

CSS 语法规则

一、CSS 简介

CSS(Cascading Style Sheet)层叠样式表，它是用来美化页面的一种语言，之前提到的 W3C 规范中的“样式”。

作用：

1、美化界面。

2、页面布局。

基本语法：

选择器{

样式规则

}

样式规则：

属性名 1：属性值 1;

属性名 2：属性值 2;

属性名 3：属性值 3;

...

选择器:是用来选择标签的，选出来以后给标签加样式。

从 Hello World 说起，

```
<p style="background-color:#FFF;color:#000;font-size:30px;">Hello World!</p>
```

这样写可以实现将原本的白底黑字 Hello World! 改成样式为 黑底白字的效果。但是这样写有一定的缺点：

缺点 1：内容、样式、行为三者未分离，代码糅杂，不符合 W3C 规范，难以维护。

缺点 2：代码冗余，不可复用。

二、如何引入 CSS?

1) 行内样式

即上述的一般形式，由于其存在的缺点，介绍下一种样式。

2) 内嵌样式，给 p 标签加样式，加在 head 标签中

```
<style type="text/css">
```

```
p{
```

```
background-color:#FFF;
```

```
color:#000;
```

```
font-size:30px;
```

```
}
```

```
</style>
```

当本页面的 p 标签都需要相同的样式时，这样写大大降低了代码量，并让其他编程人员一眼分辨出来 CSS 样式。

3) 链接样式（链入式）

在 head 标签中加入

```
<link href="../css/my.css" type="text/css" rel="stylesheet"/>
```

将 CSS 文件和 html 文件分开来放，这样增加了代码的重用性，但是每个浏览器都要单独读取 css 文件中的样式，比较慢，如果追求速度，内嵌样式会更好。

4) 导入样式

```
<style type="text/css">
```

```
@import url("路径名");
```

```
</style>
```

四种样式的优先级：

如果一个页面中四种样式同时存在，这四种都会对标签起作用，但是重复样式的属性值，只有最后一个有效。如果内嵌样式和导入样式放在一起，则内嵌样式起作用。

如果内嵌样式和导入样式分成两个 style 定义，则在 CSS 样式中的最后一个 style 起作用。

css 中注释的格式，没有单行注释，所有的注释均是/*注释内容*/

```
img{                /* 标签选择器 */
    width: 200px;
    height: 300px;
}
.pic{               /*类选择器--可重复使用*/
    width: 200px;
    height: 400px;
}
#pic{              /* id 选择器-名字是唯一的 */
    width: 300px;
    height: 400px;
}
.window img{
}

/* 选择器优先级
    id-100 > class-10 > 标签-1
*/
    background-color: pink;          /* 背景颜色 */
    color: red;                      /* 文字颜色 */
    font-size: 20px;                 /* 文字大小-默认 16px */
    /* font-weight: bold;            字体加粗 */
    font-family: '楷体';             /* 字体样式 */
    text-indent: 40px;               /* 首行缩进 */
    line-height: 25px;              /* 行高-文字垂直居中 */
    letter-spacing: 2px;             /* 文字间距-默认是 0 */
}
.txt-left{
    text-align: left;
    text-decoration: underline;      /* 文字水平对齐 */
}
```

```

.txt-center{
    text-align: center;
}
.txt-right{
    text-align: right;
}
.link{
    color: gray;
    text-decoration: none;          /* 清除下划线 */
}
.link:hover{                      /* 鼠标滑过加载 */
    color: red;
    background-color: pink;
    text-decoration: underline;    /* 添加下划线 */
}
.list{
    overflow: hidden;              /* 溢出隐藏--清除浮动 */
}

.list li{
    list-style: none;
    float: left;
    /*让当前标签脱离文档流*/
}

```

定位

```

.box{
    width: 300px;
    height: 300px;
    border: 1px solid red;
    position: relative;
    /*调整位置的四个属性: top,left,right,bottom*/
    /* position: absolute;          绝对定位 */
    /* right: 20px; */
    /* bottom: 50px; */
    /*绝对定位-脱离文档流-基于屏幕视口定位-会随页面滚动而滚动*/
    /* position: relative;          相对定位 */
    /* top: 50px; */
    /*相对定位-不会脱离文档流-基于自身定位-浏览器会保留当前标签的初始位置(不会对
    其他标签布局造成影响)*/
    /* position: fixed;              固定定位 */
    /* right: 50px; */
    /* bottom: 20px; */
    /*固定定位-会脱离文档流-基于浏览器视口-不会随页面滚动而滚动*/
}

```

```
}  
.box1{  
    width: 100px;  
    height: 100px;  
    background-color: green;  
    position: absolute;  
    z-index: 1;  
    /*z-index 只能与定位搭配，用来调整 z 轴位置(垂直层级)*/  
}  
.box2{  
    width: 100px;  
    height: 100px;  
    background-color: blue;  
    position: absolute;  
    right: 20px;  
    bottom: 20px;  
    /**  
    *如果给当前标签设置绝对定位，浏览器解析时会从当前标签出发；  
    *逐级往上去寻找其祖先级，如果有任意一个定位，就会以当前定位标签为基准定位；  
    *如其祖先级有多个定位，则采取就近原则  
    */  
}
```