

# Bootstrap 的组件

## 六、组件

组件就是 boot 封装好的一些内容但它们有自己特有的样式组件有的是有动态交互效果的，因此有的需要 js 参与

### 1.按钮

- 可以作为按钮的标签：`<a>`、`<button>`、`<input>`
- `.btn` 按钮的基本类，渲染了基本的背景、边框、过渡等样式，必须加

#### (1)颜色样式

按钮的颜色都在 Bootstrap 中广泛应用，需要有印象。另外自己配色可以参照

- `.btn-default` 默认样式
- `.btn-success` 成功
- `.btn-primary:hover` 按钮的悬停样式，根据按钮颜色指定



#### (2)按钮轮廓

- `.btn-outline-success` 成功
- `.btn-outline-warning` 警告
- `.btn-outline-info:hover` 按钮的悬停样式，根据按钮颜色指定



### (3)块级按钮

块级按钮`.btn-block`会随着外层容器的宽度变化，因为它的宽度是外层的100%，多用于移动端，比如：支付，下一步，登录等

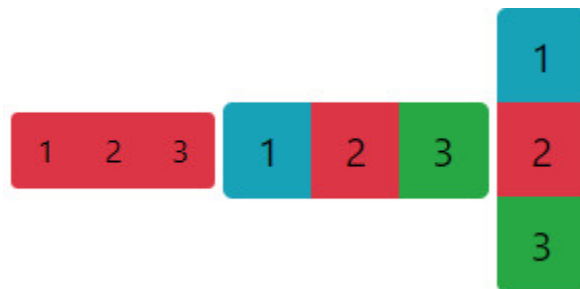
## 2.按钮组

按钮组就是把按钮放在一个组里显示

-`.btn-group` 按钮组（基类）

-`.btn-group-lg`和`.btn-group-sm`按钮组尺寸

-`.btn-group-vertical`按钮组竖列显示，不需要写基类`.btn-group`



```
<div class="container">
  <!-- 普通的按钮 -->
  <button class="btn">按钮</button>
  <button class="btn btn-danger btn-sm">按钮</button>
  <button class="btn btn-success btn-lg">按钮</button>
  <button class="btn btn-outline-danger">按钮</button>
  <button class="btn btn-block btn-primary">块级按钮</button>
  <!-- 按钮组 -->
  <!-- 一说组，必须找一个元素包起来 btn-group 按钮组 -->
  <div class="btn-group">
    <button class="btn bg-danger">1</button>
    <button class="btn bg-danger">2</button>
    <button class="btn bg-danger">3</button>
  </div>
  <div class="btn-group btn-group-lg">
    <button class="btn bg-info">1</button>
    <button class="btn bg-danger">2</button>
    <button class="btn bg-success">3</button>
  </div>
  <!-- 竖排按钮 btn-group-vertical -->
  <div class="btn-group-vertical btn-group-lg">
    <button class="btn bg-info">1</button>
    <button class="btn bg-danger">2</button>
    <button class="btn bg-success">3</button>
  </div>
</div>
```

### 3.徽章

是一种小型的用于计数和打标签的组件，比如邮件未读提示，购物车角标等等,一般涵盖在标签中用 span 标签来表示徽章

- `.badge` 徽章（基类）
- `.badge-pill` 胶囊徽章，可以写基类可以不写，不写会变大
- `.badge-primary` 徽章主题颜色

你好,吴道华 新用户

收件箱 2

收件箱 2

发件箱

草稿箱 2

```
<div class="container">
  <p>你好,吴道华 <span class="badge bg-success text-white">新用户</span></p>
  <button class="btn btn-primary">收件箱 <span class="badge bg-danger">2</span>
</button>
  <!-- 小练习 -->
  <div class="btn-group-vertical btn-group-lg">
    <button class="btn btn-info">收件箱 <span class="badge bg-
danger">2</span></button>
    <button class="btn btn-warning">发件箱</button>
    <button class="btn btn-success">草稿箱 <span class="badge bg-
danger">2</span></button>
  </div>
</div>
```

### 4.卡片

- `.card` 卡片，最外层容器
- `.card-header` 卡片头部信息
- `.card-img-top` 卡片顶部图片 和 `.card-img-bottom` 卡片的底部图片
- `.card-body` 卡片主体部分
- `.card-title` 卡片标题格式

- `.card-text` 卡片文字介绍格式

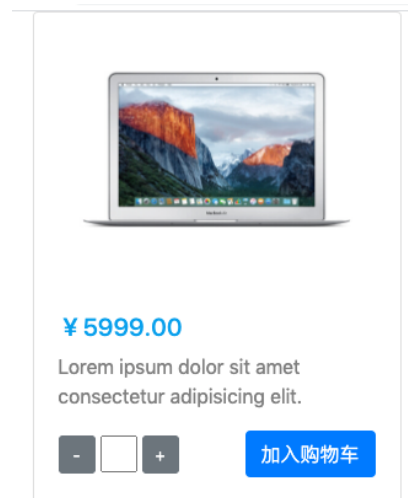
- `.card-footer` 卡片的底部

```
<div class="card box">
  <h3 class="card-header">一个商品</h3>
  
  <div class="card-body">
    <h4 class="card-title">美的炒菜锅</h4>
    <p class="card-text">美的（Midea）典雅金麦饭石色涂层不粘炒锅家用炒锅电磁炉
  </p>
    <button class="btn btn-danger">加入购物车</button>
  </div>
  <div class="card-footer">暂时无货</div>
</div>
```

练习 使用卡片做栅格布局多张卡片效果

```
<style>
  .zdy-w {width: 300px;}
  /* 学子的卡 */
  .xz-card h5{color: #0aa1ed;}
  .xz-card p{color: #808080;}
  .xz-shop input {
    width: 30px;
  }
  .xz-shop >div>button {
    width: 30px;
  }
</style>

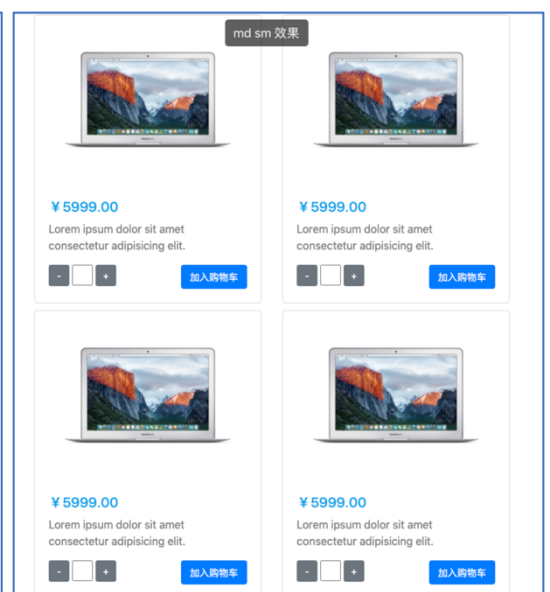
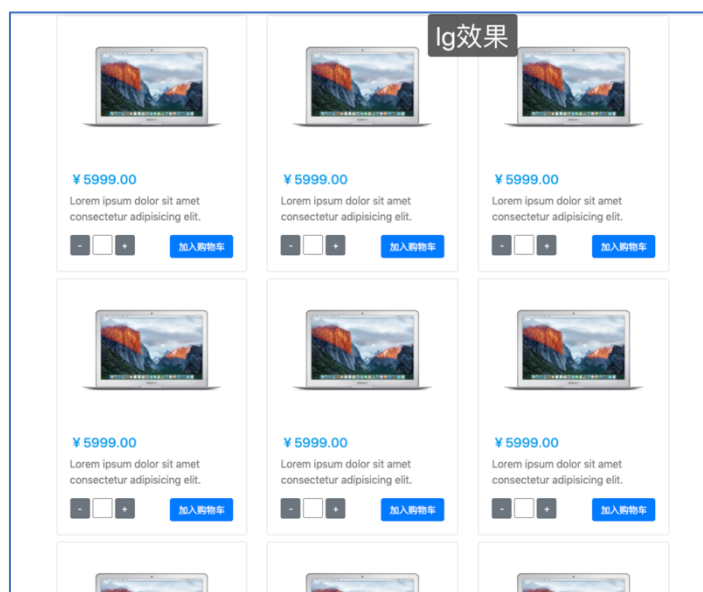
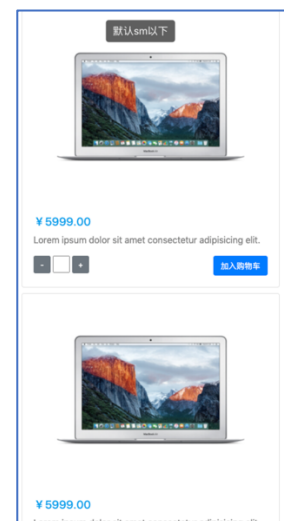
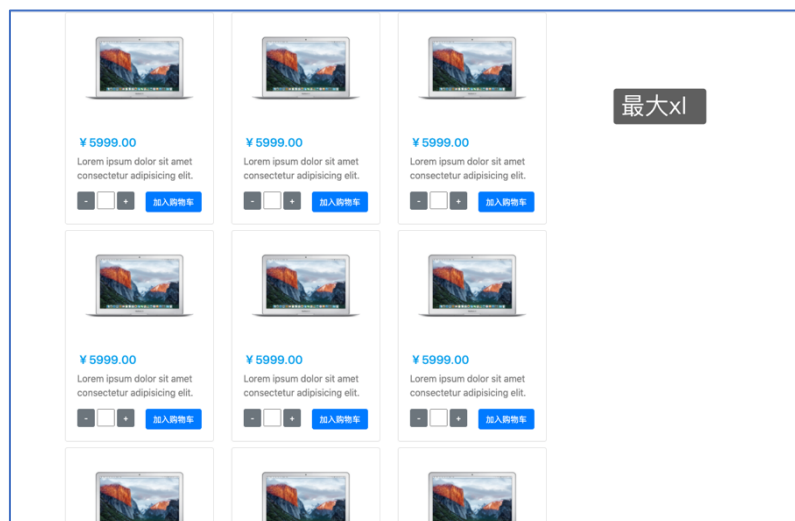
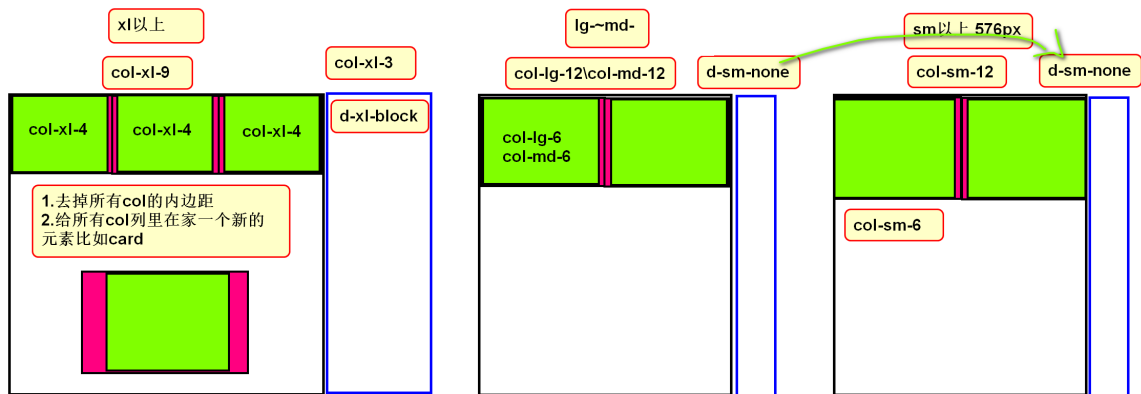
<div class="container">
  <div class="card zdy-w xz-card">
    
    <div class="card-body">
      <h5> ¥ 5999.00</h5>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.</p>
      <div class="xz-shop d-flex justify-content-between">
        <div class="clearfix">
          <button class="btn btn-sm btn-secondary float-left mt-1">-</button>
          <input type="text" class="float-left mt-1 ml-1 mr-1">
          <button class="btn btn-secondary btn-sm float-left mt-1">+</button>
        </div>
        <button class="btn btn-primary">加入购物车</button>
      </div>
    </div>
  </div>
</div>
```



```

</div>
<button class="btn btn-primary">加入购物车</button>
</div>
</div>
</div>
</div>

```



## 5.面包屑导航

面包屑导航 ( Breadcrumb ) 用于指示当前页面在导航层级中的位置

- `.breadcrumb` 面包屑导航外层属性
- `.breadcrumb-item` 面包屑导航的项目
- `.active` 代表当前的项目

```
<style>

/*找到分隔符的 css , 把 content 改了*/

.breadcrumb-item+.breadcrumb-item::before {
  content: ">" !important;
}
</style>
<nav aria-label="breadcrumb">
  <ol class="breadcrumb">

    <li class="breadcrumb-item">首页</li>

    <li class="breadcrumb-item">学习用品</li>

    <li class="breadcrumb-item">笔记本电脑</li>

    <li class="breadcrumb-item active">联想</li>

  </ol>
</nav>
```

## 6.弹窗

组件中涉及自定义属性，通过自定义属性控制交互行为。

因为自定义属性的存在，因此使用自定义属性做 js 交互

自定义属性`data-dismiss="alert"`通过 data 属性：通过数据 API 添加可取消功能，只需要向关闭按钮添加 `data-dismiss="alert"`，就会自动为警告框添加关闭功能。

- `.alert` 设置元素为警告框组件

- `.alert-info` 颜色
- `.close` 用于警告框内单独的标签设置按钮为可关闭；
- `data-dismiss="alert"`是给关闭按钮设置的自定义属性事件

```
<div class="container">
  <!-- 一个单独的提示窗没有任何交互的 -->
  <div class="alert alert-primary">这是一个提示窗</div>
  <!-- 有关闭标识 -->
  <div class="alert alert-danger">
    这是一个有关闭标识的提示窗
    <span class="close" data-dismiss="alert">x</span>
  </div>
</div>
```

## 7.折叠

折叠是一种效果，有一个类似按钮的元素，点按之后出现对应的内容。需要一个按钮类的组件来进行显示元素的控制。之前在学习 css 的时候有一个选择器叫 target，和这个类似。使用`data-toggle="collapse"`自定义属性作为切换的按钮控制折叠元素，同时要用选择器锁定被折叠元素`data-target="#show"`。

这个选择器如果使用 id，则会控制对应 id 元素的打开和折叠，如果是类名，则可以控制两个显示区域的切换。

在 boot 4.6 以上版本，有单独的手风琴效果，4.0 没有

- 需要两组元素，第一组是控制元素，控制显示和隐藏的区域
- 控制元素需要使用`data-toggle="collapse"`设置折叠控制，`data-target="#show1"`关

联折叠元素

- `.collapse`被折叠元素，需要有被控制的选择器，如 id 或者 class

```
<div class="container">
  <div>
```

```
<!--
```

执行效果必要：

1. data-toggle="collapse" 切换折叠的动作（给对应的 id 元素增加一个叫 show 的类）
2. data-target="#需要切换的 id 名字" 关联 id
- 3.如需要切换两个显示元素 data-target=".相同的类名" 需要使用类来进行两个元素的切换效果

```
-->
```

```
    <button class="btn btn-primary" data-target="#hello1" data-toggle="collapse">折叠
1</button>
```

```
    <button class="btn btn-danger" data-target="#hello2" data-toggle="collapse">折叠
2</button>
```

```
    <button class="btn btn-success" data-target=".mybox" data-toggle="collapse">切换
1/2</button>
```

```
    <div class="mybox collapse" id="hello1">11111111</div>
```

```
    <div class="mybox collapse" id="hello2">2222222</div>
```

```
    </div>
```

```
</div>
```

## 手风琴效果

```
<div class="container">
```

```
    <!-- 手风琴效果最外层必须加 id，手风琴效果必须要把每个显示的元素关联它的最
    顶端元素的 id data-parent="#sfq"-->
```

```
    <!-- 可以使用卡片做手风琴的一组（选项和内容） -->
```

```
    <div id="sfq">
```

```
        <!-- 每张卡是一组 -->
```

```
        <div class="card">
```

```
            <button class="btn btn-primary" data-toggle="collapse" data-target="#a">手风
            琴的第一项点击按钮</button>
```

```
            <div id="a" class="collapse" data-parent="#sfq">第一项内容</div>
```

```
        </div>
```

```
        <div class="card">
```

```
            <button class="btn btn-primary" data-toggle="collapse" data-target="#b">手风
            琴的第二项点击按钮</button>
```

```
            <div id="b" class="collapse" data-parent="#sfq">第二项内容</div>
```

```
        </div>
```

```
        <div class="card">
```

```
            <button class="btn btn-primary" data-toggle="collapse" data-target="#c">手风
            琴的第三项点击按钮</button>
```

```
            <div id="c" class="collapse" data-parent="#sfq">第三项内容</div>
```

```
        </div>
```

```
    </div>
```

```
</div>
```



## 8. 下拉菜单

下拉菜单与折叠类似，需要一个控制元素，关联开合的下拉组

- `.dropdown` 下拉菜单区域
- 控制按钮
  - `.dropdown-toggle` 向下三角
  - `data-toggle="dropdown"` 控制下拉的控件
- 菜单区域
  - `.dropdown-menu` 菜单列表
  - `.dropdown-item` 菜单列表项

```
<div class="container">
  <div class="dropdown">
    <button class="btn btn-danger dropdown-toggle" data-toggle="dropdown">下拉
    菜单</button>
    <div class="dropdown-menu">
      <a href="#" class="dropdown-item">111</a>
      <a href="#" class="dropdown-item">222</a>
      <a href="#" class="dropdown-item">333</a>
    </div>
  </div>
</div>
```

## 9. 列表组

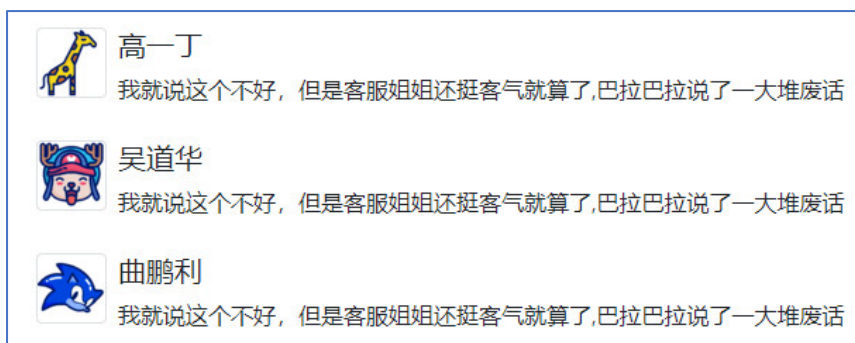
- `.list-group` 列表组 ( 父级 )
- `.list-group-item` 列表项目
- `.list-group-item-action` 列表项目的悬停效果
- `.active` 被选项
- `.list-group-item-primary` 列表项的颜色

曹操
刘备
孙权

```
<div class="container">
  <ul class="list-group">
    <li class="list-group-item list-group-item-danger list-group-item-action active">曹操</li>
    <li class="list-group-item list-group-item-action list-group-item-danger">刘备</li>
    <li class="list-group-item list-group-item-action list-group-item-danger">孙权</li>
  </ul>
</div>
```

## 10.媒体对象

构建高度重复的组件，例如博客评论，推文等。



- `.media` 媒体对象，创建重复模块
- `.media-body` 媒体对象的内容区域
- 直接加入 `<img>` 图片 `.media-body` 的内容会在它一侧展示，可以通过 `<img>` 和 `.media-body` 位置调换来让图片在左还是在右显示
- `.list-unstyled` 使用媒体对象列表可以包裹多个媒体对象

```
<div class="container">
  <ul class="list-unstyled">
    <li class="media">
      
      <div class="media-body">
        <h5 class="mt-2">高一丁</h5>
        <p>我就说这个不好，但是客服姐姐还挺客气,巴拉巴拉说了一大堆废话</p>
      </div>
    </li>
  </ul>
</div>
```

```
        </div>
      </li>
    <li class="media">
      <div class="media-body">
        <h5 class="mt-2">吴道华</h5>
        <p>我就说这个不好，但是客服姐姐还挺客气,巴拉巴拉说了一大堆废话</p>
      </div>
      
    </li>
  </ul>
</div>
```

## 11.导航

### - 普通导航

- `.nav` 导航 (父级的基类)
- `.nav-item` 导航项目 (多用 li)
- `.nav-link active` 导航链接 (一般用于 li 中的 a) 以及 当前的
- `.justify-content-center` 导航的居中对齐
- `.justify-content-end` 导航的右侧逐起

### - 胶囊导航

- `.nav` 导航 (父级的基类)
- `.nav-pills` 胶囊效果 (父级)
- `.nav-item` 导航项目 (多用 li)
- `data-toggle="pill"` 使用胶囊切换 (li)
- `.nav-link active` 导航链接 (一般用于 li 中的 a) 以及 当前的

### - 选项卡

- `.nav` 导航 (父级的基类)
- `.nav-tabs` 导航为选项卡 (父级)

- `.nav-link` tab 链接

- `data-toggle="tab" href="#home"` 具有切换的功能 以及 显示目标的 id , 依靠 id

关联

- `.tab-content` 显示目标的父级

- `.tab-pane` 显示部分隐藏

- `.show` 显示

- `.active` 默认选择

- `id="home"` 显示内容对应 id

```
<div class="container">
  <!-- 普通导航 -->
  <ul class="nav justify-content-end">
    <li class="nav-item"><a href="#" class="nav-link active">首页</a></li>
    <li class="nav-item"><a href="#" class="nav-link">学习用品</a></li>
    <li class="nav-item"><a href="#" class="nav-link">私人订制</a></li>
  </ul>
  <!-- 胶囊导航 -->
  <ul class="nav nav-pills">
    <li class="nav-item">
      <a href="#" class="nav-link active" data-toggle="pill">首页</a>
    </li>
    <li class="nav-item">
      <a href="#" class="nav-link" data-toggle="pill">学习用品</a>
    </li>
    <li class="nav-item">
      <a href="#" class="nav-link" data-toggle="pill">私人订制</a>
    </li>
  </ul>
</div>
```

滑动门 ( tab 面板 )

```
<div class="container">
  <!-- 选项卡分两个区域：选择区、内容区 -->
  <nav>
    <!-- 选择区：放 a 标签的 -->
    <div class="nav nav-tabs">
      <a href="#tab1" class="nav-link active" data-toggle="tab">商品详情</a>
```

```

        <a href="#tab2" class="nav-link" data-toggle="tab">商品评价</a>
    </div>
    <!-- 内容区：a 标签的切换区域 -->
    <div class="tab-content">
        <div class="tab-pane show active" id="tab1">
            
        </div>
        <div class="tab-pane" id="tab2">
            <ul class="list-unstyled">
                <!-- 一组媒体对象 -->
                <li class="media">
                    
                    <div>
                        <h5>宝玉</h5>
                        <p>我买这个东西不是为了需要，而是为了显示自己的实力</p>
                    </div>
                </li>
                <li class="media">
                    
                    <div>
                        <h5>二旺</h5>
                        <p>宝玉买了我就买了</p>
                    </div>
                </li>
                <li class="media">
                    
                    <div>
                        <h5>芬儿</h5>
                        <p>我真的需要一个电脑，这个看着还可以，用用再说</p>
                    </div>
                </li>
            </ul>
        </div>
    </div>
</nav>
</div>

```

## 12.导航条

导航条或者叫导航栏和导航的区别是:

1.导航是横向，导航条可以是横向和纵向。

2.导航的结构 `ul>li>a`,导航条的结构最外层需要有一个包裹元素如 `div,nav`

3.boot 网页的导航，缩小页面看效果，导航条缩小看效果，导航条缩小后可以响应式收缩

导航内容变成三条线,叫做断点分组

- `.navbar`` 导航条外层 (基类)

- `.navbar-dark bg-dark`` 深色主题

- `.navbar-light bg-light`` 浅色主题

- logo 和项目介绍

  - `.navbar-brand`` 用于 logo 和项目名的外层

- 响应式

  - `.navbar`` 导航条外层 (基类)

  - `.navbar-expand{-sm|-md|-lg|-xl}`` 菜单为竖向排列没加入该类变为横向，同样用于响应式(必要)

- 可隐藏的导航菜单

  - `.collapse .navbar-collapse`` 断点分组和隐藏导航栏内容，与缩小菜单相关，在导航父级设置

  - 如果需要响应式隐藏导航，需要设置 id 与缩小菜单关联，如：``id=navMenu``

- 缩小菜单

  - 一般使用 `<button>` 作为缩小菜单

  - `.navbar-toggler`` 缩小菜单 (缩小时展示的菜单)

  - ``data-toggle="collapse" data-target="#navMenu`` 需要切换属性以及关联可切换的目标菜单,前提必须有对应 id 的菜单才生效

  - 一般使用 `<span>` 设置缩小菜单的图标 `.navbar-toggler-icon``

```

<style>
  /* md 为分界线，其实控制的小于 md 以下的宽度 */
  @media (max-width:768px) {
    .xz-search {
      margin: 10px 0;
      width: 100% !important;
    }
  }
  .xz-search { width: 55%; }
  .xz-search input {
    outline: none;
    background-image: url(/img/header/search.png);
    background-repeat: no-repeat;
    background-position: right;
    background-size: auto 100%;
  }
  .xz-menu a:nth-child(n+4) {
    font-size: 12px;
    color: #808080;
    line-height: 28px;
  }
</style>

```

```

<body>
  <div class="navbar-part">
    <div class="container">
      <div class="navbar navbar-light bg-light navbar-expand-lg">
        <!-- logo 区 -->
        <a href="#" class="navbar-brand"></a>
        <!-- 小菜单 lg -->
        <button class="navbar-toggler" data-toggle="collapse" data-target="#xzmenu">
          <span class="navbar-toggler-icon"></span>
        </button>
        <!-- 大菜单 lg 以上 -->
        <div id="xzmenu" class="collapse navbar-collapse justify-content-between">
          <!-- 搜索框 -->
          <div class="xz-search">
            <input type="text" class="border-0 w-100">
          </div>
          <!-- 菜单列表 -->
          <div class="xz-menu nav">
            <a href="#" class="nav-item nav-link"></a>
            <a href="#" class="nav-item nav-link"></a>
            <a href="#" class="nav-item nav-link"></a>

```

```
        <a href="#" class="nav-item nav-link">注册</a>
        <a href="#" class="nav-item nav-link">登录</a>
      </div>
    </div>
  </div>
</div>
</div>
</body>
```

### 13.分页

- `.pagination` 分页的父级效果



- `.page-item` 分页项

- `.active` 被激活的项目（因为没有js的参与所以不能切换 active）

- `.disabled` 禁用的，比如第一页时的上一页，和最后页时的下一下

- `.page-link` 分页项中的链接

- 对齐方式`.justify-content-center`居中，`.justify-content-end`居右

```
<div class="container">
  <ul class="pagination">
    <li class="page-item disabled">
      <a href="#" class="page-link">上一页</a>
    </li>
    <li class="page-item active">
      <a href="#" class="page-link">1</a>
    </li>
    <li class="page-item">
      <a href="#" class="page-link">2</a>
    </li>
    <li class="page-item">
      <a href="#" class="page-link">3</a>
    </li>
    <li class="page-item">
      <a href="#" class="page-link">下一页</a>
    </li>
  </ul>
</div>
```



## 14.巨幕

巨幕就是某个图片或者某个颜色扩展至整个视口，主要用于大型活动和广告的宣传（非重点，因为有圆角所以自定义背景图基本不用）

- `.jumbotron` 超大巨幕，可以不写在`.container`或者`.container-fluid`中
- `.jumbotron`可以包裹`.container`或者`.container-fluid`
- 因`.jumbotron`有圆角，因此，加`.jumbotron-fluid` 可以去除巨幕的圆角

```
<div class="jumbotron jumbotron-fluid">
  <div class="container">
    <h1 class="display-4">巨目的标题</h1>
    <p>随便写点文字巴拉巴拉吧拉</p>
  </div>
</div>
```

## 15.轮播

轮播图比较常见，效果也很多，而他的控制往往通过js 来做。

- 父级
  - `.carousel`定位
  - `.slide`平滑的过渡效果，不需要可以删掉
  - `data-ride="carousel"` 负责利用定时器切换内部项目（js），不需要可以删掉
- 展示图
  - `.carousel-inner` 展示图大区域位置
  - `.carousel-item` 展示区域中的展示项目
  - `.active` 被激活的展示项目
- 左右控制按钮
  - 左右切换使用 a 标签，利用`href="#banner"`来控制轮播

- `.carousel-control-prev` 上一个 ( 在轮播中定位 )
- `.carousel-control-next` 下一个 ( 在轮播中定位 )
- `.carousel-control-prev-icon`和`.carousel-control-next-icon` 可点击的按钮
- `data-slide="prev"`向左控制按钮
- `data-slide="next"`向右控制按钮
- 指示符
  - `.carousel-indicators`指示符区域，在轮播中的位置
  - `data-slide-to="0"` 指示符项目控制的对应展示项目的位置，从 0 开始
  - `.active` 指示符激活
  - `data-target="#banner"` 指示符需要控制轮播
  - 在 boot 中`.carousel-indicators li`设置了控制符的样式
  - `.carousel-indicators .active`控制 active 的样式

```

<style>
  .xz-kz {
    width: 50px;
    height: 100px;
    top: 50%;
    transform: translateY(-50%);
    border-radius: 5px;
  }
  .xz-kz:hover {background-color: rgba(0,0,0,0.1);}
  .xz-zsf li {
    width: 10px;
    height: 10px;
    box-shadow: 0 0 2px 1px #ccc;
    border-radius: 50%;
  }
  .xz-zsf .active {
    background-color: #19a8f7;
  }
</style>
<body>

```

```
<div class="container">
  <!-- 轮播的最外层父级 -->
  <div class="carousel slide" data-ride="carousel" id="banner">
    <!-- 轮播放图片的区域 -->
    <div class="carousel-inner">
      <div class="carousel-item active">
        
      </div>
      <div class="carousel-item">
        
      </div>
      <div class="carousel-item">
        
      </div>
      <div class="carousel-item">
        
      </div>
    </div>
    <!-- 左右切换按钮 -->
    <a href="#banner" class="carousel-control-prev xz-kz" data-slide="prev">
      <span class="carousel-control-prev-icon"></span>
    </a>
    <a href="#banner" class="carousel-control-next xz-kz" data-slide="next">
      <span class="carousel-control-next-icon"></span>
    </a>
    <!-- 指示符 -->
    <ul class="carousel-indicators xz-zsf">
      <li data-target="#banner" data-slide-to="0" class="active"></li>
      <li data-target="#banner" data-slide-to="1"></li>
      <li data-target="#banner" data-slide-to="2"></li>
      <li data-target="#banner" data-slide-to="3"></li>
    </ul>
  </div>
</div>
```