

# CSS 介绍

---

## 1. 什么是 CSS

CSS全称为Cascading Style Sheets，译为层叠样式表。样式定义如何显示HTML元素。样式通常存储在样式表中。

CSS 是能够真正做到网页表现与内容分离的一种样式设计语言。相对于传统 HTML 的表现而言，CSS 能够对网页中的对象的位置排版进行像素级的精确控制，支持几乎所有的字体字号样式，拥有对网页对象和模型样式编辑的能力，并能够进行初步交互设计，是目前基于文本展示最优秀的表现设计语言。

## 2. 发展历史

从 1990 年代初 HTML 被发明开始，样式表就以各种形式出现了，不同的浏览器结合了它们各自的样式语言，读者可以使用这些样式语言来调节网页的显示方式。一开始样式表是给读者用的，最初的 HTML 版本只含有很少的显示属性，读者来决定网页应该怎样被显示。

但随着 HTML 的成长，为了满足设计师的要求，HTML 获得了很多显示功能。随着这些功能的增加外来定义样式的语言越来越没有意义了。

1994 年哈坤·利提出了 CSS 的最初建议。伯特·波斯（Bert Bos）当时正在设计一个叫做“Argo”的浏览器，他们决定一起合作设计 CSS。

当时已经有过一些样式表语言的建议了，但 CSS 是第一个含有“层叠”的主意的。在 CSS 中，一个文件的样式可以从其他的样式表中继承下来。读者在有些地方可以使用他自己更喜欢的样式，在其他地方则继承，或“层叠”作者的样式。这种层叠的方式使作者和读者都可以灵活地加入自己的设计，混合各人的爱好。

哈坤于 1994 年在芝加哥的一次会议上第一次展示了 CSS 的建议，1995 年他与波斯一起再次展示这个建议。当时 W3C 刚刚创建，W3C 对 CSS 的发展很感兴趣，它为此组织了一次讨论会。哈坤、波斯和其他一些人（比如微软的托马斯·雷尔登）是这个项目的主要技术负责人。1996 年底，CSS 已经完成。1996 年 12 月 CSS 要求的第一版本被出版。

1997 年初，W3C 内组织了专门管 CSS 的工作组，其负责人是克里斯·里雷。这个工作组开始讨论第一版中没有涉及到的问题，其结果是 1998 年 5 月出版的第二版要求。到 2007 年为止，第三版还未完备。

## 3. 版本历史

- CSS1 —— 作为一项 W3C 推荐，CSS1 发布于 1996 年 12 月 17 日。1999 年 1 月 11 日，此推荐被重新修订。
- CSS2 —— 作为一项 W3C 推荐，CSS2 发布于 1999 年 1 月 11 日。CSS2 添加了对媒介（打印机和听觉设备）和可下载字体的支持。

- CSS3 —— CSS3 计划将 CSS 划分为更小的模块。

## 如何使用 CSS

---

### 4. 第一种内联样式

通过HTML元素的style属性来设置CSS样式，语法如下：

```
1 | style="css属性:css属性值;"
```

示例代码：

```
1 | <!DOCTYPE html>
2 | <html>
3 |   <head>
4 |     <title>01_第一种使用方式.html</title>
5 |   </head>
6 |   <body>
7 |     <!-- style="css属性:css属性值;" -->
8 |     <div style="color:red;" >atguigu</div>
9 |   </body>
10 | </html>
```

### 5. 第二种内联样式

通过HTML页面的style元素来设置CSS样式，语法如下：

```
1 | <style type="text/css">
2 |   选择器 {
3 |     属性名 : 属性值;
4 |   }
5 | </style>
```

示例代码：

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>02_第二种使用方式.html</title>
5     <meta http-equiv="content-type" content="text/html; charset=UTF-8">
6     <!--
7       style标签:封装样式内容
8       * type:指定使用样式,值为"text/css"
9       * 设置CSS语法:
10         选择器 {
11           属性名 : 属性值;
12         }
13     -->
14     <style type="text/css">
15       div {
16         color : red;
17       }
18     </style>
19   </head>
20   <body>
21     <div>atguigu</div>
22   </body>
23 </html>

```

## 6. 外联样式

通过HTML页面的link元素来引入外部CSS样式，语法如下：

```

1 | <link href="css文件路径" rel="stylesheet" type="text/css" />

```

示例代码：

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>04_第四种使用方式.html</title>
5     <meta http-equiv="content-type" content="text/html; charset=UTF-8">
6     <!--
7       link标签:
8         * href:引入外部css文件路径
9         * rel:设置引入文件为样式文件
10        * type:指定使用样式,值为"text/css"
11      -->
12     <link href="div.css" rel="stylesheet" type="text/css" />
13   </head>
14   <body>
15     <div>atguigu</div>
16   </body>
17 </html>

```

## CSS 语法

### 7. 语法格式

```
1 | 选择器 {属性名称 : 属性值; 属性名称 : 属性值;...}
```

语法特点:

- CSS声明总是以键值对(key\value)形式存在。
- CSS声明总是以分号(;)结束。
- 声明组以大括号({})括起来。
- 为了让CSS可读性更强, 每行只描述一个属性。

### 8. CSS 注释

注释是用来解释你的代码, 并且可以随意编辑它, 浏览器会忽略它。CSS注释以 `/*` 开始, 以 `*/` 结束。

```

1 <style type="text/css">
2   /* 这是一个注释 */
3   div {
4     /* 这是另一个注释 */
5     color : red;
6   }
7 </style>

```

## 9. 值得注意的问题

### 值的不同写法和单位

例如在设置字体颜色时，以下几种方式效果相同。

- 第一种方式

```
1 | #show1 {color : red;}
```

- 第二种方式

```
1 | #show2 {color : #ff0000;}
```

像上面这种使用十六进制设置颜色时，如果两两相同，可以写成如下格式：

```
1 | #show2 {color : #f00;}
```

- 第三种方式

```
1 | #show3 {color : rgb(255,0,0);}
```

上面的格式还可以写成如下格式：

```
1 | #show3 {color : rgb(100%,0%,0%);}
```

值得注意的是：当使用 RGB 百分比时，即使当值为 0 时也要写百分比符号。但是在其他的情况下就不需要这么做了。

### 值为若干单词，记得写引号

```
1 | div {font-family : Courier, "Courier New", monospace;}
```

### 多重声明

如果要定义不止一个声明，则需要用分号将每个声明分开。

### 空格和大小写

大多数样式表包含不止一条规则，而大多数规则包含不止一个声明。多重声明和空格的使用使得样式表更容易被编辑。

## 块级元素与内联元素

---

大多数 HTML 元素被定义为块级元素或内联元素。

### 10. 块级元素

块级元素在浏览器显示时，通常会以新行来开始（和结束）。

我们已经学习过的块级元素有：<h1>，<p>，<ul>，<table> 等。

值得注意的是：<p> 标签中不能包含任何块级元素。

### 11. <div> 元素

<div> 元素是块级元素，它可用于组合其他 HTML 元素的容器。

<div> 元素没有特定的含义。除此之外，由于它属于块级元素，浏览器会在其前后显示折行。

如果与 CSS 一同使用，<div> 元素可用于对大的内容块设置样式属性。

<div> 元素的另一个常见的用途是文档布局。

值得注意的是：<div> 取代了使用表格定义布局的老式方法。

### 12. 内联元素

内联元素在显示时通常不会以新行开始。

我们已经学习过的内联元素有：<td>，<a>，<img> 等。

值得注意的是：<a> 标签中不能包含 <a> 标签，但可以包含其他任何元素。

### 13. <span> 元素

<span> 元素是内联元素，可用作文本的容器。<span> 元素也没有特定的含义。

当与 CSS 一同使用时，<span> 元素可用于为部分文本设置样式属性。

值得注意的是：内联元素中不能包含块级元素。一般都是块级元素中包含内联元素。

## CSS 选择器

---

## 14. 元素选择器

元素选择器就是通过 HTML 页面的元素名称来设置 CSS 样式。具体语法如下：

```
1 | 元素名称 { 属性名 : 属性值; }
```

示例代码：

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

## 15. 类选择器

类选择器就是通过 HTML 元素的 class 属性值来设置 CSS 样式。具体语法如下：

```
1 | .class属性值 { 属性名 : 属性值; }
```

示例代码：

```
1 | .myDiv {color : red; }
```

## 16. ID选择器

ID选择器就是通过 HTML 元素的 id 属性值来设置 CSS 样式。具体语法如下：

```
1 | #id属性值 { 属性名 : 属性值; }
```

示例代码：

```
1 | #show1 {color : red; }
```

## 17. 属性选择器

属性选择器就是通过 HTML 元素的属性名称来设置 CSS 样式。具体语法如下：

```
1 | [属性名称] { 属性名 : 属性值; }
```

示例代码：

```
1 | [name] {color : red; }
```

选择器	描述
[attribute]	用于选取带有指定属性的元素。
[attribute=value]	用于选取带有指定属性和值的元素。
[attribute~=value]	用于选取属性值中包含指定词汇的元素。
[attribute^=value]	匹配属性值以指定值开头的每个元素。
[attribute\$=value]	匹配属性值以指定值结尾的每个元素。
[attribute*=value]	匹配属性值中包含指定值的每个元素。

## 18. 后代选择器

后代选择器就是设置 HTML 页面的指定元素的后代元素（中间使用空格）的 CSS 样式。具体语法如下：

```
1 | 祖先元素 后代元素 { 属性名 : 属性值; }
```

示例代码：

```
1 | div em {color : red; }
```

## 19. 子元素选择器

子选择器就是设置 HTML 页面的指定元素的子元素的 CSS 样式。具体语法如下：

```
1 | 祖先元素 > 子元素 { 属性名 : 属性值; }
```

示例代码：

```
1 | div > em {color : red; }
```

## 20. 相邻元素选择器

相邻元素选择器就是设置 HTML 页面的指定元素的下一个兄弟元素的 CSS 样式。具体语法如下：



```
1 | 指定元素 + 兄弟元素 { 属性名 : 属性值; }
```

示例代码:

```
1 | div + div {color : red; }
```

## 21. 其他内容

### 选择器分组

选择器分组就是将不同选择器相同声明的内容“压缩”在一起，得到更简洁的样式表。

```
1 | /* no grouping */
2 | h1 {color:blue;}
3 | h2 {color:blue;}
4 | h3 {color:blue;}
5 | h4 {color:blue;}
6 | h5 {color:blue;}
7 | h6 {color:blue;}
8 |
9 | /* grouping */
10 | h1, h2, h3, h4, h5, h6 {color:blue;}
```

### 通配符选择器

CSS2 引入了一种新的简单选择器 - 通配选择器（universal selector），显示为一个星号（\*）。该选择器可以与任何元素匹配，就像是一个通配符。

```
1 | * {color:red;}
```

值得注意的是: 通配符选择器的性能并不好。

## 伪类

CSS 伪类用于向某些选择器添加特殊的效果。

属性	描述
:active	向被激活的元素添加样式。
:focus	向拥有键盘输入焦点的元素添加样式。
:hover	当鼠标悬浮在元素上方时，向元素添加样式。
:link	向未被访问的链接添加样式。
:visited	向已被访问的链接添加样式。
:first-child	向元素的第一个子元素添加样式。

## 22. 锚伪类

```
1 | a:link {color: #FF0000}    /* 未访问的链接 */
2 | a:visited {color: #00FF00} /* 已访问的链接 */
3 | a:hover {color: #FF00FF}   /* 鼠标移动到链接上 */
4 | a:active {color: #0000FF}  /* 选定的链接 */
```

## 23. :first-child伪类

```
1 | p:first-child {font-weight: bold;}
```

# 选择器的优先级

在实际开发中，经常会为同一个标签设置多个样式。使用不同方式的选择器，由于其优先级不同，导致最终的效果也会不同。

选择器的优先级，具体如下：

- 内联样式的优先级为 1000
- ID 选择器的优先级为 100
- 类选择器的优先级为 10
- 元素选择器的优先级为 1
- 通配符选择器的优先级为 0
- 继承的样式没有任何优先级

值得注意的是：

- 当一个样式中包含多个选择器时，需要将所有的选择器的优先级进行相加。
- 优先级高的选择器优先显示，选择器的计算不会超过其最大的数量级。

- 如果两个选择器的优先级相同，则哪个选择器最后定义才有效。
- 如果某个选择器使用了 `!important`，则改选择器的优先级最高。

---

## 版权说明

本笔记的内容免费开源，任何人都可以免费学习、分享，甚至可以进行修改。但需要注明作者及来源，并且不能用于商业。

本笔记采用[知识共享署名-非商业性使用-禁止演绎 4.0 国际许可协议](#)进行许可。

---

## 联系方式



可以问职业问题。  
不要问技术问题。



可以问职业问题。  
可以问技术问题。



可以问职业问题。  
可以问技术问题。

