# CSS基础入门

第2天课堂笔记（本课程共8天）

# 目录

[CSS基础入门 1](#_Toc29659)

[目录 2](#_Toc18370)

[一、 复习 3](#_Toc16966)

[二、 css属性 5](#_Toc18497)

[1、 文字 5](#_Toc3371)

[color 5](#_Toc21388)

[font-size 5](#_Toc13306)

[font-family 5](#_Toc24368)

[line-height 5](#_Toc639)

[font-weight 7](#_Toc19056)

[font-style 7](#_Toc15088)

[text-indent 7](#_Toc13991)

[text-align 7](#_Toc10580)

[text-decoration 8](#_Toc30598)

[三、 盒模型 8](#_Toc21298)

[1、盒模型初步 8](#_Toc661)

[2、 padding 10](#_Toc11690)

[3、 margin 11](#_Toc16219)

[4、 border 12](#_Toc21864)

# 复习

1. html杂项

注释：

|  |
| --- |
| 1. <!-- 这里是注释 --> |

快捷键ctrl+/。

字符实体/转义字符

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

废弃标签：font、b、u、i、del、em、strong、<br /> 、<hr />。

1. css

css：cascading style sheet。层叠式样式表。

层叠式有三个体现。

①一个标签元素可以被多个选择器选中。样式会叠加作用。

②css样式具有继承性。祖先元素设置的文字样式，会继承给它的后代。

③层叠性：多个选择作用在一个元素，同样的属性会出现层叠性。

书写位置：行内式、内嵌式、外链式。

目前学内嵌式。

|  |
| --- |
| 1. <style type=”text/css”>   css样式书写  </style> |

css的书写语法：

|  |
| --- |
| 1. <style type=”text/css”>   选择器{  k:v;  k:v;  k:v;  k:v;  k:v;  k:v;  }   1. </style> |

选择器：就是我们要给它添加样式的元素标签。通过一些方法得到这个标签。

基础选择器：标签选择器、类选择器、id选择器、通配符。

高级选择器：后代选择器、交集选择器、并集选择器。

标签选择器：我们的标签名。

单独的标签选择器，选中页面中所有的p元素。

应用：清除、设置默认样式。

|  |
| --- |
| 1. p{ color:red;} |

id选择器：通过元素的一个id属性选择这个元素。

选择方法：#id。

同一个id在页面中只能出现1次。

id命名规则：必须以字母开头（严格区分大小写），可以有数字、下划线、横线。

id只能控制一个，实战中不会用。id留给js加特效、行为。

|  |
| --- |
| 1. #ps{ font-size: 30px;} |

类选择器：通过元素的class属性来获取这个元素。

选择方法：.class。

class：多个元素共有相同的class，一个标签拥有多个class。

|  |
| --- |
| 1. .classname{ border:1px solid red;} |

可以多选元素，同时设置相同的样式，是我们工作常用的选择器。

通配符：\*。会选中页面中所有的元素。

一般用来清除比较普遍的默认样式。

|  |
| --- |
| 1. \*{ margin: 0; padding: 0;} |

后代选择器：选择的是一个元素的后代所有符合的元素。

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

选中的是div盒子后代的所有p。并不是只有子代的p。

|  |
| --- |
| 1. .box1 div.box2 .box3 p |

选中的是类为box1的元素后代的类为box2的div的后代的类叫box3的元素的后代的所有p。

交集选择器：选择的是符合所有约束条件的元素。

|  |
| --- |
| 1. div.box2 |

必须是div元素，而且必须有一个class叫做box2.

|  |
| --- |
| 1. div.box1.box2.box3 |

选的是div，它的class必须同时有box1/box2/box3.

并集选择器：多个元素有相同的样式，而且不能用同一个选择器选中。

多个选择器之间用逗号隔开。

|  |
| --- |
| 1. p,div,ul,ol{} |

相当于

|  |
| --- |
| 1. p{} 2. div{} 3. ul{} 4. ol{} |

继承性：从他的祖先元素继承而来的属性。必须是文字属性。

继承性遵循就近原则。

如果一样近，先比较权重，权重相同看书写顺序。

层叠性：元素被多个选择器选中，相同的属性会被层叠。权重大的层叠掉权重小的。

权重一样大，看书写顺序。

！important可以提升权重，但是不影响就近原则。提升的是单个属性的权重。

# css属性

css属性重点：

文字

盒模型

背景

浮动

定位

## 文字

### color

color:文字的颜色。

色值：#ff0000 rgb(255,0,0) 颜色名（查手册html颜色名）

|  |
| --- |
| 1. ff=16\*15 + 15 = 255 |

实际工作：设计图会提供颜色色值，直接复制就可。

自己模仿网站：需要用工具吸取颜色。

### font-size

font-size：字号。

单位：px。

浏览器有默认最小的字号，如果低于这个字号，显示默认的最小字号。

谷歌Chrome：最小字号12px。

IE：随意。

实际工作中：看设计图。

如果设计图没有图层。写两个相同的文字，调整字号，如果能重合就是这个字号。



fw的设置：文字里面有一个消除锯齿选项。改成，不消除锯齿。

### font-family

font-family：字体

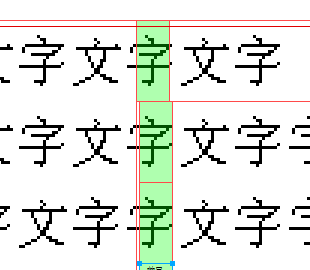
中文：”微软雅黑”、”宋体”。

英文：Arial。

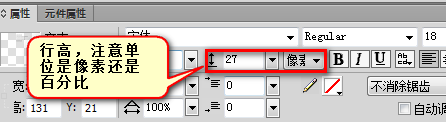
### line-height

line-height：行高。

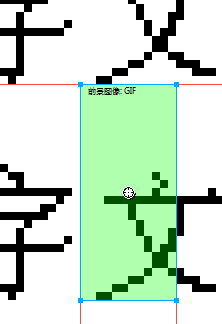
行高是一行文字占的实际高度。在此高度内，文字是垂直居中。



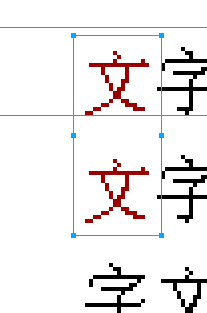
行高怎么找：看设计图。



方法1：从上一行文字的底边量到下一行文字的底边。



方法2：写两行相同的文字，进行对比。



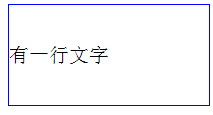
设置：属性值有两种表示方法。像素、百分比。

行高在书写的时候，一定要写在字号属性的后面。用字号来计算百分比。

|  |
| --- |
| 1. .box{ 2. border: 1px solid red; 3. color: #555; 4. font-size: 16px; 5. line-height: 200%; 6. } |

单行文字在盒子内居中：让行高等于盒子高。

|  |
| --- |
| 1. p{ 2. border: 1px solid blue; 3. width: 200px; 4. height: 100px; 5. font-size: 20px; 6. line-height: 100px; 7. } |



行高和字号、字体可以写在一起。font的符合属性。

|  |
| --- |
| 1. font: 20px/100px "微软雅黑"; |

### font-weight

font-weight：加粗属性。

默认值：normal。 值是400

加粗：bold。 值是700

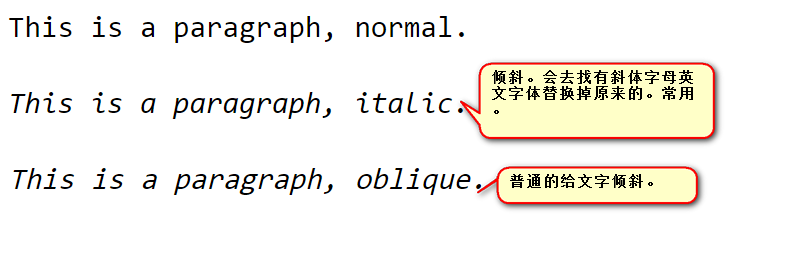
数值：100,200，……700.

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

### font-style

font-style：文字样式。设置是否倾斜。

|  |
| --- |
| 1. .normal{ 2. font-style: normal; 3. } 4. .italic{ 5. font-style: italic; 6. } 7. .oblique{ 8. font-style: oblique; 9. } |



复合写法：单独属性之间用空格隔开

|  |
| --- |
| 1. font: italic bold 20px/100px "微软雅黑"; |

文本控制的属性：左右居中、缩进、装饰。

### text-indent

text-indent：文本缩进。

我们常用的缩进两个汉字。

|  |
| --- |
| 1. text-indent: 2em; |

直接写像素值。

|  |
| --- |
| 1. text-indent: 80px; |

可以书写百分比。是父级元素宽度的百分比。

|  |
| --- |
| 1. text-indent: 10%; |

### text-align

text-align：设置文本整体居左、居右、居中。

属性值：left左边、right右边、center中间。

默认值是：

|  |
| --- |
| 1. text-align:left; |

让文本水平居中，对文字是否单行没要求。

|  |
| --- |
| 1. text-align: center; |

### text-decoration

text-decoration：文本的修饰。



默认的属性值（除了a元素）都是none。

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

文本这些属性都是根据我们的设计图和用户需要来进行设置的。

记单词：family字体

Arial、Microsoft Yahei、SimSun字体类型

line-height行高

weight重量 normal正常 bold加粗

style样式 normal无特殊 italic斜体 oblique斜体

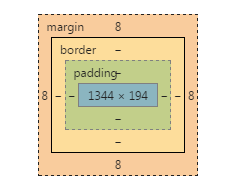
indent缩进

text-align水平方向对齐 left左 right右 center居中

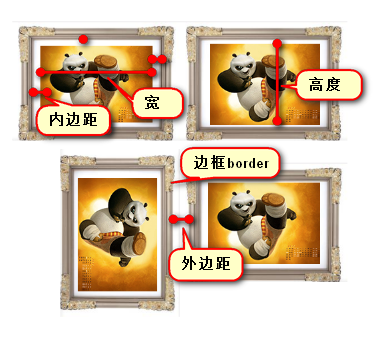
text-decoration文字装饰 none无 underline下划线 overline上划线 line-through删除线

# 盒模型

## 1、盒模型初步



真正的一个盒子的模型：宽度、高度、边框、内边距、外边距



内容宽度：width

内容高度：height

边框：border

内边距：padding

外边距：margin

可以写内容的区域：

宽=width

高=height

盒子实际占位的区域：

宽=width + padding-left + padding-right + border\*2

高=height + padding-top + padding-bottom + border\*2

例：计算实际占有宽高

|  |
| --- |
| 1. .box{ 2. width: 500px; 3. height: 400px; 4. padding: 40px; 5. border: 10px dashed #333; 6. } |

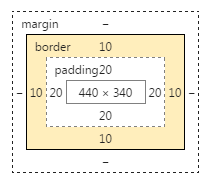
实际占有宽= 500 + 40 + 40 + 10 + 10 = 600px

实际占有高= 400 + 40 + 40 + 20 = 500px

我们实际工作中，实际占有宽高是确定。设置width、height，用实际占有的宽高减去内边距和边框。

|  |
| --- |
| 1. .box{ 2. width: 440px; 3. height: 340px; 4. padding: 20px; 5. border: 10px dashed #333; 6. } |

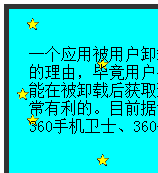
实际宽为500px，实际高位400px。



练习：实际宽高分别是560px、450px。内边距是20px。边框是5像素。

背景渲染的时候，渲染的区域包含内容区和内边距。

|  |
| --- |
| 1. background-color: #00ffff; 2. background-image: url(images/star.gif); |



## padding

padding是一个复合属性。

拆分成四个方向书写单独属性。

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

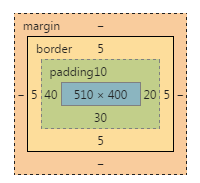
四个方向的属性值可以不同。

还可以写成复合属性。

复合属性padding书写时顺序：上、右、下、左。

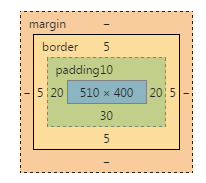
四值法：上、右、下、左。

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



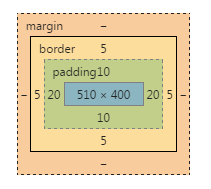
三值法：上、右左、下

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



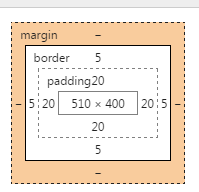
二值法：上下、右左

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



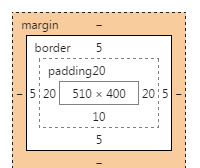
单值法：上右下左

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

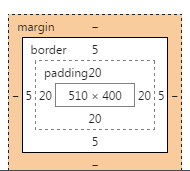


实际过程中，先写一个单值法，特殊的方向用单独属性设置。

|  |
| --- |
| 1. padding: 20px; 2. padding-bottom: 10px; |



|  |
| --- |
| 1. padding-bottom: 10px; 2. padding: 20px; 会层叠掉前面的10px |



## margin

外边距是盒子之间的距离。

设置和书写方法和padding一样。

四个单独属性；

|  |
| --- |
| 1. margin-top 2. margin-right 3. margin-bottom 4. margin-left |

复合属性

|  |
| --- |
| 1. margin: 20px 30px 40px 50px; 2. margin: 10px 20px 30px; 3. margin: 10px 20px; 4. margin: 20px; |

## border

border：边框。

也是一个复合属性。

两种拆分方式：

①按照类型划分：

宽度、线的类型、线的颜色。

|  |
| --- |
| 1. border-width: 10px; 2. border-style: dashed; 3. border-color: #ff0; |

写成复合属性：

|  |
| --- |
| 1. border: 10px dashed #ff0; |

线型： solid 实线

dashed 虚线

dotted 点线

double 双线，取决于线宽。

groove 边框凹槽效果

ridge 边框凸起效果

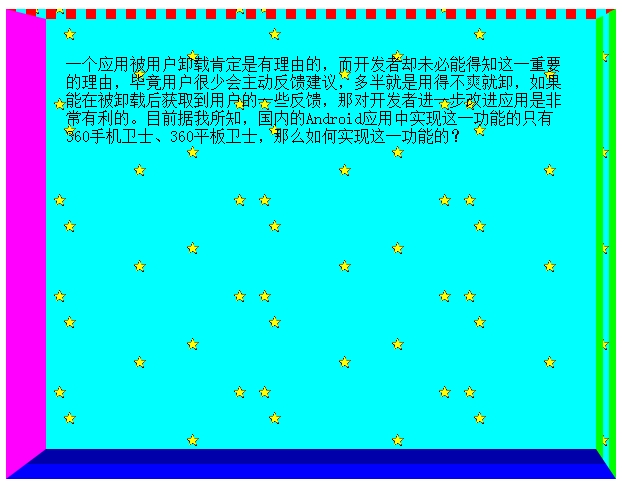
inset 内容区域凹陷效果

outset 内容区域突出效果

②按方向划分：

需要写全三个类型属性值。

|  |
| --- |
| 1. border-top: 10px dotted #f00; 2. border-right: 20px double #0f0; 3. border-bottom: 30px groove #00f; 4. border-left: 40px solid #f0f; |



上面这种没法合成一个复合属性。

③可以对上面两个分类综合书写。

书写时方向写在前面。

|  |
| --- |
| 1. border-left-width: 50px; |

线宽根据方向可以书写：四值、三值、二值、单值。

例子：用代码画一个三角形。

|  |
| --- |
| 1. .box{ 2. width: 0; 3. height: 0; 4. border: 30px solid #fff; 5. border-top-color: #f00; 6. border-bottom: none; 7. } |



如果边框没有可以设置属性值为：none。

④border-collapse属性

用来合并表格的边框。

帮助我们做单线表格。

|  |
| --- |
| 1. table,tr,td{ 2. border: 1px solid #333; 3. border-collapse: collapse; 4. } |

属性值：separate 单独，默认值，线会分开

collapse 塌陷，线会合并。

