中山大學
SUN YAT-SEN UNIVERSITY

Lecture 04. HTML & CSS III

Modern Web Programming
(http://my.ss.sysu.edu.cn/wiki/display/WEB/ **supported by Deep Focus**)

School of Data and Computer Science, Sun Yat-sen University

HTML5 elements

aritcle

表 2-4 article 元素

HTML 元素名	用途	语法
article	文章	<article>文章内容</article>

源代码 2-8 article 元素示例

```
<h1>会当凌绝顶</h1>
<article>
  <h2>五岳</h2>
  <h3>泰山：会当凌绝顶，一览众山小</h3>
  <p>  泰山……</p>
  ……
</article>
<article>
  <h2>四大佛教名山</h2>
  <h3>峨眉山：峨眉天下秀</h3>
  <p> 峨眉山……</p>
</article>
```

header

表 2-5 header 元素

HTML 元素名	用途	语法
header	文章或其它内容的头部	<header>头部内容</header>

源代码 2-9 header 元素示例

```
<h1>会当凌绝顶</h1>
<article>
  <header>
    <h2>五岳</h2>
    <h3>-- 五岳小知识</h3>
  </header>
  <h3>泰山：会当凌绝顶，一览众山小</h3>
  <p>泰山……</p>
  ……
</article>
```

hgroup

表 2-6 hgroup 元素

HTML 元素名	用途	语法
hgroup （header group）	标题组	<hgroup> …. … </hgroup>

源代码 2-10 hgroup 元素示例

```
<h1>会当凌绝顶</h1>
<article>
  <header>
    <hgroup>
      <h2>五岳</h2>
      <h3>-- 五岳小知识</h3>
    </hgroup>
  </header>
  <h3>泰山：会当凌绝顶，一览众山小</h3>
  <p>泰山……</p>
```

footer

表 2-7 footer 元素

HTML 元素名	用途	语法
footer	文章或其它内容的脚部	<footer>脚部内容</footer>

源代码 2-11 footer 元素示例

```
<h1>会当凌绝顶</h1>
<article>
  <header>……</header>
  <h3>泰山……</h3>
  <p>……</p>
  <h3>华山……</h3>
  <p>……</p>
  <footer>晓鸣收集整理</footer>
</article>
……
```

section

表 2-8 section 元素

HTML 元素名	用途	语法
section	文章或其它内容的章节（部分）	<section>章节内容</section>

源代码 2-12 section 元素示例

```
<h1>会当凌绝顶</h1>
<article>
  <header>……</header>
  <section>
    <h3>泰山……</h3>
    <p>……</p>
  </section>
  <section>
    <h3>华山……</h3>
    <p>……</p>
  </section>
  <footer>晓鸣收集整理</footer>
</article>
……
```

aside

表 4-1 aside 元素

HTML 元素名	用途	语法
aside	网页边栏	<aside>边栏内容</aside>

源代码 4-1 aside 元素示例

```
……
<h3>泰山：会当凌绝顶，一览众山小</h3>
<aside>
    <blockquote>
    岱宗夫如何？齐鲁青未了。<br />
    ……
    -- 《望岳》 杜甫
</blockquote>
</aside>
<p> 泰山是中国五岳之首，……</p>
……
```

Semantic Web

article、header、hgroup、footer、section、aside 等元素的使用，并不会直接改变浏览器中网页的呈现。但是，这些元素说明了网页的结构，有利于我们使用样式表定义网页外观和使用脚本操纵相应元素，同时也有利于各种工具分析网页，获取信息。

detail & summary

表 2-9 details 元素、summary 元素

HTML 元素名	用途	语法
details	可隐藏的细节内容	<details> <summary>摘要</summary> 细节内容…… </details>
summary	细节内容的摘要	

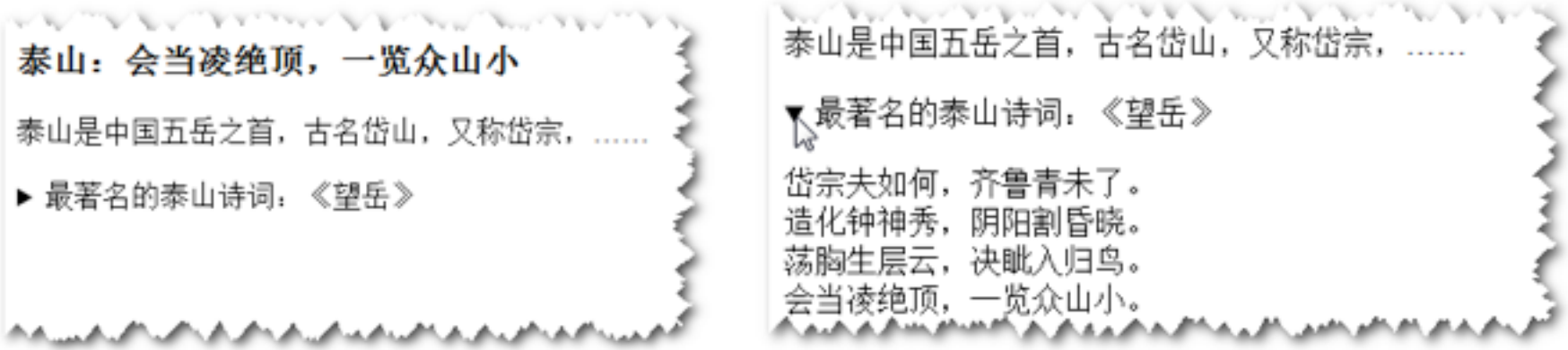


图 2-8 源代码 2-13 运行效果（左：点击打开 details 前；右：点击打开后）

detail & summary

源代码 2-13 details 和 summary 元素示例

```
<h3>泰山：会当凌绝顶，一览众山小</h3>
<p>泰山是中国五岳之首，古名岱山，又称岱宗，……</p>
<details>
    <summary>最著名的泰山诗词：《望岳》</summary>
    <p>岱宗夫如何，齐鲁青未了。<br/>
    造化钟神秀，阴阳割昏晓。<br/>
    荡胸生层云，决眦入归鸟。<br/>
会当凌绝顶，一览众山小。</p>
</details>
```

nav

表 4-1 nav 元素

HTML 元素名	用途	语法
nav	网页的导航栏	<nav>*导航链接*</nav>

源代码 4-1 nav 元素示例

```
<nav>
  <a href="/ermei/">峨眉山</a> |
  <a href="/wutai/">五台山</a> |
  <a href="/putuo/">普陀山</a> |
  <a href="/jiuhua/">九华山</a>
</nav>
```

图 4-2 源代码 4-1 运行效果

Cat Lovr BETA

<header>

Home About Contact <nav> Search

Italics Kittehs <article>

Posted 29 May 2008

Hungry kittens

42 Comments | Permalink

Rating:

Archives

<aside>

<footer>

Graphic vs. Image

图形（Graphic） vs. 图像（Image）

计算机中用两种不同的方式来表达图片。一种方式认为图片由直线、圆、矩形、曲线等轮廓线和着色区域构成，用一系列矢量表示图片，称之为图形。另一种认为图片为一系列的点和其颜色构成，用点阵数组表示图片，称之为图像。

图形适合于表示几何图形、工程图纸、数学建模的各种模型等等。它表现的对象轮廓不很复杂，色彩不是很丰富。图像适合于表现照片、绘画等含有大量明暗、场景、色彩细节，和复杂轮廓的对象。

图形描述轮廓线和着色区域，因此放大时不会失真。图像记录的是点阵信息，放大到一定程度，会产生锯齿状的失真。

svg canvas

表 2-32 svg 元素、canvas 元素

HTML 元素名	用途	语法
svg（scalable vector graphic）	嵌入图形（inline-block，参考 4.4.5b))	<svg xmlns="…" version="…"> ……svg 元素…… <svg>
canvas	嵌入图像（inline-block）	<canvas id="myCanvas"></canvas>

svg canvas

源代码 2-46 svg 元素示例

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <circle cx="100" cy="50" r="40" stroke="black" stroke-width="2"
fill="red" />
    <text x="60" y="110" fill="red">泰山落日圆</text>
</svg>
```

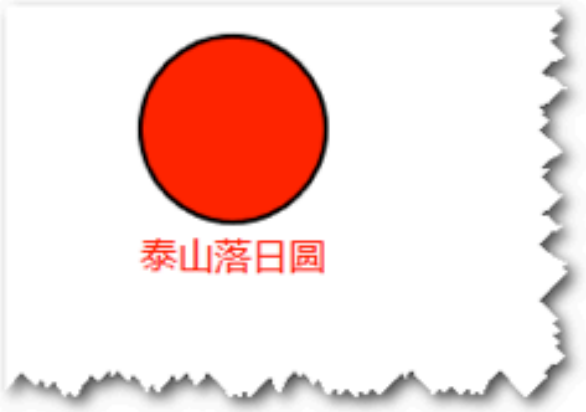


图 2-33 源代码 2-46 运行效果

svg canvas

源代码 2-47 canvas 元素示例

```
<canvas id="myCanvas"></canvas>

<script type="text/javascript">
var canvas=document.getElementById('myCanvas');
var ctx=canvas.getContext('2d'); // 绘制二维图形
ctx.fillStyle='#FF0000'; // 填充色为红色
ctx.beginPath();
// 圆弧从0度到2π（360度），圆心距页面左边100像素、上边50像素，半径40像素
ctx.arc(100,50,40,0,Math.PI*2,true);
ctx.closePath();
ctx.fill();
ctx.font = '16px 宋体'
// 文本位置距页面左边60像素、上边110像素
ctx.fillText('泰山落日圆',60, 110)
</script>
```

video source track

表 2-34 video 元素、source 元素、track 元素

HTML 元素名	用途	语法
video	嵌入网页的视频（inline-block，参考 4.4.5b））	<video width='...' height='...' ...> <source type='...' src='...'/> <source type='...' src='...'/> <track kind='...' src='...'/> </video>
source	多种视频/音频文件源（可选）	
track*	视频字幕（可选）	

video source track

源代码 2-48 video 元素、source 元素示例

```
<video width="500" height="375" controls autoplay
poster="avs/taishan.jpg">
    <source type="video/mp4" src="avs/taishan.mp4" />
    <source type="video/webm" src="avs/taishan.webm" />
</video>
```

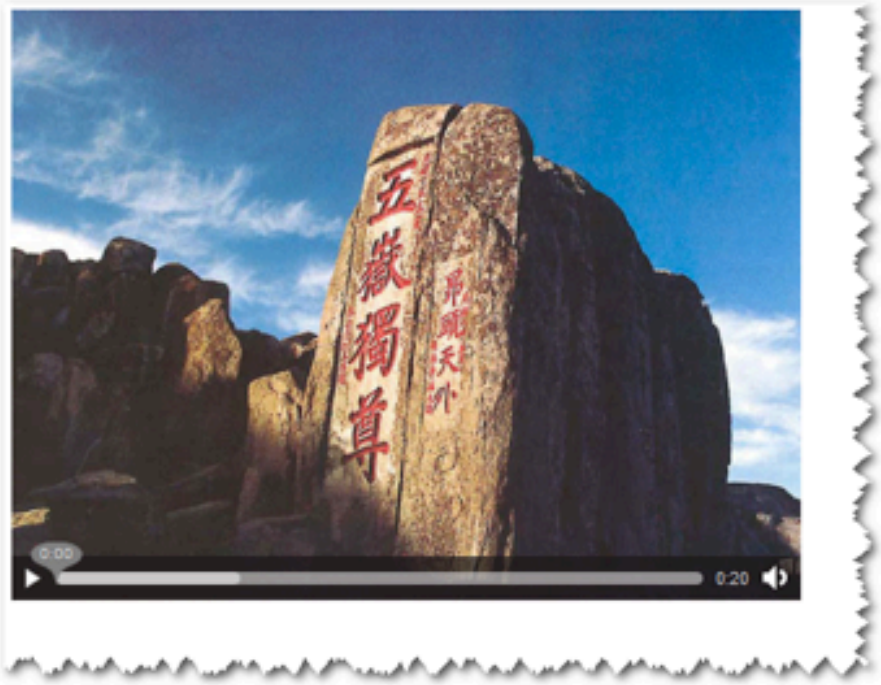


图 2-35 源代码 2-48 运行效果（左：视频播放前；右：视频播放中）

audio

源代码 2-50 audio 元素示例（可能需要修改 Web 服务器 MIME 配置，见 b)节）

```
<audio controls autoplay>
    <source type="audio/mp3" src="avs/Adele-Rolling.mp3" />
    <source type="video/webm" src="avs/Adele-Rolling.ogg" />
</audio>
```

CSS 3

CSS3 Borders

Introduction

With CSS3, you can create rounded borders, add shadow to boxes, and use an image as a border - without using a design program, like Photoshop

Browser Support

Property	Browser Support				
border-radius					
box-shadow					
border-image		-moz-	-webkit-	-webkit-	-o-

CSS3 Borders

Border Syntax

border-(sub-property): (top) (right) (bottom) (left);

sub-property	we can use width, color, style and radius as sub-properties of Border.
top	This value gets assigned to the top portion of the border
right	This value gets assigned to the right portion of the border
bottom	This value gets assigned to the bottom portion of the border
left	This value gets assigned to the left portion of the border

Here in sub-property we can use width, color, style and radius as sub-types of Border.
Four values need be to defined.

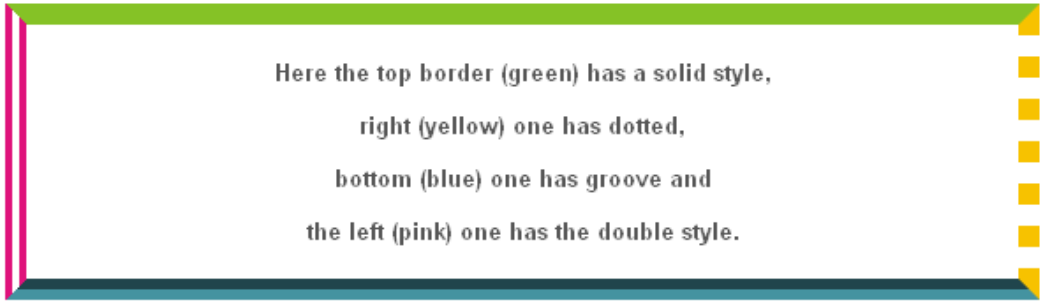
CSS3 Borders

Border Color

Syntax : border-color: (top) (right) (bottom) (left);
Example: border-color: #85C226 #F7C200 #4493A0 #DF127B ;

Border Styles Important

This property must be used if you are using any of the Border properties.
Without using this border-style property you won't see any borders at all.
Syntax : border-style: (top) (right) (bottom) (left);
Example: border-style: solid dotted groove double;

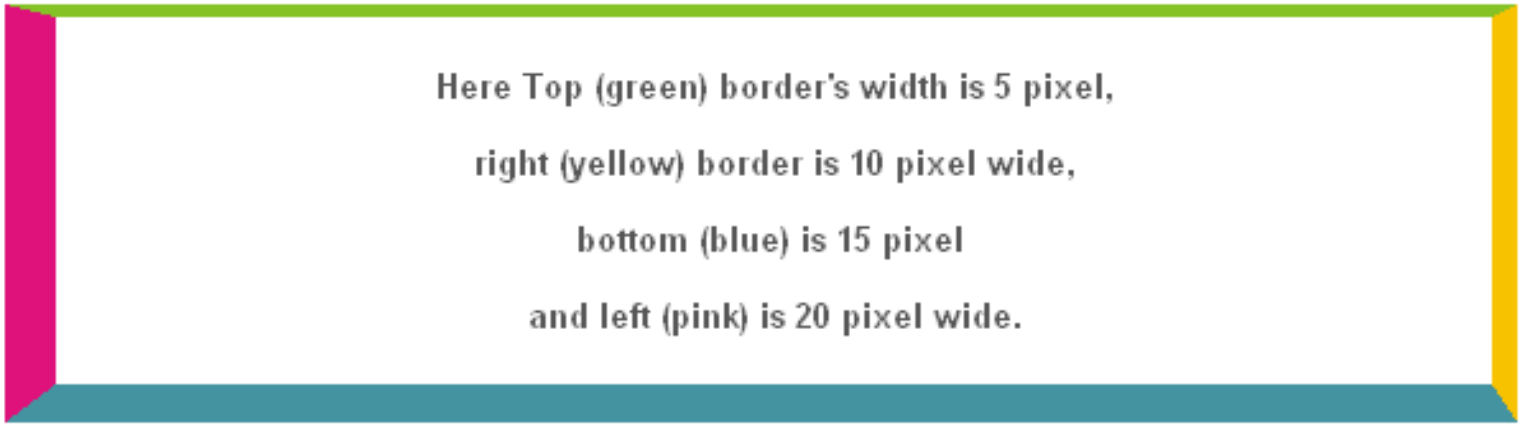


CSS3 Borders

Border Width

Syntax : border-width: (top) (right) (bottom) (left);

Example: border-width: 5px 10px 15px 20px;



CSS3 Borders

Border Radius

Syntax : border-radius: (top-left) (top-right) (bottom-right) (bottom-left);
Example: border-radius: 0px 30px 70px 20px;

For displaying round corners Border Radius property is used.
IE9 and Opera use the same code. For Firefox a prefix " **-moz-** " needs to be added before the code. And for Webkit Browsers " **-webkit-** " needs to be added before the code.

Box-shadow property for all the browsers.

For IE and Opera	box-shadow:0px 30px 70px 20px;
For Mozilla Firefox	-moz-box-shadow:0px 30px 70px 20px;
For Webkit Browsers	-webkit-box-shadow:0px 30px 70px 20px;

No Border radius (0px) has been applied on the top-left (green-pink) corner.
Border radius (30px) has been applied on the top-right (green-yellow) corner.
Border radius (70px) has been applied on the bottom-right (blue-yellow)corner.
Border radius (20px) has been applied on the bottom-left (blue-pink) corner.

CSS3 Borders

Hemisphere

Use values of width, height and border as defined above. Only add these codes to your existing stylesheet or make a new one.


```
#hemisphere{
    border-radius:100px 100px 0 0 ;
    -moz-border-radius:100px 100px 0 0 ;
    -webkit-border-radius:100px 100px 0 0 ;
 }
```

CSS3 Borders

Advanced Hemisphere

Now we'll add more effects to the border by applying different values for width and using different border style.


```
#adv-hemisphere{
    border-width: 4px 30px 4px 30px;
    border-style: groove ridge dashed groove;
    border-color: #cc0000;
    border-radius:100px 100px 0 0 ;
    -moz-border-radius:100px 100px 0 0 ;
    -webkit-border-radius:100px 100px 0 0 ;
 }
```

CSS3 Borders

Round Rectangle

Here all the corners have the same border radius (20px).


```
#round-rectangle{
    border-radius: 20px;
    -moz-border-radius: 20px;
    -webkit-border-radius: 20px;
 }
```

CSS3 Borders

Advanced Round Rectangle

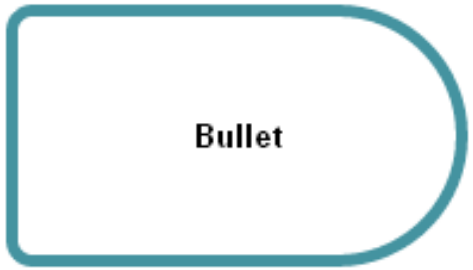
We'll make top border's and bottom border's width zero pixel, Use same border-radius (40px) for both top corners and for both the bottom corners (60px).


```
#adv-round-rectangle{
    border-width: 0px 12px 0px 12px;
    border-style: double;
    border-color: violet;
    border-radius:40px 40px 60px 60px;
    -moz-border-radius:40px 40px 60px 60px;
    -webkit-border-radius:40px 40px 60px 60px;
 }
```

CSS3 Borders

Bullet

Same border radius (20px) is assigned to the corners on the left, and same border radius (100px) is assigned to the right corners.




```
#bullet{
    border-radius: 20px 100px 100px 20px;
    -moz-border-radius: 20px 100px 100px 20px;
    -webkit-border-radius: 20px 100px 100px 20px;
 }
```

CSS3 Borders

Advanced Bullet

Here Groove style border and different colors for each border is used and the code is same as above.


```
#adv-bullet{
    border-width: 12px;
    border-style: groove;
    border-color: red blue green black;
    border-radius:20px 100px 100px 20px;
    -moz-border-radius:20px 100px 100px 20px;
    -webkit-border-radius:20px 100px 100px 20px;
}
```

CSS3 Borders

Advanced Leaf

Background color (orange) is used for the . Solid style border is used here same color (blue) for the top and right border and same color (green) is used for bottom and left border.


```
#adv-leaf{
    background:orange;
    border-width:4px 30px 4px 30px;
    border-style: solid;
    border-color: #56c6d9 #56c6d9 #fe2192 #fe2192;
    border-radius:0 120px 0 120px;
    -moz-border-radius:0 120px 0 120px;
    -webkit-border-radius:0 120px 0 120px;
 }
```

CSS3 Background

we will be taking a look at the new background properties. These include background size, using more than one background for an element, and background origin (which effects the position of a background).

The new features allow greater control of the background element and will provide designers with a whole array of new possibilities.

Multiple Backgrounds

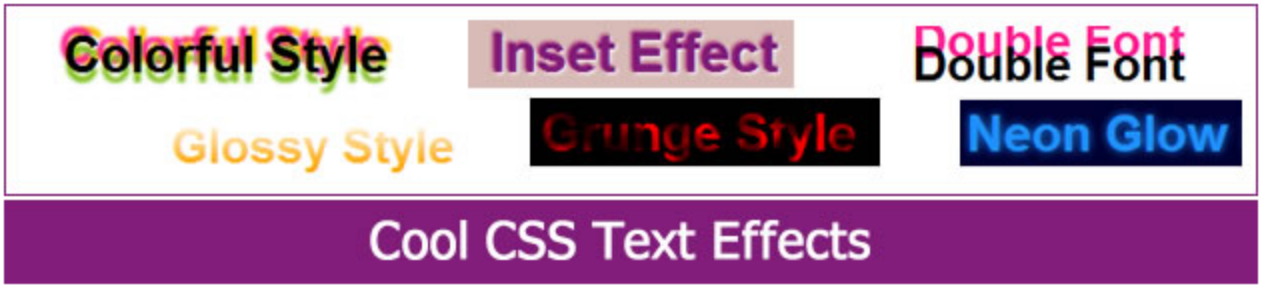
The new ability to use multiple backgrounds is a great time saver, allowing you to achieve effects which previously required more than one div. Whether it will be possible to combine this with background-size will be interesting to see.

The example below uses one background for the top border, one repeated vertically for the left and right borders and a third at the bottom.

```
1.multiplebackgrounds {
2height: 150px;
3width: 270px;
4padding: 40px 20px 20px 20px;
5background: url(top.gif) top left no-repeat,
6url(bottom.gif) bottom left no-repeat,
7url(middle.gif) left repeat-y;
8}
```

CSS3 Text effects

In this tutorial we will learn how to use CSS Styles to give stylish effects to text.
Learn how to create shadow effects, inset effects, neon style effects, glossy styles, grunge style effects and more!



CSS3 Text effects

The HTML code

Just put your text inside any " div " or " span "and give it an id="style". For the example below, we are working with the text "CSS Text Effects".

```
1   <div id="style">
2
3      CSS Text Effects
4
5   </div>
```

Apply shadow to the text inside the with the id "style".

The Syntax

text-shadow:(x-offset) (y-offset) (blur-radius) (color)

X-Offset: To position the shadow along the x-axis.
Y-Offset: To position the shadow along the y-axis.
Blur-radius: To set the amount of blur.
Color: To set the color of the shadow.

Here are some examples of the text effects using CSS.

CSS3 Text effects

The Colorful Style

Three colored shadows are used here. So we'll have to define three shadow values, each separated by a comma.


```
#style{
text-shadow: 0 6px 4px #85c226,
             -3px -5px 4px #fe2192,
             3px -5px 4px #f7c200;
}
```

The Double Font Style

A single shadow is used here. The Pink(#fe2192) colored shadow has a y-offset value set at -15 pixel.


```
#style{
text-shadow:0px -15px 0 #fe2192;
}
```

CSS3 Text effects

The Neon Glow Style

A single Blue(#1E90FF) colored shadow is used here with a Blur Radius of 7 pixel.
The text color is also the same Blue(#1E90FF) color.

CSS Text Effects

```
#style{
text-shadow: 0 0 7px #1E90FF;
background:#000000;
color: #1E90FF;
}
```

The Inset Effect

Two shadows are used here, each of them moved 1 pixel along the x-y axis in opposite directions.
The Background and the Text have the same Grey(#CCCCCC) Color.

The Dark Grey(#666666) colored shadow is moved to the top left corner.
The White(#FFFFFF) colored shadow is moved to the bottom right corner, to give it an inset look.

CSS Text Effects

CSS3 Text effects

The Outset Style

We'll do the exact opposite of what we did for the Inset Effect.

The Dark Grey(#666666) colored shadow is moved to the bottom right corner.
The White(#FFFFFF) colored shadow is moved to top left corner, to give it an inset look.

CSS Text Effects

```
#style{
background: #CCCCCC;
color: #CCCCCC;
text-shadow: 1px 1px 3px #666666,
             -1px -1px 3px #FFFFFF;
}
```

CSS3 fonts

→ The CSS Code

The Syntax:

The

```
@font-face {
 font-family:font-name;
 src: url("folder-name/font");
}
```

Explanation:

@font-face {

→ With this code we will define a new font-name and include the fonts that we have downloaded using src.

font-family:font-name;

→ Here any name can be assigned to the font.

src: url("folder-name/font");

→ The location of the font (in this case the folder named *fonts*)

The HTML Code

Lets change the fonts inside a div with id "change".

Here's the div element with some text.

```
<div id="change">
   Apply new font on this text.
</div>
```

CSS3 fonts

The CSS Code

Now to change the font of the text inside the div element with id "change", we'll have to define the new fonts and use them with font-family property.

We'll have to define the new font twice.

Once for IE and once for other CSS Browsers.

Suppose we downloaded the font Rockfont, then we will have two fonts.

One will have an extension **TTF (rockfont.ttf)** and the other **EOT (rockfont.eot)**.

The

```
1   @font-face {
2    font-family:rockfont;
3    src: url("fonts/rockfont.eot"); /* EOT file for IE */
4   }
5
6   @font-face {
7    font-family:rockfont;
8    src: url("fonts/rockfont.ttf"); /* TTF file for CSS3 browsers */
9   }
10
11  #change{
12   font-family:rockfont;
13  }
14
```

CSS3 fonts

Explanation

Line 1 & Line 6 › @font-face {

→ Used to define new font family.

Line 2 & Line 7 › font-family:rockfont;

→ Font name is assigned here (you can use any name here).

Line 3 › src: url("fonts/rockfont.eot");

→ Location of the font file with respect to this html file.

→ Notice the extension "eot". This is for IE.

Line 8 › src: url("fonts/rockfont.ttf");

→ Location of the font file with respect to this html file.

→ Notice the extension "eot". This is for all the browsers.

Line 11 › #change{ ;

→ The id of the div.

Line 12 › font-family:rockfont;

→ Newly Created font-family *rockfont* which is going to be assigned to the div with id "change" .

→ All the text inside this div will have the same rockfont font.

The Result

CSS3 shadows

These effects can be made using the **Css Box Shadow Property**. There are two types of shadow effect that are possible using Css Box Shadow Property. Look at the images above.
The first one has Css Shadow Property applied on the outer side of the div element.
The Second one has Css Shadow Property applied to it on the inside of div element.

This Property works for IE9, Opera, Firefox and Webkit browsers like Safari, Chrome.
IE9 and Opera use simillar code to display the shadow while for Firefox and Webkit Browsers, a prefix needs to be added before the code.

Lets have a look at the code:

To create Outer Shadow:

box-shadow:X-Offset Y-Offset Blur Blur-Offset Color;
Ex: box-shadow: 0px 0px 20px 10px #00CED1;

To create Inner Shadow:

box-shadow:Inset X-Offset Y-Offset Blur Blur-Offset Color;
Ex: box-shadow:inset 0px 0px 20px 10px #00CED1;

CSS3 shadows


```
#example0{
    box-shadow:0px 0px 20px 0px #000;
    -moz-box-shadow:0px 0px 20px 0px #000;
    -webkit-box-shadow:0px 0px 20px 0px #000;
}
```


```
#example1{
    box-shadow:0px 0px 20px 10px #000;
    -moz-box-shadow:0px 0px 20px 10px #000;
    -webkit-box-shadow:0px 0px 20px 10px #000;
}
```

CSS3 shadows


```
#example2{
    box-shadow:20px 0px 20px 10px #000;
    -moz-box-shadow:20px 0px 20px 10px #000;
    -webkit-box-shadow:20px 0px 20px 10px #000;
}
```


```
#example3{
    box-shadow:20px 20px 20px 10px #000;
    -moz-box-shadow:20px 20px 20px 10px #000;
    -webkit-box-shadow:20px 20px 20px 10px #000;
}
```

CSS3 shadows

Inner Shadows

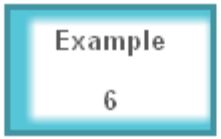
Example 4

```
#example4{
    box-shadow:inset 0px 0px 20px 0px #000;
    -moz-box-shadow:inset 0px 0px 20px 0px #000;
    -webkit-box-shadow:inset 0px 0px 20px 0px #000;
}
```

Example 5

```
#example5{
    box-shadow:0px 0px 10px 5px #000;
    -moz-box-shadow:0px 0px 10px 5px #000;
    -webkit-box-shadow:0px 0px 10px 5px #000;
}
```

CSS3 shadows




```
#example6{
    box-shadow:inset 5px 0px 5px 5px #222;
    -moz-box-shadow:inset 5px 0px 5px 5px #222;
    -webkit-box-shadow:inset 5px 0px 5px 5px #222;
}
```


```
#example7{
    box-shadow:inset 5px 5px 5px 5px #222;
    -moz-box-shadow:inset 5px 5px 5px 5px #222;
    -webkit-box-shadow:inset 5px 5px 5px 5px #222;
}
```

CSS3 Transforms & Transitions

Transforms

CSS3 transform property lets you translate, rotate, scale, or skew any element on the page.
While some of these effects were possible using previously existing CSS features
(like relative and absolute positioning), CSS3 gives you unprecedented control over many more aspects of an element's appearance.

Translation

Translation functions allow you to move elements left, right, up, or down.

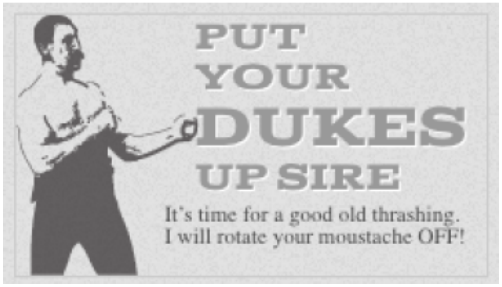
These functions are similar to the behaviour of position: relative; where you declare top and left.
When you employ a translation function, you're moving elements without impacting the flow of the document.

Unlike position: relative, which allows you to position an element either against its current position or against a parent or other ancestor, a translated element can only be moved relative to its current position.

The translate(x,y) function moves an element by x from the left, and y from the top:

CSS3 Transforms & Transitions

Examples



let’s say we want to move the word “dukes” over to the right when the user hovers over it,

```
#ourExample h1:hover span {
color: #484848;
-webkit-transform: translateX(40px);
-moz-transform: translateX(40px);
-ms-transform: translateX(40px);
-o-transform:translateX(40px);
transform: translateX(40px);
}
```

CSS3 Transforms & Transitions

Scaling

The scale(x,y) function scales an element by the defined factors horizontally and vertically, espectively. If only one value is provided, it will be used for both the x and y scaling.

Lets scale the same example

```
#ourExample h1:hover span {
color: #484848;
-webkit-transform: translateX(40px) scale(1.5);
-moz-transform: translateX(40px) scale(1.5);
-ms-transform: translateX(40px) scale(1.5);
-o-transform:translateX(40px) scale(1.5);
transform: translateX(40px) scale(1.5);
}
```

CSS3 Transforms & Transitions

Rotation

The rotate() function rotates an element around the point of origin (as with scale,by default this is the element’ s center), by a specified angle value. Generally, angles are declared in degrees, with positive degrees moving clockwise and negativemoving counter-clockwise.

Lets scale the same example

```
#ourExample h1:hover span {
color: #484848;
-webkit-transform:rotate(10deg) translateX(40px) scale(1.5);
-moz-transform:rotate(10deg) translateX(40px) scale(1.5);
-ms-transform:rotate(10deg) translateX(40px) scale(1.5);
-o-transform:rotate(10deg) translateX(40px) scale(1.5);
transform:rotate(10deg) translateX(40px) scale(1.5);
}
```

CSS3 Transforms & Transitions

Skew

The skew(x,y) function specifies a skew along the X and Y axes. As you'd expect, the x specifies the skew on the X axis, and the y specifies the skew on the Y axis.

If the second parameter is omitted, the skew will only occur on the X axis:

```
-webkit-transform: skew(15deg, 4deg);
-moz-transform: skew(15deg, 4deg);
-ms-transform: skew(15deg, 4deg);
-o-transform: skew(15deg, 4deg);
transform: skew(15deg, 4deg);
```

Applying the above styles to a heading, for example, results in the skew shown in

As with `translate` and `scale`, there are axis-specific versions of the skew transform: `skewx()` and `skewy()`.

CSS3 Transforms & Transitions

Transitions

Transitions allow the values of CSS properties to change over time, essentially providing simple animations. For example, if a link changes color on hover, you can have it gradually fade from one color to the other, instead of a sudden change

Here are the steps to create a simple transition using only CSS:

1. Declare the original state of the element in the default style declaration.
2. Declare the final state of your transitioned element; for example, in a hover state.
3. Include the transition functions in your default style declaration, using a few different properties: transition-property, transition-duration,transition-timing-function, and transition-delay.

list of properties that can be transitioned

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

CSS3 Transforms & Transitions

Transition-duration

The transition-duration property sets how long the transition will take. You can specify this either in seconds (s) or milliseconds (ms). We'd like our animation tobe fairly quick, so we'll specify 0.2 seconds,

or 200 milliseconds:

```
-webkit-transition-duration: 0.2s;
-moz-transition-duration: 0.2s;
-o-transition-duration: 0.2s;
transition-duration: 0.2s;
```

CSS3 Transforms & Transitions

Transition-timing-function

The transition-timing-function lets you control the pace of the transition in even more granular detail.

Do you want your animation to start off slow and get faster, start off fast and end slowe

You can specify one of the key terms ease, linear, ease-in, ease-out, or easein-out.

The best way to familiarize yourself with them is to play around and try them all.

```
-webkit-transition-timing-function: ease-out;
-moz-transition-timing-function: ease-out;
-o-transition-timing-function: ease-out;
transition-timing-function: ease-out;
```

CSS3 Transforms & Transitions

Transition-delay

Finally, by using the transition-delay property, it’ s also possible to introduce a delay before the animation begins. Normally, a transition begins immediately, so the default is 0. Include the number of milliseconds (ms) or seconds (s) to delay the transition:

```
-webkit-transition-delay: 250ms;
-moz-transition-delay: 250ms;
-o-transition-delay: 250ms;
transition-delay: 250ms;
```

Negative Delays

Interestingly, a negative time delay that is less than the duration of the entire transition will cause it to start immediately, but it will start partway through the animation. For example, if you have a delay of -500ms on a 2s transition, the transition will start a quarter of the way through, and will last 1.5 seconds. This might be used to create some interesting effects, so it’s worth being aware of.

CSS3 Transforms & Transitions

Transition-shorthand property

With four transition properties and three vendor prefixes, you could wind up with 16 lines of CSS for a single transition. Fortunately, as with other properties, there's a shorthand available. The transition property is shorthand for the four transition functions described above. Let's take another look at our transition so far:

```
#ad2 h1 span {
  -webkit-transition-property: -webkit-transform, color;
  -moz-transition-property: -moz-transform, color;
  -o-transition-property: -o-transform, color;
  transition-property: transform, color;
  -webkit-transition-duration: 0.2s;
  -moz-transition-duration: 0.2s;
  -o-transition-duration: 0.2s;
  transition-duration: 0.2s;
  -webkit-transition-timing-function: ease-out;
  -moz-transition-timing-function: ease-out;
  -o-transition-timing-function: ease-out;
  transition-timing-function: ease-out;
}
```

shorthand

```
#ad2 h1 span {
  -webkit-transition: -webkit-transform 0.2s ease-out;
  -moz-transition: -moz-transform 0.2s ease-out;
  -o-transition: -o-transform 0.2s ease-out;
  transition: transform 0.2s ease-out;
}
```

Note that order of the values is important and must be as follows (though you don't need to specify all four values):

1. transition-property
2. transition-duration
3. transition-function
4. transition-delay

CSS3 Transforms & Transitions

Multiple transition

With four transition properties and three vendor prefixes, you could wind up with 16 lines of CSS for a single transition. Fortunately, as with other properties, there's a shorthand available. The transition property is shorthand for the four transition functions described above. Let's take another look at our transition so far:

It's possible to specify multiple transitions when using the shorthand `transition` property also. In this case, specify all the values for each transition together, and separate each transition with commas:

```
transition: color 0.2s ease-out, transform 0.2s ease-out;
```

CSS3 Animation

Transitions animate elements over time; however, they're limited in what they can do.
You can define starting and ending states, but there's no fine-grained control over any intermediate states. CSS animations, unlike transitions, allow you to control each step of an animation via **keyframes**. If you've ever worked with Flash, you're likely very familiar with the concept of keyframes; if not, don't worry, it's fairly straightforward.

A keyframe is a snapshot that defines a starting or end point of any smooth transition.
With CSS transitions, we're essentially limited to defining the first and last keyframes. CSS animations allow us to add any number of keyframes in between, to guide our animation in more complex ways.

Animation properties:

animation-name
animation-duration
animation-timing-function
animation-iteration-count
animation-direction
animation-delay
animation-fill-mode

CSS3 Animation

To animate an element in CSS, you first create a named animation, then attach it to an element in that element's property declaration block. Animations in themselves don't do anything; in order to animate an element, you will need to associate the animation with that element.

To create an animation, use the @keyframes

Here are a few simple animations:

```
@-webkit-keyframes 'appear' {
  0% {
    opacity: 0;
  }
  100% {
    opacity: 1;
  }
}

@-webkit-keyframes 'disappear' {
  to {
    opacity: 0;
  }
  from {
    opacity: 1;
  }
}

@-webkit-keyframes 'appearDisappear' {
  0%, 100% {
    opacity: 0;
  }
  20%, 80% {
    opacity: 1;
  }
}
```

The last animation is worth paying extra attention to: we've applied the same styles to 0% and 100%, and to 20% and 80%. In this case, it means the element will start out invisible (opacity: 0;), fade in to visible by 20% of the way through the duration, remain visible until 80%, then fade out.

We've created three animations, but they aren't attached to any elements. Once we have defined an animation, the next step is to apply it to one or more elements using the various animation properties.

CSS3 Animation

Shorthand

The animation property takes as its value a space-separated list of values for the longhand animation name, animation-duration, animation-timing-function, animation-delay, animation-iteration-count, animation-direction, and animation-fill-mode properties:

```
.verbose {
  -webkit-animation-name: 'appear';
  -webkit-animation-duration: 300ms;
  -webkit-animation-timing-function: ease-in;
  -webkit-animation-iteration-count: 1;
  -webkit-animation-direction: alternate;
  -webkit-animation-delay: 5s;
  s-webkit-animation-fill-mode: backwards;
}

/* shorthand */
.concise {
  -webkit-animation: 'appear' 300ms ease-in 1 alternate 5s
➥backwards;
}
```

To declare multiple animations on an element, include a grouping for each animation name, with each shorthand grouping separated by a comma. For example:

```
.target {
  -webkit-animation:
    'animationOne' 300ms ease-in 0s backwards,
    'animationTwo' 600ms ease-out 1s forwards;
}
```

Pseudo elements

表 4-11 伪元素（pseudo-element）

伪元素	意义
:first-letter	元素中文本的第一个字符。
:first-line	元素中文本的第一行。
:before	在元素之前插入。
:after	在元素之后插入。

Pseudo elements

源代码 4-35 伪元素使用示例

```
<div>
    <h1>将进酒</h1>
    <h2>李白</h2>
    <p>
        君不见黄河之水天上来，奔流到
        君不见高堂明镜悲白发，朝如青
        ......<br/>
        与尔同消万古愁。
    </p>
</div>

h1, h2 {padding-left:3em;}
:first-letter {color:red;}
p:first-line {color:blue}
h2:before {content:url(libai.jpg); padding-right:0.5em;}
div:after {content:"-- 摘自《唐诗赏析》"; padding-left:10em;}
```

Pseudo class

表 4-12 伪类（pseudo-class）

伪元素	意义
:link	未被用户访问过的链接。
:visited	用户访问过的链接。
:hover	鼠标移过。
:active	鼠标指向链接，按下左键未松开之前。
:first-child	文档树上为父元素第一个孩子的元素。
:lang(*language*)	定义了 lang 属性的元素，并且 lang 的值为“language”，或者前缀为“language-”。注意短划线“-”不可缺少。

CSS3 selectors

Selectors are at the heart of CSS. Without selectors to target elements on the page, the only way to modify the CSS properties of an element would be to use the element's style attribute and declare the styles inline. This, of course, is ugly, awkward, and unmaintainable. So we use selectors.

Relational selectors

Descendant (E F)

You should definitely be familiar with this one. The descendant selector targets any element F that is a descendant (child, grandchild, great grandchild, and so on) of an element E.

Child (E > F)

This selector matches any element F that is a *direct child* of element E—any further nested elements will be ignored. Continuing the above example, ol > li would only target li elements directly inside the ol, and would omit those nested inside a ul.

Adjacent Sibling (E + F)

This will match any element F that shares the same parent as E, and comes *directly after* E in the markup. For example, li + li will target all li elements except the first li in a given container.

General Sibling (E ~ F)

This one's a little trickier. It will match any element F that shares the same parent as any E and comes after it in the markup. So, h1~h2 will match any h2 that follows an h1, as long as they both share the same direct parent—that is, as long as the h2 is not nested in any other element.

CSS3 selectors

Attribute selectors

E[attr$=val] (IE8+, WebKit, Opera, Mozilla)
Matches any element E whose attribute attr ends in val. In other words, the val matches the end of the attribute value.

E[attr*=val] (IE8+, WebKit, Opera, Mozilla)
Matches any element E whose attribute attr matches val anywhere within the attribute. In other words, the string val is matched anywhere in the attribute value. It is similar to E[attr~=val] above, except the val can be part of a word.

Using the same example as above, .fakelink[title~=info] {} would match any element with the class fakelink that has a title attribute containing the string info, such as "Click here for more information."

CSS3 selectors

Pseudo-classes

:enabled
A user interface element that's enabled.
:disabled
Conversely, a user interface element that's disabled.
:checked
Radio buttons or checkboxes that are selected or ticked.
:valid
Applies to elements that are valid, based on the type or pattern attributes
:invalid
Applies to empty required elements, and elements failing to match the requirements defined by the type or pattern attributes.
:in-range
Applies to elements with range limitations, where the value is within those limitations. This applies, for example, to number
and range input types with min and max attributes
:out-of-range
The opposite of :in-range: elements whose value is *outside* the limitations of their range.
:required
Applies to form controls that have the required attribute set.
:optional
Applies to all form controls that *do not* have the required attribute.
:read-only
Applies to elements whose contents are unable to be altered by the user. This is usually most elements other than form fields.
:read-write
Applies to elements whose contents are user-alterable, such as text input fields

CSS3 selectors

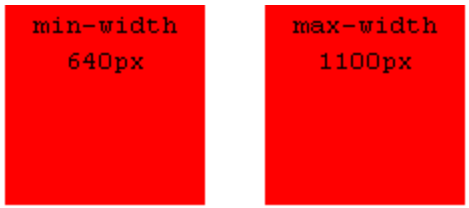
Structural Pseudo-classes

:root
The root element, which is always the html element.
E F:nth-child(n)
The element F that is the nth child of its parent E.
E F:nth-last-child(n)
The element F that is the nth child of its parent E, counting backwards from the last one. li:nth-last-child(1) would match the
last item in any list—this is the same as li:last-child (see below).
E:nth-of-type(n)
The element that is the nth element of its type in a given parent element.
E:nth-last-of-type(n)
Like nth-of-type(n), except counting backwards from the last element in a parent.
E:first-child
The element E that is the first child E of its parent. This is the same as :nthchild(1).
E:last-child
The element E that is the last child E of its parent, same as :nth-last-child(1).
E:first-of-type
Same as :nth-of-type(1).
E:last-of-type
Same as :nth-last-of-type(1).
E:only-child
An element that's the only child of its parent.
E:only-of-type
An element that's the only one of its type inside its parent element.
E:empty
An element that has no children; this includes text nodes, so hello will not be matched.
E:lang(en)
An element in the language denoted by the two-letter abbreviation (en).
E:not(exception)

CSS3 media queries

CSS2 added support for the media="screen" way of defining which stylesheet to use for which representation of the data. CSS3 adds a new feature to this functionality, by adding media queries. Basically, this means you can change stylesheets based on for instance the width and height of the viewport. In a broader sense, this means as the spec puts it: *"by using Media Queries, presentations can be tailored to a specific range of output devices without changing the content itself."*

Below are two tests, for min-width and max-width, currently only functional (and thus green) in Safari 3, Opera, and Firefox 3.1 (Alpha). This *is* however the future of web development, and could make building pages that are usable in both the mobile as the normal web a lot easier.



The CSS which should color the two divs above is as follows:

```
@media all and (min-width: 640px) { #media-queries-1 { background-color:
#0f0; } } @media screen and (max-width: 2000px) { #media-queries-2 {
background-color: #0f0; } }
```

Thank you!