Bootstrap



# 什么是bootstrap



# 为什么使用Bootstrap

移动设备优先：自 Bootstrap 3 起，框架包含了贯穿于整个库的移动设备优先的样式。

浏览器支持：所有的主流浏览器都支持 Bootstrap。

Internet Explorer Firefox Opera Google Chrome Safari

* 容易上手：只要具备 HTML 和 CSS 的基础知识，就可以开始学习 Bootstrap。
* **响应式设计：**Bootstrap 的响应式 CSS 能够自适应于台式机、平板电脑和手机。更多有关响应式设计的内容详见 Bootstrap 响应式设计。
* 响应式设计
* 它为开发人员创建接口提供了一个简洁统一的解决方案。
* 它包含了功能强大的内置组件，易于定制。
* 它还提供了基于 Web 的定制。
* 它是开源的。

# 主要学习内容

全局CSS样式，一些常用组件 ，表单，导航栏等。

# 如何使用Bootstrap

## 下载Bootstrap

Bootstrap官网：<http://www.bootcss.com/>

点击Bootstrap3中文文档-> 下载Bootstrap

地址：[http://v3.bootcss.com/getting-started/#download](http://v3.bootcss.com/getting-started/" \l "download)



下载压缩包之后，将其解压缩到任意目录即可看到以下（压缩版的）目录结构：



## 引入Bootstrap

### 方式一：连接外网



### 方式二：

|  |
| --- |
| <head>  <meta charset=*"utf-8"*>  <!--声明文档兼容模式，表示使用IE浏览器的最新模式-->  <meta http-equiv=*"X-UA-Compatible"* content=*"IE=edge"*>  <!--设置视口的宽度(值为设备的理想宽度)，页面初始缩放值<理想宽度/可见宽度>-->  <meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>  <!-- 上述3个meta标签\*必须\*放在最前面，任何其他内容都\*必须\*跟随其后！ -->  <title>Bootstrap容器介绍</title>  <!-- 引入Bootstrap核心样式文件 -->  <link href=*"../../css/bootstrap.min.css"* rel=*"stylesheet"*>  <!-- 引入jQuery核心js文件 -->  <script src=*"../../js/jquery-1.11.0.min.js"*></script>  <!-- 引入BootStrap核心js文件 -->  <script src=*"../../js/bootstrap.min.js"*></script>  </head> |

# 布局容器

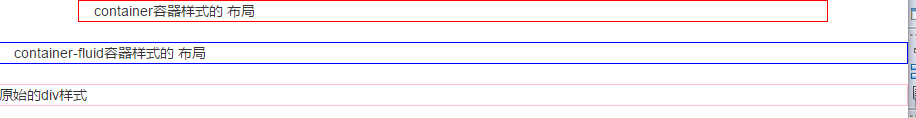
**Bootstrap需要为页面内容和栅格系统包裹一个container容器。**

**.container容器用于固定宽度并支持响应式布局的容器**

**container-fluid 容器类似于100%宽度，占据全部视口的全部**

|  |
| --- |
| <div class=*"container"*>  container容器样式的 布局  </div><br/>  <div class=*"container-fluid"*>  container-fluid容器样式的 布局  </div><br/>  <div>  原始的div样式  </div> |

在页面中显示样式



container-fluid的width是100%，而container的width是用媒体查询获得的动态尺寸。两者样式肯定是不一样的。并且由于padding的原因两者不可用嵌套，他们就是提供两种视觉风格。

# **栅格系统**

**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**

## 栅格系统的工作原理

* “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（aligment）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）”中包含了的“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。

## 媒体查询

由于手机端，平板端，pc端各种设备的屏幕尺寸大小，分辨率等各不相同。Bootstrap为各种设备尺寸提供不同的显示效果，less文件中使用以下媒体查询（media query）来创建关键的分界点阈值，来设置具体显示的样式 。

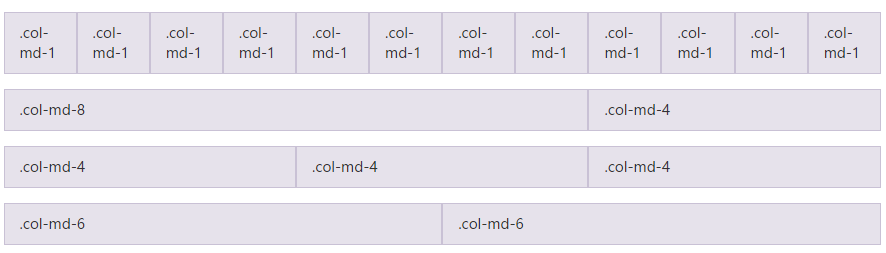
栅格参数如下：



## 堆叠的元素到水平排列

**比如元是div不加任何约束和样式都是的显示样式都是堆叠在一起的，bootstrap将栅格系统水平排列**

使用单一的一组 .col-md-\* 栅格类 所有“列（column）必须放在 ” .row 内。



|  |
| --- |
| <div class="row">  <div class="col-md-1">.col-md-1</div>  <div class="col-md-1">.col-md-2</div>  <div class="col-md-1">.col-md-3</div>  <div class="col-md-1">.col-md-4</div>  <div class="col-md-1">.col-md-5</div>  <div class="col-md-1">.col-md-6</div>  <div class="col-md-1">.col-md-7</div>  <div class="col-md-1">.col-md-8</div>  <div class="col-md-1">.col-md-9</div>  <div class="col-md-1">.col-md-10</div>  <div class="col-md-1">.col-md-11</div>  <div class="col-md-1">.col-md-12</div>  </div>  <div class="row">  <div class="col-md-8">.col-md-8</div>  <div class="col-md-4">.col-md-4</div>  </div>  <div class="row">  <div class="col-md-4">.col-md-4</div>  <div class="col-md-4">.col-md-4</div>  <div class="col-md-4">.col-md-4</div>  </div>  <div class="row">  <div class="col-md-6">.col-md-6</div>  <div class="col-md-6">.col-md-6</div>  </div> |

简单示例：将三张div里面的三张图片显示在一行

|  |
| --- |
| <div class=*"container"*>  <div class=*"row"*>  <div class=*"col-md-4"*>  <img alt=*""* src=*"../../img/1.jpg"*>  </div>  <div class=*"col-md-4"*>  <img alt=*""* src=*"../../img/1.jpg"*>  </div>  <div class=*"col-md-4"*>  <img alt=*""* src=*"../../img/1.jpg"*>  </div>  </div>  </div> |

Bootstrap将每个rows分割成12列Col-md-\* 数字相加必须要12 ，多余的列会显示到下行 。

|  |
| --- |
| <div class=*"container"*>  <div class=*"row"*>  <div class=*"col-xs-6 col-sm-4 col-md-2"*>  <!--移动设备显示两列 平板显示三列 pc显示六列 -->  <img width=*"100"* height=*"100"* alt=*""* src=*"../../img/1.jpg"*>  </div>  <div class=*"col-xs-6 col-sm-4 col-md-2"*>  <img width=*"100"* height=*"100"* alt=*""* src=*"../../img/1.jpg"*>  </div>  <div class=*"col-xs-6 col-sm-4 col-md-2"*>  <img width=*"100"* height=*"100"* alt=*""* src=*"../../img/1.jpg"*>  </div>  <div class=*"col-xs-6 col-sm-4 col-md-2"*>  <img width=*"100"* height=*"100"* alt=*""* src=*"../../img/1.jpg"*>  </div>  <div class=*"col-xs-6 col-sm-4 col-md-2"*>  <img width=*"100"* height=*"100"* alt=*""* src=*"../../img/1.jpg"*>  </div>  <div class=*"col-xs-6 col-sm-4 col-md-2"*>  <img width=*"100"* height=*"100"* alt=*""* src=*"../../img/1.jpg"*>  </div>  </div>  </div> |

# 导航栏

|  |
| --- |
| <!--导航条-->  <nav class="navbar navbar-inverse">  <div class="container-fluid">  <!-- Brand and toggle get grouped for better mobile display -->  <div class="navbar-header">  <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">  <span class="sr-only">Toggle navigation</span>  <span class="icon-bar"></span>  <span class="icon-bar"></span>  <span class="icon-bar"></span>  </button>  <a class="navbar-brand" href="#"><img src="img/logo.png" height="100%" /></a>  </div>  <!-- Collect the nav links, forms, and other content for toggling -->  <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">  <ul class="nav navbar-nav">  <li class="active">  <a href="#">首页 <span class="sr-only">(current)</span></a>  </li>  <li>  <a href="#">鞋服箱包</a>  </li>  <li>  <a href="#">情趣益智</a>  </li>  <li>  <a href="#">电动玩具</a>  </li>  <li>  <a href="#">时尚经典</a>  </li>  <li>  <a href="#">时尚经典</a>  </li>  <li>  <a href="#">时尚经典</a>  </li>  </ul>  <form class="navbar-form navbar-left">  <div class="form-group">  <input type="text" class="form-control" placeholder="Search">  </div>  <button type="submit" class="btn btn-default">搜索</button>  </form>  <ul class="nav navbar-nav navbar-right">  <li class="dropdown">  <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">Dropdown <span class="caret"></span></a>  <ul class="dropdown-menu">  <li>  <a href="#">鞋服箱包</a>  </li>  <li>  <a href="#">情趣益智</a>  </li>  <li>  <a href="#">电动玩具</a>  </li>  <li>  <a href="#">时尚经典</a>  </li>  <li role="separator" class="divider"></li>  <li>  <a href="#">Separated link</a>  </li>  </ul>  </li>  </ul>  </div>  <!-- /.navbar-collapse -->  </div>  <!-- /.container-fluid -->  </nav> |

# 表格

与原始的表格一样用，只要加上class就可以显示样式

## 表格样式

基本表格

|  |
| --- |
| <table class="table"> |

条纹表格

|  |
| --- |
| <table class=*"table table-striped"*> |

边框表格

|  |
| --- |
| <table class="table table-bordered"> |

鼠标悬停表格

|  |
| --- |
| <table class="table table-hover"> |

紧凑表格

|  |
| --- |
| <table class="table table-condensed"> |

## Tr td样式

显示不同的颜色

|  |
| --- |
| <!-- On rows 一行 -->  <tr class=*"active"*>...</tr>  <tr class=*"success"*>...</tr>  <tr class=*"warning"*>...</tr>  <tr class=*"danger"*>...</tr>  <tr class=*"info"*>...</tr>  <!-- On cells (`td` or `th`) 一个单元格-->  <tr>  <td class=*"active"*>...</td>  <td class=*"success"*>...</td>  <td class=*"warning"*>...</td>  <td class=*"danger"*>...</td>  <td class=*"info"*>...</td>  </tr> |

# 表单

单独的表单控件会被自动赋予一些全局样式。所有设置了 .form-control 类的 <input>、<textarea> 和 <select> 元素都将被默认设置宽度属性为 width: 100%;。 将 label 元素和前面提到的控件包裹在 .form-group 中可以获得最好的排列。

## 基本表单

|  |
| --- |
| <form class=*"form-inline"*>  <div class=*"form-group"*>  <label class=*"sr-only"* for=*"exampleInputEmail3"*>Email address</label>  <input type=*"email"* class=*"form-control"* id=*"exampleInputEmail3"* placeholder=*"Email"*>  </div>  <div class=*"form-group"*>  <label class=*"sr-only"* for=*"exampleInputPassword3"*>Password</label>  <input type=*"password"* class=*"form-control"* id=*"exampleInputPassword3"* placeholder=*"Password"*>  </div>  <div class=*"checkbox"*>  <label>  <input type=*"checkbox"*> Remember me  </label>  </div>  <button type=*"submit"* class=*"btn btn-default"*>Sign in</button>  </form> |

## 支持的控件

### 输入框

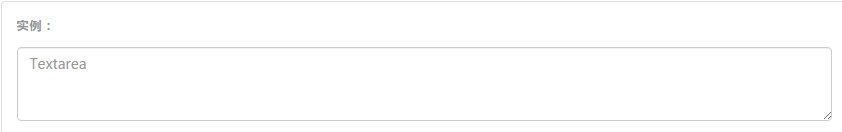
包括大部分表单控件、文本输入域控件，还支持所有 HTML5 类型的输入控件： text、password、datetime、datetime-local、date、month、time、week、number、email、url、search、tel 和 color。

|  |
| --- |
| <input type=*"text"* class=*"form-control"* placeholder=*"Text input"*> |

### 文本域

支持多行文本的表单控件。可根据需要改变 rows 属性。

|  |
| --- |
| **<textarea class="form-control" rows="3"></textarea>** |



### 多选单选框

|  |
| --- |
| <label class=*"checkbox-inline"*>  <input type=*"checkbox"* id=*"inlineCheckbox1"* value=*"option1"*> 1  </label>  <label class=*"checkbox-inline"*>  <input type=*"checkbox"* id=*"inlineCheckbox2"* value=*"option2"*> 2  </label>  <label class=*"checkbox-inline"*>  <input type=*"checkbox"* id=*"inlineCheckbox3"* value=*"option3"*> 3  </label>  <label class=*"radio-inline"*>  <input type=*"radio"* name=*"inlineRadioOptions"* id=*"inlineRadio1"* value=*"option1"*> 1  </label>  <label class=*"radio-inline"*>  <input type=*"radio"* name=*"inlineRadioOptions"* id=*"inlineRadio2"* value=*"option2"*> 2  </label>  <label class=*"radio-inline"*>  <input type=*"radio"* name=*"inlineRadioOptions"* id=*"inlineRadio3"* value=*"option3"*> 3  </label> |

### 下拉列表

对于标记了 multiple 属性的 <select> 控件来说，默认显示多选项。

|  |
| --- |
| <select multiple class=*"form-control"*>  <option>1</option>  <option>2</option>  <option>3</option>  <option>4</option>  <option>5</option>  </select> |

### 禁用状态

为输入框设置 disabled 属性可以禁止其与用户有任何交互（焦点、输入等）。被禁用的输入框颜色更浅，并且还添加了 not-allowed鼠标状态。

|  |
| --- |
| <input class=*"form-control"* id=*"disabledInput"* type=*"text"* placeholder=*"Disabled input here..."* disabled> |

### 静态控件

页面显示的一些静态的数据，相当于文本显示

如果需要在表单中将一行纯文本和 label 元素放置于同一行，为 <p> 元素添加 .form-control-static 类即可。

|  |
| --- |
| <form class=*"form-horizontal"*>  <div class=*"form-group"*>  <label class=*"col-sm-2 control-label"*>Email</label>  <div class=*"col-sm-10"*>  <p class=*"form-control-static"*>email@example.com</p>  </div>  </div>  <div class=*"form-group"*>  <label for=*"inputPassword"* class=*"col-sm-2 control-label"*>Password</label>  <div class=*"col-sm-10"*>  <input type=*"password"* class=*"form-control"* id=*"inputPassword"* placeholder=*"Password"*>  </div>  </div>  </form> |

# 按钮

## 普通的按钮



## 带有样式的按钮



|  |
| --- |
| <!-- Standard button -->  <button type="button" class="btn btn-default">（默认样式）Default</button>  <!-- Provides extra visual weight and identifies the primary action in a set of buttons -->  <button type="button" class="btn btn-primary">（首选项）Primary</button>  <!-- Indicates a successful or positive action -->  <button type="button" class="btn btn-success">（成功）Success</button>  <!-- Contextual button for informational alert messages -->  <button type="button" class="btn btn-info">（一般信息）Info</button>  <!-- Indicates caution should be taken with this action -->  <button type="button" class="btn btn-warning">（警告）Warning</button>  <!-- Indicates a dangerous or potentially negative action -->  <button type="button" class="btn btn-danger">（危险）Danger</button>  <!-- Deemphasize a button by making it look like a link while maintaining button behavior -->  <button type="button" class="btn btn-link">（链接）Link</button> |

## 按钮的尺寸

需要让按钮具有不同尺寸吗？使用 .btn-lg、.btn-sm 或 .btn-xs 就可以获得不同尺寸的按钮。

## 按钮的激活状态

当按钮处于激活状态时，其表现为被按压下去（底色更深、边框夜色更深、向内投射阴影）。对于 <button> 元素，是通过 :active 状态实现的。对于 <a> 元素，是通过 .active 类实现的。然而，你还可以将 .active 应用到 <button> 上（包含 aria-pressed="true"属性)），并通过编程的方式使其处于激活状态。

由于 :active 是伪状态，因此无需额外添加，但是在需要让其表现出同样外观的时候可以添加 .active 类。

|  |
| --- |
| <button type=*"button"* class=*"btn btn-primary btn-lg active"*>Primary button</button>  <button type=*"button"* class=*"btn btn-default btn-lg active"*>Button</button> |

## 超链接



禁用状态 不可点击



# 轮播图片

|  |
| --- |
| <div>  <div id="carousel-example-generic" class="carousel slide" data-ride="carousel">  <!-- Indicators 几张图片几个点点 -->  <ol class="carousel-indicators">  <li data-target="#carousel-example-generic" data-slide-to="0" class="active"></li>  <li data-target="#carousel-example-generic" data-slide-to="1"></li>  <li data-target="#carousel-example-generic" data-slide-to="2"></li>  <li data-target="#carousel-example-generic" data-slide-to="3"></li>  </ol>    <!-- Wrapper for slides 轮播的图片 -->  <div class="carousel-inner" role="listbox">  <div class="item active">  <img src="../img/okwu.jpg" alt="">  <div class="carousel-caption">  <p>aaaa</p><!--图片上显示的文字-->  </div>  </div>  <div class="item">  <img src="../img/okwu-athletics.jpg" alt="">  <div class="carousel-caption">    </div>  </div>  <div class="item">  <img src="../img/bartlesvillecf.jpg" alt="">  <div class="carousel-caption">    </div>  </div>  <div class="item">  <img src="../img/emancipation.jpg" alt="">  <div class="carousel-caption">    </div>  </div>  </div>    <!--显示上一个和下一个的图标-->  <!-- Controls -->  <a class="left carousel-control" href="#carousel-example-generic" role="button" data-slide="prev">  <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>  <span class="sr-only">Previous</span>  </a>  <a class="right carousel-control" href="#carousel-example-generic" role="button" data-slide="next">  <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>  <span class="sr-only">Next</span>  </a>  </div>  </div> |

# 图形图片



|  |
| --- |
| <img src="..." alt="..." class="img-rounded"><img src="..." alt="..." class="img-circle"><img src="..." alt="..." class="img-thumbnail"> |

# 情景背景

与table中用的样式是一样的

|  |
| --- |
| <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> |

# 内容的显示与隐藏

|  |
| --- |
| <div class="show">...</div>  <div class="hidden">...</div> |

# 相应设备的可见性

不同的设备的可见性与页面中的栅格系统的col-md-\* 用法一致

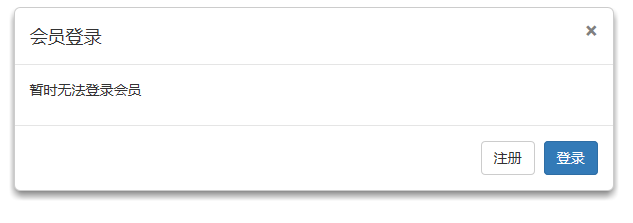


# Js插件支持一些动态效果

一些常见的模态框。

|  |
| --- |
| <!-- 模态声明，show 表示显示 -->  <div class=*"modal show"* tabindex=*"-1"*>  <!-- 窗口声明 -->  <div class=*"modal-dialog"*>  <!-- 内容声明 -->  <div class=*"modal-content"*>  <!-- 头部 -->  <div class=*"modal-header"*>  <button type=*"button"* class=*"close"* data-dismiss=*"modal"*>  <span>×</span>  </button>  <h4 class=*"modal-title"*>会员登录</h4>  </div>  <!-- 主体 -->  <div class=*"modal-body"*>  <p>暂时无法登录会员</p>  </div>  <!-- 注脚 -->  <div class=*"modal-footer"*>  <button type=*"button"* class=*"btn btn-default"*>注册</button>  <button type=*"button"* class=*"btn btn-primary"*>登录</button>  </div>  </div>  </div>  </div> |

打开页面时弹出模态框



# 定制

用户可以自定义一些样式和js等组件，这里不一一演示

