Bootstrap框架

## Bootstrap简介

是适用于移动设备有限的框架集，通过操作类class来实现页面呈现

* Bootstrap V3支持到IE7，但是IE8/IE9不支持一下样式属性，



* 需要支持下列属性， **Internet Explorer 8 需要**[**Respond.js**](https://github.com/scottjehl/Respond)**配合才能实现对媒体查询（media query）的支持。**

**引用案例：**<script src="js/respond.min.js"></script>

* 同时Bootstrap还要适用JQuery来进行底层的js铺垫

引用案例：<script src="js/jquery.min.js"></script>

## 栅格系统

Bootstrap内置了一套**响应式、移动设备优先的流式栅格系统**，随着屏幕设备或视口（viewport）尺寸的增加，**系统会自动分为最多12列**。它包含了易于使用的**预定义classe**，还有强大的mixin用于生成更具语义的布局。

* 栅格系统必须包含在 **.container 或 .container-fluid** 中

**注意：适用 .container 会居中显示，左右两边会有部分留白。而 .container-fluid 100%的占满屏幕。**

* **设置 .container 效果**

<div class="container ">



* 设置 .container-fluid 效果

<div class="container-fluid">



### 栅格选项

在使用栅格系统时，行（row）必须包含在.container或 .container-fluid 中

通过设置列（取值为1-12） -col-xs-手机 / col-sm-平板 / col-md-中屏电脑 / col-lg-大屏电脑，列最大为12列

通过屏幕的大小来设置响应式布局



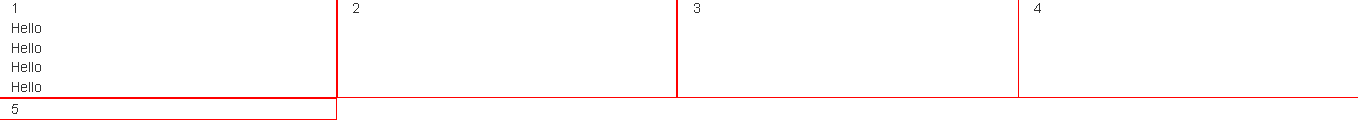
注：col-后面是规定最小适应的屏幕大小

* 栅格系统之清除浮动 **clearfix**

<div class="clearfix"></div>

* 实现列等高：

可以使用 dispaly：flex 和 flec-wrap：wrap 来实现



* 列偏移 col-offset-偏移量

元素是从左向右偏移的，偏移之后后续的元素会在前面偏移的元素之后进行排列

**示例代码：**

<div class="container-fluid">  
 <div class="row">  
 <div class="col-sm-3 col-sm-offset-2">1</div>  
 <div class="col-sm-3 col-sm-offset-1">2</div>  
 </div>  
</div>

**示例效果：**



* 列位移：不会影响其他元素，但是会与其偏移之后的元素发生重叠。

col-push-推（从左向右移动）

col-pull-拉（从右向左移动）

应用场景：可以改变元素的顺序

**代码示例：**

<div class="row">  
 <div class="col-sm-3 col-sm-push-3">1</div>  
 <div class="col-sm-3 col-sm-pull-3">2</div>  
</div>

**示例效果：**

****

## 表格

* 通过设置 class="table" 来设置表格的样式

table-responsive 设置当屏幕为小屏时候出现滚动条（做响应式表格）

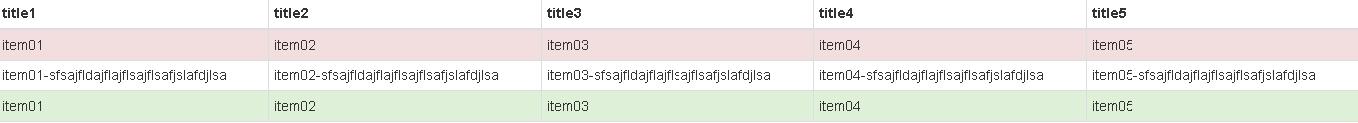
**注：此样式要设置在table父包裹器上**  
 table 设置表格默认样式 宽度为100%  
 table-bordered 设置表格边框  
 table-striped 设置表格间隔不同颜色显示  
 table-condensed 设置表格变成紧凑模式  
 table-hover 设置鼠标移入某一项时候的状态

* 设置表格行的状态  
   .active 鼠标悬停在行或单元格上时所设置的颜色  
   .success 标识成功或积极的动作  
   .warning 标识警告或需要用户注意  
   .danger 标识危险或潜在的带来负面影响的动作

**示例代码：**

<div class="table-responsive">  
 <table class="table table-striped table-bordered table-hover table-condensed">  
 <tr class="success">  
 <td>item01</td>  
 <td>item02</td>  
 <td>item03</td>  
 <td>item04</td>  
 <td>item05</td>  
 </tr>  
 </table>  
</div>

**示例效果：**

****

## 按钮

* 设置 **btn** 属性来实现按钮样式设置   
  btn-lg / btn-default / btn-sm / btn-xs 设置按钮的大小  
  btn-primary 设置按钮为“主要” （使用在提交上）  
  btn-success 设置那妞为“成功” （使用提交成功之后）  
  btn-warning 设置为“警告”  
  btn-danger 设置为“危险”  
  btn-info 设置为“信息”提示  
  btn-link 设置为“链接”

**示例代码：**

<button type="button" class="btn btn-lg btn-primary">大号按钮</button>  
<button type="button" class="btn btn-default btn-success">默认按钮</button>  
<button type="button" class="btn btn-sm btn-warning">小号按钮</button>  
<button type="button" class="btn btn-xs btn-danger">超号按钮</button>

**示例效果：**

****

**注：**disabled属性 设置按钮的状态为不可选 ，同时按钮要使用form包裹，或者通过 元素的form属性与form标签的id关联起来

## 表单

* **.form-control 设置 表单控件上 ，**

所有设置了.form-control的<input>、<textarea>和<select>元素都将被默认设置为width: 100%;

示例代码：

<form role="form">

<div class="form-group">

<label for="exampleInputEmail1">Email address</label>

<input type="email" class="**form-control**" id="exampleInputEmail1" placeholder="Enter email">

</div>

</form>

示例效果：



* .form-inline 设置在form标签上，实现所用的表单控件在一行上。

示例代码：

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

<div class="form-group">

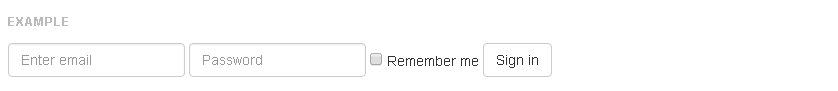
<label class="sr-only" for="exampleInputEmail2">Email address</label>

<input type="email" class="form-control" id="exampleInputEmail2" placeholder="Enter email">

</div>

</form>

示例效果：



* .form-horizontal 设置在form标签上。与 .form-group一起实现label与input控件水平分组显示

示例代码：

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

<div class="form-group">

<label for="inputEmail3" class="col-sm-2 control-label">Email</label>

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

<input type="email" class="form-control" id="inputEmail3" placeholder="Email">

</div>

</form>

示例效果：



* .form-group 表单分组选择器：

注：**form-group == row 在使用表单水平分组的时候可以把form-group看成一个row来时使用栅格系统，实现表单水平分组的布局**

* 使用复选框 checkbox 示例代码：

<div class="form-group"> //相当于 row

<div class="col-sm-offset-2 col-sm-10"> //实现栅格

<div class="checkbox"> //包裹 设置checkbox样式

<label>

<input type="checkbox"> Remember me

</label>

</div>

</div>

</div>

* 使用 sr-only 设置只在屏幕阅读器上显示（正常情况下是不显示的）

## 被支持的表单控件

### input

* 大部分表单控件、文本输入域控件。包括HTML5支持的所有类型：

text、password、datetime、datetime-local、date、month、time、week、number、email、url、search、tel和color

示例代码：

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

### Textarea （多行文本）

* 示例代码：

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

注：**rows=“3”；代表多行文版显示行数**

### Checkbox 和 radio

* 示例代码：

<div class="radio"> //设置单选框

<label>

<input type="radio" name="optionsRadios" id="optionsRadios1" value="option1" checked> Option one

</label>

</div>

<div class="checkbox"> // 设置多选框

<label>

<input type="checkbox" value=""> Option one

</label>

</div>

注：**在包裹“多选”或“单选”的label标签上设置 checkbox-inline 或 radi0-inline 会是多选和单选在同一行展**示

### Select

* 示例代码：

<select multiple class="form-control">

<option>1</option>

<option>2</option>

<option>3</option>

<option>4</option>

<option>5</option>

</select>

**示例效果：**

****

**注：multiple设置下拉列表是否平铺和多选**。 **设置 form-control 下拉菜单当独占父元素的100%的宽度，设置 from-inline之后下来在一行显示**

### 静态控件

* 在水平布局的表单中，如果需要将一行纯文本放置于label的同一行，为<p>元素添加.form-control-static即可。

示例代码：

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

<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>

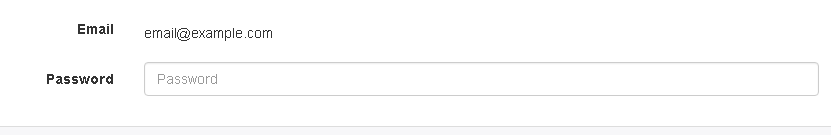
</div>

</div>

</form>

注：**通过control-label 来设置 label标签在 栅格列中垂直居中向右布局。通过设置静态控件 form-control-static 来控制静态控件的展示**

示例效果：



### 控件状态

* box-shadow设置输入框被选中（获取焦点）并对其:focus状态赋予了box-shadow样式。

示例代码：

<input class="form-control" id="focusedInput" type="text" value="This is focused...">

**注：设置 form-control之后 input会变成独占父元素的100%的宽**

* 为输入框设置**disabled**属性可以防止用户输入，并能改变一点外观，使其更直观。

示例代码：

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

示例效果：



### 控件尺寸

* 设置input控件尺寸

input-lg 设置大号的尺寸

Default input 默认尺寸

input-sm 设置小号的尺寸

示例代码：

<input class="form-control input-lg" type="text" placeholder=".input-lg">

<input class="form-control" type="text" placeholder="Default input">

<input class="form-control input-sm" type="text" placeholder=".input-sm">

示例效果：



### 调整列尺寸

* 用栅格系统中的列包裹input或其任何父元素，都可很容易的为其设置宽度。

示例代码：

<div class="row"> // 设置栅格行

<div class="col-xs-2"> // 设置列

<input type="text" class="form-control" placeholder=".col-xs-2"> </div>

<div class="col-xs-4"> // 设置列

<input type="text" class="form-control" placeholder=".col-xs-4"> </div>

</div>

示例效果：



**注：input控件通过 form-control 来控制独占100%的列，进而实现调整控件大小。**

### 帮助文本

* 通过设置 help-block 来实现帮助提示文档

示例代码：

<span class="help-block">自己独占一行或多行的块级帮助文本。</span>

示例效果：



**注：使用帮助文本，会让提示文本独占一行显示**

## 图片

* img-rounded 设置图片圆角
* img-circle 设置图片椭圆
* img-thumbnail 正常

示例代码：

<div class="row">  
 <div class="col-sm-3 col-xs-push-1">  
 <!-- img-rounded 设置图片圆角-->  
 <img src="img/air.png" alt="..." class="img-rounded ">  
 </div>  
 <div class="col-sm-6">  
 <!-- img-circle 设置图片椭圆-->  
 <img src="img/biology.jpg" alt="..." class="img-circle">  
 </div>  
 <div class="col-sm-3 ">  
 <!-- img-thumbnail 正常 -->  
 <img src="img/geography.png" alt="..." class="img-thumbnail">  
 </div>  
</div>

示例效果：



### 响应式图片

* 通过添加**.img-responsive**其实质是为图片赋予了max-width: 100%;和height: auto;属性，可以让图片按比例缩放，不超过其父元素的尺寸。

示例代码：

<img src="..." class="img-responsive" alt="Responsive image">

## 工具class

* 通过使用一些具有特有功能的class类来实现一些js的功能

### 关闭按钮

* 通过使用一个象征关闭的图标，可以用来让模式对话框和警告框消失。

示例代码：

<button type="button" class="close" aria-hidden="true">&times;</button>

示例效果：

注：如果是使用在“模泰框”的头部则自动在右上角

### Carets

* 实现一个倒三角的小图标

示例代码：

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

示例效果：\

### 快速设置浮动

* 通过设置 pull-left 或 pull-right 来设置左右浮动，但是在使用时候建议加上VIP标志（!**important**）

示例代码：

<div class="pull-left">...</div>

<div class="pull-right">...</div>

**注：在栅格系统同栅格列本身就是用浮动进行布局的**

### 图片替换

* 通过.text-hide  可以将页面元素所包含的文本内容替换为背景图。

示例代码：

<h1 class="text-hide">Custom heading</h1>

**注：替换成页面元素（input）替换成背景图之后会不显示，并不适用于图片**

## 组件

# Glyphicons 图标

* 图标是一个独立的样式，在使用的时候不能使用其他的类样式，只能使用提供的 .glyphicons 来进行引入。
* 在使用glyphicons 图标时候，把它们放在按钮，工具栏的按钮组中，导航或输入栏的前面都可以。

示例代码 ：

<button type="button" class="btn btn-default btn-lg">

<span class="glyphicon glyphicon-star"></span> Star

</button>

示例效果：



注：**glyphicons 图标为字体图标。可以为其设置另外的样式属性**

## 下拉菜单

* 下拉菜单都是包裹在 dropdown 中
* 通过设置一个按钮的自定义属性（data-toggle）来触发菜单的显示与 不显示，其中 data-toggle="dropdown" 用来调用js脚本
* 设置菜单列表的默认不显示 dropdpwn-menu
* divider 设置菜单列表的分割线
* 禁用菜单项的某一项功能，则使用**disabled 来进行设置**

示例代码：

<div class="dropdown col-offset-3">  
 <button type="button" class="btn btn-primary btn-lg dropdown-toggle" data-toggle="dropdown">  
 下拉菜单演示<span class="glyphicon glyphicon-star"></span>  
 </button>  
 <ul class="dropdown-menu">  
 <li><a href="">test1</a></li>  
 <li class=" disabled "><a href="">test2</a></li>  
 <li class="divider"></li>  
 <li class=""><a href="">test3</a></li>  
 <li class=""><a href="">test4</a></li>  
 </ul>  
</div>

示例效果：

## 输入框组

* 通过在基于文本的输入框前面，后面或是两边加上文字或按钮

input-group 设置可以使输入框前面和后面增加的文字和按钮与输入框关联在一起

* 用带有.input-group-addon的.input-group，可以给.form-control前面或后面追加元素

示例代码：

<div class="form-horizontal">  
 <div class="input-group input-group-lg col-xs-2 col-lg-offset-2">  
 <div class="input-group-addon">  
 <span class="glyphicon glyphicon-user "></span>  
 </div>  
 <input type="text" class="form-control">  
 </div>  
 <br>  
 <div class="input-group input-group-sm col-xs-4 col-xs-offset-2">  
 <input type="text" class="form-control">  
 <span class="input-group-addon btn btn-default">Go!</span>  
 </div>  
  
 <br>  
 <div class=" input-group input-group-sx col-xs-4 col-xs-offset-2">  
 <div class="input-group-addon">  
 <span class=" glyphicon glyphicon-user"></span>  
 </div>  
 <input type="text" class="form-control">  
 <span class="input-group-addon btn btn-default">Go!</span>  
 </div>  
</div>

示例效果：



* 设置输入组的尺寸

input-group-lg 大号

input-group-sm 默认

input-group-sx 小号

## 复选框与单选框

* 复选框

示例代码：

<div class="col-lg-6">

<div class="input-group">

<span class="input-group-addon">

<input type="checkbox">

</span>

<input type="text" class="form-control">

</div>

</div>

示例效果：



* 单选框

示例代码：

<div class="col-lg-6">

<div class="input-group">

<span class="input-group-addon">

<input type="checkbox">

</span>

<input type="text" class="form-control">

</div>

</div>

示例效果：



## 带下拉菜单的按钮

* 注意要是用input-group-btn 类样式 和data-toggle="dropdown"属性

示例代码：

<div class="col-lg-6">  
 <div class="input-group">  
 <div class="input-group-btn">  
 <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown">Action

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

</button>  
 <ul class="dropdown-menu">  
 <li><a href="#">Action</a></li>  
 <li><a href="#">Another action</a></li>  
 <li><a href="#">Something else here</a></li>  
 <li class="divider"></li>  
 <li><a href="#">Separated link</a></li>  
 </ul>  
 </div><!-- /btn-group -->  
 <input type="text" class="form-control">  
 </div><!-- /input-group -->  
</div>

示例效果：



# 导航

* Bootstrap中可用的导航有相似的标记，用基类.nav开头，这是相似的部分。改变修饰类可以改变样式。

## 标签页

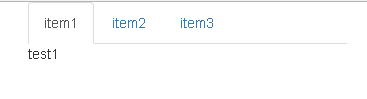
* 标签页是基于 nva-tabs 来设置的
* 设置页签似的导航条 nva nva-tabs
* 通过调用 data-toggle="tab" 来实现页签之间额相互切换，类似 ：traget 伪类的指向目标地址
* 通过 tab-pane 来时先页签内容的“显示/隐藏”；
* 通过tab-content 设置页面内容部分
* 通过 nav-pills 实现页签的 “胶囊”效果
* 通过在nav-tabs 或 nav-pills 之后加入 nav-stacked 实现垂直排列的效果
* 通过设置 nav-justifyed 来设置 页签均匀的分部在所占有的栅格上
* 通过在具体的页签上设置 disabled 来实现 页签不能被选中

示例代码：

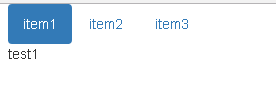
<div class="row">  
 <div class="col-xs-3 col-xs-offset-2">  
 <ul class="nav nav-tabs"> // nav nav-pills || nav-stacked  
 <li><a href="#tab1" data-toggle="tab" >item1</a></li>  
 <li><a href="#tab2" data-toggle="tab">item2</a></li>  
 <li><a href="#tab3" data-toggle="tab">item3</a></li>  
 </ul>  
 <div class="tab-content">  
 <div class="tab-pane active" id="tab1">test1</div>  
 <div class="tab-pane" id="tab2">test2</div>  
 <div class="tab-pane" id="tab3">test3</div>  
 </div>  
 </div>  
</div>

示例效果：

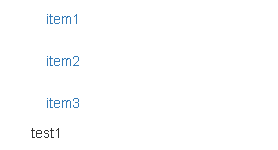
* nav-tabs 效果



* nav-pills 效果



* 加入 nav -stacked 效果



* 加入 nav-justifyrd 效果



## 下拉式的导航菜单

* 通过在某一项页签增加一个下拉财当的组件，使页签具有下拉财当的功能

示例代码：

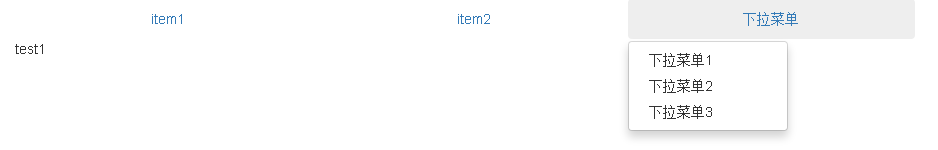
<div class="row">  
 <div class="col-xs-8 col-xs-offset-2">  
 <ul class="nav nav-pills nav-stacked nav-justified">  
 <li><a href="#tab1" data-toggle="tab" >item1</a></li>  
 <li><a href="#tab2" data-toggle="tab">item2</a></li>

//集成了下拉菜单  
 <li class="dropdown">  
 <a href="#" class="dropdown-toggle" data-toggle="dropdown"> 下拉菜单

<span class="glyphicon glyphicon-user"></span>

</a>  
 <ul class="dropdown-menu">  
 <li><a href="#tab3" data-toggle="tab">下拉菜单1</a></li>  
 <li><a href="">下拉菜单2</a></li>  
 <li><a href="">下拉菜单3</a></li>  
 </ul>  
 </li>  
 </ul>  
 <div class="tab-content">  
 <div class="tab-pane active" id="tab1">test1</div>  
 <div class="tab-pane" id="tab2">test2</div>  
 <div class="tab-pane" id="tab3">test3</div>  
 </div>  
 </div>  
</div>

示例代码：



## 导航条

* 导航条基于 navbar 来进行设置

## 模态框

通过一个按钮或者点击 来时实现模态框的触发。

* 是基于 .modal 来进行设置的，
* 在.modal的子元素上 .modal-dialog 来进行设置
* 在modal-dialog 的子元素上设置内容区域 modal-content中有三个部分：头部 modal-header

主体 modal-body

底部 modal-footer

* 显示模态框 data-toggle="modal" （设置在触发控件上）
* 关闭模态框 data-dismiss="modal" （一般设置header的头部，会默认在模态框的右边）
* 设置在modal-footer的样式，默认在底部的右边

示例代码 ：

<button type="button"  
 data-toggle="modal"  
 data-target="#myModal">Modal</button>  
 <!-- 显示模态框 data-toggle="modal"-->  
 <div class="modal" id="myModal">  
 <div class="modal-dialog">

//内容区域  
 <div class="modal-content">

//头部  
 <div class="modal-header">  
 <button type="button"  
 class="close"  
 data-dismiss="modal"  
 aria-hidden="true">&times;</button>  
 </div>  
 <!--关闭模态框 data-dismiss="modal"-->

//主体部分  
 <div class="modal-body">  
 <!--表单组件-->  
 <form action="" class="form-inline">  
 <!--输入类表单项-->  
 <label for="userNo">用户名</label>  
 <input type="text" name="userNo" id="userNo" class="form-control">  
 <label for="psw">密码</label>  
 <input type="password" name="psw" id="psw" class="form-control">  
 </form>  
 </div>  
 //尾部  
 <div class="modal-footer">  
 <button type="button" class="btn btn-primary">登录</button>  
 <button type="button" class="btn btn-danger" data-dismiss="modal">取消</button>  
 </div>  
 </div>  
 </div>  
 </div>

示例效果：



## 滚动监听

* 滚动监听，通过页面滚动的位置 实现自动到对应的页签上。

事件的触发需要在body上设置data-spy=“scroll” 和 data-target=“对象的类或者ID”

* data-target所指向的是滚动监听区

示例代码：

css样式：

.box1 {  
 height: 500px;  
 width:200px;  
 background-color: yellow;  
}.box2 {  
 height: 500px;  
 width:200px;  
 background-color: red;  
}.box3 {  
 height: 500px;  
 width:200px;  
 background-color: blue;  
}

HTML结构：

<body data-spy="scroll" data-target="#test">  
 <div class="navbar navbar-default navbar-fixed-top">  
 <div class="navbar-example" id="test">  
 <ul class="nav nav-tabs nav-pills" >  
 <li><a href="#box1">item1</a></li>  
 <li><a href="#box2">item2</a></li>  
 <li><a href="#box3">item3</a></li>  
 </ul>  
 </div>  
 </div>  
 <div class="box1" id="box1" >test1</div>  
 <div class="box2" id="box2" >test2</div>  
 <div class="box3" id="box3" >test3</div>

示例效果：

