# 1.css样式表由3种写法

## 1.内联样式，把style作为并且的属性，然后提交样式，如

|  |
| --- |
|  |

### 效果：

|  |
| --- |
|  |

## 2.内部样式，把样式代码写入head标签里面的style标签中，如

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Document</title>      <style>          body{              margin: 0;              padding: 0;              background-color: #f3f3f3;          }          div{              margin: 0;              background-color: deeppink;              text-align: center;          }          p{              border: 1px dashed #999;              background-color: aquamarine;              font-size: large;              font-family: Impact, Haettenschweiler, 'Arial Narrow Bold', sans-serif;          }      </style>  </head>  <body>      <div>          <p class="p1">I am page1</p>          <p class="p1">I am page2</p>          <p class="p1">I am page3</p>      </div>  </body>  </html> |

### 效果：

|  |
| --- |
|  |

## 3.外部样式，就是把样式保存在一个外部的css文件中，如我们新建一个index.css，把上面的代码剪切进去，然后在html里面用link标签引入

### index.css

|  |
| --- |
| body{      margin: 0;      padding: 0;      background-color: #f3f3f3;  }  div{      margin: 0;      background-color: deeppink;      text-align: center;  }  p{      border: 1px dashed #999;      background-color: aquamarine;      font-size: large;      font-family: Impact, Haettenschweiler, 'Arial Narrow Bold', sans-serif;  } |

### index.html

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>外部样式表</title>      <link rel="stylesheet" href="./index.css">  </head>  <body>      <div>          <p class="p1">I am page1</p>          <p class="p1">I am page2</p>          <p class="p1">I am page3</p>      </div>  </body>  </html> |

### 效果：和上面是一样的

|  |
| --- |
|  |

## 第三种写法是最常用的，第一种写法的优先级最高，我们以后都是用外部样式文件。

# 2.css选择器

## 1.标签选择器：element selector，如div，h1，p等等

|  |
| --- |
|  |

## 特点，一旦你用标签选择器设置看样式，所以最高标签的元素都会受到影响

|  |
| --- |
|  |

## 2.类选择器，class selector，选择特定类名的元素不管它是什么元素

|  |
| --- |
|  |

### 注意：分号不能少

|  |
| --- |
|  |

## 3.id选择器，id selector，选择使用id为指定名字的使用元素

|  |
| --- |
| **#myId{**  **color:deeppink;**  **}** |

### 效果：

|  |
| --- |
|  |

### 注意：在一个html文件中一个id只能用于一个元素，一个class可以用于任意多个元素

## 4.属性选择器，

### 1.根据属性来选择元素

下面的例子是把包含标题（title）的所有元素变为蓝色：

### 实例

[title] { color:blue; }

### 2.属性和值选择器

下面的实例改变了标题title='runoob'元素的边框样式:

### 实例

[title=runoob] { border:5px solid green; }

### 3.属性和值的选择器 - 多值

下面是包含指定值的title属性的元素样式的例子，使用（~）分隔属性和值:

### 实例

[title~=hello] { color:blue; }

下面是包含指定值的lang属性的元素样式的例子，使用（|）分隔属性和值:

### 实例

[lang|=en] { color:blue; }

|  |
| --- |
|  |
|  |

## 5.通配符选择器\*，选择所有元素，它的优先级最低

### 比如我们由这样子的一个页面

|  |
| --- |
|  |

### 效果：

|  |
| --- |
|  |

### 另外一个实例

|  |
| --- |
|  |

## 以上这几个是最常用的，还有很多，但是不太常用，有空可以看看

# 3.css颜色

## 1.颜色属性

|  |
| --- |
|  |

### 可以到这个网址里面查看有名称的颜色

#### [<named-color> - CSS：层叠样式表 | MDN (mozilla.org)](https://developer.mozilla.org/zh-CN/docs/Web/CSS/named-color)

### 如果你对这些颜色不满意，还可以来这个网站：

#### [Color Palettes for Designers and Artists - Color Hunt](https://colorhunt.co/)

### 这里还有一个rgb调色网站：

#### [RGB Mixer - Computer Science Field Guide (csfieldguide.org.nz)](https://www.csfieldguide.org.nz/en/interactives/rgb-mixer/)

# 4.字体Font

## 1.font-size的不同表达方法

|  |
| --- |
|  |

### 注意：

### 1.1em:和父元素的字体大小一样，2em：相当于父元素的字体大小\*2

### 2.在thml网页中，root元素就是指html标签

### 3.建议使用rem而不是em，因为这样子比较方便，尤其是网页比较复杂的情况

## 例子

|  |
| --- |
|  |

### 效果

|  |
| --- |
|  |

## 2.font-weight的不同表达方法

|  |
| --- |
|  |

### 注意：lighter是指父元素的font-weight -100，bolder是指父元素的font-weight+100

## 3.font-family，这个属性通常可以用多个值，用逗号隔开，意思就是如果没有找到第一个，可以继续找第二个，一直到找到为止。如：

|  |
| --- |
|  |

### 注意：如果font-family的名称有空格，需要使用双引号包裹

|  |
| --- |
|  |

### 可以到下面的网站下载免费字体

#### [Browse Fonts - Google Fonts](https://fonts.google.com/)，

#### 你需要复制字体的链接，然后在html文件里面用link并且引入，如图

|  |
| --- |
|  |

#### 引入之后，就可以通过family的值来使用

|  |
| --- |
|  |

### 当然，你也可以把字体下载下来

# 5.盒子模型，涉及到边框的宽度，边框样式，边框颜色还内边距和外边距，圆角，阴影等等

## 1.边框，和他有关的属性：border-width，border-style，border-color

### border-width

|  |
| --- |
|  |

#### 此时，你会发现，只是设置边框宽度并没有效果，因为他必须配置其他两个属性，另外属性值可以使用px或者%

### border-style，接着我们设置border-style位虚线

|  |
| --- |
|  |

#### 你会发现，此时依然没有效果，因为还差一个属性

### border-color，我们来将border-color设置位青绿色

|  |
| --- |
|  |

#### 此时，效果出来了，可见设置边框需要设置3给属性分别是宽度，边框样式和颜色，注意顺序不要搞错

## 其实，我们可以按照上面的顺序吧这三个属性合在一起写，不过此时使用的是border属性

|  |
| --- |
|  |

### 可以看到效果是一样的

## 关于设置边框的样式的属性是：上右下左，顺时针方向

### 设置p并且的四条边的看到分别是10px，15px，20px，25px，同样的道理，每一个属性都可以按照上右下左的方法来设置如图

|  |
| --- |
|  |

### 效果：

|  |
| --- |
|  |

#### 样子有点丑，不过说明是对的

## 还可以设置2个值，分别表示上边和下边，左边和右边，第一个值设置上边和下边，第二个值设置左边和右边，如我们修改一下样式如图

|  |
| --- |
|  |

### 效果：

|  |
| --- |
|  |

## 也可以设置3个值，表示上，左右，下，如

|  |
| --- |
|  |

## 效果：

|  |
| --- |
|  |

## 2.padding设置内边距，设置后其实元素的宽高没有改变，只是填充的空间把它“撑大了”

|  |
| --- |
|  |
|  |

## 3.外边距margin，也是不好分别元素宽高

|  |
| --- |
|  |

### 注意：虽然设置了内外边距后，元素的宽高没有变化，但是元素的外观是变化了，在计算位置是时候需要考虑内外边距的影响，否则达不到你要的效果

## 内外边距和边框的关系如图

|  |
| --- |
|  |

# 6.css优先顺序和继承

## 优先顺序inline>internal>external,也就是说，如果样式右冲突，internal的样式会覆盖external的样式，inline样式会覆盖internal样式，也就是优先级高的会覆盖优先级低的

|  |
| --- |
|  |
|  |
|  |

### 注意：如果你写两个样式属性名一样，但是属性值不一样，后面写的会覆盖前面写的

## CSS 优先规则1：

 最近的祖先样式比其他祖先样式优先级高。

例1：

<!-- 类名为 son 的 div 的 color 为 blue -->

<div style="color: red">

<div style="color: blue">

<div class="son"></div>

</div>

</div>

如果我们把一个标签从祖先那里继承来的而自身没有的属性叫做"祖先样式"，那么"直接样式"就是一个标签直接拥有的属性。又有如下规则：

## CSS 优先规则2：

"直接样式"比"祖先样式"优先级高。

例2：

<!-- 类名为 son 的 div 的 color 为 blue -->

<div style="color: red">

<div class="son" style="color: blue"></div>

</div>

## CSS 优先规则3：选择器的优先级

上面讨论了一个标签从祖先继承来的属性，现在讨论标签自有的属性。在讨论 CSS 优先级之前，先说说 CSS 7 种基础的选择器：

* ID 选择器， 如 #id{}
* 类选择器， 如 .class{}
* 属性选择器， 如 a[href="segmentfault.com"]{}
* 伪类选择器， 如 :hover{}
* 伪元素选择器， 如 ::before{}
* 标签选择器， 如 span{}
* 通配选择器， 如 \*{}

优先级关系：内联样式 > ID 选择器 > 类选择器 = 属性选择器 = 伪类选择器 > 标签选择器 = 伪元素选择器

#### 注意：你会发现，有些时候属性选择器的优先级会高于类选择器

例3：

// HTML

<div class="content-class" id="content-id" style="color: black"></div>

// CSS

#content-id {

color: red;

}

.content-class {

color: blue;

}

div {

color: grey;

}

最终的 color 为 black，因为内联样式比其他选择器的优先级高。

所有 CSS 的选择符由上述 7 种基础的选择器或者组合而成，组合的方式有 3 种：

* 后代选择符： .father .child{}
* 子选择符： .father > .child{}
* 相邻选择符: .bro1 + .bro2{}

当一个标签同时被多个选择符选中，我们便需要确定这些选择符的优先级。我们有如下规则：

## CSS 优先规则4：

计算选择符中 ID 选择器的个数（a），计算选择符中类选择器、属性选择器以及伪类选择器的个数之和（b），计算选择符中标签选择器和伪元素选择器的个数之和（c）。按 a、b、c 的顺序依次比较大小，大的则优先级高，相等则比较下一个。若最后两个的选择符中 a、b、c 都相等，则按照"就近原则"来判断。

例4：

// HTML

<div id="con-id">

<span class="con-span"></span>

</div>

// CSS

#con-id span {

color: red;

}

div .con-span {

color: blue;

}

由规则 4 可见，<span> 的 color 为 red。

如果外部样式表和内部样式表中的样式发生冲突会出现什么情况呢？这与样式表在 HTML 文件中所处的位置有关。样式被应用的位置越在下面则优先级越高，其实这仍然可以用规则 4 来解释。

例5：

// HTML

<link rel="stylesheet" type="text/css" href="style-link.css">

<style type="text/css">

@import url(style-import.css);

div {

background: blue;

}

</style>

<div></div>

// style-link.css

div {

background: lime;

}

// style-import.css

div {

background: grey;

}

从顺序上看，内部样式在最下面，被最晚引用，所以 <div> 的背景色为 blue。

上面代码中，**@import** 语句必须出现在内部样式之前，否则文件引入无效。当然不推荐使用 **@import** 的方式引用外部样式文件，原因见另一篇博客：[CSS 引入方式](https://www.runoob.com/w3cnote/html-import-css-method.html)。

CSS 还提供了一种可以完全忽略以上规则的方法，当你一定、必须确保某一个特定的属性要显示时，可以使用这个技术。

## CSS 优先规则5：

属性后插有 **!important** 的属性拥有最高优先级。若同时插有 **!important**，则再利用规则 3、4 判断优先级。

例6：

// HTML

<div class="father">

<p class="son"></p>

</div>

// CSS

p {

background: red !important;

}

.father .son {

background: blue;

}

虽然 .father .son 拥有更高的权值，但选择器 p 中的 background 属性被插入了 !important， 所以 <p> 的 background 为 red。

# 由于笔记太长，我们需要创建一个：[学习笔记05-css学习2](学习笔记05-css学习2.docx)