

Bootstrap 学习笔记

概述

- 由 Twitter 公司两名技术工程师开发的
- 基于 HTML、CSS、JS 的开源框架
- 代码简洁、视觉优美，可用于快速简单地构建基于 PC 及移动端设备的 Web 页面需求。
- 响应式布局：兼容 PC、手机、PAD 端
- 国内网站：<http://www.bootcss.com/>
- 缺点：不方便修改样式，对细节的不好把握，而且 Bootstrap 是由 Twitter 的开发者开发来系统化他们网站应用的样式。当你的网站应用的样式和他们不一样，这意味着你不得不覆盖掉他们的样式来。大多数网站的样式并不像 Twitter 那样。因此，如果他们安装 Bootstrap，他们会覆盖掉很多样式。

六大特点

1. 跨设备、跨浏览器
2. 响应式布局
3. 提供全面组件：导航、标签。。。.
4. 内置 JQ 插件
5. 支持 HTML5、CSS

6. 支持 LESS 动态样式

```
<head>
  <meta charset="UTF-8">
  <title>Bootstrap介绍</title>
  <link rel="stylesheet" href="css/bootstrap.min.css">
</head>
<body>

  <button class="btn btn-info">Bootstrap</button>

  <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

js放到head: 打开就会直接加载

而放到body, 打开后不会立即加载, 页面可以先显示, 再加载JS

像 CSS、JS, 在 HTML 中需要引入 bootstrap.js, bootstrap.css

注意: 必须先引用 JQ, 再引用 bootstrap.js

```
<link rel="stylesheet" href="../bootstrap.css">
<script src="../jquery-3.3.1.js"></script>
<script src="../bootstrap.js"></script>
```

CSS 概览

移动设备优先

Bootstrap 3 的设计目标是移动设备优先, 然后才是桌面设备。这实际上是一个非常及时的转变, 因为现在越来越多的用户使用移动设备。

为了让 Bootstrap 开发的网站对移动设备友好, 确保适当的绘制和触屏缩放, 需要在网页的 head 之中添加 **viewport meta** 标签:

```
<meta name="viewport" content="width=device-width,initial-scale=1.01">
```

initial-scale: 1:1 比例呈现

user-scalable=no: 禁止缩放

响应式图像

```

```

可以是图片得到更好的支持：行内块、不超出父元素、按比例缩放

全局显示

Margin: *body {margin: 0;}*

```
body {  
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;  
  font-size: 14px;  
  line-height: 1.428571429;  
  color: #333333;  
  background-color: #ffffff;  
}
```

排版

使用 `@font-family-base`、`@font-size-base` 和 `@line-height-base` 属性
作为排版样式

（@：使用的 Less）

链接样式

通过属性 `@link-color` 设置全局链接的颜色。

容器

- `.container` 大小是 0，需要自己设置

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

```
.container {  
  padding-right: 15px;  
  padding-left: 15px;  
  margin-right: auto;  
  margin-left: auto;  
}
```

- .container-fluid: 100% 宽度, 占据全部视口

页面排版

P

Bootstrap 将全局 font-size 设置为 14px, line-height 行高设置为 20px
<p> 设置为 10px,color#333。

如果不使用 Bootstrap, 那么 HTML 的字体大小颜色都取决于浏览器, 不同浏览器显示不同。

H

<h1>Bootstrap 框架</h1>	//36px
<h2>Bootstrap 框架</h2>	//30px
<h3>Bootstrap 框架</h3>	//24px
<h4>Bootstrap 框架</h4>	//18px
<h5>Bootstrap 框架</h5>	//14px
<h6>Bootstrap 框架</h6>	//12px

H 中可以添加一个 small 标签, 其大小为父 H 标签的 65%:

内联文本

- 添加标记: <mark>或者.mark 类

```
<p>Bootstrap<mark>框架</mark></p>
```

```
//添加标记, <mark>元素或.mark 类
<p>Bootstrap<mark>框架</mark></p>

//各种加线条的文本
<del>Bootstrap 框架</del> //删除的文本
<s>Bootstrap 框架</s> //无用的文本
<ins>Bootstrap 框架</ins> //插入的文本
<u>Bootstrap 框架</u> //效果同上, 下划线文本

//各种强调的文本
<small>Bootstrap 框架</small> //标准字号的 85%
<strong>Bootstrap 框架</strong> //加粗 700
<em>Bootstrap 框架</em> //倾斜
```

4. 对齐

```
//设置文本对齐
<p class="text-left">Bootstrap 框架</p> //居左
<p class="text-center">Bootstrap 框架</p> //居中
<p class="text-right">Bootstrap 框架</p> //居右
<p class="text-justify">Bootstrap 框架</p> //两端对齐, 支持度不佳
<p class="text-nowrap">Bootstrap 框架</p> //不换行
```

5. 大小写

```
//设置英文文本大小写
<p class="text-lowercase">Bootstrap 框架</p> //小写
<p class="text-uppercase">Bootstrap 框架</p> //大写
<p class="text-capitalize">Bootstrap 框架</p> //首字母大写
```

7. 地址文本

```
//设置地址, 去掉了倾斜, 设置了行高, 底部 20px
<address> I
  <strong>Twitter, Inc.</strong><br>
  795 Folsom Ave, Suite 600<br>
  San Francisco, CA 94107<br>
  <abbr title="Phone">P:</abbr> (123) 456-7890
</address>
```

8. 引用文本

//默认样式引用，增加了做边线，设定了字体大小和内外边距

```
<blockquote>
```

```
    Bootstrap 框架
```

```
</blockquote>
```

//反向

```
<blockquote class="blockquote-reverse">
```

```
    Bootstrap 框架
```

```
</blockquote>
```

9. 列表排版

//移出默认样式

```
<ul class="list-unstyled">
```

```
    <li>Bootstrap 框架</li>
```

```
    <li>Bootstrap 框架</li>
```

```
    <li>Bootstrap 框架</li>
```

```
    <li>Bootstrap 框架</li>
```

```
    <li>Bootstrap 框架</li>
```

```
</ul>
```

//设置成内联

```
<ul class="list-inline">
```

```
    <li>Bootstrap 框架</li>
```

```
    <li>Bootstrap 框架</li>
```

```
    <li>Bootstrap 框架</li>
```

```
    <li>Bootstrap 框架</li>
```

```
    <li>Bootstrap 框架</li>
```

```
</ul>
```

//水平排列描述列表

```
<dl class="dl-horizontal">
```

```
    <dt>Bootstrap</dt>
```

```
    <dd>Bootstrap 提供了一些常规设计好的页面排版的样式供开发者使用。</dd>
```

10. 代码

//内联代码

```
<code>&lt;section&gt;</code>
```

//用户输入

```
press <kbd>ctrl + ,</kbd>
```

//代码块

```
<pre>&lt;p&gt;Please input ..&lt;/p&gt;</pre>
```

Bootstrap 还列举了<var>表示标记变量，<samp>表示程序输出，只不过没有重新复

表格和按钮

```
<table>
  <thead>
    <tr>
      <th>编号</th>
      <th>姓名</th>
      <th>性别</th>
      <th>年龄</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>张三</td>
      <td>男</td>
      <td>50</td>
    </tr>
  </tbody>
</table>
```

- 基本表格样式: `<table class="table">`
- 条纹 table-striped
- 加边框: table-bordered
- 悬停鼠标: 鼠标悬停产生效果: table-hover
- `<table class="table table-striped table-bordered table-hover">`
- 状态类:

设置在 tr 中

会呈现不同颜色

5. 状态类

//可以单独设置每一行的背景样式

```
<tr class="success">
```

注：一共五种不同的样式可供选择。

样式	说明
active	鼠标悬停在行或单元格上
success	标识成功或积极的动作
info	标识普通的提示信息或动作
warning	标识警告或需要用户注意

danger	表示危险或潜在的带来负面影响的动作
--------	-------------------

- 隐藏一行：tr: sr-only

- 响应式表格

表格父元素设置为响应式，小于 768px 时会出现滚动条，大于时滚动条自动消失。

```
<body class="table-responsive">
```

按钮

1. 可作为按钮使用的标签或元素

```
<a> <input> <button>
```

- 普通按钮：btn btn-default


```

<a href="#" class="btn btn-default">A Lable</a>
<!--btn:按钮
      btn-default: 默认格式
-->
<button class="btn btn-default">Button</button>
<input type="button" class="btn btn-default" value="Button">

```

样式	说明
btn-default	默认样式
btn-success	成功样式
btn-info	一般信息样式
btn-warning	警告样式
btn-danger	危险样式
btn-primary	首选项样式
btn-link	链接样式

注意:

- 导航和导航条组件只支持<button>, 不支持 a , input
- 连接作为按钮时:
当用于当前页面触发某些功能而不是连接到其他页面或者是本页面的其他部分时, 必须要使用 role="button"
- 跨浏览器展现: 建议劲量使用 button

3. 尺寸大小

//从大到小的尺寸

<button class="btn btn-lg">Button</button>

lg:大

<button class="btn">Button</button>

sm:小

<button class="btn btn-sm">Button</button>

<button class="btn btn-xs">Button</button>

xs:超小

4. 块级按钮

//块级换行

<button class="btn btn-block">Button</button>

<button class="btn btn-block">Button</button>

5. 激活状态

// 激活按钮

```
<button class="btn active">Button</button>
```

6. 禁用状态

// 禁用按钮

```
<button class="btn active disabled">Button</button>
```

表单

Label+input

在 label 的 for 中使用 input 的 id，在点击 label 时就会使 input 获取焦点。

```
<div class="form-group">
  <label for="inp">Name</label>
  <input id="inp" class="form-control" placeholder="请输入姓名，这是placeholder">
</div>
```

- .form-group: div 使用该类，将 label+form 放入其中，可自动设置间距布局
- .form-control: input, textarea, select 加上该类后，width=100%
- Placeholder: 放在输入框中的提示语，获取焦点即消失。

注意：不要将表单组直接和输入框组混合使用。建议将输入框组嵌套到表单组中使用。

内联表单

<form> 元素添加 .form-inline 类：可使其内容左对齐并且表现为 inline-block 级别的控件

```
<form class="form-inline">
```

注意：

视口（viewport）至少在 768px 宽度时，会显示在一行，但是小于 768px 时就会认为是手机端，不会显示在一行，恢复独占样式。

一定要添加 label 标签

表单合组

\$	Amount	.00	Transfer cash
----	--------	-----	---------------

```
<form action="">
  <div class="input-group">
    <!-- 不要使用label，会影响效果-->
    <!--<label for="i">Name</label> -->
    <div class="input-group-addon">$</div>
    <input type="text" class="form-control">
    <div class="input-group-addon">结尾</div>
  </div>
</form>
```

Input-group-addon 使用注意：

- 不使用 form-group，用 input-group
- 不要使用 label
- Input-group-addon 是 div 的类，放在 input 前后

水平排列

水平排列：让表单中的元素保持水平排列：label 和 input 居于一样（注意内联是全在一行）

需要结合栅格系统。

在 form 标签添加:form-horizontal, label 添加 col-sm-2 和 control-label,

input 放入 div, div 添加 col-sm-10

```
<form class="form-horizontal">
  <div class="form-group">
    <label class="col-sm-2 control-label">NAME</label>
    <div class="col-sm-9">
      <input type="text" class="form-control">
    </div>
  </div>
  <div class="form-group">
    <label class="col-sm-2 control-label">NAME Family</label>
    <div class="col-sm-9">
      <input type="text" class="form-control">
    </div>
  </div>
</form>
```



NAME

NAME Family

对齐了, 且在一行

复选框

```
<form >
  <!-- 外围添加div, 类设置为CheckBox, 可以单行显示, 且有效果-->
  <div class="checkbox">
    <label >
      <!-- 把input放到label中, 点击文字也可以触发选中/取消-->
      <input type="checkbox">Beauty
    </label>
  </div>
  <div class="checkbox">
    <label >
      <input type="checkbox">Handsome
    </label>
  </div>
```

添加 disable 可以设置禁用

```

<div class="checkbox">
  <label>
    <input type="checkbox" disabled> Handsome
  </label>
</div>

```

disable这里不是类

单选框

与复选框用法一直，把 checkbox 变成 radio，另外同样需要给 input 添加同一个 name

```

<form>
  <!-- 外围添加div，类设置为CheckBox，可以单行显示，且有效果 -->
  <div class="radio">
    <label>
      <!-- 把input放到label中，点击文字也可以触发选中/取消 -->
      <input type="radio" name="radio"> Beauty
    </label>
  </div>
  <div class="radio">
    <label>
      <input type="radio" name="radio"> Handsome
    </label>
  </div>
</form>

```

下拉列表

Select+option

```

<select class="form-group">
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
</select>

```

校验状态


样式	说明
has-error	错误状态
has-success	成功状态
has-warning	警告状态

通过 spring 校验然后 addClass 就可以设置这些状态了

添加额外的图标

```
<form >
  <div class="form-group has-feedback">
    <label >Email</label>
    <input type="email" class="form-control">
    <span class="glyphicon glyphicon-record form-control-feedback"></span>
  </div>
</form>
```

电子邮件



样式	说明
glyphicon-ok	成功状态
glyphicon-warning-sign	警告状态
glyphicon-remove	错误状态

控制大小

Input-sm

Input-lg

图片

形状:

```



```

响应式: 添加 img-responsive

栅格系统

移动设备优先

添加：

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
```

布局容器

需要为页面内容或栅格系统添加一个类：`.container` 或 `.container-fluid`

两者不能相互嵌套



栅格系统

原理：

- “行 (row)”必须包含在 `.container` (固定宽度) 或 `.container-fluid` (100% 宽度) 中, 以便为其赋予合适的排列 (alignment) 和内补 (padding)。
- 通过“行 (row)”在水平方向创建一组“列 (column)”。
- 你的内容应当放置于“列 (column)”内, 并且, 只有“列 (column)”可以作为行 (row)”的直接子元素。
- 类似 `.row` 和 `.col-xs-4` 这种预定义的类, 可以用来快速创建栅格布局。Bootstrap 源码中定义的 `mixin` 也可以用来创建语义化的布局。
- 通过为“列 (column)”设置 `padding` 属性, 从而创建列与列之间的间隔 (gutter)。通过为 `.row` 元素设置负值 `margin` 从而抵消掉为 `.container` 元素设置的 `padding`, 也就间接为“行 (row)”所包含的“列 (column)”抵消掉了 `padding`。
- 负值的 `margin` 就是下面的示例为什么是向外突出的原因。在栅格列中的内容排成一行。
- 栅格系统中的列是通过指定 1 到 12 的值来表示其跨越的范围。例如, 三个等宽的列可以使用三个 `.col-xs-4` 来创建。
- 如果一行 (row)”中包含的“列 (column)”大于 12, 多余的“列 (column)”所在的元素将被作为一个整体另起一行排列。
- 栅格类适用于与屏幕宽度大于或等于分界点大小的设备, 并且针对小屏幕设备覆盖栅格类。因此, 在元素上应用任何 `.col-md-*` 栅格类适用于与屏幕宽度大于或等于分界点大小的设备, 并且针对小屏幕设备覆盖栅格类。因此, 在元素上应用任何 `.col-lg-*` 不存在, 也影响大屏幕设备。

实现: `.container>row>col`

```
<div class="container">
  <div class="row">
    <div class="col-xs-2 aa">1</div>
    <div class="col-xs-1 aa">1</div>
    <div class="col-xs-1 aa">1</div>
    <div class="col-xs-1 aa">1</div>
  </div>
  <div class="row" ...>
  <div class="row" ...>
  <div class="row" ...>
</div>
```

不能设置固定了
宽度, 否则col-n
显示不出多列

参数:

	超小屏幕 ($<768\text{px}$)	小屏幕 ($\geq 768\text{px}$)	中等屏幕 ($\geq 992\text{px}$)	大屏幕 ($\geq 1200\text{px}$)
栅格系统行为	总是水平排列	开始是堆叠在一起的，当大于这些阈值时将变为水平排列C		
.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			

电脑桌面：使用 md

可以都添加，在不同的设备上显示不同的列数：

[illegible]

列偏移: col-lg-offset-n

可以嵌套:

调换位置: 原理是向左向右移动位置 push 向右, pull 向左

Col-md-push-n

Col-md-pull-n

Panel

面板组件用于把 DOM 组件插入到一个盒子中。创建一个基本的面板, 只需要向 `<div>` 元素添加 `class .panel` 和 `class .panel-default` 即可

```
<div class="panel panel-default">
  <div class="panel-heading">
    <h3 class="panel-title">
      带有 title 的面板标题
    </h3>
  </div>
  <div class="panel-body">
    面板内容
  </div>
</div>
```

```
<div class="panel panel-default">
  <div class="panel-body">
    这是一个基本的面板
  </div>
  <div class="panel-footer">面板脚注</div>
</div>
```

以在面板中添加脚注, 只需要把按钮或者副文本放在带有 `class .panel-footer` 的 `<div>` 中即可

● 带语境色彩的面板:

使用语境状态类 `panel-primary`、`panel-success`、`panel-info`、`panel-warning`、`panel-danger`, 来设置带语境色彩的面板

● 带表格的面板:

为了在面板中创建一个无边框的表格，我们可以在面板中使用 class .table。假设有个 <div> 包含 .panel-body，我们可以向表格的顶部添加额外的边框用来分隔。如果没有包含 .panel-body 的 <div>，则组件会无中断地从面板头部移动到表格。

```
<div class="panel panel-default">
  <div class="panel-heading">
    <h3 class="panel-title">面板标题</h3>
  </div>
  <div class="panel-body">
    这是一个基本的面板
  </div>
  <table class="table">
    <th>产品</th><th>价格 </th>
    <tr><td>产品 A</td><td>200</td></tr>
    <tr><td>产品 B</td><td>400</td></tr>
  </table>
</div>
```

● 带列表组的面板

我们可以在任何面板中包含列表组，通过在 <div> 元素中添加 .panel 和 .panel-default 类来创建面板，并在面板中添加列表组。您可以从 [列表组](#) 一章中学习如何创建列表组。

实例

```
<div class="panel panel-default">
  <div class="panel-heading">面板标题</div>
  <div class="panel-body">
    <p>这是一个基本的面板内容。这是一个基本的面板内容。
      这是一个基本的面板内容。这是一个基本的面板内容。
      这是一个基本的面板内容。这是一个基本的面板内容。
      这是一个基本的面板内容。这是一个基本的面板内容。
    </p>
  </div>
  <ul class="list-group">
    <li class="list-group-item">免费域名注册</li>
    <li class="list-group-item">免费 Window 空间托管</li>
    <li class="list-group-item">图像的数量</li>
    <li class="list-group-item">24*7 支持</li>
    <li class="list-group-item">每年更新成本</li>
  </ul>
</div>
```

辅助类

情境文本颜色

```
<p class="text-muted">...</p>
<p class="text-primary">...</p>
<p class="text-success">...</p>
<p class="text-info">...</p>
<p class="text-warning">...</p>
<p class="text-danger">...</p>
```

样式名	
text-muted	柔和灰
text-primary	主要蓝
text-success	成功绿
text-info	信息蓝
text-warning	警告黄
text-danger	危险红

情境背景色

```
<p class="bg-primary">...</p>
<p class="bg-success">...</p>
<p class="bg-info">...</p>
<p class="bg-warning">...</p>
<p class="bg-danger">...</p>
```

关闭按钮

Class = “close”

```
<!--使用X是不规范的-->
<button>X</button>
<!--使用: &times -->
<button class="close" >&times;</button>
```

会显示在右上角

快速浮动

- Pull-left
- Pull-right

相当于 float, 不过使用 important 加强了优先级

级块浮动

Center-block

相当于 margin:n auto, 并设置了 display: block

清理浮动

```
<div class="pull-left a">左边</div>
<div class="clearfix"></div>
<div class="a">右边</div> I
```

上面例子中, 左边的浮动了, 右边的 div 浮上去, 但是文字却没浮上去, 使用 clearfix 清除

显示隐藏

```
<div class="show">show</div>
<div class="hidden">hidden</div>
```

响应式工具

在媒体查询时，针对不同的屏幕大小，有时候需要显示、隐藏部分内容，响应式工具类就提供了这种解决方案。

	超小屏幕 手机 (<768px)	小屏幕 平板 (≥768px)	中等屏幕 桌面 (≥992px)	大屏幕 桌面 (≥1200px)
.visible-xs-*	可见	隐藏	隐藏	隐藏
.visible-sm-*	隐藏	可见	隐藏	隐藏
.visible-md-*	隐藏	隐藏	可见	隐藏
.visible-lg-*	隐藏	隐藏	隐藏	可见
.hidden-xs	隐藏	可见	可见	可见
.hidden-sm	可见	隐藏	可见	可见
.hidden-md	可见	可见	隐藏	可见
.hidden-lg	可见	可见	可见	隐藏

从 v3.2.0 版本起，形如 .visible-*-* 的类针对每种屏幕大小都有了三种变体，每个针对 CSS 中不同的 display 属性，列表如下：

类组	CSS display
.visible-*-block	display: block;
.visible-*-inline	display: inline;
.visible-*-inline-block	display: inline-block;

图标菜单按钮组件

小图标组件

两百多个小图标



使用<i>和, 声明 glyphicon+具体图标

```
<i class="glyphicon glyphicon-lock"></i>
<span class="glyphicon glyphicon-lock"></span>
```

可以加到按钮等中。

注意：glyphicon 这个样式，必须要关联到 glyphs-halflings-regular.eot 等文件才能正常使用。在开发中直接把整个 dist 文件引入吧。

下拉菜单

```
<div class="dropdown">
  <button class="btn btn-default" data-toggle="dropdown">
    下拉菜单
    <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    <li><a href="#">首页</a></li>
    <li><a href="#">资讯</a></li>
    <li><a href="#">产品</a></li>
    <li><a href="#">关于</a></li>
  </ul>
</div>
```


按钮和菜单需要包裹在.dropdown 的容器里，而作为被点击的元素按钮需要设置 data-toggle="dropdown"才能有效。对于菜单部分，设置 class="dropdown-menu"才能自动隐藏并添加固定样式。设置 class="caret"表示箭头，可上可下。

//设置向上触发

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

//菜单项居右对齐，默认值是 dropdown-menu-left

```
<ul class="dropdown-menu dropdown-menu-right">
```

//设置菜单的标题，不要加超链接

```
<li class="dropdown-header">网站导航</li>
```

//设置菜单的分割线

```
<li class="divider"></li>
```

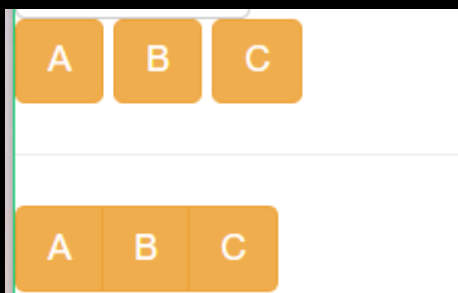
//设置菜单的禁用项

```
<li class="disabled"><a href="#">产品</a></li>
```

按钮组件

- 将按钮放到一个 group：形成整体风格

```
<div class="btn-group">  
  <button type="button" class="btn btn-default">左</button>  
  <button type="button" class="btn btn-default">中</button>  
  <button type="button" class="btn btn-default">右</button>  
</div>
```



- Btn-toolbar: 工具条

```
//将多个按钮组整合起来便于管理
<div class="btn-toolbar">
  <div class="btn-group">
    <button type="button" class="btn btn-default">左</button>
    <button type="button" class="btn btn-default">中</button>
    <button type="button" class="btn btn-default">右</button>
  </div>

  <div class="btn-group">
    <button type="button" class="btn btn-default">1</button>
    <button type="button" class="btn btn-default">2</button>
    <button type="button" class="btn btn-default">3</button>
  </div>
</div>
```

● 嵌套一个分组——按钮式下拉菜单

这个应该是很常用的，在导航栏中点击一项，然后显示下来菜单

```
<div class="btn-group">
  <button class="btn btn-default">左</button>
  <button class="btn btn-default">中</button>
  <button class="btn btn-default">右</button>
  <div class="btn-group">
    <button class="btn btn-default dropdown-toggle" data-toggle="dropdown">
      下拉菜单
      <span class="caret"></span>
    </button>
    <ul class="dropdown-menu">
      <li class="dropdown-header">网站导航</li>
      <li><a href="#">首页</a></li>
      <li><a href="#">资讯</a></li>
      <li class="divider"><a href="#">产品</a></li>
      <li class="disabled"><a href="#">关于</a></li>
    </ul>
  </div>
</div>
```

● 按钮组垂直排列：

```
//设置按钮组垂直排列
<div class="btn-group-vertical">
```

● 按钮两端对齐

```
//设置两端对齐按钮组，使用<a>标签
<div class="btn-group-justified">
  <a type="button" class="btn btn-default">左</a>
  <a type="button" class="btn btn-default">中</a>
  <a type="button" class="btn btn-default">右</a>
</div>
```

注意：不能使用 button，需要使用 a 标签

● 分裂式下拉菜单

```
//分裂式按钮下拉菜单
<div class="btn-group">
  <button type="button" class="btn btn-default">下拉菜单</button>
  <button type="button" class="btn btn-default dropdown-toggle"
data-toggle="dropdown">
    <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    <li><a href="#">首页</a></li>
    <li><a href="#">资讯</a></li>
    <li><a href="#">产品</a></li>
    <li><a href="#">关于</a></li>
  </ul>
</div>
```

```
//向上弹出式
<div class="btn-group dropdown">
```

输入框组件

前面已学过一些

按钮+表单组合 input-group-btn：十分常用

```
<div class="input-group">
  <input type="text" class="form-control">
  <div class="input-group-btn">
    <button class="btn btn-default">提交</button>
  </div>
</div>
```

提交

Input+下拉菜单

```
<div class="input-group">
  <input type="text" class="form-control">
  <div class="input-group-btn">
    <button class="btn btn-default dropdown-toggle" data-toggle="dropdown">
      下拉菜单
      <span class="caret"></span>
    </button>
    <ul class="dropdown-menu">
      <li class="dropdown-header">网站导航</li>
      <li><a href="#">首页</a></li>
      <li><a href="#">资讯</a></li>
      <li class="divider"><a href="#">产品</a></li>
      <li class="disabled"><a href="#">关于</a></li>
    </ul>
  </div>
</div>
```

导航组件

- 基于导航标签组件：nav nav-tabs

```
<ul class="nav nav-tabs">
  <li class="active"><a href="#">首页</a></li>
  <li><a href="#">资讯</a></li>
  <li><a href="#">产品</a></li>
  <li><a href="#">关于</a></li>
</ul>
```

- 胶囊式：

```
<ul class="nav nav-pills">
  <li class="active"><a href="#">首页</a></li>
  <li><a href="#">资讯</a></li>
  <li><a href="#">产品</a></li>
  <li><a href="#">关于</a></li>
</ul>
```

- 垂直胶囊式：

```
//垂直胶囊式导航
<ul class="nav nav-pills nav-stacked">
```

- 两端对齐式：

```
//导航两端对齐
<ul class="nav nav-tabs nav-justified">
```

- 禁用：

```
//禁用导航中的项目
<li class="disabled"><a href="#">关于</a></li>
```

- 带下拉菜单的导航栏：

```
<ul class="nav nav-tabs nav-justified">
  <li class="active"><a href="#">首页</a></li>
  <li><a href="#">资讯</a></li>
  <li><a href="#">产品</a></li>
  <li class="dropdown">
    <a href="#" data-toggle="dropdown">
      下拉菜单
      <span class="caret"></span>
    </a>
    <ul class="dropdown-menu">
      <li><a href="#">首页</a></li>
      <li><a href="#">资讯</a></li>
      <li><a href="#">产品</a></li>
      <li><a href="#">关于</a></li>
    </ul>
  </li>
  <li><a href="#">关于</a></li>
</ul>
```


导航条组件

- 基本格式

```
<nav class="navbar navbar-default">  
  ...  
</nav>
```

除了 navbar-default,还有 navbar-reverse 等效果

- 添加 head, brand 等

```
<nav class="navbar navbar-default">  
  <div class="container">  
    <div class="navbar-header">  
      <a href="#" class="navbar-brand">标题</a>  
    </div>  
    <ul class="nav navbar-nav">  
      <li class="active"><a href="#">首页</a></li>  
      <li><a href="#">资讯</a></li>  
      <li class="disabled"><a href="#">产品</a></li>  
      <li><a href="#">关于</a></li>  
    </ul>  
  </div>  
</nav>
```

- 添加 input

```
<form action="" class="navbar-form navbar-right">  
  <div class="input-group">  
    <input type="text" class="form-control">  
    <div class="input-group-btn">  
      <button class="btn btn-default">提交</button>  
    </div>  
  </div>  
</form>  
<button class="btn btn-default navbar-btn navbar-left">按钮</button>  
<p class="navbar-text">我是一段文本</p>  
<a href="#"></a>  
</div>
```

-

```
//导航中使用一段文本
<p class="navbar-text">我是一段文本</p>

//非导航链接，一般需要置入文本区域内
<a href="#" class="navbar-link">非导航链接</a>

//将导航固定在顶部，下面的内容会自动上移
<nav class="navbar navbar-default navbar-fixed-top">

//将导航补丁在底部
<nav class="navbar navbar-default navbar-fixed-bottom">

//静态导航，和页面等宽的导航条，去掉了圆角
<nav class="navbar navbar-default navbar-static-top">
```

路径分页标签

路径组件（面包屑导航）

一. 路径组件

路径组件也叫做面包屑导航。

//面包屑导航

```
<ol class="breadcrumb">
  <li><a href="#">首页</a></li>
  <li><a href="#">产品列表</a></li>
  <li class="active">韩版 2015 年羊绒毛衣</li>
</ol>
```

首页 / 产品列表 / 韩版2015羊绒大衣

分页组件

使用 active 即为首选项


```

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

```

翻页组件

```

<ul class="pager">
  <li><a href="#">上一页</a></li>
  <li><a href="#">下一页</a></li>
</ul>

```

```

<ul class="pager">
  <li class="previous disabled"><a href="#">上一页</a></li>
  <li class="next"><a href="#">下一页</a></li>
</ul>

```

标签

```
<h3>标签 <span class="label label-default">new</span></h3>
```

//不同色调的标签

```
<h3>标签 <span class="label label-primary">new</span></h3>
```

```
<h3>标签 <span class="label label-success">new</span></h3>
```

```
<h3>标签 <span class="label label-info">new</span></h3>
```

```
<h3>标签 <span class="label label-warning">new</span></h3>
```

```
<h3>标签 <span class="label label-danger">new</span></h3>
```

徽章

//未读信息数量徽章

```
<a href="#">信息 <span class="badge">10</span></a>
```

//按钮中使用徽章

```
<button class="btn btn-success">
```

```
    提交 <span class="badge">3</span>
```

```
</button>
```

//激活状态自动适配色调

```
<ul class="nav nav-pills">
```

```
    <li class="active">
```

```
        <a href="#">首页 <span class="badge">2</span></a>
```

```
    </li>
```

```
    <li><a href="#">资讯</a></li>
```

```
</ul>
```