

# 1.CSS简介

CSS是**层叠样式表**(Cascading Style Sheets)的简称，有时我们也会称之为CSS样式表或级联样式表。

CSS也是一种标记语言

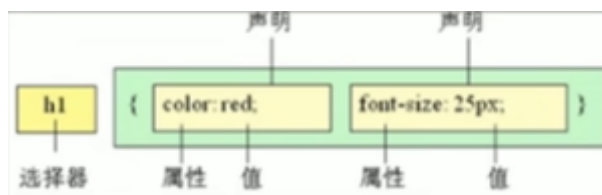
主要用于设置HTML页面中的**文本内容**(字体，大小，对齐方式等)、**图片的外形**（宽高，边框样式，边距等）以及板面的布局和**外观显示样式**

CSS可以美化HTML，让HTML更漂亮，让页面布局更简单。

**CSS最大价值：**由HTML专注去做结构呈现，样式交给CSS，即结构(HTML)与样式(CSS)相分离。

## 1.1CSS语法规范

CSS规则由两个主要的部分构成：选择器以及一条或多条声明。



- **选择器**是用于指定CSS样式的**HTML标签**，花括号内是该对象设置的具体样式
- 属性和属性值以“键值对”的形式出现
- 属性是对指定的对象设置的样式属性，例如字体大小、文本颜色等
- 属性和属性值之间用英文“`:`”分开

## 1.2CSS代码风格

### 1.2.1样式格式

#### ① 紧凑格式

```
h3 { color: deeppink;font-size: 20px;}
```

#### ② 展开格式

```
h3 {  
    color: pink;  
    font-size: 20px;  
}
```

**强烈推荐第二种格式**，因为更直观。

### 1.2.1样式大小写

一般都使用小写，特殊情况除外

### 1.2.1 空格位置

1. 属性值前面，冒号后面，保留一个空格
2. 选择器（标签）和大括号中间保留空格

## 2.CSS基础选择器

### 2.1选择器的作用

选择器就是根据不同的需求把不同的标签选出来。简单来说，就是选标签用的。

### 2.2选择器的分类

选择器分为基础选择器和复合选择器两个大类。先讲解基础选择器。

- 基础选择器是由单个选择器组成的
- 基础选择器包括：标签选择器、类选择器、id选择器和通配符选择器

### 2.3基础选择器

#### 2.3.1 标签选择器

标签选择器（元素选择器）是指用HTML标签名作为选择器，按标签名称分类，为页面中某一类标签指定统一的CSS样式。

语法：

```
1  标签名{
2      属性1: 属性值1;
3      属性2: 属性值2;
4      .....
5  }
```

缺点：不能差异化设计标签

#### 2.3.1 类选择器

差异化选择不同的标签，单独选一个或某几个标签

语法：

```
1  .类名{
2      属性1: 属性值1;
3      .....
4  }
5  /*例如：将所拥有red类的HTML元素均为红色*/
6  .red{
7      color:red;
8  }
9  结构需要**class属性**来调用class类
10 <div class="red">变红色</div>
```

类选择器口诀：样式点定义，结构类调用，一个或多个，开发最常用

注意：

- 类选择器使用“.”进行标识，后面紧跟类名（自己命的名）
- 长名或词组可以使用中横线为选择器命名
- 不要使用纯数字，中文等命名，尽量使用英文字母表示
- 命名要有意义，使用拼音尽量全拼

- 常用的命名规范：

头：header  
 内容：content/container  
 尾：footer  
 导航：nav  
 侧栏：sidebar  
 栏目：column  
 页面外围控制整体布局宽度：wrapper  
 左右中：left right center  
 登录条：loginbar  
 标志：logo  
 广告：banner  
 页面主体：main  
 热点：hot  
 新闻：news  
 下载：download  
 子导航：subnav

## 类选择器-多类名

我们可以给一个标签指定多个类名，从而达到更多的选择目的。这些类名都可以选出这个标签。简单理解就是一个标签有多个类名

### 多类名使用方式

```
1 <div class="red font20">芜湖</div>
```

- 在标签class属性中写多个类名
- 多个类名中间必须用空格分开

## 2.3.3 id选择器

id选择器可以标有特定id的HTML元素指定特定的样式

HTML元素以id属性来设置id选择器，CSS中id选择器以“#”来定义

语法：

```

1 #id名{
2     属性1: 属性值1;
3     .....
4 }
```

id选择器口诀：样式#定义，结构id调用，只能调用一次，别人切勿使用

id选择器一般用于页面唯一性的元素上，经常和JavaScript搭配使用

### 2.3.4 通配符选择器

在CSS中，通配符选择器使用“`*`”定义，它表示选取页面中所有元素（标签）

语法：

```
1 * {
2     属性1: 属性值1;
3     .....
4 }
```

- 通配符选择器不需要调用，自动给所有的元素使用样式
- 特殊情况才使用，后面讲解使用场景（以下是清楚所有的元素标签内外边距）

```
1 * {
2     margin: 0;
3     padding: 0;
4 }
```

### 2.3.5基础选择器总结

基础选择器	作用	特点	使用情况	用法
标签选择器	可以选出所有相同的标签， 比如p	不能差异化选择	较多	p { color: red;}
类选择器	可以选出1个或者多个标签	可以根据需求选择	非常多	.nav { color: red; }
id选择器	一次只能选择1个标签	ID属性只能在每个 HTML 文档中出现一次	一般和js搭配	#nav {color: red;}
通配符选择器	选择所有的标签	选择的太多，有部分不需要	特殊情况使用	* {color: red;}

## 3.CSS字体属性

CSS Fonts（字体）属性用于定义字体系列、大小、粗细和文字样式（如斜体）

### 3.1字体系列

CSS使用 `font-family` 属性定义文本的字体系列

```
1 p {font-family:"微软雅黑";}
2 div {font-family: Arial,"Microsoft Yahei","微软雅黑";}
```

语法规范：

- 各种字体之间必须使用英文状态下的逗号隔开
- 一般情况下，如果有空格隔开的多个单词组成的字体，加引号
- 尽量使用系统默认自带字体，保证在任何用户的浏览器中都能正确显示

## 3.2 字体大小

CSS使用 `font-size` 属性定义字体大小

```
1 p {  
2   font-size: 20px;  
3 }
```

- px大小是我们网页开发最常用的单位
- 谷歌浏览器默认的文字大小是16px
- 不同浏览器默认显示的字号大小不一致，尽量给出明确值大小，不要默认大小
- 可以给 `body` 指定整个页面文字的大小

## 3.3 字体粗细

CSS使用 `font-weight` 属性定义字体粗细

参数：

`normal`：正常的字体。相当于number为400.声明此值将取消之前任何设置。

`bold`：粗体。相当于number为700.也相当于b对象的作用。

`bolder`：特粗体

`lighter`：细体

number: 100 200 300 400 500 ....

```
1 font-weight: 400;  
2 /*数字后面不要加单位*/
```

实际开发中，更喜欢用数字加粗。

## 3.4 文字样式

CSS使用 `font-style` 定义文字样式

属性值	作用
normal	默认值，浏览器会显示标准的字体样式 <code>font-style: normal;</code>
italic	浏览器会显示斜体的字体样式。

**注意：**平时很少给字体加倾斜，反而要给倾斜标签 `<em>` `<i>` 改为不倾斜字体。

## 3.5 字体复合属性

```
1 font: font-style font-weight font-size/line-height font-family  
2 /*例如: */  
3 font: italic 700 16px 'Microsoft yahei'
```

**注意：**

- 使用 `font` 属性时，必须按照上面的格式顺序书写，不能更换顺序，并且各个属性间以**空格**隔开
- 不需要设置的属性可以省略（取默认值），但必须保留 `font-size` 和 `font-family` 属性，否则 `font` 属性不起作用

### 3.6字体属性总结

属性	表示	注意点
font-size	字号	我们通常用的单位是px 像素，一定要跟上单位
font-family	字体	实际工作中按照团队约定来写字体
font-weight	字体粗细	记住加粗是 700 或者 bold 不加粗 是 normal 或者 400 记住数字不要跟单位
font-style	字体样式	记住倾斜是 italic 不倾斜 是 normal 工作中我们最常用 normal
font	字体连写	1. 字体连写是有顺序的 不能随意换位置 2. 其中字号 和 字体 必须同时出现

### 4.CSS文本属性

CSS Text（文本）属性可以定义文本的**外观**，比如文本的颜色、对齐方式、装饰文本、文本缩进、行间距等

#### 4.1文本属性

```
1  div {
2      color:red;
3      color:#ff0000;/*16进制表示*/
4      color:rgb(255,0,0)
5  }
```

表示表示	属性值
预定义的颜色值	red, green, blue, 还有我们的御用色 pink
十六进制	#FF0000, #FF6600, #29D794
RGB代码	rgb(255,0,0)或rgb(100%,0%,0%)

开发中最常用的是16进制。

#### 4.2对齐文本

`text-align` 属性用于设置元素内文本内容的水平对齐方式。

属性值	解释
left	左对齐（默认值）
right	右对齐
center	居中对齐

```
1  h1 {
2      text-align:center;
3      /*本质是让h1中的文字水平居中对齐*/
4  }
```

## 4.3装饰文本

`text-decoration` 属性规定添加到文本的修饰。可以给文本加下划线，删除线，上划线等。

属性值	描述
none	默认。没有装饰线（最常用）
underline	下划线。链接 <code>a</code> 自带下划线（常用）
overline	上划线。（几乎不用）
line-through	删除线。（不常用）

```
1 a {  
2   text-decoration: none;  
3   /*取消a自带的下划线*/  
4 }
```

## 4.4文本缩进

`text-indent` 属性用来指定**文本的第一行**的缩进，通常是段落的首行缩进。

```
1 p {  
2   text-indent: 20px;  
3 }
```

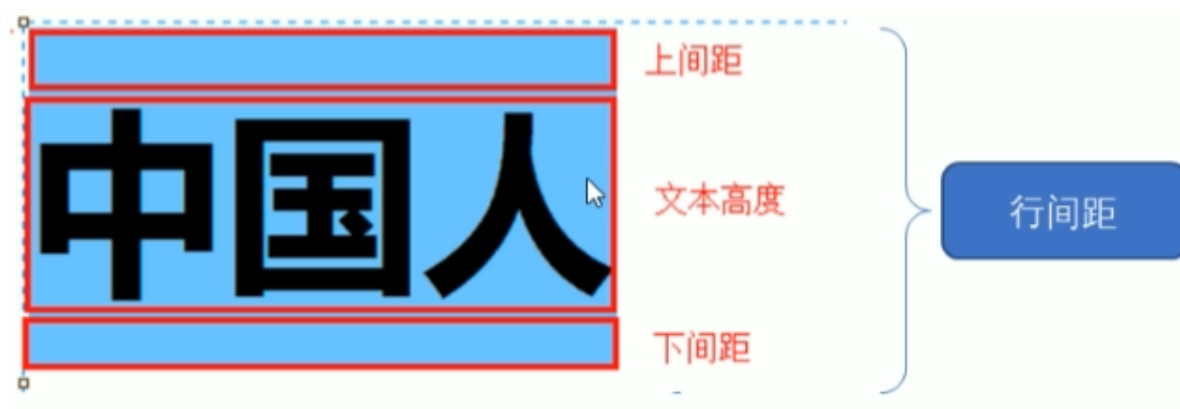
通过设置该属性，所有元素的第一行都可以缩进一个给定的长度，甚至可以是负值。

```
1 p {  
2   text-indent: 2em;  
3 }
```

`em` 是一个相对单位，就是当前元素( `font-size` )1个文字的大小，如果当前元素没有设置大小，则会按照父元素的1个文字大小。

## 4.5行间距

`line-height` 属性用于设置行间距的距离（行高）。可以控制文字行与行之间的距离。



## 4.6 文本属性总结

属性	表示	注意点
color	文本颜色	我们通常用 十六进制 比如 而且是简写形式 #fff
text-align	文本对齐	可以设定文字水平的对齐方式
text-indent	文本缩进	通常我们用于段落首行缩进2个字的距离 text-indent: 2em;
text-decoration	文本修饰	记住 添加下划线 underline 取消下划线 none
line-height	行高	控制行与行之间的距离

## 5. CSS引入方式

### 5.1 CSS的三种样式表

#### 5.1.1 内部样式表

内部样式表（内嵌样式表）是写到html页面内部，是将所有的CSS代码抽取出来，单独放到一个 `<style>` 标签中。

```
1 <style>
2     div {
3         color:red;
4     }
5 </style>
```

- `<style>` 标签理论上可以放到HTML文档的任何地方，但一般放在 `<head>` 标签中
- 通过此种方式，可以方便控制当前整个页面中的元素样式设置
- 代码结构清晰，但是并没有实现结构与样式完全分离

#### 5.1.2 行内样式表

在元素标签内部style属性中设定CSS样式，适合于修改简单的样式

```
1 <p style="color:pink;font-size:12px;">
2     青春不常在，抓紧谈恋爱！
3 </p>
```

- style是标签的属性；双引号中间要符合CSS规范
- 可以控制当前的标签设置样式

#### 5.1.3 外部样式表

适合于样式比较多的情况，核心是：样式单独写到CSS文件中，之后把CSS文件引入到HTML页面使用

方法：

- 1.新建一个后缀名为 `.css` 的文件，把所有CSS代码都放入此文件中
- 2.在HTML页面中，使用 `<link>` 标签引入这个文件



```
1 | <link rel="stylesheet" href="css文件路径">
```

属性	作用
rel	定义当前文档与被链接文档之间的关系，在这里需要指定为“stylesheet”，表示被链接的文档是一个样式表文件。
href	定义所链接外部样式表文件的URL，可以是相对路径，也可以是绝对路径。

## 5.2 CSS引入方式总结

样式表	优点	缺点	使用情况	控制范围
行内样式表	书写方便，权重高	结构样式混写	较少	控制一个标签
内部样式表	部分结构和样式相分离	没有彻底分离	较多	控制一个页面
外部样式表	完全实现结构和样式相分离	需要引入	最多，吐血推荐	控制多个页面

## Chrome调试工具

2. 使用调试工具

①

Ctrl+滚轮

可以放大开发者工具代码大小。

②

左边是 HTML 元素结构，右边是 CSS 样式。

③

右边 CSS 样式可以改动数值（左右箭头或者直接输入）和查看颜色。

④

Ctrl + 0

复原浏览器大小。

⑤

如果点击元素，发现右侧没有样式引入，极有可能是类名或者样式引入错误。

⑥

如果有样式，但是样式前面有黄色叹号提示，则是样式属性书写错误。

## 6. Emmet语法

### 6.1 快速生成HTML结构语法

1. 生成标签 直接输入标签名 按tab键即可 比如 div 然后tab 键，就可以生成 <div></div>
2. 如果想要生成多个相同标签 加上 \* 就可以了 比如 div\*3 就可以快速生成3个div
3. 如果有父子级关系的标签，可以用 > 比如 ul > li就可以了
4. 如果有兄弟关系的标签，用 + 就可以了 比如 div+p
5. 如果生成带有类名或者id名字的，直接写 .demo 或者 #two tab 键就可以了
6. 如果生成的div类名是有顺序的，可以用自增符号 \$
7. 如果想要在生成的标签内部写内容可以用 {} 表示

## 6.2 快速生成CSS样式

CSS基本采用简写形式即可

1. w200 按tab，可以生成 width: 200px;
2. lh26 按tab，可以生成 line-height: 26px

## 6.3 快速格式化代码

vscode快捷键: `Alt + shift + f`

也可以设置 当我们 保存页面的时候自动格式化代码:

- 1) 文件 -----> 【首选项】-----> 【设置】;
- 2) 搜索emmet.include;
- 3) 在settings.json下的【用户】中添加以下语句:

```
"editor.formatOnType": true,  
"editor.formatOnSave": true
```

只需要设置一次即可，以后都可以自动保存格式化代码

## 7. 复合选择器

在 CSS 中，可以根据选择器的类型把选择器分为**基础选择器**和**复合选择器**，复合选择器是建立在基础选择器之上，对基本选择器进行组合形成的。

- 复合选择器可以更准确、更高效的选择目标元素（标签）
- 复合选择器是由两个或多个基础选择器，通过不同的方式组合而成的
- 常用的复合选择器包括：后代选择器、子选择器、并集选择器、伪类选择器等等

