# CSS基础入门

第1天课堂笔记（本课程共8天）

# 目录

[CSS基础入门 1](#_Toc32225)

[目录 2](#_Toc27578)

[一、 复习 3](#_Toc9712)

[二、 html杂项 3](#_Toc11643)

[1、 注释 3](#_Toc28680)

[2、 字符实体 3](#_Toc27077)

[3、 废弃标签 4](#_Toc4496)

[三、 css的基础知识 4](#_Toc3993)

[1、css的概念 4](#_Toc22515)

[2、css的作用 5](#_Toc3959)

[3、 css的一些样式 5](#_Toc28522)

[4、 css的书写位置 6](#_Toc21379)

[5、css的一些小事项 6](#_Toc5391)

[四、 基础选择器 7](#_Toc22994)

[1、 标签选择器 7](#_Toc13952)

[2、 id选择器 7](#_Toc2559)

[3、 类选择器 8](#_Toc21403)

[4、 通配符 9](#_Toc26739)

[五、 高级选择器 9](#_Toc23320)

[1、 后代选择器 9](#_Toc7442)

[2、 交集选择器 10](#_Toc16660)

[3、 并集选择器 10](#_Toc12330)

[六、 继承性和层叠性 11](#_Toc18119)

[1、 继承性 11](#_Toc14278)

[2、 层叠性 12](#_Toc25841)

# 复习

1. 列表

无序列表：ul>li

有序列表：ol>li

定义列表：dl>dt+dd

都是组标签，必须成对出现。嵌套关系不能乱。

1. div和span。两个布局的盒子。
2. 表格。table>caption+(thead>tr>th)+(tbody>tr>td)

|  |
| --- |
| 1. <table> 2. <caption></caption> 3. <thead> 4. <tr> 5. <th></th> 6. </tr> 7. </thead> 8. <tbody> 9. <tr> 10. <td></td> 11. </tr> 12. </tbody> 13. </table> |

4、表单

|  |
| --- |
| 1. 单行文本框<input type="text" /> 2. 密码框<input type="password" /> 3. 单选框<input type="radio" /> 4. 复选框<input type="checkbox" /> 5. 普通按钮<input type="button" /> 6. 提交按钮<input type="submit" /> 7. 重置按钮<input type="reset" /> 8. 多行文本框<textarea cols="30" rows="10"></textarea> 9. 下拉菜单<select><option ></option></select> |

# html杂项

## 注释

注释是给程序员看的，不会出现在页面显示里。

|  |
| --- |
| 1. <!--header部分--> |

可以提高我们代码的可读性。

用途：标注某一部分的开始结束，不想要的代码可以注释。给别人看。

sublime快捷键：ctrl+/

## 字符实体

有一些特殊符号有特殊的功能。如果写在普通文本里面，不会正常显示你想要的效果。

也叫做转义字符。可以直接帮我们输出特殊符号的文本。

常见的：

大于号> &gt; greater than

小于号< &lt; less than

无换行空格 &nbsp; nobreaking space

版权符号 &copy; copyright

|  |
| --- |
| 1. 整体格式&关键词; |

还有很多字符实体。查W3C手册。



## 废弃标签

废弃标签有：b、u、i、font、em、strong、del、br、hr

|  |
| --- |
| 1. <b>我是一个b标签</b><br /> 2. <u>我是一个u标签</u><br /> 3. <i>我是一个i标签</i><br /> 4. <del>我是一个del标签</del><br /> 5. <em>我是一个em标签</em><br /> 6. <strong>我是一个strong标签</strong><br /> 7. <hr size="10" width="300px" color="red"/> |

b：bold。

u：underline

i：italic

del：delete。line-through

em：emphasis

br：breaking

hr：horizontal rule

# css的基础知识

## 1、css的概念

css：cascading style sheet。层叠式样式表。有两个关键词“层叠式”，“样式”。

web前端三层：

html 结构层

css 样式层

Javascript 行为层

css也是有w3c组织维护更新的。

css目前发展到css3版本，我们学习的是css2.1版本。

## 2、css的作用

css包括的部分：选择器（选择哪个标签进行样式设置）、样式属性（给标签添加什么属性）。

css的属性是逐条排列的，有什么样式就直接写什么属性。

|  |
| --- |
| 1. <style type="text/css"> 2. div{ 3. width: 100px; 4. height: 100px; 5. color: gold; 6. background-color: lightblue; 7. } 8. </style> |

css的作用：

①可以给文字添加各种样式。

②给盒子添加样式用于布局。

## css的一些样式

①文字样式：

|  |
| --- |
| 1. div{ 2. color: #ff0000; 3. font-size: 24px; 4. font-family: "微软雅黑"; 5. } |

color：表示文字的颜色。可以用十六进制或者单词的方式表示、rgb格式。

|  |
| --- |
| 1. #ff0000 红色 2. red 红色 3. rgb(255,0,0) 红色 |

font-size：字号。这种多个单词组合的词，在css里面需要用-横线链接

实际上font是一个复合属性。单一属性书写时注意横线连接。

font-family：字体。

字体：中文字体、英文字体。

英文：Arial

中文字体：不同的电脑内安装的字体不同，为了保证所有用户浏览器显示字体相同，一般用计算机自带字体。黑体、宋体、微软雅黑。默认的是宋体。

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

英文字体写在前面，使用的中文字体写在后面，最后写一个保留的备用字体。字体用英文的双引号包裹，每个字体之间用逗号隔开。

网页有时追求加载速度，中文字体写成英文单词。

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

②盒子的样式。

盒子实体化：宽、高、背景色、边框

|  |
| --- |
| 1. width: 200px; 2. height: 200px; 3. background-color: rgb(0,255,255); 4. border: 1px solid black; |

背景是一个复合属性。

background-color：背景色。色值表示十六进制、单词、rgb、rgba。

border也是一个复合属性，边框。包含三部分：

线的宽度：1px。像素pixel。

线的类型：实线型solid。虚线型dashed。

线的颜色：色值的三种表示法。

|  |
| --- |
| 1. border: 1px solid black; |

## css的书写位置

css的引入有三种方式：行内式、内嵌式、外链式。

行内式：写在标签上面的，有一个style属性，涵盖了标签的所有css样式属性。

一般是后台程序员使用。

|  |
| --- |
| 1. <div style="color:purple">我是一行文字</div> |



内嵌式：css样式写在head标签内部的一个style标签里面。

style是一个双标签，css样式写在标签内部。

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

style标签有一个type属性，text/css表示内部书写的是纯文本类型的css代码。type属性要么写全写对，要么可以不写。

外链式：从外部引入一个css文件。学项目实战的时候来讲。

## 5、css的一些小事项

键值对的概念：

html的键值对：k=”v”.

css的键值对：k:v;

分号的重要性：分号必须写。

每一个属性之间必须用分号隔开。

css对空格、换行、缩进，也不敏感。

|  |
| --- |
| 1. div{width: 200px;height: 200px;background-color: rgb(0,255,255);border: 10px dashed black;font-size: 24px;font-family: "Arial","微软雅黑","宋体";} 2. p{color: red;} |

在书写的时候：每个属性都换行缩进。为了清晰易读。

在传到网页之前：对代码进行压缩，删除所有的空格、换行，让代码在一行显示。体积变小，加载速度变快。

# 基础选择器

选择器就是你想给他加样式的html元素或标签。

## 标签选择器

以标签本身作为选择器选择元素。标签是什么就写什么。

|  |
| --- |
| 1. p{ 2. color: red; 3. } 4. h3{ 5. color: green; 6. } |

p、h3都是我们的标签名，直接书写可以选择我们页面上所有的同种的标签。

不会管我们的嵌套有多少层，就算嵌套的层数再多再深都能选择中。

|  |
| --- |
| 1. <div> 2. <p>我是一个段落</p> 3. <p>我是一个段落</p> 4. <p>我是一个段落</p> 5. <p>我是一个段落</p> 6. </div> 7. <div> 8. <div> 9. <div> 10. <div> 11. <h3>我是一个深层次的h3</h3> 12. </div> 13. </div> 14. </div> 15. </div> |

我们一般情况不会同时更改所有的同种标签的样式。用途：是在清空默认样式、设置默认样式的时候。

## id选择器

通过一个html元素的id属性选择这个元素，添加样式。

id选择器书写：#开头，后面加id的属性值。

只选择某一个元素。

|  |
| --- |
| 1. #para{ 2. font-size: 30px; 3. } |

id的属性值命名规则：必须以字母开头（严格区分大小写，a和A表示的是不同的）、可以有数字、下划线和横线。

|  |
| --- |
| 1. <p id="P1\_-2">我是一个段落</p> |

注意：#后面必须紧跟id名，不能有空格。

一个id名在一个html页面只能出现一次。不论是不是相同的标签或不同的标签。

下面是错误的：

|  |
| --- |
| 1. <h3 id="para1">我是一个h3</h3> 2. <p id="para1">我是一个段落</p> |

层叠性体现1：我们可以通过p标签选择器和id选择器同时选中一个标签元素，给它添加不同的样式，这些样式都会在标签上面进行显示。

|  |
| --- |
| 1. <p id="P1\_-2">我是一个段落</p> |

标签选择器不好用：选中所有的元素。

id选择器也不好用：选中一个元素。

## 类选择器

通过元素上面的class属性的值可以选中相同类名的所有标签元素。

书写方式：以点.开头，后面紧跟类名。

可以选中一个或多个元素。

|  |
| --- |
| 1. **.**warning{ 2. color: orange; 3. } |

所有的以warning为类名的元素都被选中了



优势：可以给我们想要的元素添加统一的类型，设置统一的样式。才是我们真正常用的css选择器。

特殊应用：我们要学会使用“原子类”（公共类）。我们一般页面上会设置很多原子类，用到的时候给元素添加这个类名。原子类的属性尽量的少。如果一个标签有多个样式，可以添加多个原子类的类名。

|  |
| --- |
| 1. .warning{ 2. color: orange; 3. } 4. .fs{ 5. font-size:30px; 6. } 7. <p class="warning fs">我是一个段落</p> |

添加多个类名：一个class属性可以有多个值，每个值之间用空格隔开。

类名也有命名规则：同id一样，必须以字母开头（严格区分大小写），可以有数字、下划线和横线。

实际工作中，用类选择器的情况居多，用id选择器控制css样式的情况少。

id一般默认都有js行为。

**类上样式，id上行为。**

## 通配符

通配符选择器：\*。

可以选择我们所有的包括body在内的全部的标签。

|  |
| --- |
| 1. \*{ 2. color: red; 3. } |

具体使用：用它来快速清除默认样式。

效率不高，不是实际工作中要用的。

# 高级选择器

在基础选择器基础上，进行的引申。

## 后代选择器

通过有嵌套关系的标签选择或者类选择器，确定一个大的范围，在范围内再次进行选择，可以再多次选择。

书写规则：用空格隔开每一层的选择器。

|  |
| --- |
| 1. <div> 2. <div class="box"> 3. <p>我是一个段落</p> 4. <p>我是一个段落</p> 5. <p>我是一个段落</p> 6. <p>我是一个段落</p> 7. </div> 8. </div> |

在style里面用后代选择选中类名叫做box的div里面的p。

|  |
| --- |
| 1. .box p{ 2. color: green; 3. } |

如果是下面情况，同样能选中p元素。说明：后代表示的不止包含子级，还包括所有的孙子级，曾孙级……。

|  |
| --- |
| 1. <div class="box"> 2. <div> 3. <p>我是一个段落</p> 4. <p>我是一个段落</p> 5. <p>我是一个段落</p> 6. <p>我是一个段落</p> 7. </div> 8. </div> |

我们的嵌套关系不管有多少层，只要是嵌套在内的，都会被选择中。

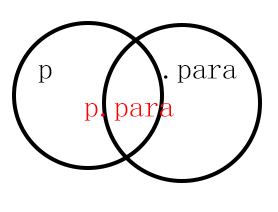
后代选择器可以连续写它的祖先级，可以把祖先级补充完整。尽可能的缩小范围，精确选择我们要的标签。

|  |
| --- |
| 1. .box div p{ 2. color: green; 3. } |

不管怎么写，html结构必须有对应的嵌套关系。

## 交集选择器

选中两个选择器都拥有的元素。



例：我想找一个标签为p元素，且类名叫para的元素

交集选择器写法：p.para。将两个选择器连续书写，中间没有任何空格。

|  |
| --- |
| 1. <div class="box"> 2. <p>我是第二层的p</p> 3. <div class="para"> 4. <p>我是一个段落</p> 5. <p class="para">我是一个段落</p> 6. <p>我是一个段落</p> 7. <p>我是一个段落</p> 8. </div> 9. </div> |

交集选择器可以正确选择。多种选择器的混用。

|  |
| --- |
| 1. p.para{ 2. color: yellow; 3. } |

注意：可以连续交集。

|  |
| --- |
| 1. p.para.f20{ 2. color: purple; 3. } 4. <p class="para ps f20">我是一个段落</p> |

表示既是p元素，类名也得叫para，类名还的叫f20。不满足任意一个条件，都不能选中。

连续交集从IE7开始兼容。

|  |
| --- |
| 1. .para.f20 可用的 2. #box.ps 不用，ID本来就有一个，用id选择器就可以直接选中 |

跟我们的后代选择器区分

|  |
| --- |
| 1. .box.ps 交集选择器，又是box，又是ps 2. .box .ps 后代选择器，box里面的ps。 |

## 并集选择器

选中多个元素，每个元素选择器之间用逗号隔开。表示“或”。

|  |
| --- |
| 1. <p>i am p!</p> 2. <p>i am p!</p> 3. <p>i am p!</p> 4. <h3>i am h3</h3> 5. <h3>i am h3</h3> 6. <h3>i am h3</h3> 7. <div>i am div</div> 8. <div>i am div</div> 9. <div>i am div</div> |

选中所有的p、h3、div，里面的文字都变红色

|  |
| --- |
| 1. p,h3,div{ 2. color: red; 3. } |

只要看到逗号，就是并集选择器。

|  |
| --- |
| 1. .ps,.h3s,.divs{ 2. color: green; 3. } |

# 继承性和层叠性

## 继承性

css里，有一些属性，如果给父级元素设置了，那么它的后代元素将都继承这个属性。

|  |
| --- |
| 1. <div class="box"> 2. <h3>标题</h3> 3. <p>段落</p> 4. <div> 5. <p>段落</p> 6. </div> 7. </div> |

给div.box盒子设置的样式，有一部分继承给了内部的元素。

|  |
| --- |
| 1. .box{ 2. width: 200px; 3. height: 200px; 4. background-color: #ccc; 5. color: blue; 6. font-size: 18px; 7. font-family: "微软雅黑"; 8. } |



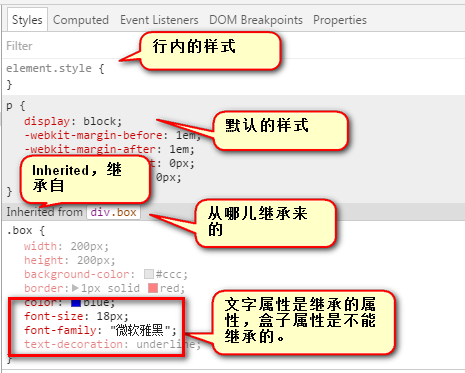
p、h3标签，从他的祖先元素继承了文字的属性，没有继承盒子的属性。

只有文字的属性能够继承，盒子的属性不能继承。

文字属性：color、font-系列、line-系列、text-系列。

盒子属性：background-系列、width、height、border-系列、浮动、定位等。

如何看控制台：



继承性是比较好的一个属性。

我们可以简化css代码，有的属性可以设置给body，body内部的标签都会继承这个属性。

层叠性体现2：css的继承性。

## 层叠性

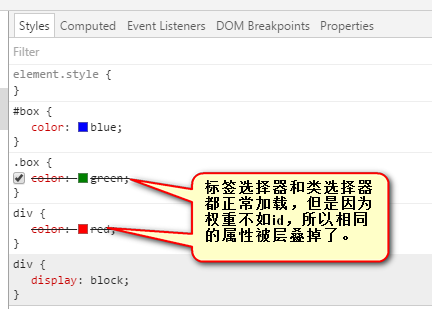
同一个元素可以被不同的选择器选中，设置样式。

问题：同一个属性，我听谁的？

涉及一个叫做选择器权重的概念。

|  |
| --- |
| 1. <div class="box" id="box"> 2. 我是一个div，看看我到底是什么颜色 3. </div> |

|  |
| --- |
| 1. <style type="text/css"> 2. div{ 3. color: red; 4. font-size: 30px; 5. } 6. .box{ 7. color: green; 8. } 9. #box{ 10. color: blue; 11. } 12. </style> |



最终显示蓝色，因为id选择器权重最高，相同的属性层叠掉了其他的选择器设置的属性。

权重：选择器的针对性越强，权重越高。

权重排序：id>class>标签。

层叠性：如果同时选中了一个元素，权重高的层叠掉权重低的属性。

①如果选中了这个元素。计算选择器的权重。

计算方法：比较id、类、标签的数量。依次比较，如果id的数量多，就权重最大。如果id的数量相同，比较类的数量，类多的权重大。如果类的数量也相同，比较标签的数量，标签多的权重大。标签还相同，那么看css里面的书写顺序，最后写的就是我们的样式。

实际上可以按数量比：1个id=255个类

|  |
| --- |
| 1. （id的数量，类的数量，标签的数量） |

看颜色：

|  |
| --- |
| 1. <div class="box" id="box"> 2. <div class="box2" id="box2"> 3. <div class="box3" id="box3"> 4. 我是一个div，看看我到底是什么颜色 5. </div> 6. </div> 7. </div> |

不同的选择器选中了同一个元素，比较权重。

|  |
| --- |
| 1. .box .box2 .box3{ 2. color: red; 3. } 4. .box div div{ 5. color: green; 6. } 7. .box #box2 .box3{ 8. color: blue; 9. } |



结果：蓝色。

如果权重相同，最后出现的就是我们显示的。

|  |
| --- |
| 1. <style type="text/css"> 2. #box .box2 .box3{ 3. color: red; 4. } 5. .box #box2 .box3{ 6. color: blue; 7. } 8. .box .box2 #box3{ 9. color: green; 10. } 11. </style> |



②如果没有选中元素，看继承性。

如果没有选中，也有权重，权重为0.不能进行权重的比较。

比较的是选择器离目标元素的距离。距离近的就是我们继承的属性。

下面：都没有选中目标元素，样式靠继承。遵循就近原则。距离不同，谁离的目标元素近，选谁的。

|  |
| --- |
| 1. <style type="text/css"> 2. body{ 3. color: red; 4. } 5. div .box2{ 6. color: green; 7. } 8. #box{ 9. color: blue; 10. } 12. </style> |

如果距离相同，就需要再次比较权重。权重高的是被选择。

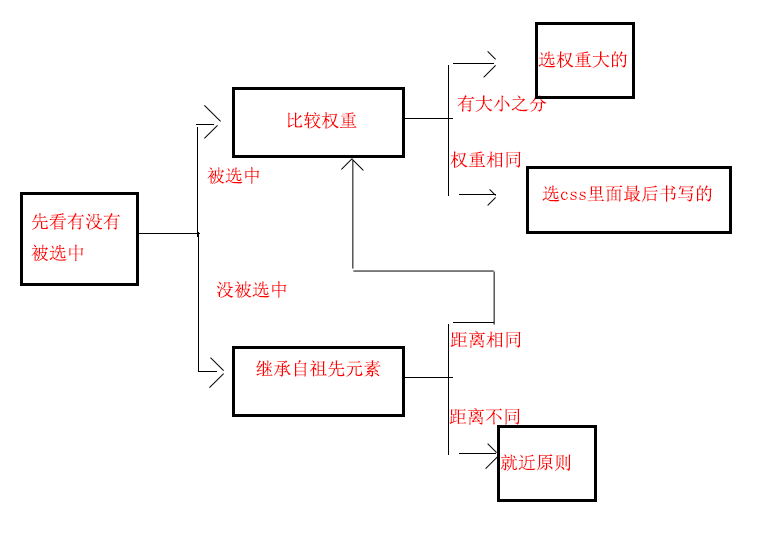
|  |
| --- |
| 1. .box .box2{ （0,2,0） 2. color: red; 3. } 4. #box #box2{ （2，0,0） 5. color: blue; 6. } 7. #box .box2{ （1，1,0） 8. color: green; 9. } |

继承来的元素都是一个，就比较权重。蓝色的权重大。



如果：距离相同，权重相同，谁写在最后面听谁的。

两种情况要记清：



提示：如果是并集选择器，你的拆开来看。

③特殊的关键词。！important。

如果选中了元素，它的权重最高。

如果没有选中元素，它不影响就近原则。

important作用的是单个的属性，而不是一个选择器。