

CSS 层叠样式表

1 CSS 代码写在哪？

1.1 内联样式表：HTML 标签内

```
<p style="color:blue;">落霞与孤鹜齐飞，秋水共长天一色</p>
```

1.2 内部样式表：head 标签内，title 标签后面

```
<style type="text/css">
    p {
        font-size: 80px;
    }
</style>
```

1.3 外部样式表：独立的 CSS 文件中

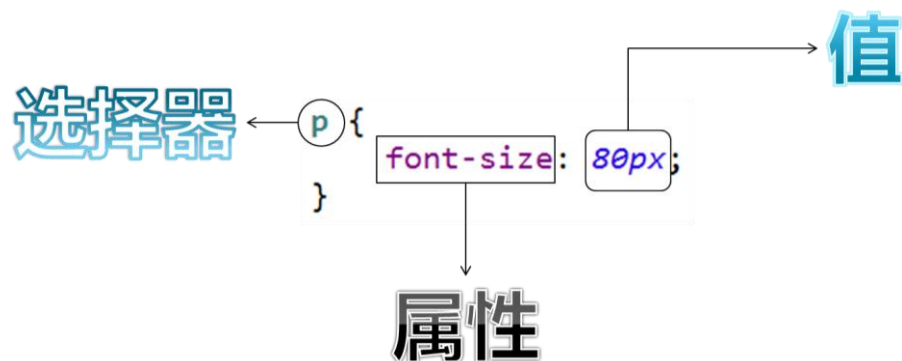
```
p {
    border-style: solid;
    border-width: 1px;
    border-color: black;
    text-align: center;
}
```

使用 link 标签引入到当前文档中

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

2 CSS 基本语法

2.1 CSS 语法由三部分构成：选择器、属性和值：selector {property: value}



3 CSS 选择器：根据 id 值定位 HTML 元素

3.1 id 选择器

```
<ul>
    <li>普通列表项</li>
    <li>普通列表项</li>
    <li>普通列表项</li>
    <li id="tuHao">土豪列表项</li>
    <li>普通列表项</li>
</ul>
#tuHao {
    font-size: 50px;
}
```

```
}
```

- 普通列表项
- 普通列表项
- 普通列表项
- 土豪列表项
- 普通列表项

3.2 类选择器：如果想将一组元素设定为相同的样式，可为它们设置相同的 class 属性值，再通过 CSS 类选择器指定相同的样式

```
<ul>
  <li>普通列表项</li>
  <li>普通列表项</li>
  <li>普通列表项</li>
  <li class="special">特殊列表项</li>
  <li class="special">特殊列表项</li>
  <li class="special">特殊列表项</li>
  <li>普通列表项</li>
  <li>普通列表项</li>
</ul>
.special {
  font-size: 50px;
}
```

- 普通列表项
- 普通列表项
- 普通列表项
- 特殊列表项
- 特殊列表项
- 特殊列表项
- 普通列表项
- 普通列表项

3.3 选择器分组：不同的元素使用相同的样式时，可以使用“,” 隔开，表示并列关系

```
<h1>标题1</h1>
<h2>标题2</h2>
<h3>标题3</h3>
<h4>标题4</h4>
```

```
<h5>标题5</h5>
<h6>标题 6</h6>
h2,h3 {
    border-style: solid;
    border-width: 1px;
    border-color: black;
}
```

标题1

标题2

标题3

标题4

标题5

标题6

3.4 派生选择器：使用“父元素 子元素”的格式定位到特定父元素下的子元素

```
<div>
    <p>div下的段落</p>
    <p>div下的段落</p>
    <p>div下的段落</p>
</div>
<p>普通段落</p>
<p>普通段落</p>
<p>普通段落</p>
div p {
    background-color: blue;
    color: white;
    font-size: 50px;
}
```

div下的段落

div下的段落

div下的段落

普通段落

普通段落

普通段落

- 3.5 id 选择器结合派生选择器：“#id 值 子元素”使用 id 选择器定位到特定元素后，定位其特定子元素

```
<div>
    <p>段落</p>
    <p>段落</p>
</div>
<div id="targetDiv">
    <p>目标段落</p>
    <p>目标段落</p>
</div>
<div>
    <p>段落</p>
    <p>段落</p>
</div>
#targetDiv p {
    background-color: blue;
    color: white;
    font-size: 50px;
}
```

段落

段落

目标段落

目标段落

段落

段落

- 3.6 类选择器结合派生选择器

```
<div>
    <p>段落</p>
</div>
<div class="targetDiv">
    <p>目标段落</p>
</div>
<div>
    <p>段落</p>
</div>
<div class="targetDiv">
    <p>目标段落</p>
```

```
</div>
.targetDiv p {
    font-style: italic;
    font-size: 30px;
}
```

段落

目标段落

段落

目标段落

3.7 元素也可以基于它们的类而被选择

```
<p class="big">段落</p>
<p class="big">段落</p>
<table>
    <tr>
        <td>单元格</td>
        <td>单元格</td>
        <td>单元格</td>
    </tr>
    <tr>
        <td class="big">单元格</td>
        <td class="big">单元格</td>
        <td class="big">单元格</td>
    </tr>
    <tr>
        <td>单元格</td>
        <td>单元格</td>
        <td>单元格</td>
    </tr>
</table>
td.big {
    background-color: #cccccc;
    color: #0000ff;
}
.big {
    font-style: italic;
}
```

段落

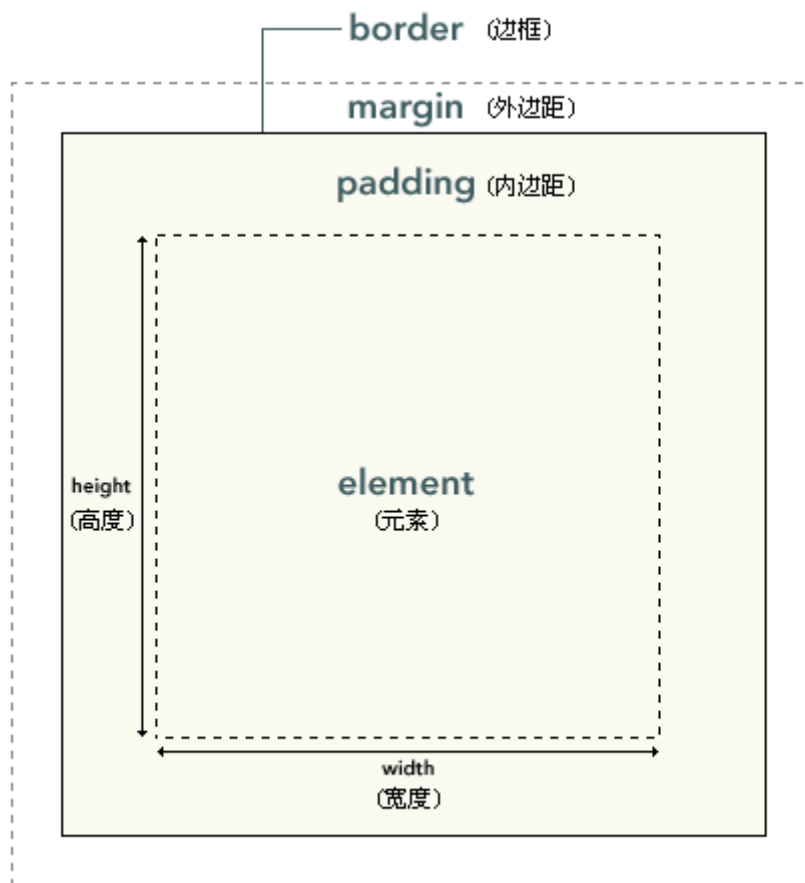
段落

单元格 单元格 单元格

单元格 单元格 单元格

单元格 单元格 单元格

4 CSS 框模型



W3School.com.cn

行框：前后不换行，不能指定宽度[a、span、strong 等标签，或通过 display : inline 指定]

落花有意，广告时间流水无情

落花有意，广告时间流水无情

默认效果：

转为块框：display : block;

可以指定宽度了：

落花有意，

流水无情

块框：前后换行[div、table、p、h1~h6 等，或通过 display : block 指定]

其实世上本没有路，<div>走的人多了</div>，也便成了路

其实世上本没有路，

走的人多了

默认效果：，也便成了路

转为行框：display: inline;

其实世上本没有路，走的人多了，也便成了路
宽度设定失效：

行内框：前后不换行，但能指定宽度[通过 display : inline-block 指定]

其实世上本没有路，<div id="widthDiv">走的人多了</div>，也便成了路

转为行内框：display: inline-block;

在行内，前后无换行，但是能够设置宽度：

其实世上本没有路，走的人多了，也便成了路

5 CSS 定位机制

5.1 文档流

5.1.1 行框：按顺序水平摆放，放不下出现滚动条

5.1.2 块框：按顺序垂直摆放，放不下出现滚动条

5.2 相对定位，原来在文档流中的位置仍然保留，当前位置是相对于原始位置偏移后的结果

```
position: relative;
```

```
#pos {  
    position: relative;  
    top: 5px;  
    left: 55px;  
}
```



5.3 绝对定位

```
position: absolute;
```

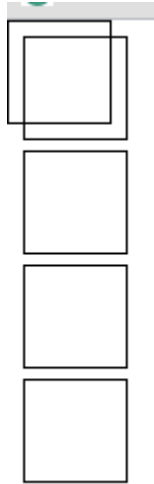
从文档流中删除其原来的位置，就好像该元素从来都不存在一样。它当前所处的位置如果和其他元素重合则会遮盖住其他元素的显示，这就是 CSS 中层的概念。

当前位置的定位有两种情况：

①父容器以及祖先容器未定位：相对于浏览器左上角

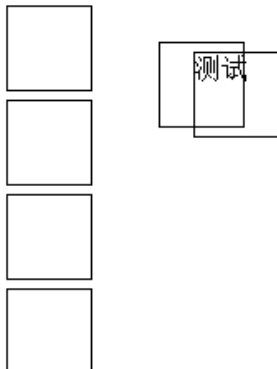
```
#pos {
```

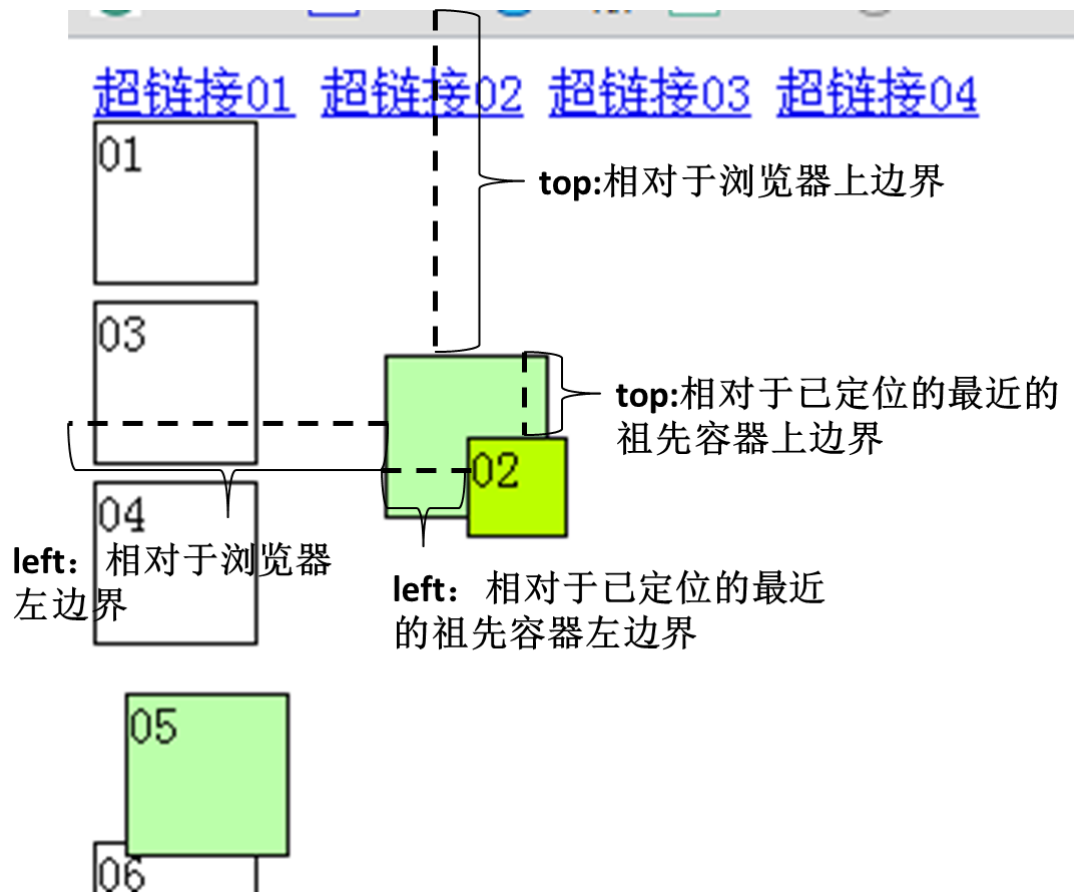
```
position: absolute;  
top: 0px;  
left: 0px;  
}
```



②存在已定位的祖先元素：相对于最接近的已定位的祖先元素

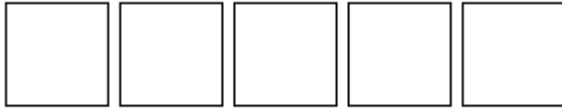
```
<div>&nbsp;</div>  
<div>&nbsp;</div>  
<div id="pos"><div id="test">测试</div></div>  
<div>&nbsp;</div>  
<div>&nbsp;</div>  
#pos {  
    position: absolute;  
    top: 30px;  
    left: 100px;  
}  
#test {  
    position: absolute;  
    top: 5px;  
    left: 20px;  
}
```





5.4 浮动：脱离文档流，向上浮起一层，所有同一容器内的浮动元素根据 `float` 属性值按顺序进行排列，例如如果都是 `float:left` 则按从左向右的顺序排列。也会遮盖住文档流中未浮动的正常元素

```
<div>&nbsp;</div>
<div>&nbsp;</div>
<div>&nbsp;</div>
<div>&nbsp;</div>
<div>&nbsp;</div>
div {
  border-style: solid;
  border-width: 1px;
  border-color: black;
  width: 50px;
  height: 50px;
  margin-right: 5px;
  float: left;
}
```



补充:

设置元素透明

filter:alpha(opacity=90);

-moz-opacity:0.9;

-khtml-opacity:0.9;

opacity:0.9;

设置表格线的细线效果

```
table {  
    border-spacing: 1px;  
    background-color: black;  
}
```

```
td,th {  
    background-color: white;  
}
```

使控件不可用

```
<input type="button" value="添加" class="btn" disabled="disabled" />
```

使元素隐藏

display: none;

visibility: hidden;