

[TOC]

# 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;"
```

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;

### 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

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

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

[TOC]

## 一、尺寸与颜色单位

### 1. 尺寸单位

- px 像素单位
- % 百分比，参照父元素对应属性的值进行计算
- em 字体尺寸单位，参照父元素的字体大小计算，1em=16px
- rem字体尺寸单位,参照根元素的字体大小计算，1rem=16px

## 2. 颜色单位

- 英文单词: red, green, blue
- rgb(r,g,b) 使用三原色表示，每种颜色取值0~255
- rgba(r,g,b,alpha) 三原色每种取值0~255，alpha取值0（透明）~1（不透明）
- 十六进制表示：以#为前缀，分为长十六进制和短十六进制。
  - 长十六进制：每两位为一组，代表一种三原色；每位的取值范围0~9，a~f 例：red rgb(255,0,0) #ff0000
  - 短十六进制：由3位组成，每一位代表一种三原色，浏览器会自动对每一位进行重复扩充，仍然按照长十六进制解析 例：#000 #fff #f00

## 2. CSS 盒模型

---

### 1. 内容尺寸

- 一般情况下，为元素设置width/height，指定的是内容框的大小
- 内容溢出：内容超出元素的尺寸范围，称为溢出。默认情况下溢出部分仍然可见，可以使用overflow调整溢出部分的显示,取值如下：

取值	作用
visible	默认值，溢出部分可见
hidden	溢出部分隐藏
scroll	强制在水平和垂直方向添加滚动条
auto	自动在溢出方向添加可用滚动条

## 2. 边框

### 1. 边框实现

语法：

```
border:width style color;
```

边框样式为必填项，分为：

样式取值	含义
solid	实线边框

样式取值	含义
dotted	点线边框
dashed	虚线边框
double	双线边框

## 2. 单边框设置

分别设置某一方向的边框，取值：width style color;

属性	作用
border-top	设置上边框
border-bottom	设置下边框
border-left	设置左边框
border-right	设置右边框

## 3. 网页三角标制作

1. 元素设置宽高为0
2. 统一设置四个方向透明边框
3. 调整某个方向边框可见色

## 4. 圆角边框

1. 属性：border-radius 指定圆角半径
2. 取值：像素值或百分比
3. 取值规律：

一个值	表示统一设置上右下左
四个值	表示分别设置上右下左
两个值	表示分别设置上下 左右
三个值	表示分别设置上右下，左右保持一致

## 5. 轮廓线

1. 属性：outline
2. 取值：width style color
3. 区别：边框实际占位，轮廓不占位
4. 特殊：取none可以取消文本输入框默认轮廓线

## 6. 盒阴影

1. 属性：box-shadow
2. 取值：offsetX offsetY blur (spread) color;

3. 使用： 不管是浏览器窗口还是元素自身都可以构建坐标系，统一以左上角为原点，向右向下为X轴和Y轴的正方向

offsetX	取像素值，阴影的水平偏移距离
offsetY	取像素值，阴影的垂直偏移距离
blur	取像素值，表示阴影的模糊程度，值越大越模糊
spread	选填，取像素值，阴影是否需要延伸
color	设置阴影颜色,默认为黑色

### 3. 内边距

1. 属性： padding
2. 作用： 调整元素内容框与边框之间的距离
3. 取值：

20px;	一个值表示统一设置上右下左
20px 30px;	两个值表示分别设置(上下) (左右)
20px 30px 40px;	三个值表示分别设置上右下，左右保持一致
20px 30px 40px 50px;	表示分别设置上右下左

4. 单方向内边距,只能取一个值：

```
padding-top
padding-right
padding-bottom
padding-left
```

### 4. 外边距

1. 属性： margin
2. 作用： 调整元素与元素之间的距离
3. 特殊： 1) margin:0; 取消默认外边距 2) margin:0 auto;左右自动外边距，实现元素在父元素范围内水平居中 3) margin:-10px;元素位置的微调
4. 单方向外边距： 只取一个值 margin-top margin-right margin-bottom margin-left
5. 外边距合并： 1) 垂直方向 1. 子元素的margin-top作用于父元素上 解决： 为父元素添加顶部边框； 或为父元素设置padding-top:0.1px; 2. 元素之间同时设置垂直方向的外边距，最终取较大的值 2) 水平方向 块元素对盒模型相关属性（width,height,padding,border,margin）完全支持; 行内元素对盒模型相关属性不完全支持，不支持width/height,不支持上下边距 行内元素水平方向上的外边距会叠加显示 带有默认边距的元素： body,h1,h2,h3,h4,h5,h6,p,ul,ol{ margin:0; padding:0; list-style:none; }



[TOC]

# 一、布局方式

---

## 1. 标准流/静态流

默认布局方式,按照代码书写顺序及标签类型从上到下,从左到右依次显示

## 2. 浮动布局

主要用于设置块元素的水平排列

### 1) 属性

```
float
```

### 2) 取值

可取left或right, 设置元素向左浮动或向右浮动

```
float:left/right;
```

### 3) 特点

- 元素设置浮动会从原始位置脱流,向左或向右依次停靠在其他元素边缘,在文档中不再占位
- 元素设置浮动,就具有块元素的特征,可以手动调整宽高
- "文字环绕":浮动元素遮挡正常元素的位置,无法遮挡正常内容的显示,内容围绕在浮动元素周围显示

### 4) 常见问题

子元素全部设置浮动,导致父元素高度为0,影响父元素背景色和背景图片展示,影响页面布局

### 5) 解决

- 对于内容固定的元素,如果子元素都浮动,可以给父元素固定高度(例:导航栏)
- 在父元素的末尾添加空的块元素。设置clear:both;清除浮动
- 为父元素设置overflow:hidden;解决高度为0

## 3. 定位布局

结合偏移属性调整元素的显示位置

### 1) 属性

position

## 2) 取值

可取`relative`（相对定位）/`absolute`（绝对定位）/`fixed`（固定定位）

```
position: relative/absolute/fixed
```

## 3) 偏移属性

设置定位的元素可以使用偏移属性调整距离参照物的位置

<code>top</code>	距参照物的顶部
<code>right</code>	距参照物的右侧
<code>bottom</code>	距参照物的底部
<code>left</code>	距参照物的左侧

## 4) 分类

- `relative` 相对定位

元素设置相对定位,可参照元素在文档中的原始位置进行偏移,不会脱离文档流

- `absolute` 绝对定位

1. 绝对定位的元素参照离他最近的已经定位的祖先元素进行偏移,如果没有,则参照窗口进行偏移
2. 绝对定位的元素会脱流,在文档中不占位,可以手动设置宽高

使用绝对定位: "父相子绝": 父元素设置相对定位,子元素绝对定位,参照已定位的父元素偏移。

- `fixed` 固定定位

1. 参照窗口进行定位,不跟随网页滚动而滚动
2. 脱离文档流

## 5) 堆叠次序

元素发生堆叠时可以使用 `z-index` 属性调整已定位元素的显示位置,值越大元素越靠上:

- 属性: `z-index`
- 取值: 无单位的数值,数值越大,越靠上
- 堆叠:

1. 定位元素与文档中正常元素发生堆叠，永远是已定位元素在上
2. 同为已定位元素发生堆叠，按照 HTML 代码的书写顺序，后来者居上

## 二、背景属性

---

### 1. 背景颜色

```
background-color: red;
```

### 2. 背景图片相关

#### 1) 设置背景图片

```
background-image : url("路径");
```

设置背景图片，指定图片路径，如果路径中出现中文或空格，需要加引号

#### 2) 设置背景图片的重复方式

默认背景图片从元素的左上角显示，如果图片尺寸与元素尺寸不匹配时，会出现以下情况：

1. 如果元素尺寸大于图片尺寸，会自动重复平铺，直至铺满整个元素
2. 如果元素尺寸小于图片尺寸，图片默认从元素左上角开始显示，超出部分不可见

```
background-repeat:repeat/repeat-x/repeat-y/no-repeat
```

取值：

repeat 默认值，沿水平和垂直方向重复平铺  
repeat-x 沿X轴重复平铺  
repeat-y 沿Y轴重复平铺  
no-repeat 不重复平铺

#### 3) 设置背景图片的显示位置

默认显示在元素左上角

```
background-position:x y;
```

取值方式：

### 1. 像素值

设置背景图片的在元素坐标系中的起点坐标

### 2. 方位值

水平 : `left/center/right`

垂直 : `top/center/bottom`

注: 如果只设置某一个方向的方位值, 另外一个方向默认为`center`

### 3. 百分比

类似于方位值, 根据百分比计算背景图片的显示坐标。

计算方式:

横坐标 = (元素宽度 - 背景图片宽度) \* `x%`

纵坐标 = (元素高度 - 背景图片高度) \* `y %`

特殊值:

`0% 0%` 左上角

`100% 100%` 右下

`50% 50%` 居中显示

精灵图技术: 为了减少网络请求, 可以将所有的小图标拼接在一张图片上, 一次网络请求全部得到; 借助于 `background-position` 进行背景图片位置的调整, 实现显示不同的图标

## 4) 设置背景图片的尺寸

```
background-size:width height;
```

取值方式:

### 1. 像素值

1. `500px 500px`; 同时指定宽高

2. `500px`; 指定宽度, 高度自适应

### 2. 百分比

百分比参照元素的尺寸进行计算

1. `50% 50%`; 根据元素宽高, 分别计算图片的宽高

2. `50%`; 根据元素宽度计算图片宽高, 图片高度等比例缩放

## 3. 背景属性简写

```
background:color url("") repeat position;
```

注意:

1. 如果需要同时设置以上属性值, 遵照相应顺序书写

2. `background-size` 单独设置

# 三、文本属性

## 1. 字体相关

### 1) 设置字体大小

```
font-size:20px;
```

### 2) 设置字体粗细程度

```
font-weight:normal;
```

取值：

1. normal（默认值）等价于400
2. bold（加粗）等价于700

### 3) 设置斜体

```
font-style:italic;
```

### 4) 设置字体名称

```
font-family:Arial,"黑体";
```

取值：1. 可以指定多个字体名称作为备选字体,使用逗号隔开 2. 如果字体名称为中文,或者名称中出现了空格,必须使用引号 例：

```
font-family:Arial;  
font-family:"黑体","Microsoft YaHei",Arial;
```

### 5) 字体属性简写

```
font : style weight size family;
```

注意：1. 如果四个属性值都必须设置,严格按照顺序书写 2. size family 是必填项

## 2. 文本样式

## 1) 文本颜色

```
color:red;
```

## 2) 文本装饰线

```
text-decoration:none;
```

取值: underline 下划线 overline 上划线 line-through 删除线 none 取消装饰线

## 3) 文本内容的水平对齐方式

```
text-align:center;
```

取值:

left(默认值)	左对齐
center	居中对齐
right	右对齐
justify	两端对齐

## 4) 行高

```
line-height:30px;
```

使用: 文本在当前行中永远垂直居中,可以借助行高调整文本在元素中的垂直显示位置  $\text{line-height} = \text{height}$  设置一行文本在元素中垂直居中  $\text{line-height} > \text{height}$  文本下移显示  $\text{line-height} < \text{height}$  文本靠上显示 特殊: line-height 可以采用无单位的数值,代表当前字体大小的倍数,以此计算行高

## 5) font属性简写2

```
font : size/line-height family;
```