

# 样式的继承

子元素会继承父元素的样式

## 单位

像素: 200px

百分比: 相对于父元素的大小

em: 相对于元素的字体的大小计算

rem: 相对于根元素的字体大小

## 文档流

### 块元素（块级元素）

在页面中，独占一行，从上往下，垂直排列，默认情况下的宽度，就是浏览器的宽度，高度会被内容撑开

### 行元素（行内元素）

不独占一行，在页面中，从左往右依次排列，高度和宽度，被元素撑开，宽度会自动换行

## 盒子模型

内容区域：元素的内容，width, height

边框：盒子的边缘框架，框里面是盒子内部，外面是外部

内边距(padding)：内容区域和边框之间的距离

外边距(margin):不影响盒子的显示的大小的，影响盒子的位置

```
.box { width: 300px; /*内容区域宽度*/ height: 300px; /*内容区域高度*/ background-color: blue; border: 15px red solid; /*边框 粗细15px 颜色红色 实线 */ padding: 50px 70px 80px 100px; /*内边距: 上50px 右70px 下80px 左100px*/ padding: 150px; /*上下左右都是150px */ padding: 50px 150px; /*前面一个是上下 50px，左右是150px*/ margin: 100px; /*外边距 100px*/ }
```

## 水平布局

子元素在父元素中，必须要满足

左右margin的宽度 + 左右border的宽度 + 左右padding的宽度 + 内容宽度 = 父亲的宽度

如果其中是auto的，那么浏览器自动计算其宽度 margin : 0

auto, 那么, 自动计算左、右margin的宽度为(父亲宽度-孩子宽度) / 2

水平居中: 父元素一定要有宽度

```
.innerBox {  
  margin: 0 auto  
}
```

## 垂直布局

默认情况下, 父元素的高度被内容撑开。

如果父元素设置了高度和宽度, 子元素如果过长, 会产生溢出。处理溢出, overflow

```
.box {  
  width: 200px;  
  height: 200px;  
  background-color: aqua;  
  overflow: auto;  
}
```

## 浮动

浮动float: 子元素可以移动到父元素的左边或者右边 float:left  
, 让元素可以水平排列

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8" />  
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />  
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
    <title>Document</title>  
    <style>  
      header {  
        width: 1000px;  
        margin: 0 auto;  
        height: 80px;  
        background-color: #af251b;  
        color: white;  
        text-align: center;  
        line-height: 80px;  
        margin-bottom: 10px;  
      }  
  
      main {
```

```

width: 1000px;
height: 300px;
background-color: yellow;
margin: 0 auto;
margin-bottom: 10px;
}

footer {
width: 1000px;
height: 30px;
background-color: blue;
margin: 0 auto;
}

main .nav {
width: 80px;
height: 300px;
background-color: aqua;
float: left;
}

main .content {
width: 910px;
height: 300px;
background-color: darkmagenta;
margin-left: 10px;
float: left;
}
</style>
</head>
<body>
  <header>苏州大学</header>

  <main>
    <div class="nav">
      导航
    </div>

    <div class="content">
      内容
    </div>

  </main>
  <footer>底部</footer>
</body>
</html>

```

## Flex布局

当前最常用，会在移动开发学习，大家自学

# CSS框架

---

别人帮你设计好很多样式，你直接去用

bootstrap

tailwind

# bootstrap框架

---