



CSS简介

1. HTML主要做结构，显示元素内容
2. CSS美化HTML，布局网页
3. CSS最大价值：由HTML专注去做结构呈现，样式交给CSS，即结构（HTML）与样式相分离。

CSS语法规规范

使用HTML时，需要遵从一定的规范，CSS也是如此。要想熟练地使用CSS对网页进行修饰，首先需要了解CSS样式规则。

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，花括号内是对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
- 多个“键值对”之间用英文";"进行区分

CSS代码风格

以下代码书写疯狂不是强制规范，而是符合实际开发书写方式

1. 样式格式书写
使用展开的方式书写

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

2. 样式大小写
选择器、属性名、属性值关键字全部使用小写字母，特殊情况除外。

```
h3 {  
  color: pink;  
}
```

3. 空格规范

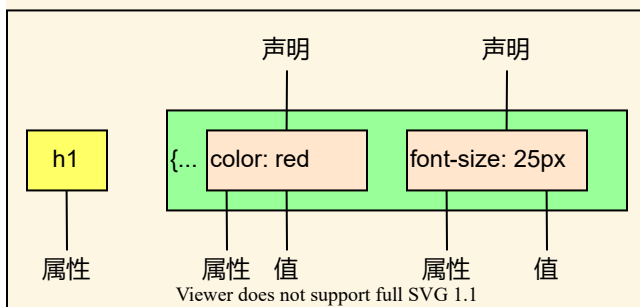
```
h3 {  
  color: pink;  
}
```

- ① 属性前面，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

CSS基础选择器

CSS选择器的作用

选择器（选择符）就是根据不同需求把不同的标签选出来，这就是选择器的作用。简单来说，就是选择标签用的。



以上CSS做了两件事：

1. 找到所有的h1标签。选择器（选对人）。
2. 设置这些标签的样式，比如颜色为红色（做对事）。

选择器分类

选择器分为基础选择器和复合选择器两个大类。

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

标签选择器

标签选择器（元素选择器）是指用HTML标签名称作为选择器，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：

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

作用

标签选择器可以某一类标签全部选择出来，比如所有的<div>标签和所有的标签

优点

能快速为页面中同类型的标签统一设置样式

缺点

不能设计差异化样式，只能选择全部的当前标签

类选择器

如果想要差异化选择不同的标签，单独选一个或者某几个标签，可以使用类选择器。

类选择器在HTML中以class属性表示，在CSS中，类选择器以一个点"."号显示。

语法：

```
.类名 {  
  属性1: 属性值1;  
  ...  
}
```

例如，将所有用于red类的HTML元素均为红色。

```
.red {  
  color: red;  
}
```

注意：

- 类选择器使用"."（英文点号）进行标识，后面紧跟类名（自定义，我们自己命名的）。
- 可以理解为给这个标签起了一个名字来表示。
- 长名称或词组可以使用中横线来为选择器命名。

```
.star-sing {  
  color: green;  
}
```

- 不要使用纯数字、中文等命名，尽量使用英文字母来表示。
- 命名要有意义，尽量使别人一眼就知道这个类名的目的。

类选择器的的口诀：样式点定义，结构类（class）调用，一个或多个，开发最常用

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名字。

1. 多类名使用方式

- 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

```
\<div class="red font20">亚瑟\</div>
```

2. 多类名的使用场景

- 可以把一些标签元素相同的样式（共同的部分放到一个类里面）。
- 这些标签都可以调用这个公共的类，然后再调用自己独有的类。

id选择器

id选择器可以为标有特定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：

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

例如，将id为nav的元素中的内容设置为红色。

```
#nav {  
    color: red;  
}
```

id选择器和类选择器的区别

1. 类选择器(class)好比人的名字，一个人可以有多个名字，同时一个名字也可以被多个人使用。
2. id选择器好比人的身份证号码，全国唯一的，不可以重复
3. id选择器和类选择器最大的不容在于使用次数上
4. 类选择器在修改样式中用得最多，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。
5. id选择器的优先级比类选择器更高。

id选择器只能在一个位置被调用一次。

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

```
* {
  属性1: 属性值1;
  ...
}
```

- 通配符选择器不需要调用，自动就给所有的元素使用样式
- 特殊情况才使用。

基础选择器总结

基础选择器	作用	特点	使用情况
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
类选择器	可以选出一个或多个标签	可以根据需求选择	非常多
id选择器	一次只能选择一个标签	id属性只能在每个HTML文档中出现一次	一般和js搭配使用
通配符选择器	选择所有标签	选择的太多，有部分不需要	特殊情况使用

- 每个基础选择器都有使用场景，都需要掌握
- 如果是修改样式，类选择器是使用最多的

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）。

字体系列

CSS 使用font-family属性定义文本的字体系列。

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

- 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
- 最常见的几个字体："Microsoft Yahei", tahoma,arial,"Hiragino Sans GB"

字体大小

CSS 使用 font-size 属性定义字体大小。

```
p{
    font-size: 20px;
}
```

- px(像素) 大小是我们网页的最常用的单位
- 谷歌浏览器默认的文字大小为16px
- 不同浏览器可能默认显示的自豪大小不一致，我们尽量给一个明确值大小，不要默认大小。
- 可以给body指定整个页面文字的大小。

字体粗细

CSS 使用font-weight属性设置文字的粗细。

值	说明
normal	正常粗细。与400等值。
bold	加粗。与700等值。

值	说明
lighter	比从父元素继承来的值更细(处在字体可行的粗细值范围内)。
bolder	比从父元素继承来的值更粗 (处在字体可行的粗细值范围内)。
<number>	一个介于 1 和 1000 (包含) 之间的 <number> 类型值。更大的数值代表字体重量粗于更 (或一样粗)。一些常用的数值对应于通用的字体重量名称，如章节常见粗细值名称和数值对

文字样式

CSS 使用 font-style 属性设置文本的风格

```
P {  
  font-style: normal;  
}
```

值	说明
normal	默认值，浏览器会显示标准的字体样式 font-style: normal
italic	浏览器会显示斜体的字体样式

字体复合属性

字体属性可以把文字样式综合来写，这样可以更节约代码

```
body {  
  font: font-style font-weight font-size/line-height font-family;  
}
```

- 使用font属性时，必须按上面语法格式中的顺序书写，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属性将不起作用。

CSS文本属性

CSS Text（文本）属性可定义文本的外观，比如文本的颜色、对齐文本、装饰文本、文本缩进、行间距等。

文本颜色

color 属性用于定义文本的颜色

```
div {
  color: red;
}
```

表示方式	属性值
预定义的颜色值	red,green,blue ...
十六进制	#ff0000, #ff6600, #29d794
RGB代码	rgb(255,0,0)或rgb(100%, 0%, 0%)

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。

```
div {
  text-align: center;
}
```

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

装饰文本

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

```
div {
  text-decoration: underline;
}
```

表示方式	属性值
none	默认，没有装饰线
underline	下划线。链接a自带下划线(常用)
overline	上划线
line-through	删除线

文本缩进

text-indent属性用来指定文本的第一行的缩进， 通常是将段落的首行缩进。

```
div {
  text-indent: 10px;
}
```

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， 甚至该长度可以是负值。

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

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。

行间距

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

```
p {  
  line-height: 26px;  
}
```

CSS 引入方式

内部样式表

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

```
<style>  
  p {  
    line-height: 30px;  
  }  
</style>
```

- <style> 标签理论上可以放在HTML文档的任何地方，但一般会放在文档的<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。
- 使用内部样式表设定CSS，通常也被称为嵌入式引入，这种方式是我们练习时常用的方式。

行内样式表

行内样式表（内联样式表）是在元素标签内部的style属性中设定CSS样式。适合修改简单样式。

```
\<div style="color: red; font-size: 32px">青青河边草</div>
```

- style 其实就是标签的属性
- 在双引号中间，写法要符合CSS规范
- 可以控制当前的标签设置样式
- 由于书写凡是，并且没有体现出结构与样式分离的思想，所以不推荐大量使用，只有对当前元素添加简单样式的时候，可以考虑使用。
- 使用行内样式设定CSS，通常也被称为行内式引入。

外部样式表

实际开发都是外部样式表。适合样式比较多的情况，核心是：样式单独写道CSS文件中，之后把CSS文件引入到HTML页面中使用。

引入外部样式表分为两步：

1. 新建一个后缀名为 .css 的样式文件，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。

```
\<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
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

- 使用外部样式表设定CSS，通常也被称为外链式引入，这种方式是开发中常用的方式。

Emmet语法

快速生成html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`，再按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

快速生成CSS样式语法

CSS样式采取简写形式即可

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

CSS 的复合选择器

在CSS中，复合选择器就是对基本选择器进行组合形成的。

- 复合选择器可以更准确、更高效地选择目标元素（标签）
- 复合选择器是由两个或以上的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

后代选择器

后代选择器又称为包含选择器，可以选择父元素里面的子元素。其写法就是把外层标签写在前面，内层标签写在后面，中间用空格分隔。当标签发生嵌套时，内层标签就成为外层标签的后代。

```
元素1 元素2 { 样式声明 }
```

上述语法表示选择元素1里面的所有元素2（后代元素）

例如：

```
ul li {样式声明} /* 选择ul里面所有的li标签元素 */
```

- 元素1和元素2中间用空格隔开
- 元素1是父级，元素2是子级，最终选择的是元素2
- 元素2可以是儿子，也可以是孙子等，只要是元素1的后代即可。

子选择器

子元素选择器（子选择器）只能选择作为某元素的最近一级子元素。简单理解就是选亲儿子元素。

语法：

```
元素1>元素2 {样式声明}
```

上述语法表示选择元素1里面的所有直接后代（子元素）元素2。

例如：

```
div > p {样式声明} /* 选择div里面所有最近一级p标签元素 */
```

- 元素1和元素2中间用大于号隔开
- 元素1是父级，元素2是子级，最终选择的是元素2
- 元素2必须是亲儿子，其孙子、重孙子之类都不归它管，因此也可以叫亲儿子选择器。

并集选择器

并集选择器可以选择多组标签，同时为它们定义相同的样式。通常用于集体声明。

并集选择器是各选择器通过英文逗号(,)连接而成，任何形式的选择器都可以作为并集选择器的一部分。

语法：

```
元素1, 元素2 {样式声明}
```

上述语法表示选择元素1和元素2.

例如：

```
ul,div {样式声明}  /* 选择ul和div标签元素 */
```

- 元素1和元素2中间用逗号隔开
- 逗号可以理解为和的意思
- 并集选择器通常用于集体声明

伪类选择器

伪类选择器用于向某些选择器添加特殊的效果，比如给链接添加特殊效果，或选择第1个，第n个元素。

伪类选择器书写最大特点是用冒号(:)表示，比如: hover、:first-child。

伪类选择器很多，比如链接伪类、结构伪类等。

链接伪类选择器

```
a:link      /* 选择所有未被访问的链接 */
a:visited   /* 选择所有已被访问的链接 */
a:hover     /* 选择鼠标指针位于其上的链接 */
a:active    /* 选择活动链接（鼠标按下未弹起的链接） */
```

链接伪类选择器注意事项：

1. 为了确保生效，请按照LVHA的顺序声明 :link - :visited - :hover - :active。
2. 因为a链接在浏览器中具有默认样式，所以在实际工作中要给链接单独指定样式。

focus伪类选择器

:focus 伪类选择器用于选取获得焦点的表单元素。

焦点就是光标，一般情况<input>类表单元素才能获取，因此这个选择器也主要针对于表单元素

来说。

```
input:focus {
  background-color: red;
}
```

复合选择器总结

选择器	作用	特征	使用情况	分割符号
后代选择器	用来选择后代元素	可以是子孙后代	较多	符号是空格 如：
子代选择器	选择最近一级元素	只能是亲儿子	较少	符号是大于号 如：
并集选择器	选择某些相同样式的元素	可以用于集体声明	较多	符号是逗号 如：
链接伪类选择器	选择不同状态的链接	跟链接相关	较多	重点记住a{}和a:hover
:focus选择器	选择获得光标的表单	跟表单相关	较少	input:focus记得

元素显示模式

元素显示模式就是元素（标签）以什么方式进行显示，比如<div>自己占一行，比如一行可以放多个

HTML元素一般分为块元素和行内元素。

块元素

常见的块元素有：<h1>~<h6>、<p>、<div>、、、等，其中<div>是最典型的块元素。

块元素的特点：

- 1. 比较霸道，自己独占一行。
- 2. 高度、宽度、外边距以及内边距都可以控制

3. 宽度默认是容器的100%
4. 是一个容器及盒子，里面可以放行内或者块元素

注意：

- 文字类的元素内不能使用块元素
- `<p>`标签主要用于存放文字，因此`<p>`里面不能放块级元素，特别是不能放`<div>`
- 同理，`<h1>~<h6>`等都是文字类块级标签，里面也不能放其他块级元素

行内元素

常见的行内元素有`<a>`、``、``、``、`<i>`、``、`<s>`、`<ins>`、`<u>`、``等，其中``标签是典型的行内元素。有的地方也将行内元素成为内联元素。

行内元素的特点：

1. 相邻行内元素在一行上，一行可以显示多个。
2. 高、宽直接设置是无效的。
3. 默认宽度就是它们本身内容的宽度
4. 行内元素只能容纳文本或其他行内元素

注意：

- 链接里面不能再放链接
- 特殊情况链接`<a>`里面可以放块级元素，但是给`<a>`转换下一块级元素最安全。

行内块元素

在行内元素中有几个特殊的标签——``、`<input />`、`<td>`，它们同时具有块元素和行内元素的特点。有些资料将它们称为行内块元素。

行内块元素的特点：

1. 和相邻行内块元素（行内块）在一行上，但是他们之间会有空白缝隙。一行可以显示多个（行内元素特点）。
2. 默认宽度就是它本身内容的宽度（行内元素特点）。
3. 宽度、高度、外边距以及内边距都可以控制（块元素特点）。

元素显示模式转换

特殊情况下，它们需要元素模式的转换，简单理解：一个模式的元素需要另外一种模式的特性，比如想要增加链接<a>的触发范围。

- 转换为块元素：display:block;
- 转换为行内元素：display:inline;
- 转换为行内块元素：display: inline-block;

CSS的背景

通过CSS背景属性，可以给页面元素添加背景样式。
背景属性可以设置背景颜色、背景图片、背景平铺、背景图片位置、背景图像固定等。

背景颜色

background-color属性定义了元素的背景颜色。

```
background-color: 颜色值;
```

一般情况下元素背景颜色默认值是transparent（透明），我们也可以去手动指定背景颜色为透明色。

背景图片

background-image属性描述了元素的背景图像。实际开发常见于logo或者一些装饰性的小图片或者是超大的背景图片，优点是非常便于位置控制。（精灵图也是一种运用场景）

```
background-image: none | url (url)
```

参考值	作用
none	无背景图片(默认的)

参考值	作用
url	使用绝对或相对地址指定背景图像

背景平铺

如果需要在HTML页面上对背景图像进行平铺，可以使用background-repeat属性。

```
background-repeat: repeat | no-repeat | repeat-x | repeat-y
```

背景图片位置

利用background-position属性可以改变图片在背景中的位置。

```
background-position: x y;
```

参数代表的意思是：x坐标和y坐标。可以使用方位名词或者精确单位。

参数值	说明
length	百分数 由浮点数字和单位标识符组成的长度值
position	top center bottom left center right 方位名词

1. 参数是方位名词
 - 如果指定的两个值都是方位名词，则两个值前后顺序无关，比如left top 和 top left 效果一致。
 - 如果只指定了一个方位词，另一个省略，则第二个值默认居中对齐。
2. 参数是精确单位
 - 如果参数值是精确坐标，那么第一个肯定是x坐标，第二个一定是y坐标。
 - 如果只指定一个数值，那该数值一定是x坐标，另一个默认垂直居中。
3. 参数是混合单位
 - 如果指定的两个值是精确单位和方位名词混合使用，则第一个值是x坐标，第二个值是y坐标。

背景图像固定（背景附着）

background-attachment 属性设置背景图像是否固定或者随着页面的其余部分滚动。

background-attachment后期可以制作视差滚动的效果。

```
background-attachment: scroll | fixed
```

参数	作用
scroll	背景图像随对象内容滚动
fixed	背景图像固定

背景属性复合写法

为了简化背景属性的代码，我们可以将这些属性合并简写在同一个属性background中。从而节约代码量

当使用简写属性时，没有特定的书写顺序，一般习惯约定顺序为：

background: 背景颜色 背景图片地址 背景平铺 背景图像滚动 背景图片位置；

```
background: transparent url(image.jpg) repeat-y fixed top;
```

这是我们实际开发中我们提倡的写法。

背景色半透明

CSS3 为我们提供了背景颜色半透明的效果。

```
background: rgba(0,0,0,0.3)
```

- 最后一个参数是alpha透明度，取值范围在0~1之间。
- 背景半透明是指盒子背景半透明，盒子里面的内容不受影响
- 该属性为CSS3新增属性

CSS 的三大特性

CSS 有三个非常重要的特性：层叠性、继承性、优先级。

层叠性

相同选择器设置相同的样式，此时一个样式就会覆盖（层叠）另一个冲突的样式。层叠性主要解决样式冲突的问题。

层叠性原则：

- 样式冲突，遵循就近原则，哪个样式离结构近，就执行哪个样式
- 样式不冲突，不会层叠

继承性

CSS 中的继承：子标签会继承父标签的某些样式，如文本颜色和字号。

- 恰当地使用继承可以简化代码，降低CSS样式的复杂性。
- 子元素可以继承父元素的样式(text-,font-,line-这些元素开头的可以继承，以及color属性)

行高的继承：

```
body {  
  font: 12px/1.5 ...  
}
```

- 行高可以跟单位也可以不跟单位