

Validate和Bootstrap

课前默写

- 1、写出jQuery的语法
- 2、写出jQuery常用事件
- 3、写出jQuery的效果
- 4、写出jQuery对HTML的操作
- 5、写出jQuery的遍历操作
- 6、写出jQuery的Ajax操作

课程回顾

- 1、什么是jQuery
- 2、jQuery的使用
- 3、jQuery的语法
- 4、jQuery常用事件
- 5、jQuery的效果
- 6、jQuery对HTML的操作
- 7、jQuery的遍历操作
- 8、jQuery的Ajax操作

今日内容

- 1、Validate插件的使用
- 2、Bootstrap的栅格系统
- 3、Bootstrap常用的排版
- 4、Bootstrap的表单
- 5、Bootstrap的组件

教学目标

- 1、掌握Validate插件的使用
- 2、掌握Bootstrap的栅格系统
- 3、掌握Bootstrap常用的排版
- 4、掌握Bootstrap的表单
- 5、熟练Bootstrap的组件

第一章 validate插件

1.1 validate概述

validate: jquery的一个插件,依赖jquery使用, Validation是历史最悠久的jQuery插件之一, 经过了全球范围内不同项目的验证, 并得到了许多Web开发者的好评。作为一个标准的验证方法库, Validation拥有如下特点:

1. 内置验证规则: 拥有必填、数字、Email、URL和信用卡号码等19类内置验证规则
2. 自定义验证规则: 可以很方便地自定义验证规则
3. 简单强大的验证信息提示: 默认了验证信息提示, 并提供自定义覆盖默认的提示信息的功能
4. 实时验证: 可能通过keyup或blur事件触发验证, 而不仅仅在表单提交的时候验证

1.2 validate使用步骤

使用步骤:

1. 导入jquery文件
2. 导入validate.js
3. 页面加载成功后!对表单进行验证! `$("选择器").validate();`
4. 在validate中编写校验规则
`$("选择器").validate(
 rules:{},
 messages:{}
});`

1.3 校验规则

默认校验规则

(1)required:true	必填字段
(2)remote:"check.php"	使用ajax方法调用check.php验证输入值
(3)email:true	必须输入正确格式的电子邮件
(4)url:true	必须输入正确格式的网址
(5)date:true	必须输入正确格式的日期 日期校验ie6出错, 慎用
(6)dateISO:true	必须输入正确格式的日期(ISO), 例如: 2009-06-23, 1998/01/22 只验证格式, 不验证有效性
(7)number:true	必须输入合法的数字(负数, 小数)
(8)digits:true	必须输入整数
(9)creditcard:	必须输入合法的信用卡号
(10)equalTo:"#field"	输入值必须和#field相同
(11)accept:	输入拥有合法后缀名的字符串 (上传文件的后缀)
(12)maxlength:5	输入长度最多是5的字符串(汉字算一个字符)
(13)minlength:10	输入长度最小是10的字符串(汉字算一个字符)
(14)rangelength:[5,10]	输入长度必须介于 5 和 10 之间的字符串"(汉字算一个字符)"
(15)range:[5,10]	输入值必须介于 5 和 10 之间
(16)max:5	输入值不能大于5
(17)min:10	输入值不能小于10

1.4 validate练习

核心代码:

```
$(function(){  
    $("#formId").validate({  
        rules:{  
            //1. 校验元素的name属性  username:"校验器" 使用单一的校验器  
            //2. 校验元素的name属性  username:{校验器:"值",校验器:"值"}  
            username:"required",  
            password:{required:true,digits:true},  
            repassword:{equalTo:"[name='password']"},  
            zxz:{min:3,required:true},  
            shuzhiqujian:{range:[5,10],required:true}  
        },  
  
        messages:{
```

```

        username:"xxx",
        password:{required:"req",digits:"dddd"},
        zxz:{min:"最小值应该大于{0}"},
        shuzhiqujian:{range:"值应该在{0}-{1}之间!"}
    }
});
})

```

实现步骤:

- 1.导入jquery.js和validate.js, messages_zh.js中文提示
- 2.加载完成 进行验证username必填
- 3、用户名必须设置

```

$(function(){
    $("#formId").validate({
        rules:{
            //1.校验元素的name属性 username:"校验器" 使用单一的校验器
            //2.校验元素的name属性 username:{校验器:"值",校验器:"值"}
            username:"required"
        },
        messages:{}
    });
})

```

5.messages

name的属性:提示信息
 name的属性:{校验器:"xx","校验器":"xxx"}
 username:"xxx",
 password:{required:"req",digits:"dddd"}
 此处可以导入messages中文提示库!

6.密码须为数字

```
password:{required:true, digits:true}
```

7.重复密码

```
equalTo:"#field" repassword:{equalTo:["name='password'"]}
```

8.最小值

min 注意 需要添加必填

9.动态修改提示的值

0是索引!!

zxz:{min:"最小值应该大于{0}"}

第二章 bootstrap

2.1 bootstrap概述

bootstrap简介:

bootstrap 是一个用于快速开发 Web 应用程序和网站的前端框架。bootstrap 是基于 HTML、CSS、JAVASCRIPT 的。bootstrap 是由 Twitter 的 Mark Otto 和 Jacob Thornton 开发的。bootstrap 是 2011 年八月在 GitHub 上发布的开源产品。

bootstrap特点

1 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:

2 所有的主流浏览器都支持 bootstrap。

Internet Explorer Firefox Opera Google Chrome Safari

3 容易上手: 只要您具备 HTML 和 CSS 的基础知识, 您就可以开始学习 bootstrap。

4 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。

2.2 案例:响应式的页面

需求:

创建一套页面!根据上网设备不同自动调节显示的效果!

m.taobao.com

taobao.com

bootstrap的使用步骤:

1. 下载bootstrap

<http://www.bootcss.com/>

2. 导入bootstrap.css文件

3. 导入jquery.js

4. 导入bootstrap.js

<!--导入css-->

<link rel="stylesheet" href="css/bootstrap.min.css" />

<!-- 导入jquery-->

<script type="text/javascript" src="js/jquery-1.11.0.js" ></script>

<!--导入css.js -->

<script type="text/javascript" src="js/bootstrap.min.js" ></script>

5. 导入支持移动设备 支持缩放 其实是1.0

<meta name="viewport" content="width=device-width, initial-scale=1">

6. 将所有内容放到布局容器中

.container 类用于固定宽度并支持响应式布局的容器。

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

```
...
```

```
</div>
```

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

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

```
...
```

```
</div>
```

测试1: 直接写container显示不全

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

```
  <div style="border: 1px solid red;">111111</div>
```

```
</div>
```

2.3 bootstrap的组成

1.全局css的样式 2.组件 3.js插件

2.4 栅格系统

2.4.1 什么是栅格系统

Bootstrap 提供了一套响应式、移动设备优先的流式网格系统，随着屏幕或视口 (viewport) 尺寸的增加，系统会自动分为最多12列。

注意： Bootstrap将每一行分成12份！

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。它包含了易于使用的预定义类，还有强大的mixin 用于生成更具语义的布局。

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。下面就介绍一下 Bootstrap 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 (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-*

2.4.2 媒体查询功能

判断是什么上网设备

小屏幕 (平板, 大于等于 768px)

```
@media (min-width: @screen-sm-min) { ... }
```

中等屏幕 (桌面显示器, 大于等于 992px)

```
@media (min-width: @screen-md-min) { ... }
```

大屏幕 (大桌面显示器, 大于等于 1200px)

```
@media (min-width: @screen-lg-min) { ... } col-lg-2
```

大屏幕 大于1200 col-lg-2

中屏幕 大于992<1200 col-md-3

小屏幕 大于768<992 col-sm-6

最小屏 小于768 col-xs-12

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 开始是堆叠在一起的, 当大于这些阈值时将变为水平排列

.container 最大宽度	None (自动)	750px	970px	1170px
-----------------	-----------	-------	-------	--------

类前缀	.col-xs-	.col-sm-	.col-md-	.col-lg-
-----	----------	----------	----------	----------

可以class中拼接多个列数限制

练习: 栅格 大屏显示6个 中 显示4个 小平 显示 2 最小 显示 1一个

```
<div class="col-lg-1 col-md-2">.col-lg-1</div>
```

2.5 排版

2.5.1 标题

标题: HTML 中的所有标题标签,

到

均可使用。另外, 还提供了 .h1 到 .h6 类, 为的是给内联 (inline) 属性的文本赋予标题的样式。

```
<h1>我是h1</h1>我是跟随者  
<!-- 要写成行内快-->  
<span class="h1">我是h1</span>我是跟随者
```

2.5.2 mark标签

you can use the mark tag to <mark> hight hight</mark>

del标签

you can use the mark tag to hight hight</>

2.5.3 对齐

通过文本对齐类, 可以简单方便的将文字重新对齐。

```
<p class="text-left">Left aligned text.</p>
<p class="text-center">Center aligned text.</p>
<p class="text-right">Right aligned text.</p>
<p class="text-justify">Justified text.</p>
<p class="text-nowrap">No wrap text.</p>
```

2.5.4 表格

table 表格 table-striped 表格隔行变色 table-hover 悬浮变色

```
<div class="table-responsive">
  <table class="table">
    ...
  </table>
```

2.5.5 行状态

通过这些状态类可以为行或单元格设置颜色。

Class	描述
.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
.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	标识危险或潜在的带来负面影响的动作

2.6 表单

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。

2.6.1 基本实例

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>

-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 <input>、<textarea> 和 <select> 添加 class = "form-control" 。

form-group 会将label和input上下排列

form-control 会自动将input填满屏幕 并添加点击高亮效果

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1"
placeholder="Email">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1"
placeholder="Password">
  </div>
</div>
```

```

        <label for="exampleInputFile">File input</label>
        <input type="file" id="exampleInputFile">
        <p class="help-block">Example block-level help text here.</p>
    </div>
    <div class="checkbox">
        <label>
            <input type="checkbox"> Check me out
        </label>
    </div>
    <button type="submit" class="btn btn-default">Submit</button>
</form>

```

2.6.2 内联表单

为 `<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。只适用于视口 (viewport) 至少在 768px 宽度时 (视口宽度再小的话就会使表单折叠)。

```

<form class="form-inline">
  <div class="form-group">
    <label for="exampleInputName2">Name</label>
    <input type="text" class="form-control" id="exampleInputName2" placeholder="Jane Doe">
  </div>
  <div class="form-group">
    <label for="exampleInputEmail2">Email</label>
    <input type="email" class="form-control" id="exampleInputEmail2"
placeholder="jane.doe@example.com">
  </div>
  <button type="submit" class="btn btn-default">Send invitation</button>
</form>

```

2.6.3 水平表单

通过为表单添加 `.form-horizontal` 类, 并联合使用 Bootstrap 预置的栅格类, 可以将 label 标签和控件组水平并排布局。这样做将改变 `.form-group` 的行为, 使其表现为栅格系统中的行 (row), 因此就无需再额外添加 `.row` 了。

```

<form class="form-horizontal">
  <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>
  </div>
  <div class="form-group">
    <label for="inputPassword3" class="col-sm-2 control-label">Password</label>
    <div class="col-sm-10">
      <input type="password" class="form-control" id="inputPassword3"
placeholder="Password">
    </div>
  </div>
  <div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">

```



```

        <div class="checkbox">
            <label>
                <input type="checkbox"> Remember me
            </label>
        </div>
    </div>
</div>
<div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
        <button type="submit" class="btn btn-default">Sign in</button>
    </div>
</div>
</form>

```

案例: 显示带引导的

```

<form class="form-inline">
    <div class="form-group">
        <label class="sr-only" for="exampleInputAmount">Amount (in dollars)</label>
        <div class="input-group">
            <div class="input-group-addon">$</div>
            <input type="text" class="form-control" id="exampleInputAmount"
placeholder="Amount">
            <div class="input-group-addon">.00</div>
        </div>
        <button type="submit" class="btn btn-primary">Transfer cash</button>
    </div>
</form>

```

2.6.4 单选框和复选框

通过将 `.checkbox-inline` 或 `.radio-inline` 类应用到一系列的多选框 (checkbox) 或单选框 (radio) 控件上, 可以使这些控件排列在一行。

```

<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>

```

2.6.5 下拉列表

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

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

2.6.6 静态控件

```
<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>
```

2.6.7 校验状态

Bootstrap 对表单控件的校验状态，如 error、warning 和 success 状态，都定义了样式。使用时，添加 .has-warning、.has-error 或 .has-success 类到这些控件的父元素即可。任何包含在此元素之内的 .control-label、.form-control 和 .help-block 元素都将接受这些校验状态的样式。

```
<div class="form-group has-success">
  <label class="control-label" for="inputSuccess1">Input with success</label>
  <input type="text" class="form-control" id="inputSuccess1" aria-
describedby="helpBlock2">
  <span id="helpBlock2" class="help-block">A block of help text that breaks onto a new
line and may extend beyond one line.</span>
</div>
<div class="form-group has-warning">

  <label class="control-label" for="inputWarning1">Input with warning</label>
```

```

    <input type="text" class="form-control" id="inputWarning1">
  </div>
  <div class="form-group has-error">
    <label class="control-label" for="inputError1">Input with error</label>
    <input type="text" class="form-control" id="inputError1">
  </div>
  <div class="has-success">
    <div class="checkbox">
      <label>
        <input type="checkbox" id="checkboxSuccess" value="option1">
        Checkbox with success
      </label>
    </div>
  </div>
  <div class="has-warning">
    <div class="checkbox">
      <label>
        <input type="checkbox" id="checkboxWarning" value="option1">
        Checkbox with warning
      </label>
    </div>
  </div>
  <div class="has-error">
    <div class="checkbox">
      <label>
        <input type="checkbox" id="checkboxError" value="option1">
        Checkbox with error
      </label>
    </div>
  </div>

```

添加额外的图标

你还可以针对校验状态为输入框添加额外的图标。只需设置相应的 `.has-feedback` 类并添加正确的图标即可。

反馈图标 (feedback icon) 只能使用在文本输入框 `<input class="form-control">` 元素上。
需要导入font字体库

```

<div class="form-group has-success has-feedback">
  <label class="control-label" for="inputSuccess2">Input with success</label>
  <input type="text" class="form-control" id="inputSuccess2" aria-
describedby="inputSuccess2Status">
  <span class="glyphicon glyphicon-ok form-control-feedback" aria-hidden="true"></span>
  <span id="inputSuccess2Status" class="sr-only">(success)</span>
</div>

```

2.6.8 按钮

`class = btn` 将任何东西变成按钮 需要配合 `btn-default` a标签也可以

1. 按钮颜色

```

<!-- 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>

```

2. 成组button

```

<div class="btn-group" data-toggle="buttons">
    <label class="btn btn-primary">
        <input type="radio" name="options" id="option1"> 选项 1
    </label>
    <label class="btn btn-primary">
        <input type="radio" name="options" id="option2"> 选项 2
    </label>
    <label class="btn btn-primary">
        <input type="radio" name="options" id="option3"> 选项 3
    </label>
</div>

```

2.6.9 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

```


<p>
    <button type="button" class="btn btn-primary btn-lg"> (大按钮) Large button</button>
    <button type="button" class="btn btn-default btn-lg"> (大按钮) Large button</button>
</p>
<p>
    <button type="button" class="btn btn-primary"> (默认尺寸) Default button</button>
    <button type="button" class="btn btn-default"> (默认尺寸) Default button</button>
</p>
<p>
    <button type="button" class="btn btn-primary btn-sm"> (小按钮) Small button</button>
    <button type="button" class="btn btn-default btn-sm"> (小按钮) Small button</button>
</p>
<p>
    <button type="button" class="btn btn-primary btn-xs"> (超小尺寸) Extra small
button</button>

    <button type="button" class="btn btn-default btn-xs"> (超小尺寸) Extra small

```

```
button</button>
</p>
```

2.6.10 图片形状

响应式图片!随着窗体大小改变大小  Responsive image 通过为 元素添加以下相应的类, 可以让图片呈现不同的形状。

```



```

2.6.11 辅助类

一. 情境文本颜色

通过颜色来展示意图, Bootstrap 提供了一组工具类。这些类可以应用于链接, 并且在鼠标经过时颜色还可以加深, 就像默认的连接一样。

```
Fusce dapibus, tellus ac cursus commodo, tortor mauris nibh.
Nullam id dolor id nibh ultricies vehicula ut id elit.
Duis mollis, est non commodo luctus, nisi erat porttitor ligula.
Maecenas sed diam eget risus varius blandit sit amet non magna.
Etiam porta sem malesuada magna mollis euismod.
Donec ullamcorper nulla non metus auctor fringilla.
```

```
<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>
```

二. 情境背景色

和情境文本颜色类一样, 使用任意情境背景色类就可以设置元素的背景。链接组件在鼠标经过时颜色会加深, 就像上面所讲的情境文本颜色类一样。

```
<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>
```

2.6.12 浮动

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

2.6.13 可见类

	(<768px)	(≥768px)	(≥992px)	(≥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

2.7 组件

无数可复用的组件，包括字体图标、下拉菜单、导航、警告框、弹出框等更多功能

2.7.1 图片使用Glyphicons 字体图标

```
<span class="glyphicon glyphicon-save-file"></span>
```

2.7.2 下拉菜单

```
<div class="dropdown">
  <!-- 下拉框 触发按钮 -->
  <button class="btn btn-success dropdown-toggle" type="button" id="dropdownMenu1"
data-toggle="dropdown" aria-haspopup="true" aria-expanded="true">
    下拉框
    <!-- 下拉框图片 -->
    <span class="glyphicon glyphicon-arrow-down"></span>
  </button>
  <!-- 下拉菜单 -->
  <ul class="dropdown-menu" aria-labelledby="dropdownMenu1">
    <li><a href="#">Action</a></li>
    <li><a href="#">Another action</a></li>
    <li><a href="#">Something else here</a></li>
    <li role="separator" class="divider"></li>
    <li><a href="#">Separated link</a></li>
  </ul>
</div>
```

2.7.3 按钮组

把一系列的.btn按钮放入.btn-group。

btn-group 内部嵌套 btn

```
<div class="btn-group" role="group" aria-label="...">
  <button type="button" class="btn btn-default">Left</button>
  <button type="button" class="btn btn-default">Middle</button>
  <button type="button" class="btn btn-default">Right</button>
</div>
```

2.7.4 toolbar 按钮工具栏

把一组 `<div class="btn-group">` 组合进一个 `<div class="btn-toolbar">` 中就可以做成更复杂的组件

```
<div class="btn-toolbar" role="toolbar" aria-label="...">
  <div class="btn-group" role="group" aria-label="...">...</div>
  <div class="btn-group" role="group" aria-label="...">...</div>
  <div class="btn-group" role="group" aria-label="...">...</div>
</div>
```

2.7.5 尺寸

只要给 `.btn-group` 加上 `.btn-group-*` 类, 就省去为按钮组中的每个按钮都赋予尺寸类了, 如果包含了多个按钮组时也适用。

```
<div class="btn-group btn-group-lg" role="group" aria-label="...">...</div>
<div class="btn-group" role="group" aria-label="...">...</div>
<div class="btn-group btn-group-sm" role="group" aria-label="...">...</div>
<div class="btn-group btn-group-xs" role="group" aria-label="...">...</div>
```

2.7.6 带分割线的下拉框

```
<div class="btn-group">
  <button id="show" type="button" class="btn btn-danger">Action</button>
  <button type="button" class="btn btn-danger dropdown-toggle" data-toggle="dropdown"
    aria-haspopup="true" aria-expanded="false">
    <span class="caret"></span>
    <span class="sr-only">Toggle Dropdown</span>
  </button>
  <ul id="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 role="separator" class="divider"></li>
    <li><a href="#">Separated link</a></li>
  </ul>
</div>
</div>
```

点击切换button显示内容

```
var $arr=$("#ul li");
$.each($arr, function() {
  $(this).click(function(){
    var $t = $(this).text();
    ($("#show").text(t);
  });
});
```

2.7.7 标签式导航栏

```
<li role="presentation" class="active"><a href="#" data-toggle="tab">Home</a></li>
```

```

<ul class="nav nav-tabs">
  <li role="presentation" class="active"><a href="#" data-toggle="tab">Home</a></li>
  <li role="presentation"><a href="#" data-toggle="tab">Profile</a></li>
  <li role="presentation"><a href="#" data-toggle="tab">Messages</a></li>
</ul>

<div id="myTabContent" class="tab-content">
  <div class="tab-pane fade in active" id="h5">
    <p>Html5最近比较火</p>
  </div>
  <div class="tab-pane fade" id="java">
    <p>java是高级语言，是最好的语言</p>
  </div>
  <div class="tab-pane fade" id="android">
    <p>android是最受大众欢迎的智能机品牌</p>
  </div>
</div>

```

2.7.8 胶囊导航

```

<ul class="nav nav-pills ">
  <li class="active"><a href="#h5" data-toggle="tab">HTML5</a></li>
  <li><a href="#java" data-toggle="tab">JAVAEE</a></li>
  <li><a href="#android" data-toggle="tab">ANDROID</a></li>
</ul>
<div id="myTabContent" class="tab-content">
  <div class="tab-pane fade in active" id="h5">
    <p>Html5最近比较火</p>
  </div>
  <div class="tab-pane fade" id="java">
    <p>java是高级语言，是最好的语言</p>
  </div>
  <div class="tab-pane fade" id="android">
    <p>android是最受大众欢迎的智能机品牌</p>
  </div>
</div>

```

2.7.9 导航条

data-toggle="tab"

倒航条自带响应效果 缩小显示成手机效果

```
<li role="presentation" class="active"><a href="#" data-toggle="tab">Home</a></li>
```

2.7.10 路径导航 面包屑导航

```

<ol class="breadcrumb">
  <li><a href="#">Home</a></li>
  <li><a href="#">2013</a></li>
  <li class="active">十一月</li>
</ol>

```

2.7.11 标签


```

<span class="label label-default">Default</span>
<span class="label label-primary">Primary</span>
<span class="label label-success">Success</span>
<span class="label label-info">Info</span>
<span class="label label-warning">Warning</span>
<span class="label label-danger">Danger</span>

```

2.7.12 徽章 提示消息数

```

<a href="#">Inbox <span class="badge">42</span></a>
<button class="btn btn-primary" type="button">
  Messages <span class="badge">4</span>
</button>

```

2.7.13 巨幕

```

<div class="jumbotron">
  <h1>Hello, world!</h1>
  <p>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</p>
  <p><a class="btn btn-primary btn-lg" href="#" role="button">Learn more</a></p>
</div>

```

2.7.14 缩略图

```

<div class="container" style="margin-top: 30px;">
  <div class="row">
    <div class="col-md-4">
      <a href="#" class="thumbnail"> </img></a>
      <div class="caption">
        <h4>HTML入门</h4>
        <h6>html是最好的静态网页语言</h6>
      </div>
    </div>
  </div>
</div>

```

2.7.15 进度条

```

<div class="progress">
  <div id="progressbar" class="progress-bar progress-bar-success"
    role="progressbar" aria-valuenow="60" aria-valuemin="0"
    aria-valuemax="100" style="width: 10%;">
    <span class="sr-only">90% 完成 (成功) </span>
  </div>
</div>

```

2.7.16 well

```

<div class="well well-lg">您好, 我在大的 Well 中! </div>
<div class="well well-sm">您好, 我在小的 Well 中! </div>

```

2.8 jQuery插件

aria-labelledby属性

当想要的标签文本已在其他元素中存在时，可以使用aria-labelledby，并将其值为所有读取的元素的id。如下：

当u1获取到焦点时，屏幕阅读器是会读：“选择您的职位”

但是如果我们没有给输入框设置label时，当其获得焦点时，屏幕阅读器会读出aria-label属性的值，aria-label不会在视觉上呈现效果。

HTML5针对html tag增加的属性：role 和 aria-*。

role的作用是描述一个非标准的tag的实际作用。比如用div做button，那么设置div 的 role="button"，辅助工具就可以认出这实际上是个button。

ARIA Roles

Use the ARIA role attribute to indicate that a generic tag is playing the role of a standard widget like a button.

而aria-*的作用就是描述这个tag在可视化的情境中的具体信息。比如，

```
<div role="checkbox" aria-checked="checked"></div>
```

辅助工具就会知道，这个div实际上是个checkbox的角色，为选中状态。

Add ARIA for screen readers

ARIA attributes provides semantic information to screen readers that is normally conveyed visually.

Note that using ARIA does not automatically implement the standard widget behavior, you'll still need to add focus management and keyboard navigation yourself.

2.8.1 弹出框

```
<button type="button" class="btn btn-default" title="Popover title"
        data-container="body" data-toggle="popover" data-placement="left"
        data-content="左侧的 Popover 中的一些内容">
    左侧的 Popover
</button>

$(function () {
    $('[data-toggle="popover"]').popover();
});
```

2.8.2 提示框

这是一个 默认的 Tooltip.

这是一个 左侧的 Tooltip.

这是一个 顶部的 Tooltip.

```
<script>
$(function () { $('[data-toggle="tooltip"]').tooltip(); });
</script>
```

2.8.3 轮播图

默认五秒自动轮播

```
<!-- Indicators
      需要几张图片!就需要几个 li
      还需要几个class等于item的div
-->
<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>
</ol>
<!-- Wrapper for slides -->
<div class="carousel-inner" role="listbox">
  <div class="item active">
    
    <!-- 文字描述-->
    <div class="carousel-caption">
      ...
    </div>
  </div>
</div>
```

自定义轮播

```
$('.carousel').carousel({
  interval: 2000
})
```

作业题

- 1、基于Bootstrap实现一个统一的后台管理系统页面

面试题

- 1、为什么使用bootstrap?
- 2、什么是bootstrap栅格系统?
- 3、编写一段响应式表格代码?