

## CSS 基础使用

- 一、CSS介绍
- 二、CSS使用方式
  - 1. 行内样式/内联样式
  - 3. 外链样式表
- 三、样式表特征
  - 1. 层叠性
  - 2. 继承性
  - 3. 样式表的优先级-就近原则
- 四、CSS 选择器
  - 1. 作用
  - 2. 分类：
    - 1. 标签选择器-全局
    - 2. id选择器-个别用
    - 3. class选择器/类选择器
    - 4. 群组选择器
    - 5. 后代选择器
    - 6. 子代选择器
    - 7. 伪类选择器
  - 3. 选择器的优先级
- 五、标签分类及嵌套
  - 1. 块元素
  - 2. 行内元素
  - 3. 行内块元素
  - 4. 嵌套原则

# CSS 基础使用

## 一、CSS介绍

CSS全称为：Cascading Style Sheets，意为层叠样式表，与HTML相辅相成，实现网页的排版布局与样式美化

## 二、CSS使用方式

### 1. 行内样式/内联样式

借助于style标签属性，为当前的元素添加样式声明

```
<标签名 style="样式声明">
```

CSS样式声明：由CSS属性和值组成  
例：

```
style="属性:值;属性:值;" 同名属性只运行一次
```

常用CSS属性：

- 设置文本颜色 color:red;

- 设置背景颜色 background-color:green;
- 设置字体大小 font-size:32px;

```
<!-- 行内样式 -->
<div style="background-color:greenyellow;color: red;font-size: 36px;" >这是
测试样式的文本</div>
```

### ### 2. 内嵌样式

借助于style标签，在HTML文档中嵌入CSS样式代码，可以实现CSS样式与HTML标签之间的分离。同时需借助于CSS选择器到HTML 中匹配元素并应用样式

示例：

```
<!-- 内联样式 -->
<style>
  div{
    /* 宽度200px */
    width:200px;
    /* 高度200px */
    height:200px;
    /* 背景色：粉色 */
    background-color: pink;
    /* 字体颜色：红色 */
    color:red;
  }
</style>
```

选择器 ：通过标签名或者某些属性值到页面中选取相应的元素，为其应用样式

示例：

```
```css
/*标签选择器 ：根据标签名匹配所有的该元素*/
p{
  color:red;
}
```

## 3. 外链样式表

- 创建外部样式表文件 后缀使用.css
- 在HTML文件中使用标签引入外部样式表

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

- 样式表文件中借助选择器匹配元素应用样式

## 三、 样式表特征

### 1. 层叠性

多组CSS样式共同作用于一个元素

### 2. 继承性

后代元素可以继承祖先元素中的某些样式

例：大部分的文本属性都可以被继承

### 3. 样式表的优先级-就近原则

优先级用来解决样式冲突问题。同一个元素的同一个样式(例如文本色)，在不同地方多次进行设置，最终选用哪一种样式？此时哪一种样式表的优先级高选用哪一种。

- 行内样式的优先级最高
- 文档内嵌与外链样式表,优先级一致,看代码书写顺序,后来者居上
- 浏览器默认样式和继承样式优先级较低

## 四、CSS 选择器

### 1. 作用

匹配文档中的某些元素为其应用样式

### 2. 分类：

#### 1. 标签选择器-全局

根据标签名匹配文档中所有该元素

语法：

```
标签名{
    属性:值;
}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <!-- 内联样式 -->
    <!-- <style>
        div{
            /* 宽度200px */
            width:200px;
            /* 高度200px */
            height:200px;
            /* 背景色: 粉色 */
            background-color: pink;
            /* 字体颜色: 红色 */
            color:red;
        }
    </style> -->
    <link rel="stylesheet" href="index.css">
</head>
<body>
    <div>这是测试样式的文本</div>
    <div>这是测试样式的文本</div>
</body>
</html>
```

## 2. id选择器-个别用

根据元素的 id 属性值匹配文档中唯一的元素，id具有唯一性，不能重复使用  
语法：

```
#id属性值{  
  
}  
  
<div id="blue_div">蓝色背景的div</div>  
  
#blue_div{  
    background-color: blue;  
}
```

注意：

id属性值自定义,可以由数字，字母，下划线，- 组成，不能以数字开头;  
尽量见名知意，多个单词组成时，可以使用连接符，下划线，小驼峰表示

## 3. class选择器/类选择器

根据元素的class属性值匹配相应的元素,class属性值可以重复使用,实现样式的复用  
语法：

```
.class属性值 {  
  
}
```

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <meta http-equiv="X-UA-Compatible" content="ie=edge">  
    <title>Document</title>  
    <style>  
        .yellow{  
            color:yellow;  
        }  
        .center{  
            text-align:center;  
        }  
    </style>  
  
</head>  
<body>  
    <p class="yellow center">p标签中的字体</p>  
    <div class="yellow center">div标签中的字体</div>  
</body>  
</html>
```

特殊用法：

## 1. 类选择器与其他选择器结合使用

注意标签与类选择器结合时,标签在前,类选择器在后

例: a.c1{ }

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    .yellow{
      color:yellow;
    }
    .center{
      text-align:center;
    }
    /* 所有p中带有yellow类的元素 */
    p.yellow{
      font-size:40px
    }
    /* 查找同时带有yellow和center两种类的元素 */
    .yellow.center{
      background-color:red;
    }
  </style>

</head>
<body>
  <p class="yellow center">p标签中的字体</p>
  <div class="yellow center">div标签中的字体</div>
</body>
</html>
```

## 2. class属性值可以写多个,共同应用类选择器的样式

例:

.c1{ }

.c2{ }

## 4. 群组选择器

为一组元素统一设置样式

语法:

```
selector1,selector2,selector3{
}
```

注: 群组可以是id,可以是类,用逗号分隔。

例如: #p1,.c1class,body

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>群组选择器</title>
<style>
    /* *表示通用选择器，匹配页面上所有的元素 */
    *{
        margin:0;
        padding:0;
    }
    body,p,ul{
        /* 取消边距 */
        margin:0;
        padding:0;
    }

</style>
</head>
<body>
    <p></p>
    <div>这是div中的文字</div>
    <ul>
        <li>1</li>
        <li>2</li>
        <li>3</li>
    </ul>
</body>
</html>

```

## 5. 后代选择器

匹配满足选择器的所有后代元素(包含直接子元素和间接子元素)

语法：

```

selector1 selector2{
}

```

匹配selector1中所有满足selector2的后代元素

## 6. 子代选择器

匹配满足选择器的所有直接子元素

语法：

```

selector1>selector2{
}

```

## 7. 伪类选择器

为元素的不同状态分别设置样式,必须与基础选择器(标签,id,class)结合使用

分类：

:link 超链接访问前的状态  
:visited 超链接访问后的状态  
:hover 鼠标滑过时的状态  
:active 鼠标点按不抬起时的状态(激活)  
:focus 焦点状态(文本框被编辑时就称为获取焦点)

```
#d2:hover{
    color:red;
    /* transition: 发生变化的样式，完成变化的时间 */
    /* linear表示匀速，从开始到结束速度一致。 */
    transition:all 0.5s;
    /* 鼠标划过时变成小手 */
    cursor:pointer;
}
```

使用：

```
a:link{
}
a:visited{
}
.cl:hover{ }
```

```
<style>
  a{
    /* 默认text-decoration: underline */
    /* 取消超链接的下划线 */
    text-decoration: none
  }
  a:link{
    background-color: greenyellow;
  }
  a:visited{
    background-color: greenyellow
  }
  a:hover{
    text-decoration: underline
  }
  a:active{
    /* 鼠标按住不放，文本颜色变成红色。 */
    color:red;
  }
</style>
```

```
<style>
  /* 获取焦点 */
  input:focus{
    background-color: pink;
  }
</style>
```

注意：

- 1. 超链接如果需要为四种状态分别设置样式,必须按照以下顺序书写

```
:link
:visited
:hover
:active
```

- 2. 超链接常用设置：

```
a{
    /*统一设置超链接默认样式(不分状态)*/
}
a:hover{
    /*鼠标滑过时改样式*/
}
```

### 3. 选择器的优先级

使用选择器为元素设置样式,发生样式冲突时,主要看选择器的权重,权重越大,优先级越高

选择器	权重
标签选择器	1
(伪)类选择器	10
id选择器	100
行内样式	1000

复杂选择器(后代,子代,伪类)最终的权重为各个选择器权重值之和  
群组选择器权重以每个选择器单独的权重为准，不进行相加计算  
例：

```
/*群组选择器之间互相独立，不影响优先级*/
body,h1,p{ /*标签选择器权重为 1 */
    color:red;
}
.c1 a{ /*当前组合选择器权重为 10+1 */
    color:green;
}
#d1>.c2{ /*当前组合选择器权重为 100+10 */
    color:blue;
}
```

## 五、 标签分类及嵌套

### 1. 块元素

独占一行,不与元素共行;可以手动设置宽高,默认宽度与与父元素保持一致  
例：body div h1~h6 p ul ol li form, table(默认尺寸由内容决定)



## 2. 行内元素

可以与其他元素共行显示;不能手动设置宽高,尺寸由内容决定

例: `span label b strong i s u sub sup a`

## 3. 行内块元素

可以与其他元素共行显示,又能手动调整宽高

例: `img input button` (表单控件)

## 4. 嵌套原则

1. 块元素中可以嵌套任意类型的元素  
p元素除外,段落标签只能嵌套行内元素,不能嵌套块元素
2. 行内元素中最好只嵌套行内或行内块元素

# `display`可以控制元素的隐藏和显示。

```
span{
  background-color: red;
  /* 将span以块元素的方式显示 */
  /* display是设置行内的显示方式 */
  display: block;
  /* 将span元素以行内块元素的方式显示 */
  display: inline-block;
  /* display取值none, 不显示元素。 */
  display: none;
  height: 200px;
}
</style>
```

```
<body>
  <!-- p元素如果嵌套其他块元素 -->
  <!-- 在浏览器中渲染到其他块元素后就会停止 -->
  <!-- p元素就会变成两段, 由浏览器自动补全。 -->
  <p>
    这是一个段落标签
    <div>
      这是p中的div。
    </div>
  </p>
</body>
```

