

## CSS 基础使用

- 一、CSS介绍
- 二、CSS使用方式
  - 1. 行内样式/内联样式
  - 2. 内嵌样式
  - 3. 外部样式表
- 三、样式表特征
  - 1. 层叠性
  - 2. 继承性
  - 3. 样式表的优先级
- 四、CSS 选择器
  - 1. 作用
  - 2. 分类：
    - 1. 标签选择器
    - 2. id选择器
    - 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

行内样式 用户自定义样式（内嵌样式） 浏览器自定义样式 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;"
```

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;

```
css.html
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <!--引入外部样式表-->
  <link rel="stylesheet" href="index.css">
  <style>
    /*内嵌样式,借助style标签书写样式声明,
    实现样式与结构的分离*/
    /*使用选择器匹配文档元素,为其应用样式*/
    /*标签选择器:根据标签名匹配元素应用样式*/
    h1{
      background:green;
    }
    span{
      color:orange;
    }
    span{
      color:pink;
    }
  </style>

</head>
<body>
  <!--行内样式,借助style标签属性设置样式规则-->
  <!--CSS样式声明 属性名:属性值;-->
  <h1 style="font-size:16px;color:red;">老祁营救小泽失败</h1>
  <h1 style="font-size:16px;color:red;">隔壁老王在扒图</h1>
  <h2>
    测试<span>老王</span>
  </h2>
  <span>老王2</span>
  <div style="color:green;">
    <p style="color:blue;">
      <span style="color:gray;">子元素可以继承父元素或祖先元素的文本样式</span>
    </p>
  </div>
</body>
</html>
```

## 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：

```
<style>
    选择器{
        属性:值;
        属性:值;
    }
</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：

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

## 3. 外部样式表

- 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

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
- 文档内嵌与外部样式表 优先级一致,看代码 (css) 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

根据标签名匹配文档中所有该元素 语法：

```
标签名{  
    属性:值;  
}
```

#### 2. id选择器

根据元素的 id 属性值匹配文档中惟一的元素, id具有唯一性, 不能重复使用 语法：

```
#id属性值{  
  
}
```

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 {  
  
}
```

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{}
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{  
}
```

#### 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)
```

使用：

```
a:link{  
}  
a:visited{  
}  
.c1:hover{ }
```

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

```
css.html
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```

<meta name="viewport"
      content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
<!--引入外部样式表-->
<link rel="stylesheet" href="index.css">
<style>
    /*内嵌样式,借助style标签书写样式声明,
    实现样式与结构的分离*/
    /*使用选择器匹配文档元素,为其应用样式*/
    /*标签选择器:根据标签名匹配元素应用样式*/
    h1{
        background:green;
    }
    span{
        color:orange;
    }
    span{
        color:pink;
    }
</style>

</head>
<body>
    <!--行内样式,借助style标签属性设置样式规则-->
    <!--CSS样式声明 属性名:属性值;-->
    <h1 style="font-size:16px;color:red;">老祁营救小泽失败</h1>
    <h1 style="font-size:16px;color:red;">隔壁老王在扒图</h1>
    <h2>
        测试<span>老王</span>
    </h2>
    <span>老王2</span>
    <div style="color:green;">
        <p style="color:blue;">
            <span style="color:gray;">子元素可以继承父元素或祖先元素的文本样式</span>
        </p>
    </div>
</body>
</html>

```

```

```index.css
h1{
    width:200px;
    height:200px;
}
span{
    color:red;
}

```

```

` selector.html
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <style>
        /*1. 根据标签名匹配所有相应元素应用样式*/
        h1{
            color:green;
        }
        /*2. 根据元素的id属性值匹配(id选择器)*/
        #d1{
            /*匹配id为"d1"的元素*/
            font-size:40px;
        }
        #d2{
            font-size:42px;
        }
        /*3. 类选择器: 根据元素的class属性值匹配到相应元素*/
        .c1{
            background: orange;
        }
        .c2{
            width:200px;
        }
        /*组合选择器*/
        h2.c1{
            /*匹配类名为c1的h2元素*/
            height:200px;
        }
        /*4. 群组选择器: 将选择器组合起来, 统一为元素设置样式*/
        h2,h3,h4{
            font-size:20px;
        }
        #d1, .c1{}
        /*5. 后代选择器: 匹配所有满足选择器的后代元素*/
        div span{
            color:red;
        }
        /*6. 子代选择器: 匹配所有满足选择器的直接后代*/
        div>span{
            background:cyan;
        }
        /*7. 伪类选择器: 为元素的不同状态分别设置样式,
        需要和基础选择器结合使用*/
        /*
        超链接伪类:
            :link(超链接访问前的状态)

```



```

        :visited(超链接访问后的状态)
    动态伪类:
        :hover(鼠标悬停时的状态)
        :active(激活状态, 鼠标点按时的状态)
        :focus(表单控件的焦点状态, 输入框的编辑状态)
    */
    /*
    a:link{
        color:black;
    }
    a:visited{
        color:gray;
    }
    a:hover{
        font-size:28px;
    }
    a:active{
        background:orange;
    }
    */
    /*
        1. 统一设置超链接四种状态下的样式
        2. 单独设置鼠标悬停时的样式
    */
    a{
        color:black;
        /*取消超链接默认下划线*/
        text-decoration:none;
    }
    a:hover{
        color:orange;
    }
    /*创建h5标签, 设置初始尺寸和背景颜色;
    鼠标悬停时修改背景颜色*/
    h5{
        width:200px;
        height:200px;
        background:pink;
    }
    h5:hover{
        background:green;
    }
    /*取消表单控件在焦点状态下的轮廓线*/
    input:focus{
        outline:none;
    }

</style>
</head>
<body>
    <input type="text">
    <h5></h5>
    <a href="01_form.html">表单应用</a>

```

```

<div>
  <span>div->span</span>
  <p>
    <span>div->p->span</span>
  </p>
</div>

<!--基础选择器:标签,id,类-->
<!--class属性可以取多个值,使用空格隔开-->
<h1 id="d2" class="c1 c2">老祁</h1>
<h1 id="d3" class="c1">小泽</h1>
<h1 id="d1">老王</h1>
<h2 class="c1 c2">MM</h2>
<h2>参考</h2>
<h3>测试</h3>
<h4>测试2</h4>
</body>
</html>

```

```

```selector_property.html
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    .c1{
      color:blue;
    }
    #d1{
      color:pink;
    }
    h1{
      color:red;
    }
    h1{
      color:green;
    }

    /*十六进制表示颜色,以#前缀,每两位为一组,
    代表一种三原色(r,g,b),每位字符取值范围0~9,a~f
    ff:(255)*/*

    #d1{/*100*/
      color:#997679;
    }
    div h1{/*1+1*/
      color:red;

```

```

    }
    div .c1{/*1+10*/
        color:green;
    }
    div #d1{/*1+100*/
        /*#000000*/
        color:#000;
    }

</style>
</head>
<body>
    <div>
        <h1 id="d1" class="c1">小泽Maria</h1>
    </div>
</body>
</html>

```

```

```tag_type.html
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <style>
        table, form, span, b, a, input{
            width:300px;
            height:300px;
            background:pink;
        }
    </style>
</head>
<body>
    <h1>标题</h1>
    <p>段落</p>
    <ul>
        <li>列表项</li>
    </ul>
    <table>
        <tr>
            <td>单元格</td>
        </tr>
    </table>
    <form action="">
        此处为表单
        <input type="text">
    </form>
    <div>容器标签</div>

```

```

<span>span</span>
<b>加粗</b>
<a href="">超链接</a>
<div>
  <div>
    <h1>
      <span></span>
      <button></button>
    </h1>
  </div>
</div>
<p>
  <h1>1. 参照效果图实现表单练习</h1>
  <h1>2. 实现下拉菜单的隐藏与显示
    (display:none(隐藏)/block(显示))</h1>
  <h1>3. 模拟横向导航栏( 淘宝),
    鼠标悬停时修改导航项文本色或背景色</h1>
</p>
</body>
</html>

```

\*\*文本属性可以被继承，但是超链接的文本颜色需要单独设置，表单控件的文本样式也需要单独设置。\*\*

•

## 01\_list\_work下拉菜单

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>document</title>
  <style>
    /*ul, ol是块元素*/
    ul, ol{
      /*取消默认的内边距*/
      padding:0;
      /*取消列表的项目符号*/
      list-style:none;
    }
    ul{
      width:200px;
      background:orange;
      /*设置水平居中*/
      text-align:center;
      /*设置显示形状,default表示箭头,pointer表示手指,text表示文本*/
      cursor:default;
    }
    ol{

```

```

        /*默认隐藏*/
        display:none;
    }
    .address:hover ol{
        /*设置显示*/
        display:block;
    }
    ol li:hover{
        background:pink;
    }
}

</style>
</head>
<body>
    <ul>
        <li class="address">
            北京
            <ol>
                <li>北京</li>
                <li>上海</li>
                <li>广州</li>
                <li>深圳</li>
            </ol>
        </li>
    </ul>
</body>
</html>

```

## 导航栏的设置

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>document</title>
    <style>
        #nav{
            width:500px;
            height:50px;
            background:orange;
            color:#fff;
            /*处理水平方向上由于换行导致的空隙
            子元素需要手动设置可见大小*/
            font-size:0px;
        }
        a{
            font-size:24px;
            color:#fff;
            text-decoration:none;
            /*转换元素类型,display:block块元素,inline行内元素
            inline-block表示行内块元素,none隐藏*/
            display:inline-block;
        }
    </style>

```

```
        width:100px;
        height:50px;
        text-align:center;
        /*一行文本在当前行中永远垂直居中,可以设置行高与元素高度
        一致,实现一行文本的垂直居中效果*/
        line-height:50px;

    }
    a:hover{
        background:green
    }
</style>
</head>
<body>
    <!-- 导航栏整体尺寸为500*50,包含五个导航项-->
    <div id="nav">
        <!-- 文本属性可以被继承,但是超链接的文本颜色
        需要单独设置,表单控件的文本样式也需要单独设置-->
        <!--<span>测试</span>-->
        <!--<input type="text">-->
        <!-- 代码中的换行会被解析为一个空白字符,影响布局-->
        <a href="">天猫</a>
        <a href="">天猫</a>
        <a href="">天猫</a>
        <a href="">天猫</a>
        <a href="">天猫</a>
    </div>
    <div id="main"></div>
</body>
</html>
```