一：CSS3简介：

如同人类的的进化一样，CSS3是CSS2的“进化”版本，在CSS2基础上，增强或新增了许多特性， 弥补了CSS2的众多不足之处，使得Web开发变得更为高效和便捷 动画 圆角 阴影 边框图片 …

Css当js用

Js当后台语言用

二：CSS3现状：

1. 浏览器支持程度差，需要添加私有前缀(移动端)
   1. requestFullScreen
   2. webkitRequestFullScreen
   3. mozRequestFullScreen
   4. msRequestFullScreen
   5. oRequestFullScreen

2、移动端支持优于PC端

3、不断改进中

4、应用相对广泛

三：选择器：

CSS3新增了许多灵活查找元素的方法，极大的提高了查找元素的效率和精准度。CSS3选择器与jQuery中所提供的绝大部分选择器兼容

1. 属性选择器：
2. E[attribute] 表示存在attr属性即可；

div[class]

1. E[attr=val] 表示属性值完全等于val；

div[class=mydemo]

1. E[attr\*=val] 表示的属性值里包含val字符并且在“任意”位置；

div[class\*=mydemo]

1. E[attr^=val] 表示的属性值里包含val字符并且在“开始”位置；

div[class^=mydemo]

1. E[attr$=val] 表示的属性值里包含val字符并且在“结束”位置；

div[class$=demos]

1. 伪类选择器-伪元素选择器：
   1. 之前学习的：a:hover a:link a:active a:visited
   2. 以某元素相对于其父元素或兄弟元素的位置来获取无素的结构伪类
2. E:first-child:查找E这个元素的父元素的第一个子元素E
3. E:last-child:最后一个子元素
4. E:nth-child(n): 第n个子元素，计算方法是E元素的全部兄弟元素
5. E:nth-last-child(n): 同E:nth-child(n) 相似，只是倒着计算
6. E:nth-child(even): 所有的偶数
7. E:nth-child(odd): 所有的奇数
8. E:nth-of-type(n):指定类型
9. E:empty 选中没有任何子节点的E元素，注意，空格也算子元素
10. E:target 结合锚点进行使用，处于当前锚点的元素会被选中
11. 重点说明：n遵循线性变化，其取值0、1、2、3、4、... 但是当n<=0时，选取无效
12. 案例代码：

/\*第一个li元素\*/  
li:first-child{  
 color: red;  
}  
/\*最后一个元素\*/  
li:last-child{  
 color: green;  
}  
/\*获取第10个元素\*/  
li:nth-child(10){  
 color: orange;  
}  
/\*获取倒数第3个li元素\*/  
li:nth-last-child(3){  
 color: purple;  
}  
/\*获取索引顺序为6的倍数的li元素\*/  
li:nth-child(6n){  
 text-decoration: underline;  
 border: 1px solid red;  
}  
/\*获取所有索引为偶数的li元素\*/  
li:nth-child(even){  
 border: 1px solid black;  
}  
/\*获取前5个li元素\*/  
li:nth-child(-n+5){  
 background-color: #ddd;  
}

* 1. n可是多种形式：nth-child(2n)、nth-child(2n+1)、nth-child(-n+5)等

1. 伪元素选择器：
   1. 重点：E::before、E::after
      1. 是一个行内元素，需要转换成块:display:block float:\*\* position:
      2. 必须添加content,哪怕不设置内容，也需要content:””
      3. E:after、E:before 在旧版本里是伪类，在新版本里是伪元素，新版本下E:after、E:before会被自动识别为E::after、E::before，按伪元素来对待，这样做的目的是用来做兼容处理
      4. E::before: 定义在一个元素的内容之前插入[content](http://www.dreamdu.com/css/property_content/)属性定义的内容与样式
      5. E::after: 定义在一个元素的内容之后插入[content](http://www.dreamdu.com/css/property_content/)属性定义的内容与样式
      6. 注意：
         1. IE6、IE7与IE8（怪异模式Quirks mode）不支持此伪元素
         2. CSS2中 E:before或者E:after，是属于伪类的，并且没有伪元素的概念，CSS3中 提出伪元素的概念 E::before和E::after，并且归属到了伪元素当中，伪类里就不再存在E:before或者 E:after伪类
   2. E::first-letter文本的第一个字母或字(不是词组)
   3. E::first-line 文本第一行
   4. E::selection 可改变选中文本的样式

四：颜色设置： 11111111 00000000 0-255

HTML5中添加了一些新的颜色的表示方式

1.RGBA：说得简单一点就是在RGB的基础上加进了一个通道Alpha。RGBA在RGB的基础上多了控制alpha透明度的参数。以上R、G、B三个参数，正整数值的取值范围为：0 - 255。百分数值的取值范围为：0.0% - 100.0%。超出范围的数值将被截至其最接近的取值极限。并非所有浏览器都支持使用百分数值。A参数，取值在0~1之间，不可为负值。RGBA比元素设置CSS的透明度更好，因为单独的颜色可以在不影响整个元素的透明度，他不会影响到元素其他的属性，比如说边框，字体同时也不会影响到其他元素的相关透明度

1. **语法:**
   * R：红色值。正整数 | 百分数
   * G：绿色值。正整数 | 百分数
   * B：蓝色值。正整数| 百分数
   * A：透明度。取值0~1之间
2. **使用示例：**

div{  
 width: 200px;  
 height: 200px;  
 background-color: rgba(10,20,245,0.5);  
 color: white;  
}

2. HSLA(H,S,L,A)：

1. **此色彩模式与**[**HSL**](http://www.css88.com/book/css/values/color/hsl.htm)**相同，只是在**[**HSL**](http://www.css88.com/book/css/values/color/hsl.htm)**模式上新增了Alpha透明度**
2. **语法：**
   * H：Hue(色调,色相)。0(或360)表示红色，120表示绿色，240表示蓝色，也可取其他数值来指定颜色。取值为：0 – 360,过渡为：(红橙黄绿青蓝紫红)
   * S：Saturation(饱和度)。取值为：0.0% - 100.0%
   * L：Lightness(亮度)。取值为：0.0% - 100.0%，50%是平衡值
   * A：Alpha透明度。取值0~1之间。
3. **示例**

span{  
 width: 200px;  
 height: 200px;  
 display: block;  
 background-color: hsla(360,100%,50%,0.6);  
 color: white;  
}

**3.关于透明度的补充说明：**

1. opacity只能针对整个盒子设置透明度，子盒子及内容会继承父盒子的透明度
2. transparent 不可调节透明度，始终完全透明
3. 使用rgba 来控制颜色，相对opacity ，不具有继承性

五：文本(shadow阴影)：

[text-shadow](http://www.w3.org/Style/Examples/007/text-shadow.en.html#translations)还没有出现时，大家在网页设计中阴影一般都是用photoshop做成图片，现在有了css3可以直接使用[text-shadow](http://www.w3.org/Style/Examples/007/text-shadow.en.html#translations)属性来指定阴影。这个属性可以有两个作用，产生阴影和模糊主体。这样在不使用图片时能给文字增加质感

1. **语法：**

**text-shadow ： none | <length> none | [<shadow>, ] \* <shadow> 或none | <color> [, <color> ]\***

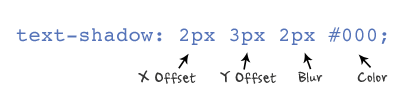
**也就是：**

**text-shadow:[颜色(Color) x轴(X Offset) y轴(Y Offset) 模糊半径(Blur)],[颜色(color) x轴(X Offset) y轴(Y Offset) 模糊半径(Blur)]...**

**或者**

**text-shadow:[x轴(X Offset) y轴(Y Offset) 模糊半径(Blur) 颜色(Color)],[x轴(X Offset) y轴(Y Offset) 模糊半径(Blur) 颜色(Color)]...**

1. **取值：**
   1. <length>：长度值，可以是负值。用来指定阴影的延伸距离。其中X Offset是水平偏移值，Y Offset是垂直偏移值
   2. <shadow>：阴影的模糊值，不可以是负值，用来指定模糊效果的作用距离
   3. <color>：指定阴影颜色，也可以是rgba透明色。
   4. 图示：



1. **说明：**

可以给一个对象应用一组或多组阴影效果，方式如前面的语法显示一样，用逗号隔开。text-shadow: X-Offset Y-Offset Blur Color中X-Offset表示阴影的水平偏移距离，其值为正值时阴影向右偏移，如果其值为负值时，阴影向左偏移；Y-Offset是指阴影的垂直偏移距离，如果其值是正值时，阴影向下偏移反之其值是负值时阴影向顶部偏移；Blur是指阴影的模糊程度，其值不能是负值，如果值越大，阴影越模糊，反之阴影越清晰，如果不需要阴影模糊可以将Blur值设置为0；Color是指阴影的颜色，其可以使用rgba色

1. **一些效果案例代码：**

.demo{  
 width: 600px;  
 padding: 30px;  
 background-color: #666;  
 margin:20px auto;  
 text-align: center;  
 font:bold 80px/100% "微软雅黑";  
 color: #fff;  
}  
/\*侧阴影效果\*/  
.demo1{  
 text-shadow: 2px 2px 2px #ff0000;  
}  
/\*辉光效果\*/  
.demo2{  
 text-shadow: 0 0 30px red;  
}  
/\*多层辉光效果\*/  
.demo3{  
 text-shadow:0 0 5px #fff,0 0 15px #fff,0 0 40px #fff ,0 0 70px red ;  
}  
/\*苹果经典效果\*/  
.demo4{  
 color: black;  
 text-shadow: 0 1px 1px #fff;  
}  
/\*浮雕效果\*/  
.demo5{  
 color: #ccc;  
 text-shadow: -1px -1px 0px #fff,-2px -2px 0px #eee,1px 1px 0px #444,2px 2px 0px #333;  
}  
/\*模糊字效果\*/  
.demo6{  
 color: transparent; /\*将本身设置为透明\*/  
 text-shadow: 0 0 6px #ff9966;  
}

六：盒模型：

1.在默认情况下，CSS设置的盒子宽度仅仅是内容区的宽度，而非盒子的宽度。同样，高度类似。真正盒子的宽度（在页面呈现出来的宽度）和高度，需要加上一些其它的属性。例如：

1. padding + border + width = 盒子的宽度
2. padding + border + height = 盒子的高度

很明显，这不直观，很容易出错，造成网页中其它元素的错位。

2. CSS3中可以通过box-sizing 来指定盒模型，即可指定为content-box、border-box，这样我们计算盒子大小的方式就发生了改变

1. content-box:对象的实际宽度等于设置的width值和border、padding之和
2. border-box： 对象的实际宽度就等于设置的width值，即使定义有border和padding也不会改变对象的实际宽度

3.浏览器的兼容性：

IE8及以上版本支持该属性，Firefox 需要加上浏览器厂商前缀-moz-，对于低版本的IOS和Android浏览器也需要加上-webkit-

七：边框圆角

border-radius可以通过值来定义样式相同的角，也对每个角分别定义

1.值的说明：

border-radius：\*px: 将创建四个大小一样的圆角

border-radius:\*px \*px \*px \*px: 四个值分别表示左上角、右上角、右下角、左下角

border-radius:\*px \*px：第一个值表示左上角、右下角；第二个值表示右上角、左下角

border-radius:\*px \*px \*px：第一个值表示左上角；第二个值表示右上角、左下角；第三个值表示右下角

2.单个圆角的设置：除了同时设置四个圆角以外，还可以单独对每个角进行设置。对应四个角，CSS3提供四个单独的属性：

\* border-top-left-radius

\* border-top-right-radius

\* border-bottom-right-radius

\* border-bottom-left-radius

这四个属性都可以同时设置1到2个值。如果设置1个值，表示水平半径与垂直半径相等。如果设置2个值，第一个值表示水平半径，第二个值表示垂直半径

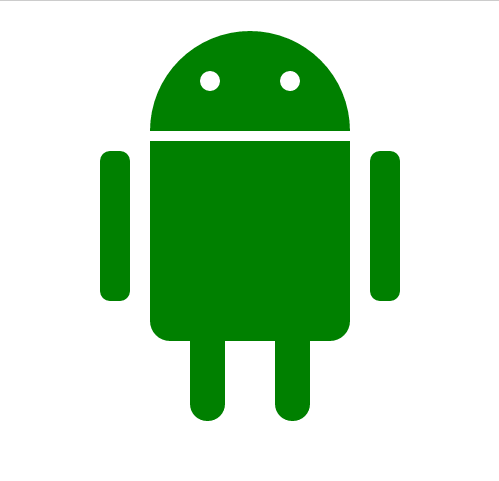
3.补充：创建两个值的非对称圆角还有一种书写方式：如border-radius:20px/10px;表示在水平方向上20px,在垂直方向上10px;具体说明如下：可分别设置长、短半径，以“/”进行分隔，遵循“1，2，3，4”规则，“/”前面的1~4个用来设置横轴半径（分别对应横轴1、2、3、4位置 ），“/”后面1~4个参数用来设置纵轴半径（分别对应纵轴1、2、3、4位置 ）

4.示例代码：

div{  
 width: 200px;  
 /\*height: 200px;\*/  
 height: 100px; /\*为椭圆设置的高度\*/  
 background-color: red;  
 margin: 100px auto;  
 /\*四个方向的圆角值相同\*/  
 /\*border-radius: 100px;\*/  
 /\*左上，右下为20px 右上，左下为40px\*/  
 /\*border-radius: 20px 40px;\*/  
 /\*左上为20px 右上，左下为40px 右下为80px\*/  
 /\*border-radius: 20px 40px 80px;\*/  
 /\*左上：20px 右上：40px 右下：80px 左下：100px\*/  
 /\*border-radius: 20px 40px 80px 100px;\*/  
  
 /\*设置某一个方向上的圆角\*/  
 /\*border-bottom-left-radius: 100px;\*/  
  
 /\*非对称圆角--制作椭圆\*/  
 border-top-left-radius: 100px 50px;  
 border-top-right-radius: 100px 50px;  
 border-bottom-right-radius: 100px 50px;  
 border-bottom-left-radius: 100px 50px;  
  
 /\*非对称圆角的简写方式\*/  
 /\*四个方向上都是一致的非对称圆角\*/  
 /\*border-radius: 100px/50px;\*/  
 /\*也可以这样\*/  
 /\*border-radius: 100px 80px 60px 20px/20px 60px 80px 100px;\*/  
 /\*等价于：\*/  
 /\*border-top-left-radius: 100px 20px;  
 border-top-right-radius: 80px 60px;  
 border-bottom-right-radius: 60px 80px;  
 border-bottom-left-radius: 20px 100px;\*/  
}

八：案例：android机器人

1. 效果：



1. Div结构：

<div class="content">  
 <!--身体头部-->  
 <div class="header"></div>  
 <!--身体主体-->  
 <div class="body"></div>  
 <!--脚-->  
 <div class="footer"></div>  
</div>

1. 样式：主要使用伪元素选择器

<style>  
 \*{  
 padding: 0;  
 margin: 0;  
 }  
 body{  
 background-color: #ccc;  
 }  
 .content{  
 width: 500px;  
 height: 450px;  
 margin: 50px auto;  
 background-color: #fff;  
 padding-top:30px;  
 }  
 .header{  
 width: 200px;  
 height: 100px;  
 margin: 0 auto;  
 background-color: green;  
 border-radius: 100px 100px 0 0;  
 position: relative;  
 }  
 .header::before,  
 .header::after{  
 content: "";  
 position: absolute;  
 width: 20px;  
 height: 20px;  
 background-color: #fff;  
 border-radius: 10px;  
 top: 40px;  
 }  
 .header::before{  
 left:50px;  
 }  
 .header::after{  
 right:50px;  
 }  
 .body{  
 width:200px;  
 height: 200px;  
 background-color: green;  
 margin: 0 auto;  
 margin-top:10px;  
 border-radius: 0 0 20px 20px;  
 position: relative;  
 }  
 .body::before,  
 .body::after{  
 content: "";  
 position: absolute;  
 top: 10px;  
 width: 30px;  
 height: 150px;  
 background-color: green;  
 border-radius: 10px;  
 }  
 .body::before{  
 left:-50px;  
 }  
 .body::after{  
 right:-50px;  
 }  
 .footer{  
 width:200px;  
 height:200px;  
 margin:0 auto;  
 position: relative;  
 }  
 .footer::before,  
 .footer::after{  
 content: "";  
 position: absolute;  
 top: 0px;  
 width: 35px;  
 height: 80px;  
 background-color: green;  
 border-radius: 0 0 40px 40px;  
 }  
 .footer::before{  
 left:40px;  
 }  
 .footer::after{  
 right:40px;  
 }  
</style>

九：边框阴影：

* + 1. box-shadow 属性向边框添加一个或多个阴影。
    2. 语法：box-shadow: h-shadow v-shadow blur spread color inset;

注释：box-shadow 向框添加一个或多个阴影。该属性是由逗号分隔的阴影列表，每个阴影由 2-4 个长度值、可选的颜色值以及可选的 inset 关键词来规定。省略长度的值是 0。

* + 1. 属性值说明：

|  |  |
| --- | --- |
| **值** | **描述** |
| *h-shadow* | 必需。水平阴影的位置。允许负值。 |
| *v-shadow* | 必需。垂直阴影的位置。允许负值。 |
| *blur* | 可选。模糊距离。 |
| *spread* | 可选。阴影的尺寸。值越大，阴影的扩散面积越大 |
| *color* | 可选。阴影的颜色。 |
| inset | 可选。将外部阴影 (outset) 改为内部阴影。 |

* + 1. 在添加阴影的时候，最好考虑到不同浏览器的兼容
    2. 案例示例：

/\*为item添加边框阴影\*/  
div[class=item]:nth-child(-n+4){  
 box-shadow: 1px 1px 3px 0px #bbb;  
}  
/\*可以同时添加多个边框阴影\*/  
div[class=item]:last-child{  
 box-shadow: 1px 1px 2px #ff0000 inset,-1px -1px 2px #ff0000 inset;  
}

十：边框图片

1. 功能：将图片规定为包围 div 元素的边框
   1. 定义和用法： border-image 属性是一个简写属性，用于设置以下属性

border-image-source

border-image-slice

border-image-width

border-image-outset

border-image-repeat

* 1. 属性说明：

|  |  |
| --- | --- |
| 值 | 描述 |
| border-image-source | 用在边框的图片的路径。 |
| border-image-slice | 图片边框向内偏移—裁切。 |
| border-image-width | 图片边框的宽度。 |
| border-image-outset | 边框图像区域超出边框的量。 |
| border-image-repeat | 图像边框是否应平铺(repeated)、铺满(rounded)或拉伸(stretched)。 |

1. 案例：任意拉伸的按钮：通过一个按钮的背景图片制作出任意大小的按钮，实现背景填充效果：
   * + 1. 效果：

* + - 1. 代码：

.downLoad{  
 width:80px;  
 height: 30px;  
 display: block;  
 text-align: center;  
 line-height: 30px;  
 border: 1px solid #ccc;  
 /\*设置图片边框背景\*/  
 border-image: url("../images/btn\_bg.png");  
 /\*设置裁切区域，同时设置填充模式\*/  
 border-image-slice: 10 fill;  
 /\*设置边框的大小，这个一般与裁切区域大小一致，否则就发生缩放\*/  
 border-image-width: 10px;  
 /\*设置边框的重复模式\*/  
 border-image-repeat: round;  
}