

# day18

## 3.边框【9:13】

### (1) 边框的编写

- `.border` 四个边的边框，浅灰色很细，是基础类
- `.border-{fx}` 单边边框，top, bottom, start, end, 这些类是基类

```
<span class="border"></span>
<span class="border-top"></span>
<span class="border-end"></span>
<span class="border-bottom"></span>
<span class="border-start"></span>
```

### (2) 去掉边框线

将单边或者所有边的边框线去掉，一般是指原本就有边框线需要去掉的某些元素。

```
<span class="border-0"></span>
<span class="border-top-0"></span>
<span class="border-end-0"></span>
<span class="border-bottom-0"></span>
<span class="border-start-0"></span>
```

### (3) 边框颜色

边框的颜色，前提是要有基础类 `.border` 或单边的基础类

```
<span class="border border-primary"></span>
<span class="border border-secondary"></span>
```

## (4) 边框宽度

边框宽度的粗细，boot中为我们定义了5个，`.border-{num}` 数字1-5，边框级别的单位是px。

```
<span class="border border-1"></span>
<span class="border border-2"></span>
<span class="border border-3"></span>
<span class="border border-4"></span>
<span class="border border-5"></span>
```

## (5) 边界半径

元素添加圆角，默认圆角很小，但是圆形和胶囊型特殊

- `.rounded` 四个方向的小圆角，单独方向的圆角 `.rounded-{fx}`
- `.rounded-circle` 圆形
- `.rounded-pill` 胶囊型

```







```

## 4.间距【9:54】

间距里包含内、外间距

## (1) 间距的基础类

单独的基础类无法直接出现，需要加间距的距离

- `m-` 对于设置的类 `margin`
- `p-` 对于设置的类 `padding`

距离分为五个长度，指间距的四个方向

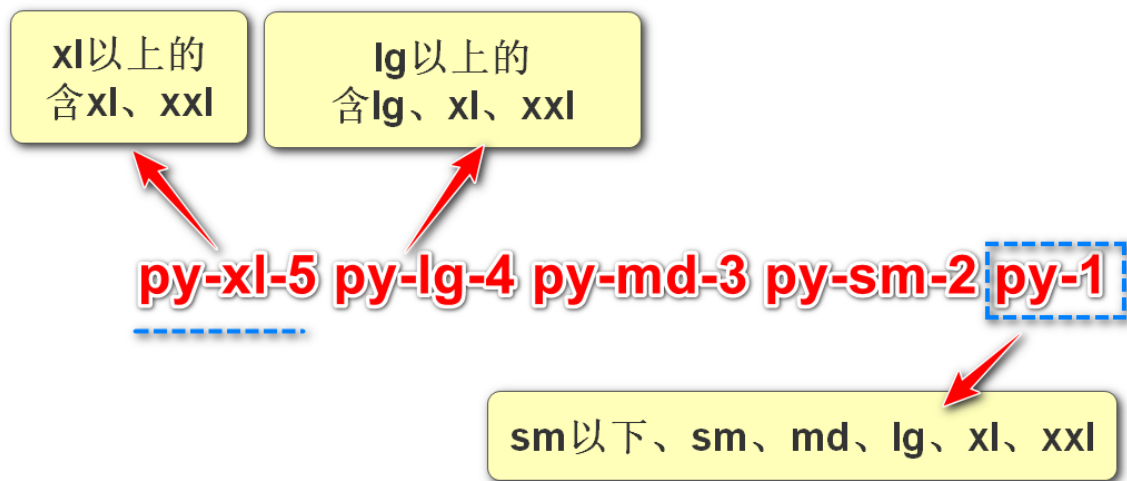
- `0-` 对于消除 `m-0` 或 `p-0` 设置为的类 `0` 大小
- `1-` 用于设置 `m-1` 或 `p-1` 的类 `0.25rem`
- `2-` 用于设置 `m-2` 或 `p-2` 的类 `0.5rem`
- `3-` 用于设置 `m-3` 或 `p-3` 的类 `1rem`
- `4-` 用于设置 `m-4` 或 `p-4` 的类 `1.5rem`
- `5-` 用于设置 `m-5` 或 `p-5` 的类 `3rem`
- `auto-` 对于设置 `margin` 为自动的类

## (2) 间距的方向

- `t-` 对于设置 `mt-{num}` 或 `pt-{num}` 上内外间距
- `b-` 对于设置 `mb-{num}` 或 `pb-{num}` 下内外间距
- `s-` (开始) 用于设置 `ms-{num}` 或 `ps-{num}` 左内外间距
- `e-` (结束) 对于设置 `me-{num}` 或 `pe-{num}` 右内外间距
- `x-` 对于同时设置左右内外间距 `mx-{num}` 和 `px-{num}`
- `y-` 对于同时设置上下内外间距 `my-{num}` 和 `py-{num}`

## (3) 响应式写法

`{内外间距}{方向}-{类中缀}-{数字}` 这些类使用 `sm`、`md`、`lg`、`xl` 和 `xxl`。



```
<div class="py-xl-5 py-lg-4 py-md-3 py-sm-3 py-1">随便写文字</div>
```

响应式写法，注意范围，注意可以简写的地方

## 5.展示【10:54】

展示就是display属性，以下是常用属性

- `.d-{value}` value指常用的none, inline, inline-block, block, flex
- `.d-{类中缀}-{value}` 包含 sm, md, lg, xl, and xxl

```
<!-- 图去幽灵宽度100% -->
<div class="container">
  <div class="zdy-w">
    
  </div>
</div>
<div class="container">
  <div class="zdy-h w-75 bg-primary d-inline-
block"></div><div class="zdy-h w-25 bg-danger d-none
d-md-inline-block"></div>
  <!-- 所有时候消失，md以上显示 -->
</div>
```

## 6.浮动【11:37】

### (1) 浮动属性

给元素增加浮动属性，但是注意，元素浮动后，其父元素会高度坍塌，需要清除浮动

```
<div class="float-start">左浮动</div>
<div class="float-end">右浮动</div>
<div class="float-none">不浮动</div>
```

### (2) 响应式浮动

float-{类中缀}-{方式},响应式是sm、md、lg、xl、xxl

```
<div class="float-sm-start"></div>
```

### (3) 清除浮动

当子元素浮动，父元素高度坍塌，因此要靠清除浮动类解决。

给父元素加 `.clearfix` 类

```
<div class="container bg-warning clearfix">
  <div class="zdy-h w-25 bg-primary float-start"></div>
  <div class="zdy-h w-25 bg-danger float-end">
</div>
</div>
```

## 7.定位【14:01】

### (1) 定位的常用基本属性

```
<div class="position-relative">相对定位</div>
<div class="position-absolute">绝对定位</div>
<div class="position-fixed">固定定位</div>
```

还有两种特定的，在有需要的特定情境下使用，不需要单独指定top和bottom

- `fixed-top` 吸顶导航，固定定位在视口顶端的
- `fixed-bottom` 吸底导航，固定定位在视口底端的

### (2) 排列元素

方向：

财产是以下之一：

- `top` - 对于垂直顶端位置
- `start` - 对于水平 `left` 位置

- `bottom`- 对于垂直底端位置
- `end`- 对于水平 `right` 位置

位置:

- `0`- 用于 `0` 边缘位置
- `50`- 用于 `50%` 边缘位置
- `100`- 用于 `100%` 边缘位置

### (3) 居中

使用转换实用程序类使元素居中 `.translate-middle` 双方向回移自己的50%

- `.translate-middle` 双方向回移自己的50%
- `.translate-middle-x` 轴方向向回移自己宽度的50%
- `.translate-middle-y` 轴方向向回移自己高度的50%

```
<div class="zdy-box position-relative">
  <div class="position-absolute top-50
start-50 translate-middle"></div>
</div>
```

## 练习

```
<div class="youjian bg-primary py-2 text-
white text-center position-relative">
  邮件
  <span class="position-absolute top-0
start-100 translate-middle bg-danger rounded-
circle">16</span>
</div>
```

## 8.文本【14:39】

### (1) 文本的水平居中方式

```
<p class="text-start">文本的居左</p>
<p class="text-center">文本居中</p>
<p class="text-end">文本右</p>
```

文本对齐的响应式

`.text-{类中缀}-{方位}` 文本对齐的响应式, sm, md, lg, xl, xxl

### (2) 文本粗细

pc端浏览器一般会显示文字三种粗细，粗，细，正常。但是在移动终端开发时，手机能展现的粗细将会更多，且不同的系统显示略有不同。

```
<p class="fw-bold">粗体</p>
<p class="fw-bolder">加粗体</p>
<p class="fw-normal">正常体</p>
<p class="fw-light">细体</p>
<p class="fw-lighter">极细体</p>
<p class="fst-italic">斜体</p>
```

### (3) 文本修饰线

```
<p class="text-decoration-underline">有下划线</p>
<p class="text-decoration-line-through">删除线</p>
<a href="#" class="text-decoration-none">去除下划线
</a>
```



# 栅格系统【15:20】

## 1.行与列

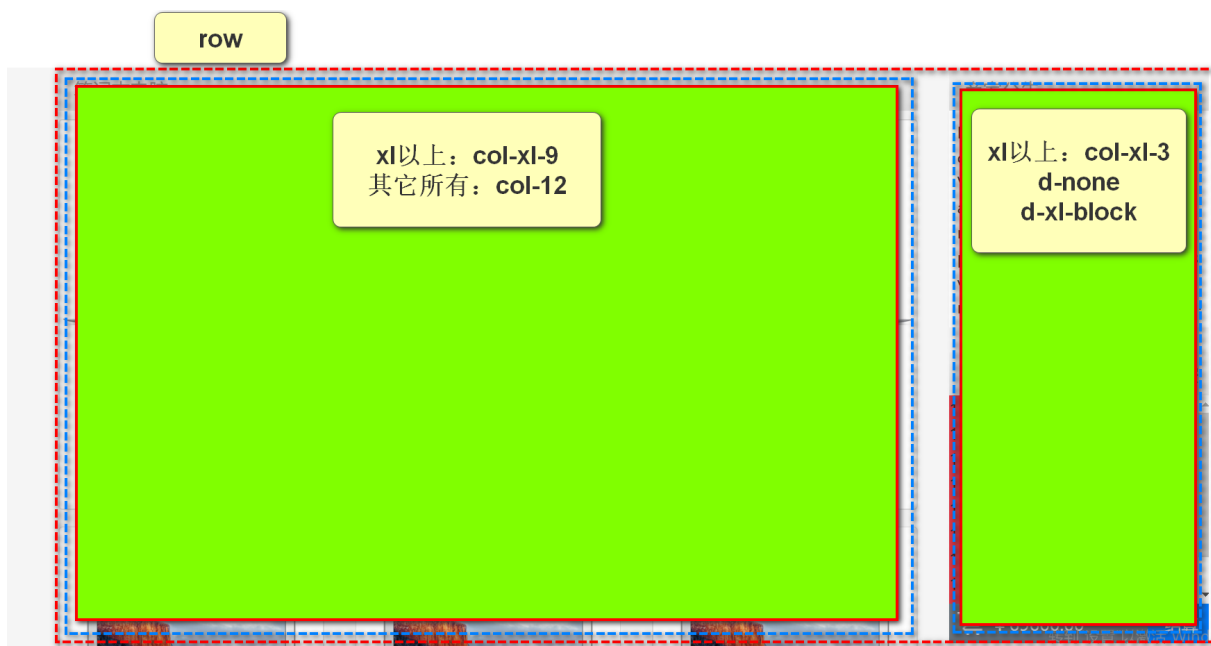
栅格系统分为行 `.row` 和列 `.col`, 行必须包裹列。可以是一行包多列, 也可以写多行包多列。

```
<div class="container">
  <div class="row">
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
  </div>
</div>
```

## 2.响应式列

	xs <576 像素	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
容器 <code>max-width</code>	无 (自动)	540像素	720像素	960像素	1140像素	1320像素
类前缀	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>	<code>.col-xxl-</code>
# 列数	12					
天沟宽度	1.5rem (左右 0.75rem)					

## 练习



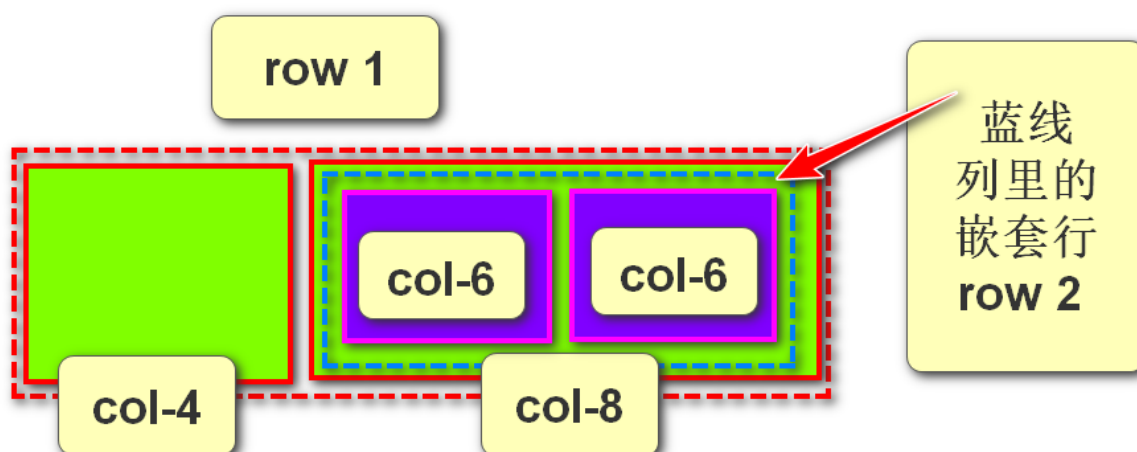
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <link rel="stylesheet" href="../css/bootstrap.css">
  <link rel="stylesheet" href="../css/bootstrap-
icons.css">
  <script src="../js/bootstrap.bundle.js"></script>
  <title>Document</title>
  <style>
    .zdy-h {
      height: 400px;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="row">
```

```

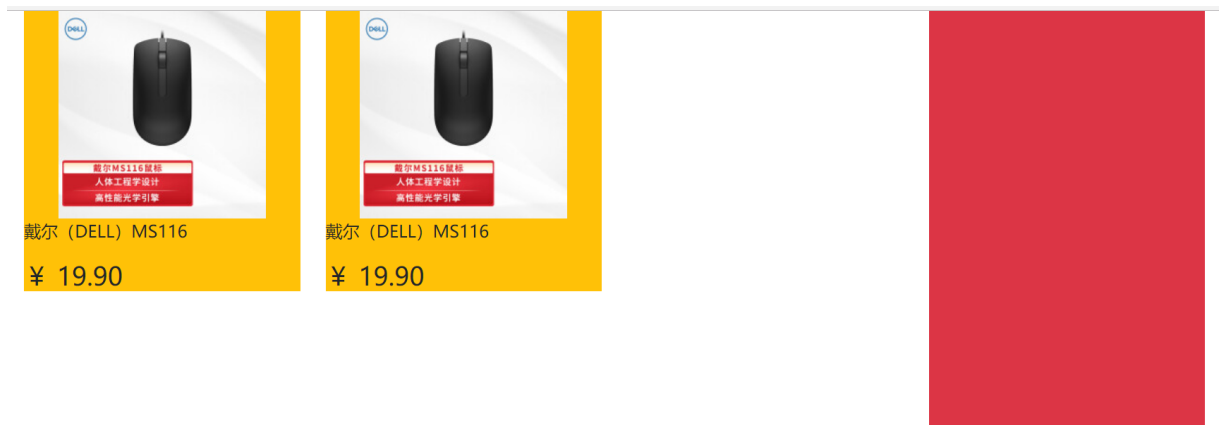
<div class="col-xl-9 col-12">
    <div class="zdy-h bg-primary"></div>
</div>
<div class="col-xl-3 d-none d-xl-block">
    <div class="zdy-h bg-danger"></div>
</div>
</div>
</div>
</body>
</html>

```

### 3. 栅格的嵌套



### 练习



写完整体效果，更换文字和图片，尽量自己写一个标题单行移除省略效果