

# CSS入门和高级技巧

第2天课堂笔记（本课程共8天）

班级：北京前端训练营1期

讲师：邵山欢

日期：2016年3月25日

邵山欢老师

微博：@邵山欢

QQ : 179427026

E-mail : shaoshanhuan@163.com

爱前端官网 ：[www.iqianduan.cn](http://www.iqianduan.cn)

# 目录

[CSS入门和高级技巧 1](#_Toc19402)

[目录 2](#_Toc18716)

[一、上节课复习 3](#_Toc2356)

[二、继承性和层叠性 7](#_Toc7989)

[2.1 继承性 7](#_Toc7943)

[2.2 层叠性 7](#_Toc19461)

[2.2.1 应用场景 8](#_Toc10715)

[2.2.2 !important提升权重 9](#_Toc25428)

[三、CSS2.1中的颜色表示法 10](#_Toc31145)

[3.1 单词来表示 10](#_Toc31581)

[3.2 rgb()表示法 10](#_Toc16725)

[3.3 十六进制表示法 11](#_Toc678)

[四、文字相关的属性 14](#_Toc26551)

[4.1 color 14](#_Toc21862)

[4.2 font-size 14](#_Toc11751)

[4.3 line-height 16](#_Toc5928)

[4.4 font-family字体 17](#_Toc15565)

[4.5 font-weight加粗 17](#_Toc14997)

[4.6 font-style属性 18](#_Toc31813)

[4.7 text-decoration属性 18](#_Toc12430)

[4.8 综合font属性 19](#_Toc26713)

[五、盒模型 20](#_Toc16627)

[5.1 盒模型整体感知 20](#_Toc32098)

[5.2 padding 22](#_Toc25295)

# 一、上节课复习

● HTML表格，table、tr、td（th）；thead、tbody；caption。

一定要会根据图形，来写表格：

|  |  |  |
| --- | --- | --- |
| 1 | 2 | |
| 3 | 4 | 5 |
| 6 | 7 |

|  |
| --- |
| 1. <table border="1"> 2. <tr> 3. <td>1</td> 4. <td colspan="2">2</td> 5. </tr> 6. <tr> 7. <td rowspan="2">3</td> 8. <td>4</td> 9. <td>5</td> 10. </tr> 11. <tr> 12. <td>6</td> 13. <td>7</td> 14. </tr> 15. </table> |

● HTML注释

|  |
| --- |
| 1. **<!--**注释写在里面**-->** |

|  |
| --- |
| 1. **<!--** <tr> 2. <td>6</td> 3. <td>7</td> 4. </tr> **-->** |

● 字符实体（转义字符）

|  |
| --- |
| 1. &nbsp; 空格 2. &gt; 大于号 3. &lt; 小于号 4. &copy; 版权符号 |

● 废弃标签：b、u、i、del、strong、em

老师昨天漏了一个挺关键的，现在补上

|  |
| --- |
| 1. <br /> |

br标签是breaking打断的意思：

|  |
| --- |
| 1. <p>床前明月光，**<br />**疑是地上霜</p> |

它是一个自封闭标签，自此我们已经遇见4个自封闭标签：

|  |
| --- |
| 1. <meta name=”keywords” content=”” /> 2. <img src=”1.jpg” /> 3. <input type=”text” /> 4. <br /> |

在2007~08年之前，所有的换行都是用<br />来完成的。而现在<br />已经被废弃，用p、div、h系列来进行换行：

|  |
| --- |
| 1. <p>床前明月光，</p> 2. <p>疑是地上霜</p> |

但是<br />也不是没用，就是有些时候，极特殊的用标签来打断语义不合适，没辙了，只能<br />，比如一个有换行的超级链接。

比如淘宝首页：

|  |
| --- |
| 1. <p> 2. <a href="">高级<br />搜索</a> 3. </p> |

高级搜索如果拆分为两个p，不合适，所以就用br打断一下。

● CSS：

cascading style sheet层叠式样式表

舞台，写代码的地方：

|  |
| --- |
| 1. <style type="text/css"> 2. </style> |

语法：

|  |
| --- |
| 1. <style type="text/css"> 2. h1{ 3. k:v; 4. k:v; 5. k:v; 6. k:v; 7. } 8. </style> |

● 选择器：

基本选择器3种：标签选择器、id选择器、类选择器

高级选择器4种：后代选择器、交集选择器、并集选择器、通配符

标签选择器：

|  |
| --- |
| 1. p{ 2. } |

注意的就是所有的p都被选择了，无论它藏的多深；页面上所有标签名都能用。

id选择器：

**id页面唯一**，只要是合法的命名，可以随便任取id。合法的命名：英语字母开头（AA和aa不同）、数字、下划线、横杠。

|  |
| --- |
| 1. <p **id=”pp”**></p> |

选择符是#

|  |
| --- |
| 1. **#pp**{ 2. } |

class选择器：

多个标签可以携带同一个class；同一个标签可以携带多个class，空格隔开。

|  |
| --- |
| 1. <p **class=”warning”**></p> |

选择符是.

|  |
| --- |
| 1. **.warning**{ 2. } |

类的使用，要注意原子类，可以把一个标签多携带几个class，简化我们的页面制作，各取所需。

后代选择器

选择的是后代，而不是儿子。

|  |
| --- |
| 1. div p |

div的后代p，都被选择

|  |
| --- |
| 1. div.haha ul.xixi li.hehe p |

有haha类的div的后代有xixi类的ul的后代有hehe类的li中的p。

交集选择器

|  |
| --- |
| 1. div.haha |

又是div，又有haha类。

并集选择器

用逗号隔开的两部分

|  |
| --- |
| 1. div.haha ul li , div.xixi p{ 2. } |

等价于

|  |
| --- |
| 1. div.haha ul li{ 2. } 3. div.xixi p{ 4. } |

通配符选择器

选择所有元素，具体用法今天下午就能遇见，清除所有元素的默认margin、padding用。

|  |
| --- |
| 1. \*{ 2. } |

# 二、继承性和层叠性

cascading style sheet， 我们对cascading这个词儿只要理解透了，css就理解透了。

实际上我们现在已经知道了cascading的第一层含义，就是同一个标签可以从多个选择器那里得到样式。样式是一层一层抹上去的。

## 2.1 继承性

继承性是css的最美的地方，它就简化了css的书写。

一些属性，如果给一个元素设置了，那么它的后代所有元素都有这个属性了，就是继承性。

哪些属性能够继承：

color

text-

font-

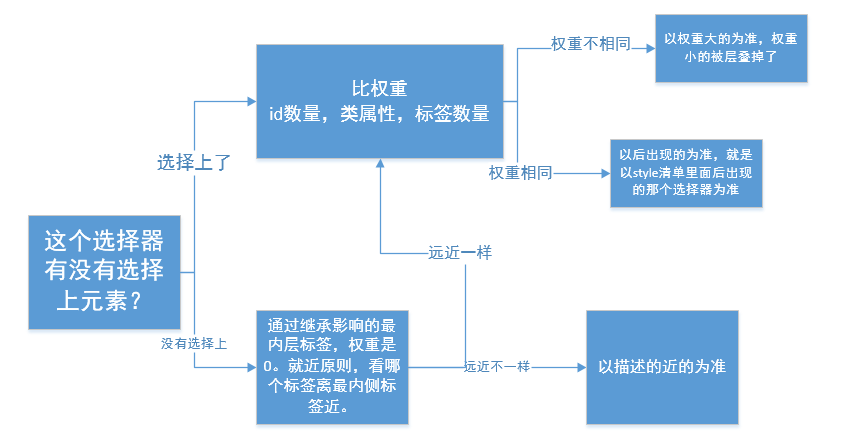
line-

特别的，要知道不能继承的属性：background-color、所有盒模型的属性（width、height、border、padding、margin）都是不继承的！

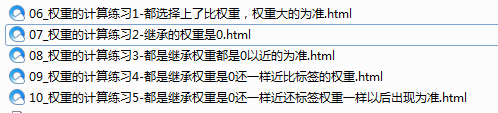
## 2.2 层叠性

层叠性就是处理冲突的能力，就比如一个标签p，用标签选择器设置文字颜色是红色，用id选择器设置文字颜色是蓝色；听谁的。听id的，标签选择器的属性就被杠掉了，术语叫做“层叠”掉了。

总结的一个图：



这里就不把各种情况写在笔记了，好好看看老师的案例：



让同学们自己研究了一个情况，就是同一个标签携带多个类名的时候，如果携带的类名有冲突，那么听谁的？

结论：只和CSS顺序有关，以后出现的为准，与HTML标签中挂类名的顺序无关。

比如：

|  |
| --- |
| 1. <p class="**spec1 spec2**">文字</p> |

或者交换两个类名的顺序写成：

|  |
| --- |
| 1. <p class="**spec2 spec1**">文字</p> |

是对权重没有任何影响的。

### 2.2.1 应用场景

在共性中有某个元素有特性。

比如，现在想让所有的li都是一个颜色，但是就第一个li颜色不同：

|  |
| --- |
| 1. <div class="nav"> 2. <ul> 3. <li class="no1">网站栏目</li> 4. <li>网站栏目</li> 5. <li>网站栏目</li> 6. <li>网站栏目</li> 7. <li>网站栏目</li> 8. <li>网站栏目</li> 9. <li>网站栏目</li> 10. </ul> 11. </div> |

正确写法：

|  |
| --- |
| 1. <style type="text/css"> 2. .nav ul li{ **← 先给所有人设置一个颜色** 3. color:blue; 4. } 5. .nav ul li**.no1**{ **← 再给这个人单独设置一个** 6. color:red; 7. } 8. </style> |

错误写法：

|  |
| --- |
| 1. <style type="text/css"> 2. .nav ul li{ 3. color:blue; 4. } 5. **.no1**{ **← 错误！不生效！因为这个权重干不过上面的.nav ul li。** 6. color:red; 7. } 8. </style> |

所以，以后一定要记住一个真理：**如果想让一个特性层叠掉共性，那么这个特性的选择器的前半部分一定要和共性的相同。**

### 2.2.2 !important提升权重

我们希望页面中的所有原子类，都是权重非常大的，这样，一旦页面中的任何一个标签，携带了这个原子类，立即有样式产生，而不会被自己的样式所层叠。所以，这时候，就可以用!important来提升权重！

|  |
| --- |
| 1. .warning{ 2. color:red **!important**; 3. } |

important是英语重要的意思。注意写法！写在分号之前，如果有多个属性要提升权重，那么必须写多个：

|  |
| --- |
| 1. .warning{ 2. color:red **!important**; 3. font-weight: bold **!important**; 4. } |

注意，**页面严禁滥用!important提升权重，只能在原子类的情况使用**！

还要注意下面几个!important的权重提升方法：

!important不影响继承性，该是0还是0。一个标签是通过继承性影响的，权重是0，加上!important也是0，也不能与已经选中了的选择器抗衡。

!important不影响就近原则，远的那个，写上!important也没用，还是以近的那个为准！

现在我们已经完全揭示了“层叠性”的意思：

1） 同一个标签可以有多个选择器作用，给他增加样式；

2） 有继承性，祖先的标签的一些属性，可以继承给后代的标签；

3） 有层叠性，当遇见冲突的时候有着严密一套法律，规定谁生效谁被杠掉。

CSS就是用代码在画画，它像工程师一样精确，像艺术家一样优美。

从现在开始，我们就要学习CSS的属性了。大致分为几类：

1） 文字样式

2） 盒模型

3） 浮动

4） 定位

5） 背景

6） 表格和列表

# 三、CSS2.1中的颜色表示法

我们大量的用到颜色，比如color、background-color、border:1px solid red;

之前我们都是用英语来描述颜色red、blue等等。

一共有三种方法：单词、rgb()、#十六进制

## 3.1 单词来表示

在HTML中能够找到这些单词表示的颜色名。不过我们一般就用常见的这么几个：

black、white、red、green、blue、yellow、pink、orange、purple、gold、gray、yellowgreen、greenyellow等等

## 3.2 rgb()表示法

|  |
| --- |
| 1. background-color: rgb(0,0,255); |

**光学显示器的三原色是红、绿、蓝，依靠他们三个的不同亮度，就能组成不一样的颜色**。每种颜色的亮度数值是0~255，一共256个数字。

计算机的显示屏是由三元色的发光晶体组成的



注意它的语法，rgb()中间用两个逗号隔开三个数字。

红色：

|  |
| --- |
| 1. background-color: rgb(255,0,0); |

绿色：

|  |
| --- |
| 1. background-color: rgb(0,255,0); |

蓝色：

|  |
| --- |
| 1. background-color: rgb(0,0,255); |

黑色：

|  |
| --- |
| 1. background-color: rgb(0,0,0); |

光学显示器什么都关掉了，就是黑色。

白色：

|  |
| --- |
| 1. background-color: rgb(255,255,255); |

特别的，当三个数字都一样的时候，就是灰色：

|  |
| --- |
| 1. background-color: rgb(111,111,111); |

每个数位都能够表示256种颜色（0~255），那么三个数位能够表示多少颜色呢？乘法原理：

256\*256\*256种颜色，16777216种颜色。

## 3.3 十六进制表示法

rgb表示法比较冗长，更常用的就是16进制表示法。

|  |
| --- |
| 1. <style type="text/css"> 2. .no1{ 3. background-color: #**000000**; 4. } 5. .no2{ 6. background-color: #**ff0000**; 7. } 8. .no3{ 9. background-color: #**00ff00**; 10. } 11. .no4{ 12. background-color: #**0000ff**; 13. } 14. </style> |

十六进制表示法以#开头，后面跟随6位数字，分为3组，分别表示红、绿、蓝的数量。

|  |
| --- |
| 1. #**ff0000** |

我们现在要介绍一下十六进制：

我们人的手指10只手指，所以人类就是10进制，逢10进1。

【10进制中】0、1、2、3、4、5、6、7、8、9 一共10个数字

【16进制中】0、1、2、3、4、5、6、7、8、9、a、b、c、d、e、f 一共16个数字。

也就是说：

16进制中的5，就是10进制中的5；

16进制中的9，就是10进制中的9；

16进制中的a，就是10进制中的10；

16进制中的b，就是10进制中的11；

16进制中的c，就是10进制中的12；

16进制中的d，就是10进制中的13；

16进制中的e，就是10进制中的14；

16进制中的f，就是10进制中的15；

16进制中的10，就是10进制中的16；

16进制中的11，就是10进制中的17；

……

位权的概念：

一个10进制的数字，比如38 ， 数字3表示拥有3个10

38 = 3 \* 10 + 8

一个16进制的数字，比如38 ，数字3表示拥有3个16

3 \* 16 + 8 = 56

也就是说，16进制中的38就是10进制中的56。

所以现在给你一个16进制数字，你就能够立即算出对应的10进制数字是多少。

16进制的25 ， 2\*16+5=37 , 就是10进制中的37

16进制的42 ， 4\*16+2=66 , 就是10进制中的66

16进制的ab , 10\*16+11 = 171 , 就是10进制中171

16进制的2b ， 2\*16+11 = 43， 就是10进制中43

16进制的ff， 15\*16+15 = 255 , 就是10进制中255

所以：

|  |
| --- |
| 1. #42ab2b |

等价于，

|  |
| --- |
| 1. rgb(66,171,43) |

所以：

|  |
| --- |
| 1. #000000 |

黑色

|  |
| --- |
| 1. #ffffff |

白色

|  |
| --- |
| 1. #ff0000 |

红色

|  |
| --- |
| 1. #111111 |

灰色

16进制写法可以简化，所有形如

|  |
| --- |
| 1. #aabbcc; |

都可以简化为：

|  |
| --- |
| 1. #abc; |

比如：

|  |
| --- |
| 1. #000000 |

等价于

|  |
| --- |
| 1. #000 |

|  |
| --- |
| 1. #6688cc |

等价于

|  |
| --- |
| 1. #68c |

|  |
| --- |
| 1. #ff0000 |

等价于

|  |
| --- |
| 1. #f00 |

所以，

|  |
| --- |
| 1. #2b2b2b |

不能简化！

大小写都一样，比如#F00等价于#ff0000

至此我们就介绍完了CSS2.1层面的颜色表示法，现在我们表示红色：

red

rgb(255,0,0)

#ff0000

#f00

# 四、文字相关的属性

## 4.1 color

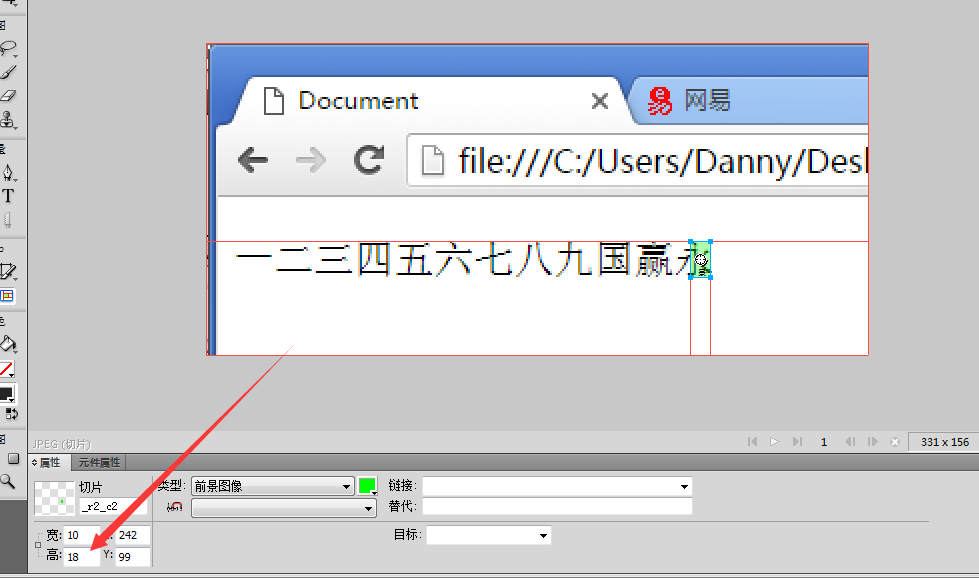
文字颜色，这个属性能够继承，能够用3种表示法来表示。

## 4.2 font-size

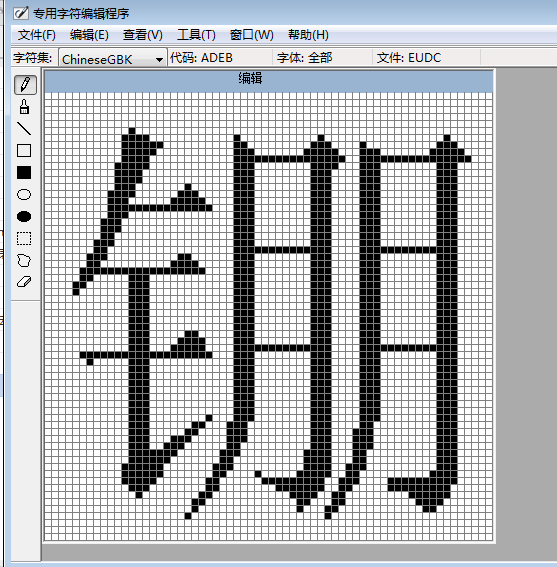
文字大小，它有单位的，现在只学习一个单位就是像素。在响应式web制作的时候，老师将讲解rem、em单位。

|  |
| --- |
| 1. font-size:20px; |

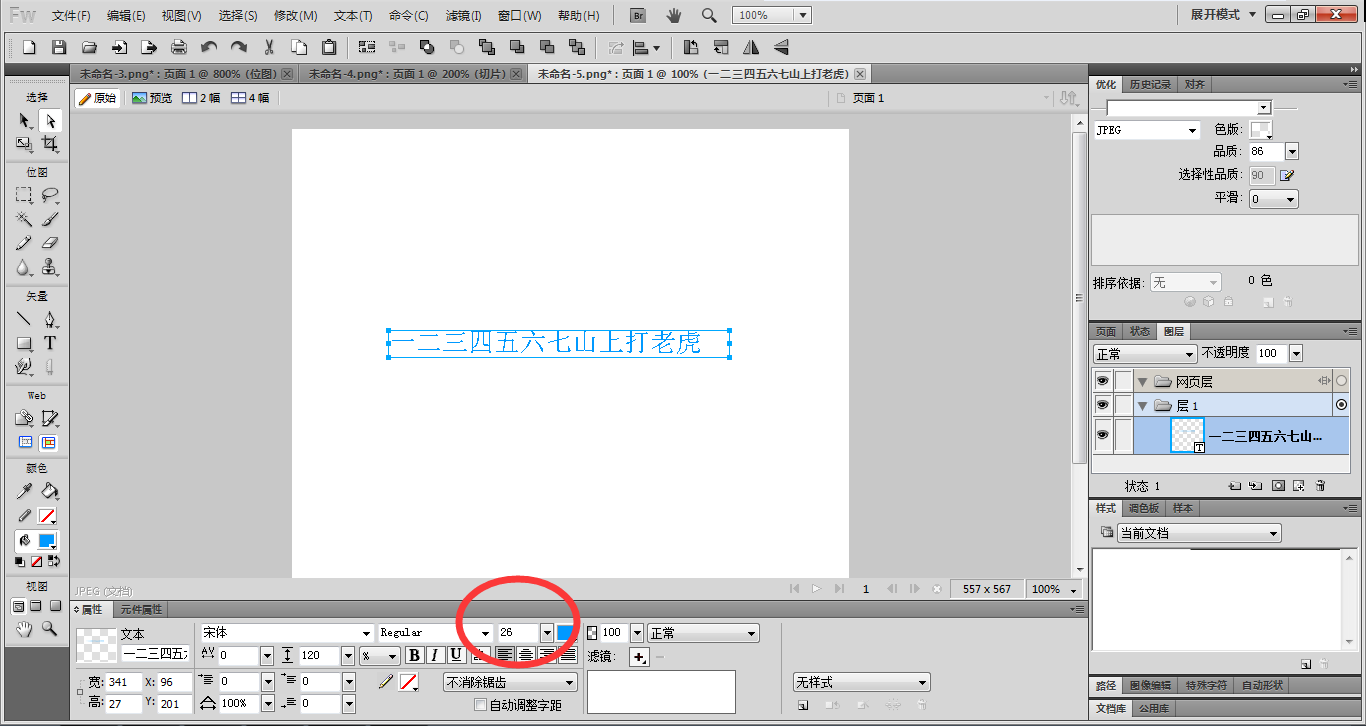
实际测量饱满汉字的真实高度却是18px。



这是因为我们的汉字在制造的时候，就不是顶天立地的：



老师，我到底要用多少号字？根据设计师的设计图。



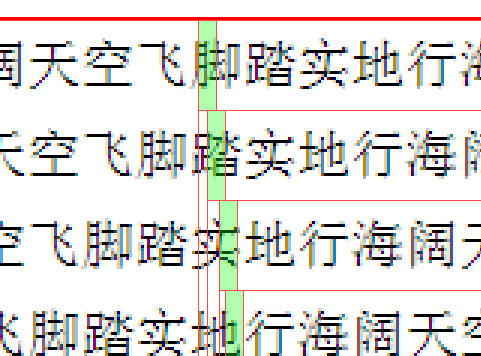
但是如果设计图中，有一些字没有图层的，那么此时比较麻烦，**绝对不要直接去量！**因为汉字不是顶天立地的，必须自己写一个汉字，然后去比较。

各个浏览器的默认字号都不一样，有的是12px、有的是14px。所以设计图上的文字，都要写字号。

Chrome浏览器支持的最小字号是12px，低于12px字号将仍以12px显示，设计师如果给你一个设计稿文字太小，直接找老板告状。

## 4.3 line-height

line-height表示行高。sublime里面的快捷键是lnh



文字所在这一行的高度，称为行高。文字在行里垂直居中。

行高到底为多少？还是那个答案：看设计图！如果设计图没有图层，要写两个文字量量。

line-height可以以px为单位，也可以用百分比为单位。

如果用百分比为单位，那么就是当前字号的百分比。也就是说：

|  |
| --- |
| 1. font-size:14px; 2. line-height:**150%**; |

等价于

|  |
| --- |
| 1. font-size:14px; 2. line-height:**21px**; |

|  |
| --- |
| 1. font-size:16px; 2. line-height:**200%**; |

等价于

|  |
| --- |
| 1. font-size:16px; 2. line-height:**32px**; |

由于字号和行高非常重要，所以可以和写在一起称为font属性

|  |
| --- |
| 1. font:14px/28px "宋体"; |

等价于：

|  |
| --- |
| 1. font-size:14px; 2. line-height:28px; 3. font-family:"宋体"; |

## 4.4 font-family字体

font-family属性就是字体，family就是家庭的意思。所有的字体都要用英语引号引用起来。

字体不是随便设置哦，必须是用户电脑里有这个字体，你才能设置，否则用户看到的是宋体。

**所以网页中，为了让所有的用户都有一致的体验，只能用宋体、微软雅黑**。黑体、楷体有的公司也用，但是不常见。

|  |
| --- |
| 1. font-family: "宋体"; |

|  |
| --- |
| 1. font-family: "微软雅黑"; |

一般来说，如果设置为微软雅黑，那么就要设置一个备选字体，备选字体一般是宋体，用逗号隔开列出。

|  |
| --- |
| 1. font-family: "微软雅黑","宋体"; |

有的服务器上面，为了追求css的加载速度，把字体名变为英语。 css中

|  |
| --- |
| 1. font-family:”Microsoft Yahei”,”SimSun”; |

等价于

|  |
| --- |
| 1. font-family: "微软雅黑","宋体"; |

英语字体写在前面：

|  |
| --- |
| 1. font-family: "Arial","Microsoft Yahei","SimSun"; |

## 4.5 font-weight加粗

bold就是粗体

|  |
| --- |
| 1. font-weight:bold; |

等价于

|  |
| --- |
| 1. font-weight: **700**; |

不加粗，要写normal这个词

|  |
| --- |
| 1. font-weight:normal; |

等价于

|  |
| --- |
| 1. font-weight:**400**; |

面试爱考。

这是能继承的属性

## 4.6 font-style属性

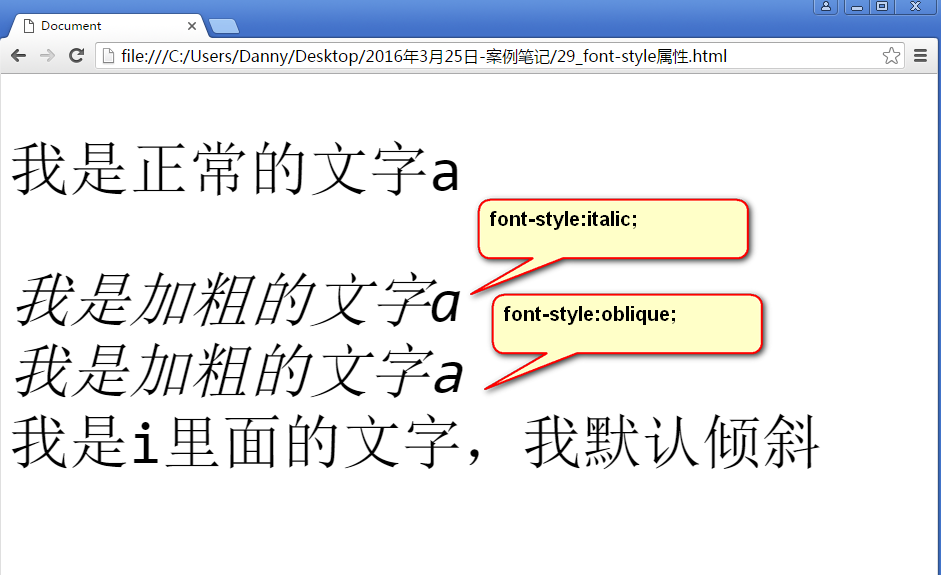
如果想让文字倾斜，使用

|  |
| --- |
| 1. font-style:italic; |

不倾斜：

|  |
| --- |
| 1. font-style: normal; |

font-style:oblique;也是倾斜，和italic的区别：

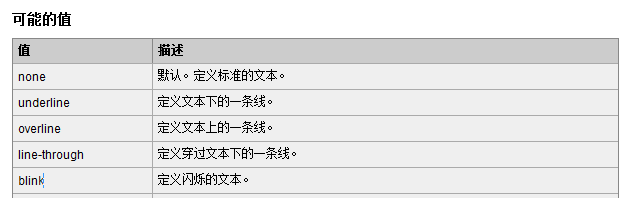


italic会找到新的斜的字型来替代；oblique就是普通倾斜。

中文一律用italic就行了。

## 4.7 text-decoration属性

字符装饰



下划线：

|  |
| --- |
| 1. text-decoration:underline; |

没有下划线：

|  |
| --- |
| 1. text-decoration:none; |

删除线：

|  |
| --- |
| 1. text-decoration:line-through; |

总结：

|  |
| --- |
| 1. font-weight:**b**old; 加粗 2. font-style:**i**talic; 倾斜 3. text-decoration:**u**nderline; 下划线 |

## 4.8 综合font属性

font属性是一个大综合属性：

|  |
| --- |
| 1. font:italic bold 12px/20px arial,sans-serif; |

等价于

|  |
| --- |
| 1. font-style:italic; 2. font-weight:bold; 3. font-size:12px; 4. line-height:20px; 5. font-family:arial,sans-serif; |

一般来说，我们不会这么综合，而是：

|  |
| --- |
| 1. font:12px/200% “Microsoft Yahei”,”SimSun”; |

其他的文本属性，比如text-indent、text-align没有那么难，所以遇见提一嘴就行了。

# 五、盒模型

## 5.1 盒模型整体感知

盒模型就是width、height、padding、border、margin外边距这么几个属性。



width：内容的宽度

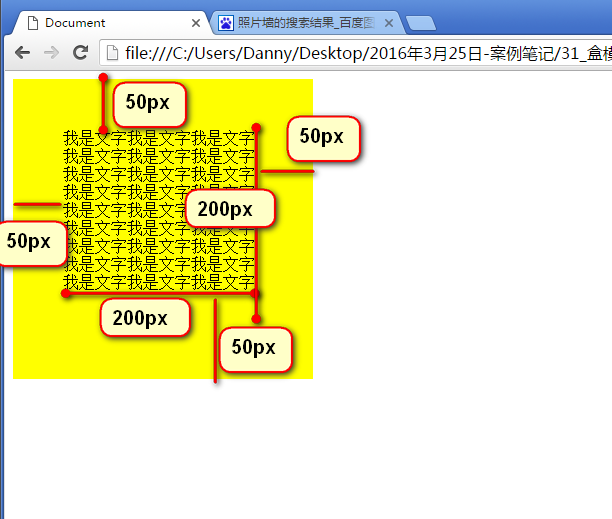
height：内容的高度

padding：内边距

border: 边框

第一个案例：

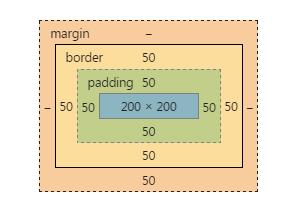
|  |
| --- |
| 1. <style type="text/css"> 2. div{ 3. width: 200px; 4. height: 200px; 5. background-color: yellow; 6. padding: 50px; 7. } 8. </style> |



加上border：

|  |
| --- |
| 1. div{ 2. width: 200px; 3. height: 200px; 4. background-color: yellow; 5. padding: 50px; 6. border:50px solid pink; 7. } |

内容、padding、border、margin是四圈。



小测试：

|  |
| --- |
| 1. div{ 2. width:100px; 3. height:100px; 4. padding:10px; 5. } |

那么这个盒子真实占有的宽度就是120px。

一定一定要养成一个习惯，就是一个盒子的width属性，不是真实占有的宽度！！

真实占有的宽度 = width + 左边padding + 右边padding + 左边border宽度 + 右边的border宽度



## 5.2 padding

内容和边框之间的距离

|  |
| --- |
| 1. padding:50px; |

四个方向的padding就都设置为50了

如果想单独设置padding：

|  |
| --- |
| 1. padding-top:10px; 2. padding-right:20px; 3. padding-bottom:30px; 4. padding-left:40px; |

等价于：

|  |
| --- |
| 1. padding:10px 20px 30px 40px; |

顺序是上、右、下、左。

如果写三个数值：

|  |
| --- |
| 1. padding:10px 20px 30px; |

此时相当于：

|  |
| --- |
| 1. padding-top:10px; 2. padding-right:20px; 3. padding-bottom:30px; 4. padding-left:20px; (和右一样) |

如果写两个属性

|  |
| --- |
| 1. padding:10px 20px; |

|  |
| --- |
| 1. padding-top:10px; 2. padding-right:20px; 3. padding-bottom:10px;(和上一样) 4. padding-left:20px; (和右一样) |