

优极限

“极限教育，挑战极限”

www.yjxxt.com

极限教育，挑战极限。优极限是一个让 95% 的学生年薪过 18 万的岗前培训公司，让我们的学员具备优秀的互联网技术和职业素养，勇攀高薪，挑战极限。公司位于上海浦东，拥有两大校区，共万余平。累计培训学员超 3 万名。我们的训练营就业平均月薪 19000，最高年薪 50 万。

核心理念：让学员学会学习，拥有解决问题的能力，拿到高薪职场的钥匙。

项目驱动式团队协作、一对一服务、前瞻性思维、教练式培养模型-培养你成为就业明星。首创的老学员项目联盟给学员充分的项目、技术支撑，利用优极限平台这根杠杆，不断挑战极限，勇攀高薪，开挂人生。

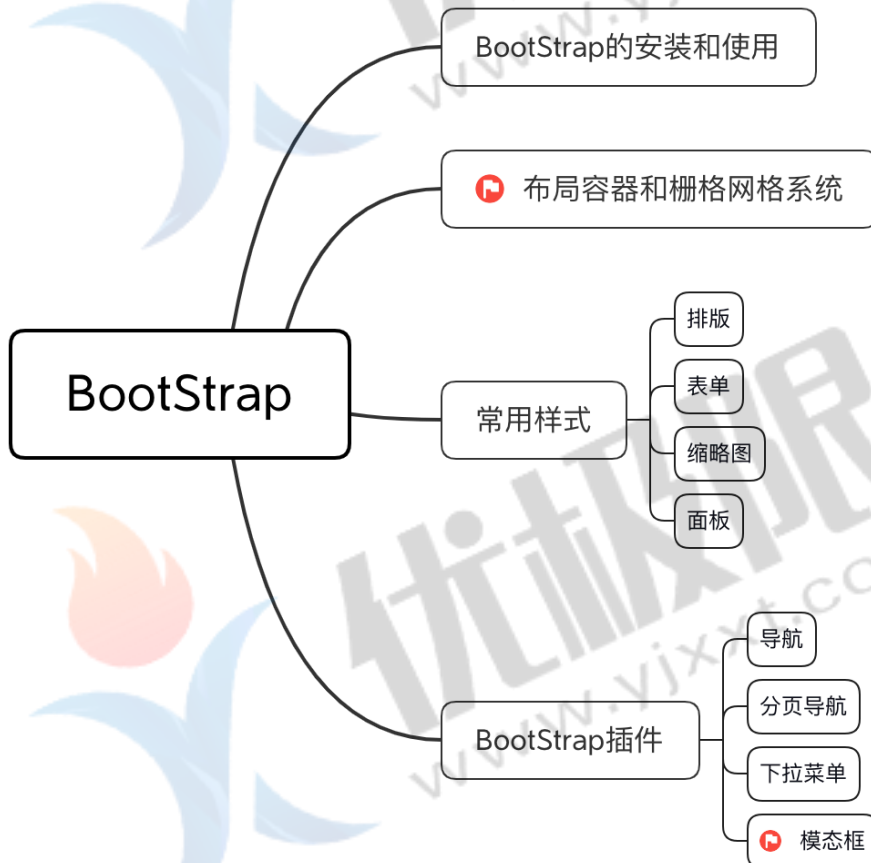
扫码关注优极限微信公众号：

（获取最新技术相关资讯及更多源码笔记）



Bootstrap

1. 主要内容



2. Bootstrap 的安装和使用

2.1. Bootstrap 介绍

官网：<http://getbootstrap.com/>

中文网：<http://www.bootcss.com/>

Bootstrap 是一套现成的 CSS 样式集合（做得还是很友好的）。是两个推特的员工干出来的。

Bootstrap 是最受欢迎的 HTML、CSS 和 JS 框架，用于开发响应式布局、移动设备优先的 WEB 项目。

2011年, twitter 的"一小撮"工程师为了提高他们内部的分析和管理能力, 用业余时间为他们产品构建了一套易用、优雅、灵活、可扩展的前端工具集 -- Bootstrap。Bootstrap 由 MARK OTTO 和 Jacob Thornton 所设计和建立, 在 github上开源之后, 迅速成为该站上最多人 watch&fork 的项目。大量工程师踊跃为该项目贡献代码, 社区惊人地活跃, 代码版本进化非常快速, 官方文档质量极其高(可以说是优雅), 同时涌现了许多基于Bootstrap 建设的网站: 界面清新、简洁;要素排版利落大方。

Bootstrap特别适合那种没有设计师的团队(甚至说没有前端的团队), 可以快速的出一个网页。

2.2. Bootstrap 特点

1. 简洁、直观、强悍的前端开发框架, html、css、javascript 工具集, 让 web 开发更速、简单。
2. 基于html5、css3的bootstrap, 具有大量的诱人特性: 友好的学习曲线, 卓越的兼容性, 响应式设计, 12列格网, 样式向导文档。
3. 自定义JQuery 插件, 完整的类库, bootstrap3 基于Less, bootstrap4 基于 Sass 的 CSS 预处理技术
4. Bootstrap 响应式布局设计, 让一个网站可以兼容不同分辨率的设备。Bootstrap 响应式布局设计, 给用户提供更好的视觉使用体验。
5. 丰富的组件

2.3. 下载与使用

1. 下载: <http://v3.bootcss.com/getting-started/>

2. 下载完成后

拷贝 dist/css 中的 bootstrap.min.css 到项目 css 中

拷贝 dist/js 中的 bootstrap.min.js 到项目的 js 中

3. 下载 jquery.js

<http://jquery.com/>

4. 在 html 中模板为:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <!--使用X-UA-Compatible来设置IE浏览器兼容模式 最新的渲染模式-->
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <!--
      viewport表示用户是否可以缩放页面;
      width指定视区的逻辑宽度;
      device-width指示视区宽度应为设备的屏幕宽度;
      initial-scale指令用于设置Web页面的初始缩放比例
      initial-scale=1则将显示未经缩放的web文档
    -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<title>Bootstrap的HTML标准模板</title>
<!-- 载入Bootstrap 的css -->
<link href="css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
  <h1>Hello, world!</h1>

  <!-- 如果要使用Bootstrap的js插件, 必须先调入jQuery -->
  <script src="js/jquery-3.4.1.js"></script>
  <!-- 包括所有bootstrap的js插件或者可以根据需要使用的js插件调用 -->
  <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

注意:

目前暂时不使用 jquery 的插件 可以不用引入 js 文件,这里是为了保证模板的完整性。

说明:

- viewport 标记用于指定用户是否可以缩放Web页面
- width 和 height 指令分别指定视区的逻辑宽度和高度。他们的值要么是以像素为单位的数字, 要么是一个特殊的标记符号。
- width 指令使用 device-width 标记可以指示视区宽度应为设备的屏幕宽度。
- height 指令使用 device-height 标记指示视区高度为设备的屏幕高度。
- initial-scale 指令用于设置Web页面的初始缩放比例。默认的初始缩放比例值因智能手机浏览器的不同而有所差异。通常情况下设备会在浏览器中呈现出整个Web页面, 设为1.0则将显示未经缩放的Web文档。

5. 参考API

<http://v3.bootcss.com/css/>

3. 布局容器和栅格网格系统

3.1. 布局容器

1、.container 类用于固定宽度并支持响应式布局的容器。

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

2、.container-fluid类用于100% 宽度, 占据全部视口 (viewport) 的容器。

```
<div class="container-fluid">
  ...
</div>
```

3.2. 栅格网格系统

Bootstrap 提供了一套响应式、移动设备优先的流式栅格系统，随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多12列。栅格系统用于通过一系列的行（row）与列（column）的组合来创建页面布局，你的内容就可以放入这些创建好的布局中。

网格系统的实现原理非常简单，仅仅是通过定义容器大小，平分12份(也有平分成24份或32份，但12份是最常见的)，再调整内外边距，最后结合媒体查询，就制作出了强大的响应式网格系统。Bootstrap 框架中的网格系统就是将容器平分成12份。

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

注意: 网格系统必须使用到css

container、row、xs (xsmall phones), sm (small tablets), md (middle desktops), lg (larger desktops) 即: 超小屏(自动), 小屏(750px), 中屏(970px)和大屏(1170px)

数据行(.row)必须包含在容器（.container）中，以便为其赋予合适的对齐方式和内距(padding)。

在行(.row)中可以添加列(.column), 只有列（column）才可以作为行容器(.row)的直接子元素，但列数之和不能超过平分的总列数，比如12。如果大于12,则自动换到下一行。

具体内容应当放置在列容器（column）之内

```
<div class="container">
  <div class="row">
    <div class="col-md-4">4列</div>
    <div class="col-md-8">8列</div>
  </div>
</div>
```

栅格参数

通过下表可以详细查看 Bootstrap 的栅格系统是如何在多种屏幕设备上工作的。

	超小屏幕 手机 (<768px)	小屏幕 平板 (≥768px)	中等屏幕 桌面显示器 (≥992px)	大屏幕 大桌面显示器 (≥1200px)
栅格系统行为	总是水平排列			
.container 最大宽度	None (自动)	750px	970px	1170px
类前缀	.col-xs-	.col-sm-	.col-md-	.col-lg-
列 (column) 数	12			
最大列 (column) 宽	自动	~62px	~81px	~97px
槽 (gutter) 宽	30px (每列左右均有 15px)			
可嵌套	是			
偏移 (Offsets)	是			
列排序	是			

3.2.1. 列组合

列组合简单理解就是更改数字来合并列（原则：列总和数不能超12，大于12,则自动换到下一行。），有点类似于表格的colspan属性。

```
<div class="container">
  <div class="row">
    <div class="col-md-4">4列</div>
    <div class="col-md-8">8列</div>
  </div>
  <div class="row">
    <div class="col-md-2">2列</div>
    <div class="col-md-10">10列</div>
  </div>
</div>
```

3.2.2. 列偏移

如果我们不希望相邻的两个列紧靠在一起，但又不想使用margin或者其他的技术手段来。这个时候就可以使用列偏移 (offset) 功能来实现。使用列偏移也非常简单，只需要在列元素上添加类名"col-md-offset-*" (其中星号代表要偏移的列组合数)，那么具有这个类名的列就会向右偏移。例如，你在列元素上添加"col-md-offset-8"，表示该列向右移动8个列的宽度 (要保证列与偏移列的总数不超过12，不然会致列断行 | 换行显示)。


```

<div class="container">
  <div class="row">
    <div class="col-md-1">1列</div>
    <div class="col-md-1">2列</div>
    <div class="col-md-1 col-md-offset-8">11列</div>
    <div class="col-md-1">12列</div>
  </div>
</div>

```

3.2.3. 列排序

列排序其实就是改变列的方向，就是改变左右浮动，并且设置浮动的距离。在Bootstrap框架的网格系统中是通过添加类名 `col-md-push-*` 和 `col-md-pull-*`（其中星号代表移动的列组合数）。往前pull，往后push。

```

<div class="container">
  <div class="row">
    <div class="col-md-1 col-md-push-10">1列</div>
    <div class="col-md-1 col-md-pull-1">2列</div>
  </div>
</div>

```

3.2.4. 列嵌套

Bootstrap框架的网格系统还支持列的嵌套。你可以在一个列中添加一个或者多个行（row）容器，然后在这个行容器中插入列。

```

<div class="container">
  <div class="row">
    <div class="col-md-2">
      我的里面嵌套了一个网格
      <div class="row">
        <div class="col-md-9">9</div>
        <div class="col-md-3">3</div>
      </div>
    </div>
    <div class="col-md-10">我的里面嵌套了一个网格
      <div class="row">
        <div class="col-md-10">10</div>
        <div class="col-md-2">2</div>
      </div>
    </div>
  </div>
</div>

```

4. 常用样式

4.1. 排版

4.1.1. 标题

Bootstrap和普通的HTML页面一样，定义标题都是使用标签

到

0.1.1.0.1. ,只不过Bootstrap覆盖了其默认的风格，使用其在所有浏览器下显示的效果一样。为了让非标题元素和标题使用相同的风格，还特意定义了.h1~.h6六个类名。同时后面可以紧跟着一行小的副标题或使用.small

```
<h1>h1. Bootstrap heading<small>副标题</small></h1>  
<div class="h1">Bootstrap标题1<span class="small">副标题</span></div>
```

0.1.2. 段落

段落是排版中另一个重要元素之一。通过.lead 来突出强调内容(其作用就是增大文本字号，加粗文本，而且对行高和margin也做相应的处理。可以使用以下标签给文本做突出样式处理：

:小号字

:加粗

:斜体

```
<p class="lead"><small>以后的</small><b>你</b>会<i>感谢</i>现在<em>努力</em>的<strong>你</strong></p>
```

0.1.3. 强调

定义了一套类名，这里称其为强调类名，这些强调类都是通过颜色来表示强调，具本说明如下：

.text-muted：提示，使用浅灰色（#999）

.text-primary：主要，使用蓝色（#428bca）

.text-success：成功，使用浅绿色（#3c763d）

.text-info：通知信息，使用浅蓝色（#31708f）

.text-warning：警告，使用黄色（#8a6d3b）

.text-danger：危险，使用褐色（#a94442）


```
<div class="text-muted">提示效果</div>
<div class="text-primary">主要效果</div>
<div class="text-success">成功效果</div>
<div class="text-info">信息效果</div>
<div class="text-warning">警告效果</div>
<div class="text-danger">危险效果</div>
```

0.1.4. 对齐效果

在CSS中常常使用text-align来实现文本的对齐风格的设置。

其中主要有四种风格：

左对齐，取值left；

居中对齐，取值center；

右对齐，取值right；

两端对齐，取值justify。

为了简化操作，方便使用，Bootstrap通过定义四个类名来控制文本的对齐风格：.text-left：左对齐 .text-center：居中对齐 .text-right：右对齐 .text-justify：两端对齐。

```
<p class="text-left">我居左</p>
<p class="text-center">我居中</p>
<p class="text-right">我居右</p>
<p class="text-justify">网格系统的实现原理非常简单，仅仅是通过定义容器大小，平分12份(也有平分成24份或32份，但12份是最常见的)，再调整内外边距，最后结合媒体查询，就制作出了强大的响应式网格系统。Bootstrap框架中的网格系统就是将容器平分成12份</p>
```

0.1.5. 列表

在HTML文档中，列表结构主要有三种：

无序列表(

- ...

)

有序列表(

1. ...

)

定义列表(

...

...

)

0.1.5.1. 去点列表

class="list-unstyled"

```
<ul class="list-unstyled">
  <li>无序项目列表一</li>
  <li>无序项目列表二</li>
</ul>
```

0.1.5.2. 内联列表

`class="list-inline"`，把垂直列表换成水平列表，而且去掉项目符号（编号），保持水平显示。也可以说内联列表就是为制作水平导航而生。

```
<ul class="list-inline">
  <li>首页</li>
  <li>java学院</li>
  <li>在线课堂</li>
</ul>
```

0.1.5.3. 定义列表

在原有的基础加入了一些样式，使用样式 `class="dl-horizontal"` 制作水平定义列表：当标题宽度超过160px时，将会显示三个省略号。

```
<dl>
  <dt>HTML</dt>
  <dd>超文本标记语言</dd>
  <dt>CSS</dt>
  <dd>层叠样式表是一种样式表语言</dd>
</dl>
<dl class="dl-horizontal">
  <dt>HTML 超文本标记语言</dt>
  <dd>HTML称为超文本标记语言，是一种标识性的语言。</dd>
  <dt>测试标题不能超过160px的宽度，否则2个点</dt>
  <dd>我在写一个水平定义列表的效果，我在写一个水平定义列表的效果。</dd>
</dl>
```

0.1.6. 代码

一般在个人博客上使用的较为频繁，用于显示代码的风格。在Bootstrap主要提供了三种代码风格：

- (1) 使用 `<code>` 来显示单行内联代码
- (2) 使用 `<pre>` 来显示多行块代码

来显示多行块代码

样式：`pre-scrollable` (height,max-height高度固定,为340px,超过存在滚动条)

- (3) 使用 `<code>` 来显示用户输入代码,如快捷键

0.1.6.1. 单行内联代码

```
<code>this is a simple code</code>
```

0.1.6.2. 快捷键

```
<p>使用<code>ctrl+s</code>保存</p>
```

0.1.6.3. 多行块代码

```
<!-- 代码会保留原本的格式，包括空格和换行 -->
<pre>
public class HelloWorld {
    public static void main(String[] args){
        System.out.println("helloworld...");
    }
}
</pre>
<!--
    显示html标签的代码需要适应字符实体
    小于号(<)要使用硬编码"&lt;"来替代，大于号(>)使用"&gt;"来替代
-->
<pre>
    &lt;ul&gt;
        &lt;li&gt;测试实体符&lt;/li&gt;
    &lt;/ul&gt;
</pre>
<!-- 当高度超过，会存在滚动条 -->
<pre class="pre-scrollable">
    <ol>
        <li>.....</li>
        <li>.....</li>
        <li>.....</li>
        <li>.....</li>
        <li>.....</li>
        <li>.....</li>
        <li>.....</li>
        <li>.....</li>
        <li>.....</li>
        <li>.....</li>
        <li>.....</li>
        <li>.....</li>
    </ol>
</pre>
```

0.1.7. 表格

0.1.7.1. 表格样式

Bootstrap为表格提供了1种基础样式和4种附加样式以及1个支持响应式的表格。在使用**Bootstrap**的表格过程中，只需要添加对应的类名就可以得到不同的表格风格：

基础样式

- 1) **.table**：基础表格

附加样式

- 1) **.table-striped**：斑马线表格
- 2) **.table-bordered**：带边框的表格
- 3) **.table-hover**：鼠标悬停高亮的表格
- 4) **.table-condensed**：紧凑型表格，单元格没内距或者内距较其他表格的内距小

0.1.7.2. tr、th、td样式

提供了五种不同的类名，每种类名控制了行的不同背景颜色

类	描述	实例
<code>.active</code>	将悬停的颜色应用在行或者单元格上	<code>#f5f5f5</code>
<code>.success</code>	表示成功的操作	<code>#dff0d8</code>
<code>.info</code>	表示信息变化的操作	<code>#d9edf7</code>
<code>.warning</code>	表示一个警告的操作	<code>#fcf8e3</code>
<code>.danger</code>	表示一个危险的操作	<code>#f2dede</code>

```

<table class="table table-bordered table-hover">
  <tr class="active">
    <th>JavaSE</th>
    <th>数据库</th>
    <th>JavaScript</th>
  </tr>
  <tr class="danger">
    <td>面向对象</td>
    <td>oracle</td>
    <td>json</td>
  </tr>
  <tr class="success">
    <td>数组</td>
    <td>mysql</td>
    <td>ajax</td>
  </tr>
</table>

```

0.2. 表单

表单主要功能是用来与用户做交流的一个网页控件，良好的表单设计能够让网页与用户更好的沟通。表单中常见的元素主要包括：文本输入框、下拉选择框、单选按钮、复选按钮、文本域和按钮等。

0.2.1. 表单控件

`.form-control` `.input-lg` (较大) `.input-sm` (较小)

0.2.1.1. 输入框 text

`.form-control`

```

<div class="col-sm-3">
  <input type="text" name="" id="" class="form-control" />
  <input type="text" name="" id="" class="form-control input-lg" />
  <input type="text" name="" id="" class="form-control input-sm" />
</div>

```

0.2.1.2. 下拉选择框 select

多行选择设置: `multiple="multiple"`

```

<div class="col-sm-3">
  <select class="form-control">
    <option>北京</option>
    <option>上海</option>
    <option>深圳</option>
  </select>
  <select class="form-control" multiple="multiple">
    <option>北京</option>
    <option>上海</option>
    <option>深圳</option>
  </select>
</div>

```

0.2.1.3. 文本域 textarea

```

<div class="col-sm-3">
  <textarea class="form-control" rows="3"></textarea>
</div>

```

0.2.1.4. 复选框 checkbox

垂直显示: `.checkbox`

水平显示: `.checkbox-inline`

```

<!-- 垂直显示 -->
<div>
  <div class="checkbox">
    <label><input type="checkbox" >游戏</label>
  </div>
  <div class="checkbox">
    <label><input type="checkbox" >学习</label>
  </div>
</div>
<!-- 水平显示 -->
<div>
  <label class="checkbox-inline">
    <input type="checkbox" >游戏
  </label>
  <label class="checkbox-inline">
    <input type="checkbox" >学习
  </label>
</div>

```

0.2.1.5. 单选框 radio

垂直显示: `.radio`

水平显示: `.radio-inline`

```

<!-- 垂直显示 -->
<div>
  <div class="radio">
    <label><input type="radio" >男</label>
  </div>
  <div class="radio">
    <label><input type="radio" >女</label>
  </div>
</div>

```

```

    </div>
</div>
<!-- 水平显示 -->
<div>
    <label class="radio-inline">
        <input type="radio" >男
    </label>
    <label class="radio-inline">
        <input type="radio" >女
    </label>
</div>

```

0.2.1.6. 按钮

1) 使用 button 实现

基础样式: btn

```
<button class="btn">按钮</button>
```

附加样式: btn-primary btn-info btn-success btn-warning btn-danger btn-link btn-default

```

<button class="btn btn-danger">按钮</button>
<button class="btn btn-primary">按钮</button>
<button class="btn btn-info">按钮</button>
<button class="btn btn-success">按钮</button>
<button class="btn btn-default">按钮</button>
<button class="btn btn-warning">按钮</button>
<button class="btn btn-link">按钮</button>

```

2) 多标签支持: 使用 a div 等制作按钮

```

<a href="#" class="btn btn-info">a标签按钮</a>
<span class="btn btn-success">span标签按钮</span>
<div class="btn btn-warning">div标签按钮</div>

```

3) 按钮大小

使用 .btn-lg、.btn-sm 或 .btn-xs 就可以获得不同尺寸的按钮

```

<button class="btn btn-primary btn-xs">超小按钮.btn-xs</button>
<button class="btn btn-primary btn-sm">小型按钮.btn-sm</button>
<button class="btn btn-primary">正常按钮</button>
<button class="btn btn-primary btn-lg">大型按钮.btn-lg</button>

```

4) 按钮禁用

方法1: 在标签中添加disabled属性

```
<button class="btn btn-danger" disabled="disabled">禁用按钮</button>
```

方法2: 在元素标签中添加类名"disabled"

```
<button class="btn btn-danger disabled">禁用按钮</button>
```


在class属性中添加disabled只是样式上禁用了，并不是真正的禁用了此按钮！

0.2.2. 表单布局

基本的表单结构是 Bootstrap 自带的，个别的表单控件自动接收一些全局样式。下面列出了创建基本表单的步骤：

- 向父元素添加 `role="form"`。
- 把标签和控件放在一个带有 `class .form-group` 的 `div` 中。这是获取最佳间距所必需的。
- 向所有的文本元素

`` 和 `<select>`
`` 添加 `class = "`

```
<form class="form-horizontal" role="form">
```

```
<div class="form-group">
```

```
<label for="email" class="control-label col-sm-2">邮箱</label>
```

```
<div class="col-sm-10">
```

```
<input type="email" class="form-control" placeholder="请输入邮箱"/>
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="pwd" class="control-label col-sm-2">密码</label>
```

```
<div class="col-sm-10">
```

```
<input type="pwd" class="form-control" placeholder="请输入密码" />
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<div class="col-sm-offset-2">
```

```
<div class="checkbox">
```

```
<label>
```

```
<input type="checkbox" />记住密码
```

```
</label>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
  <div class="col-sm-offset-2 col-sm-10">
```

```
    <button class="btn btn-default">提交</button>
```

```
  </div>
```

```
</div>
```

```
</form>
```

内联表单

将表单的控件都在一行内显示form-inline

注意label不会显示，存在的意义：如果没有为输入控件设置label标签，屏幕阅读器将无法正确识别。

```
<form class="form-inline">
  <div class="form-group">
    <label for="email" >邮箱</label>
    <input type="email" class="form-control" placeholder="请输入邮箱"/>
  </div>
  <div class="form-group">
    <label for="pwd" >密码</label>
    <input type="pwd" class="form-control" placeholder="请输入密码" />
  </div>
  <div class="form-group checkbox">
    <label><input type="checkbox" />记住密码</label>
  </div>
  <div class="form-group">
    <button class="btn btn-default">提交</button>
  </div>
</form>
```

缩略图

缩略图在电商类的网站很常见，最常用的地方就是产品列表页面。缩略图的实现是配合网格系统一起使用。同时还可以让缩略图配合标题、描述内容，按钮等。

```
<div class="container">
  <div class="row">
    <div class="col-md-3">
      <div class="thumbnail">
        
        <h3>高圆圆</h3>
        <p>出生于北京市，中国内地影视女演员、模特。</p>
      </div>
    </div>
  </div>
</div>
```

```
<button class="btn btn-default">
    <span class="glyphicon glyphicon-heart"></span>喜欢</button>
<button class="btn btn-info">
    <span class="glyphicon glyphicon-pencil"></span>评论
</button>
</div>
</div>
</div>
</div>
```

面板

默认的 `.panel` 组件所做的只是设置基本的边框 (`border`) 和内补 (`padding`) 来包含内容。

`.panel-default`: 默认样式

`.panel-heading`: 面板头

`.panel-body`: 面板主体内容

```
<div class="panel panel-success">
  <div class="panel-heading">
    .....
  </div>
  <div class="panel-body">
    .....
  </div>
</div>
```

Bootstrap 插件

导航

使用下拉与按钮组合可以制作导航

要点:

1、基本样式: .nav 与 “nav-tabs”、“nav-pills”组合制作导航

2、分类:

1)、标签型 (nav-tabs) 导航

2)、胶囊形(nav-pills)导航

3)、堆栈(nav-stacked)导航

4)、自适应(nav-justified)导航

5)、面包屑式(breadcrumb)导航 , 单独使用样式, 不与nav一起使用, 直接加入到ol、ul中即可, 一般用于导航, 主要是起的作用是告诉用户现在所处页面的位置 (当前位置)

3、状态:

1)、选中状态 active 样式

2)、禁用状态: disable

4、二级菜单

标签式导航

<p>标签式的导航菜单</p>

```
<ul class="nav nav-tabs">
  <li class="active"><a href="#">Home</a></li>
  <li><a href="#">SVN</a></li>
  <li><a href="#">iOS</a></li>
  <li><a href="#">VB.Net</a></li>
  <li><a href="#">Java</a></li>
  <li><a href="#">PHP</a></li>
</ul>
```

胶囊式导航

<p>基本的胶囊式导航菜单</p>

```
<ul class="nav nav-pills">
  <li class="active"><a href="#">Home</a></li>
  <li><a href="#">SVN</a></li>
  <li><a href="#">iOS</a></li>
  <li><a href="#">VB.Net</a></li>
  <li><a href="#">Java</a></li>
  <li><a href="#">PHP</a></li>
</ul>
```

分页导航

分页随处可见, 分为页码导航和翻页导航

页码导航: ul 标签上加 pagination [pagination-lg | pagination-sm]

翻页导航: ul 标签上加 pager

分页

```
<ul class="pagination">
  <li><a href="#">&laquo;</a></li>
  <li><a href="#">1</a></li>
  <li><a href="#">2</a></li>
  <li><a href="#">3</a></li>
  <li><a href="#">4</a></li>
  <li><a href="#">5</a></li>
  <li><a href="#">&raquo;</a></li>
</ul>
```

翻页

```
<ul class="pager">
  <li><a href="#">Previous</a></li>
  <li><a href="#">Next</a></li>
</ul>
```

下拉菜单

在使用Bootstrap框架的下拉菜单时，必须使用两个js

```
<!-- 如果要使用Bootstrap的js插件，必须先调入jQuery -->
<script src="js/jquery-3.4.1.js"></script>
<!-- 包括所有bootstrap的js插件或者可以根据需要使用的js插件调用 -->
<script src="js/bootstrap.min.js"></script>
```

要点：

1、使用一个类名为dropdown 或btn-group的div 包裹整个下拉框：

```
<div class="dropdown"></div>
```

2、默认向下dropdown，向上弹起加入 . dropup 即可

3、使用button作为父菜单，使用类名： dropdown-toggle 和自定义data-toggle属性

```
<button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown"></button>
```

4、在button中 使用font 制作下拉箭头

```
<span class="caret"></span>
```

5、下拉菜单项使用一个ul列表，并且定义一个类名为“dropdown-menu

6、分组分割线：添加类名“divider”来实现添加下拉分隔线的功能

7、分组标题：li 添加类名 “dropdown-header” 来实现分组的功能

8、对齐方式：

1)、dropdown-menu-left 左对齐 默认样式

2)、dropdown-menu-right 右对齐

9、激活状态 (.active) 和禁用状态 (.disabled)

```
<!--使用一个类名为dropdown，默认向下dropdown，向上弹起加入 . dropup 即可-->
<div class="dropdown ">
```

```

<!--使用button作为父菜单，使用类名：dropdown-toggle 和自定义data-toggle属性-->
<button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown">
    喜欢频道 <span class="caret"></span><!-- 下拉箭头-->
</button>
<!-- 下拉菜单项使用一个ul列表，并且定义一个类名为“dropdown-menu-->
<ul class="dropdown-menu"> <!--dropdown-menu-right右对齐-->
    <!--分组标题：li 添加类名 “dropdown-header” 来实现分组的功能-->
    <li class="dropdown-header">----科普----</li>
    <li>
        <a href="#">人与自然</a>
    </li>
    <!--分组分割线：<li>添加类名“divider”来实现添加下拉分隔线的功能-->
    <li class="divider"></li>
    <li class="dropdown-header">----搞笑----</li>
    <li>
        <a href="#">欢乐喜剧人</a>
    </li>
    <li>
        <a href="#">快乐大本营</a>
    </li>
    <li class="divider"></li>
    <li class="disabled"> <!--禁用状态-->
        <a href="#">生活大爆炸</a>
    </li>
</ul>
</div>

```

模态框

模态框 (Modal) 是覆盖在父窗体上的子窗体。通常，目的是显示来自一个单独的源的内容，可以在不离开父窗体的情况下有一些互动。子窗体可提供信息、交互等。

用法

1. 通过 data 属性：在控制器元素（比如按钮或者链接）上设置属性 data-toggle="modal"，同时设置 data-target="#identifier" 或 href="#identifier" 来指定要切换的特定的模态框（带有 id="identifier"）。
2. 通过 JavaScript：使用这种技术，可以通过 JavaScript 来调用带有 id="identifier" 的模态框：

```
$('#identifier').modal(options);
```

实例

```

<h2>创建模态框 (Modal) </h2>
<!-- 按钮触发模态框 -->
<button class="btn btn-primary btn-lg" data-toggle="modal" data-target="#myModal">
    开始演示模态框
</button>

```



```
<!-- 模态框 (Modal) -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-
hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;
      </button>

      <h4 class="modal-title" id="myModalLabel">模态框 (Modal) 标题</h4>
    </div>
    <div class="modal-body">在这里添加一些文本</div>
    <div class="modal-footer">
      <button type="button" class="btn btn-default" data-dismiss="modal">关闭</button>
      <button type="button" class="btn btn-primary">提交更改</button>
    </div>
  </div><!-- /.modal-content -->
</div><!-- /.modal -->
</div>
```

方法

方法	描述	实例
Toggle: .modal('toggle')	手动切换模态框。	<code>\$('#identifier').modal('toggle');</code>
Show: .modal('show')	手动打开模态框。	<code>\$('#identifier').modal('show');</code>
Hide: .modal('hide')	手动隐藏模态框。	<code>\$('#identifier').modal('hide');</code>