  
  transform: rotateX(angle) ;  
  transform: rotateY(angle) ;  
  transform: rotateZ(angle) ;  
}
```

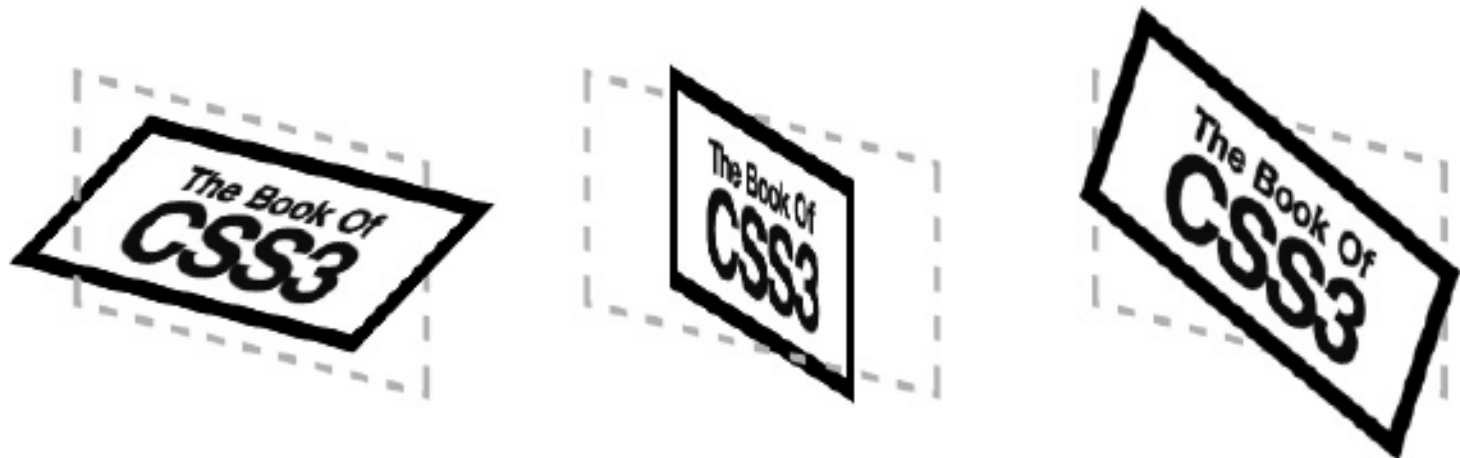
لنشاهد المثال الآن:

# 3D Transformations

مثال 1: لاحظ هنا أن كل من  $x$ ,  $y$ ,  $z$  يمثل محور.. لاحظ كيف شكل ال rotate في ال 3D

- 
- ❶ `.trans-x { transform: rotateX(-60deg); }`
  - ❷ `.trans-y { transform: rotateY(22.5deg); }`
  - ❸ `.trans-z { transform: rotateZ(22.5deg); }`
- 

You can see the results in Figure 14-4.



# 3D Transformations

مثال 2:

```
<!-- Example 2 -->
```

```
<div class="ex-46-2" style="-webkit-transform: rotateY(67deg) rotateX(67deg) rotateZ(67deg); transform: rotateY(67deg) rotateX(67deg) rotateZ(67deg);">
```

2D

3D





# 3D Transformations

طبعا يمكنك اختصار الأسلوب السابق على شكل `Rotate3d(x, y, z, angle)`.. لكن سأساعدك على عملية البحث والإطلاع على هذه الصيغة.. (الإختلاف فقط أن المتغيرات ستمثل نقاط على المتجه وال `angle` هي الزاوية ..)

أيضا الخصائص التي تعلمناها سابقا والتي لا تختلف عن الأسلوب الذي ذكرناه هي دالة ال `translate` وال `scale`..

```
E { transform: scale3d(scaleX, scaleY, scaleZ); }
```

```
E { transform: scale3d(0, 2, 5); }
```

```
E { transform: translate3d(translateX, translateY, translateZ); }
```

```
E { transform: translate3d(0px, 20px, 50px); }
```

# 3D Transformations

قد يكون تطبيق الخصائص السابقة على شكل 3D غير ظاهر لك بسهولة لكنها مهمة وتختلف عند تطبيق عملية فعلي يحتوي العديد من الخصائص .. وعندما قمت بوضع مثال ال rotate فكانت لتوضيح فكرة الإختلاف عن تطبيق معين  $^*_*$  ..  
الآن لمشاهدة الشكل 3D لل translate & scale ..



Figure 14-7: Scaling on different and multiple axes

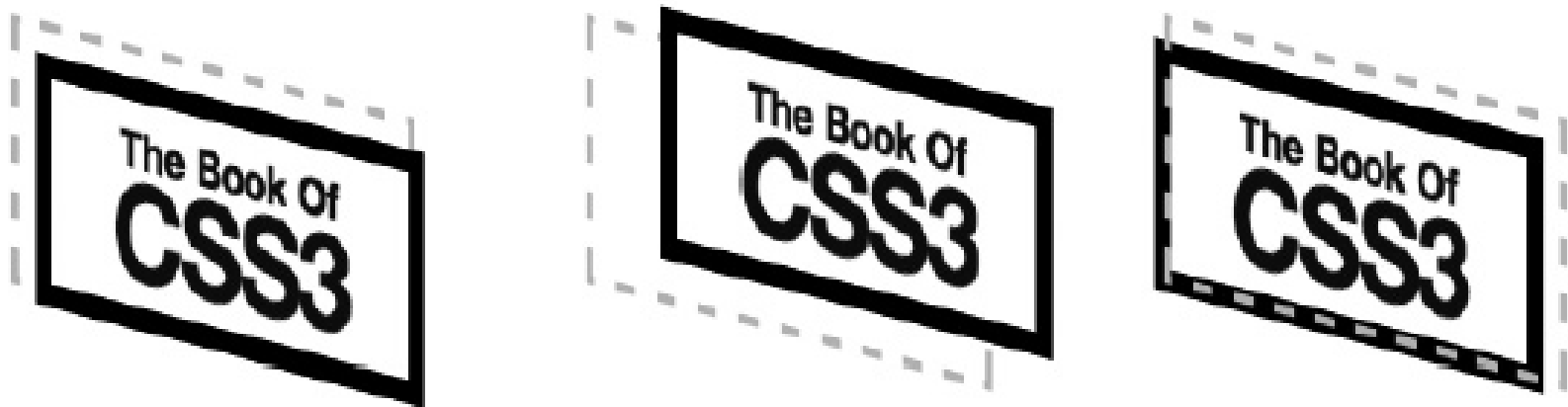
- ❶ `.trans-xy { transform: scaleX(1.5) scaleY(1.5); }`
- ❷ `.trans-z { transform: scaleZ(3) translateZ(10px); }`
- ❸ `.trans-xyz { transform: scale3d(1.25,1.25,4) translateZ(10px); }`

# 3D Transformations

وال `translate` .. لاحظ هنا، ان سبب أخذ هذه الصور من الكتاب كما هو، لتوضيح كيفية ظهور التنفيذ .. أو التطبيق ..

- 
- ❶ `.trans-xy { transform: translateX(10px) translateY(10px); }`
  - ❷ `.trans-z { transform: translateZ(40px); }`
  - ❸ `.trans-xyz { transform: translate3d(10px,0,-20px); }`
- 

You can see the results in Figure 14-6.



# 3D Transformations

Perspective: تقوم هذه الخاصية على تحديد مقدار العمق اللازم لعنصر معين عن زاوية النظر أو يمكننا القول بأنها موقع العنصر بالنسبة لموقع العرض ، شاهد الصيغة العامة:

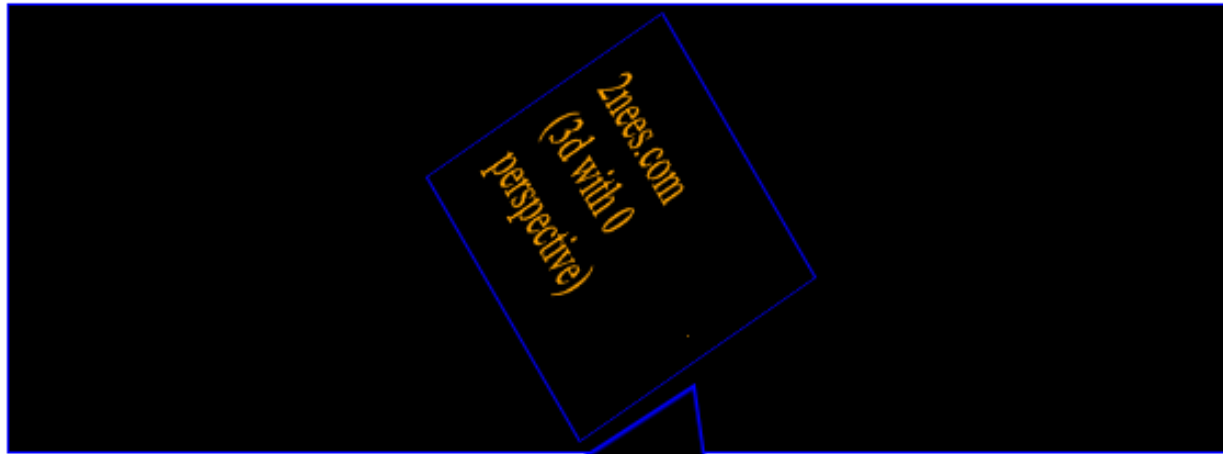
```
E { perspective: depth; }  
div { perspective: 500px; }
```

طبعا من المهم هنا أيضا استخدام ال vendor ، بالإضافة الى أن القيمة الافتراضية هنا هي none أو 0  
والآن .. لنشاهد مثلا  $\wedge\_ \wedge$

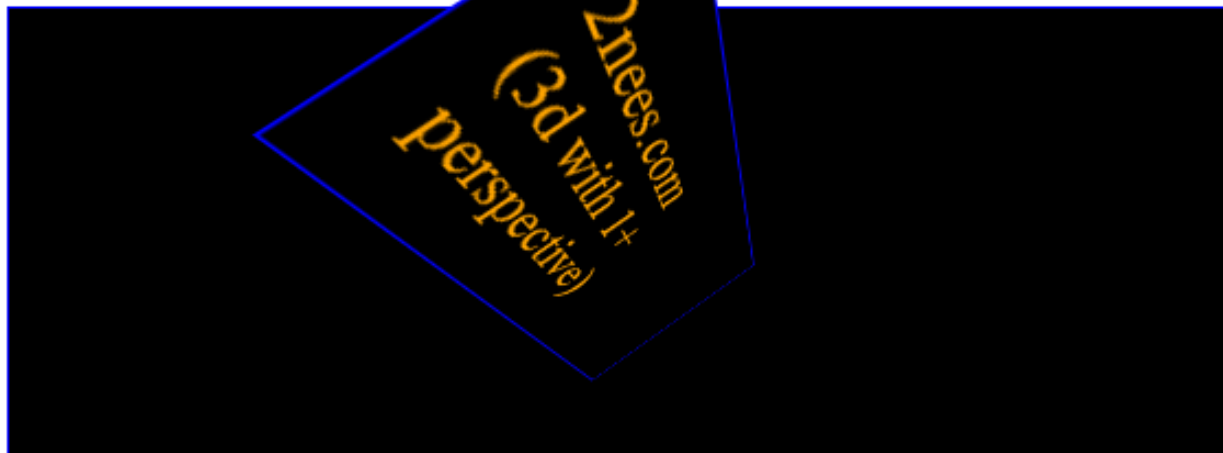
# 3D Transformations

مثال:

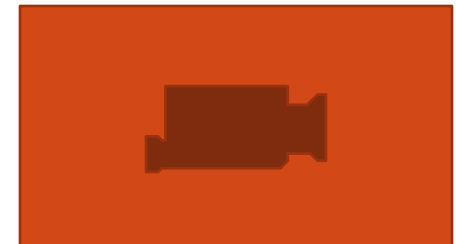
3d  
with  
no  
presp  
ective



3d  
with  
presp  
ective



شاهد المثال:



أَعُوذُ بِاللَّهِ مِنَ الشَّيْطَانِ الرَّجِيمِ  
يَا أَيُّهَا الَّذِينَ آمَنُوا إِذَا قِيلَ لَكُمْ تَفَسَّحُوا فِي الْمَجَالِسِ فَأْفْسَحُوا يَفْسَحِ اللَّهُ لَكُمْ ۗ  
وَإِذَا قِيلَ انشُرُوا فَانشُرُوا يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ  
دَرَجَاتٍ ۗ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ (11)  
صدق الله العظيم

سورة [ المجادلة : 11 ]

# الخاتمة

بحمد الله تعالى وفضله ومنه ورحمته علينا، نضع الآن قطراتنا الأخيرة على هذه الدورة المتواضعة، سائلين المولى عز وجل أن نكون وفقنا فيما فيه خير لنا ولكم، فقد كانت هذه المسيرة فيها من الجهد والعمل الكثير والحمد لله، وهذا الجهد يجب أن يتوافق بين طرفين ..الأول من الكتاب والثاني منك أنك، وبما أنك وصلت الى هنا، فنستطيع القول بأنك قد أجتزت جزءا كبيرا من عالم تتساقط الويب بإذن الله تعالى، لكن ما هذا الى نقطة انطلاق لك، وهناك المزيد والمزيد مما عليك أن تتعلمه ووتقنه، وفقنا الله وإياكم لكل خير...

أيضا أريد التنويه الى نقطة مهمة، وان هذا العمل عمل بشري، يصيب ويخطئ، فإن أصبنا فبفضل من الله تعالى، وإن أخطئنا فمن أنفسنا، ولا أزيد

على مقال عماد الاصفهاني:

رايت انه لا يكتب انسان كتابا في يومه الا قال في غده لو غير هذا لكان احسن

ولو زيد كذا لكان يستحسن

ولو قدم هذا لكان افضل ولو ترك هذا لكان اجمل وهذا من اعظم العبر وهو

دليل على استيلاء النقص على جملة البشر..

وصل اللهم وسلم على الحبيب المصطفى محمد صل الله عليه وسلم

اللهم اغفر لي ولوالدي وارحمهما كما ربياني صغيرا  
اللهم اغفر للمسلمين والمسلمات .. الأحياء منهم والأموات إنك سميع رحيم  
مجيب الدعاء

اللهم ارزقنا وارزق المسلمين الرزق الحلال الطيب الكثير المبارك واجعلنا  
ربنا بهذا الرزق أقرب إليك ..

اللهم آمين.

أخوكم أنيس حكمت أبو حميد.

2nees.com

aneeshikmat@2nees.com



وآخر دعوانا أن الحمد لله رب العالمين