

# css基础

---

## 1、css基础概念

CSS (Cascading Style Sheet)即层叠样式表，是用来控制网页的外观的一门技术。

好处：

- 功能强大
- 将内容展示和样式控制分离
  - 降低耦合度 解耦
  - 让分工协作更容易
  - 提高开发

html是实现对网页结构的定义，通过css样式表可以实现如何显示 HTML 元素，也即是实现对html元素的装饰效果。

## 2、css引入的三种方式

在HTML中引入CSS共有3种方式：

(1) 外部样式表； (2) 内部样式表； (3) 内联样式表；

### • 2.1 内联样式表

即在标签中通过属性style的方式实现，只对当前元素中的内容起作用，但是这种方式结构和表现有着强耦合关系，不利于后期的维护，开发中不推荐使用。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>测试css</title>
  </head>
  <body>
    <span style="color:
green;">span</span>
  </body>
</html>
```

## • 2.2 内嵌样式表

即一般在head标签中通过**style标签**进行定义，复用性相对较好，但是也只能在当前页面生效。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>测试css</title>
    <style type="text/css">
      span {
        font-size: 30px;
        color: #2a9bff;
      }
    </style>
  </head>
  <body>
    span
  </body>
</html>
```

```
        </style>
    </head>
<body>
    <!-- 可以发现将所有span元素都被添加了指定的
    样式-->
    <span>我是一个span元素</span>
    <span>我是一个span元素</span>
    <span>我是一个span元素</span>
</body>
</html>
```

- **2.3 外部样式表（结构和表现分离）**

即先新建一个css文件，再在html中通过link元素引入，当样式需要被应用到很多页面的时候，外部样式表将是理想的选择。

在实际开发当中，为了提升网站的性能和维护性，一般都是使用外部样式表。 -

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>测试css</title>
    <link rel="stylesheet"
type="text/css" href="my.css"/>
  </head>
  <body>
    <span>我是span元素</span>
  </body>
</html>
```

```
span {
  font-size: 30px;
  color: #2a9bff;
}
```

### 3、不同样式表的优先级

- 当同一个元素被多个样式装饰时，如果有出现样式相同，则采取“**就近原则**”，即那个样式离此元素最近，则采用谁。

默认： 内联样式 > 内嵌样式 > 外部样式

## 4、css中的注释

`/* */` 实现多行注释

## 5、css样式的继承

继承是指我们设置上级(父级)的CSS样式，上级（父级）及以下的子级（下级）都具有此属性。

**注：一般只有文字文本具有继承特性，如文字大小、文字加粗、文字颜色、字体等。**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>测试css</title>
  </head>
  <body>
    <!--会发现给div加了样式，是会影响到span
    中的-->
    <div style="color: green;">
      <span>span</span>

    <!--
      假如设置父级文字样式后，其多个子
      级中，
      可能有些子级颜色不想与父级相同，
      这个时候只需对对应
```

子级设置需要颜色即可。

```
-->  
<span style="color:  
red;">span</span>  
</div>  
</body>  
</html>
```

## 6、CSS选择器

### 6.1、选择器简介

筛选具有相似特征的元素

我们使用外部样式表 和 内部样式表 时，如果想为指定的元素添样式，就需要依靠选择器来实现。

把某一个你想要的标签选中的方式就是所谓的“选择器”，也就是说选择器就是一种选择元素的方式。

### 6.2、元素的 id 与 class属性

- id属性

id属性被赋予了标识页面元素的唯一身份。如果一个页面出现了多个相同id属性取值，CSS选择器或者JavaScript就会因为无法分辨要控制的元素而最终报错。

- class属性

class，顾名思义，就是“类”，我们可以为同一个页面的相同元素或者不同元素设置相同的class，然后使得相同的class具有相同的CSS样式。

如果你要为两个元素或者两个以上元素定义相同的样式，建议使用class属性。

- **元素原则器**：选择具有相同名称的元素。

- 类选择器优先级高于选元素选择器

## 6.3、选择器基本语法

```
选择器：{ 属性1: 值1;  
          属性2: 值2;.....  
          }
```

{ } : 我们也叫声明，声明是由一组或多组 属性 和属性值 的键值对组成

## 6.4、元素选择器

元素选择器，就是“选中”相同的元素，然后对相同的元素设置同一个CSS样式。

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>元素选择器</title>  
    <style type="text/css">  
      span {
```

```
        font-size: 30px;
        color: #2a9bff;
    }
</style>
</head>
<body>
    <span>我是span元素</span>
    <p>我是p元素</p>
    <span>我是span元素</span>
</body>
</html>
```

## 6.5、id选择器

我们可以为元素设置一个id，然后针对这个id的元素进行CSS样式操作。

**注：** 在同一个页面中，不允许出现两个相同的id。

id名前面必须要加上前缀“#”，否则该选择器无法生效。

ID属性不要以数字开头，数字开头的ID在有些浏览器中可能不起作用

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>选择器</title>
    <style type="text/css">
```



```
        #span1 {
            font-size: 30px;
            color: #2a9bff;
        }
    </style>
</head>
<body>
    <span id="span1">我是span元素</span>
</body>
</html>
```

## 6.6、class选择器

class选择器，也就是“类选择器”。我们可以对“相同的元素”或者“不同的元素”设置一个class（类名），然后针对这个class的元素进行CSS样式操作。

**注：** class名前面必须要加上前缀“.”（英文点号），否则该选择器无法生效。

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>类选择器</title>
        <style type="text/css">
            .class1{
                font-size: 50px;
            }
            .class2{
                color: red;
            }
        </style>
    </head>
    <body>
        <div class="class1">
            <div class="class2">
                我是class选择器
            </div>
        </div>
    </body>
</html>
```

```
    }

    </style>
</head>
<body>
    <span class="class1 class2">我是span
元素</span>
    <p class="class1">我是p元素</p>
    <span>我是span元素</span>
</body>
</html>
```

## 6.7、派生选择器

也叫做上下文选择器，即通过依据元素在其位置的上下文关系来定义样式的选择器。

派生选择器允许你根据文档的上下文关系来确定某个标签的样式。通过合理地使用派生选择器，我们可以使HTML 代码变得更加整洁。

关系可以有父子关系、兄弟关系。

一个元素在另一个元素的内部，那他们是父子关系

一个元素和另一个元素处于同级，那他们是兄弟关系

- (1) 后代选择器

**语法： 父选择器 空格 子选择器**

**含义：** 表示选中某个元素下的 所有后代元素 （子元素、孙子辈元素等都要）

**注：** 父元素与子元素必须用空格隔开

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>后代选择器</title>
    <style type="text/css">
      li strong {
        font-style: italic;
        font-weight: normal;
      }

    </style>
  </head>
  <body>
    <p>
      <strong>
        我是粗体字，不是斜体字，因为我
        不在列表当中，所以这个规则对我不起作用
      </strong>
    </p>

    <ol>
      <li>
```

```
        <strong>我是斜体字。这是因为
strong 元素位于 li 元素内 </strong>
    </li>
    <li>
        我是正常的字体
    </li>
    <li>
        <span>
            <strong>我嵌套很多层是斜体字。这
            是因为 strong 元素位于 li 元素内 </strong>
        </span>
    </li>
</ol>
</body>
</html>
```

- **(2) 子元素选择器**

**语法： 父选择器 > 子选择器**

说明： 筛选子选择器下的父选择器的

含义： 选中某个元素下的直系子元素， 再往下级的子元素不算

与后代选择器相比， 子元素选择器范围更小，对后代选择器来说， 子元素、孙子辈元素都是后代，而子元素选择器只管直系后代。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>子元素选择器</title>
    <style type="text/css">
      div > span {
        color: #2a9bff;
      }
    </style>
  </head>
  <body>
    <div>
      <span>我是直系子元素，我会变色
    </span>

      <p>
        <span>我不是直系子元素，我不会
        变色</span>
      </p>
    </div>
  </body>
</html>
```

## 6.8: 伪类选择器:

- 选择一些元素具有的状态
  - 语法: 元素: 状态{
    - 如:

- 状态：
  - link: 初始化状态
  - visited: 被访问过的状态
  - active: 正在访问状态
  - hover: 鼠标悬浮状态

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>伪类选择器</title>
    <style>
      /*让超链接点击之前是红色*/
      a:link{
        color:red;
      }

      /*让超链接点击之后是橙色*/
      a:visited{
        color:orange;
      }

      /*鼠标悬停，放到标签上的时候*/
      a:hover{
        color:green;
      }

      /*鼠标点击链接，但是不松手的时候*/
      a:active{
```

```
        color:black;
    }
</style>
</head>
<body>
    <a href="#">国科科技</a>
</body>
</html>
```

## 7、常见文字样式

属性	说明
font-family	字体类型, 如“微软雅黑”、“宋体”、“Times New Roman”等
font-size	字体大小, 值为像素值
font-weight	字体粗细, 一般仅用到bold（加粗）、normal（正常）这两个属性值
font-style	字体斜体, 值为 normal（正常）、italic（斜体）
color	字体颜色

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>文字样式</title>
```

```
<style>
    span{
        font-family:微软雅黑;
        font-size: 30px;
        font-weight:bold;
        font-style:italic;
        color: #2a9bff;
    }
</style>
</head>
<body>
    <span>我是span元素</span>
</body>
</html>
```

## 8、文本样式



属性	描述
<a href="#">text-decoration</a> 属性	下划线 (underline)、删除线 (line-through)、顶划线 (overline)
<a href="#">text-transform</a> 属性	文本大小写，针对英文。大写 (uppercase)、小写 (lowercase)、首字母大写 (capitalize)
<a href="#">text-indent</a> 属性	段落首行缩进，值为像素值
<a href="#">text-align</a> 属性	文本水平对齐方式，左对齐 (left，默认值)、居中 (center在·)、右对齐 (right)
<a href="#">line-height</a> 属性	行高，一行文本的高度， 值为像素值
<a href="#">letter-spacing</a>	字距
<a href="#">word-spacing</a>	词距

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>文本样式测试</title>
```

```
<style>
    /*添加下划线注释*/
    #span1{text-
decoration:underline;}

    /*首字母大写*/
    #span2{text-
transform:capitalize;}

    /*除了使用&nbspsp实现缩进，    也可以通
过缩进属性设置缩进*/
    #span2{text-indent:28px;}

    /*
        实现将文本居中显示
        此时我们没有设置div的宽度，居中
        会撑大整个div，它应该是整个浏览器窗口居中的
    */
    #div1{text-align: center;}

    /*
        实现将文本居中显示
        此时我们设置了div的宽度，它应该
        相对div容器宽度居中的
    */
    #div2{
        text-align: center;
        width: 400px;
        border: blue 1px solid;
    }
```

```
        /*
           不会出现任何效果，text-align要在块级元素中才能生效
        */
        #span6{text-align: center;}

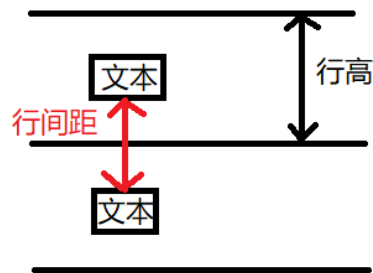
    </style>
</head>
<body>
    <span id="span1">我是span元素1</span>
</br>
    <span id="span2">hello world</span>
</br>
    <span id="span3">我是span元素3</span>
</br>

    <div id="div1">
        <span>我是span元素4</span>
    </div>
    <div id="div2">
        <span>我是span元素5</span>
    </div>
    <span id="span6">我是span元素6</span>
</body>
</html>
```

**关于行高与行间距问题：**

**行高：** 一行的高度

**行间距：** 两行文本之间的距离



如上图所示，在html中的一行里，文本并不会将整行给占满，只会占据行中间的一部分，有一个文本到另一个文本之间的距离就是行间距，所以我们可以得到行间距公式：

$$\text{行高} - \text{文本高度} = \text{行间距}$$

在html中，没有专门的属性来操作行间距，不过根据上面的公式，我们可以实现对行间距的控制。

```
<!DOCTYPE html>
<html>
<head>
```

```
<title>line-height属性</title>
<style type="text/css">
    #p1{line-height:14px;}
    #p2{line-height:18px;}
    #p3{line-height:20px;}
</style>
</head>
<body>
    <p id="p1">水陆草木之花，可爱者甚蕃。晋陶渊明
独爱菊。自李唐来，世人甚爱牡丹。予独爱莲之出淤泥而不
染，濯清涟而不妖，中通外直，不蔓不枝，香远益清，亭亭
净植，可远观而不可亵玩焉。</p><hr/>
    <p id="p2">水陆草木之花，可爱者甚蕃。晋陶渊明
独爱菊。自李唐来，世人甚爱牡丹。予独爱莲之出淤泥而不
染，濯清涟而不妖，中通外直，不蔓不枝，香远益清，亭亭
净植，可远观而不可亵玩焉。</p><hr/>
    <p id="p3">水陆草木之花，可爱者甚蕃。晋陶渊明
独爱菊。自李唐来，世人甚爱牡丹。予独爱莲之出淤泥而不
染，濯清涟而不妖，中通外直，不蔓不枝，香远益清，亭亭
净植，可远观而不可亵玩焉。</p>
</body>
</html>
```

## 9、边框样式

- ①、简介

在网页中，边框随处可见，任何 **块元素和行内元素** 都可以设置边框属性。例如，div元素可以设置边框，img元素也可以设置边框，table元素也可以设置边框，span元素同样也可以设置边框等等。

- **②、边框属性**

我们可以思考一下，对于一个方框，你能设置它的哪些样式呢？

简单思考一下，其实我们可以得到如下结论，可以设置边框的如下样式：

- (1)、边框的宽度**
- (2)、边框的外观（实线、虚线）**
- (3)、边框的颜色**

属性	描述
border-width	边框的宽度，值为像素值
border-style	边框的外观
border-color	边框的颜色，值为颜色值

注意，设置一个元素的边框必须要同时设置border-width、border-style、border-color这三个属性，这个元素的边框才能在浏览器显示出来。

```
<!DOCTYPE html>
```

```
<html>
  <head>
    <meta charset="utf-8">
    <title>测试边框属性</title>

    <style type="text/css">

      #div1{
        border-color:
black;
        border-width:
1px;
        border-
style:solid;
        height: 400px;
        width: 400px;
      }

      /* 边框样式简化写法: */
      #div2{
        border: red 1px
solid;
        height: 300px;
        width: 300px;
      }

    </style>
  </head>
  <body>
    <div id="div1"> </div>
```

```
        <div id="div2"> </div>
    </body>
</html>
```

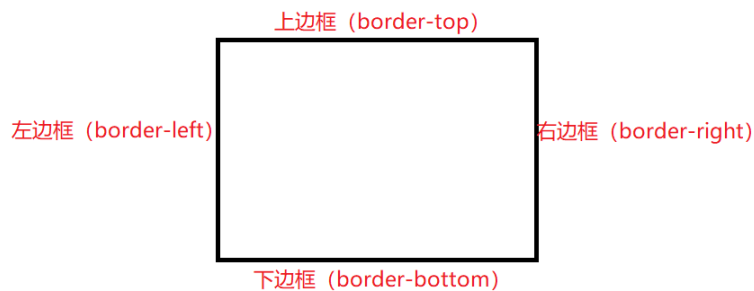
- ③、border-style属性的值

属性值	说明
none	无样式
hidden	与“none”相同。bug应用于表除外。对于表，hidden用于解决边框冲突
solid	实线
dashed	虚线
dotted	点线
double	双线，双线的宽度等于border-width值

- ④、边框的局部样式

在上述的案例中，我们设置样式时，都是同时对边框的四边生效，那么如果我们想对某条边设置样式呢？





如上图所示，在CSS中，我们可以分别针对上下左右四条边框设置单独的样式。

比如，对上边框设置样式：

```
border-top-width:5px;  
border-top-style:solid;  
border-top-color:red;
```

其他几边的边框也是同理操作。

## 10、背景样式

在CSS中，**背景样式主要包括背景颜色和背景图像**。现在几乎全部都是使用CSS的background属性来控制元素的背景颜色和背景图像。

- (1)、background-color设置背景颜色

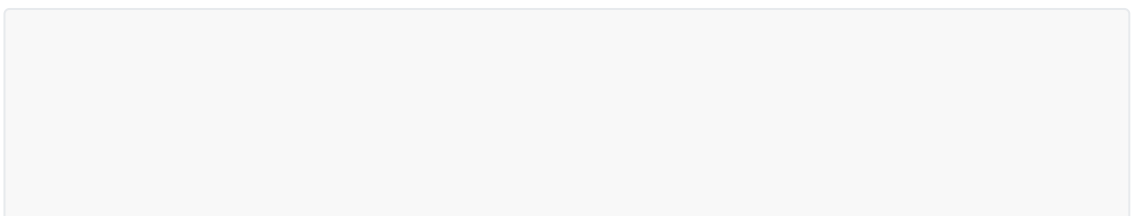
```
background-color: gray;
```

- (2)、color 与 background-color

color是专门用来设置字体的颜色， background-color 是设置元素的背景颜色

```
<!DOCTYPE html>
<html>
<head>
  <title>color和background-color区别
</title>
  <style type="text/css">
    #div1
    {
      width:290px;
      color:white;
      background-color:red;
    }
  </style>
</head>
<body>
  <div id="div1">
    元素文本颜色color值为white<br/>
    元素背景颜色background-color值为red
  </div>
</body>
</html>
```

- **(3) 、 background-image 指定背景图片**



```

<!DOCTYPE html>
<html>
<head>
    <title>color和background-color区别
</title>
    <style type="text/css">

        /*
            1、如果需要设置一个背景图像，必须为这个
            属性设置一个 URL 值：
            2、给某个元素设置背景图像，元素要有一定的
            宽度和高度，背景图片才会显示出来；
        */

        #div1{
            width:290px;
            height:220px;
            background-image:
url(eg_bg_04.gif);
        }
    </style>
</head>
<body>
    <div id="div1"> 给当前区域添加一个背景图
</div>
</body>
</html>

```

- (4)、背景图片重复问题 background-repeat

有些时候，图片本身比较小，元素宽高较大，图片不能填满整个元素时，我们可能会需要图片自己复制并铺满整个元素。此时通过background-repeat属性来实现：

属性值	描述
no-repeat	表示不平铺
repeat	默认值，表示在水平方向（x轴）和垂直方向（y轴）同时平铺
repeat-x	表示在水平方向（x轴）平铺
repeat-y	表示在垂直方向（y轴）平铺

```
<!DOCTYPE html>
<html>
  <head>
    <title>background-repeat属性</title>
    <style type="text/css">
      /*设置div元素的共同样式*/
      div{
        width:200px;
        height:100px;
```

```
        background-
image:url("../img/flower.jpg");
        text-align:center;
        border:1px solid gray;
    }

    /*设置3个div元素的个别样式*/
    #div2{background-repeat:repeat-x;}
    #div3{background-repeat:repeat-y;}
    #div4{background-repeat:no-repeat;}

    hr{border-color:red;}
</style>
</head>

<body>
    <div id="div1">
        默认背景图平铺满整个元素
    </div>
    <hr/>
    <div id="div2">
        对背景图片在x方向上进行平铺
    </div>
    <hr/>
    <div id="div3">
        对背景图片在y方向上进行平铺
    </div>
    <hr/>
    <div id="div4">
        对背景图片不平铺
    </div>
```

```
</body>
```

```
</html>
```