目录

[什么是库？ 5](#_Toc535344946)

[什么是框架？ 5](#_Toc535344947)

[BootStrap框架介绍 6](#_Toc535344948)

[模板 6](#_Toc535344949)

[栅格系统的介绍 8](#_Toc535344950)

[容器： 8](#_Toc535344951)

[概念： 8](#_Toc535344952)

[改变视口看看下面代码在什么时候堆叠和水平排列 9](#_Toc535344953)

[删格的组合使用 9](#_Toc535344954)

[组合使用的案例 9](#_Toc535344955)

[列偏移 10](#_Toc535344956)

[组合偏移 10](#_Toc535344957)

[列排序 11](#_Toc535344958)

[列排序和列偏移的区别 11](#_Toc535344959)

[删格嵌套 12](#_Toc535344960)

[清除浮动 12](#_Toc535344961)

[响应式工具 13](#_Toc535344962)

[概念 13](#_Toc535344963)

[可见工具类 13](#_Toc535344964)

[隐藏工具类 13](#_Toc535344965)

[打印工具类 13](#_Toc535344966)

[快速浮动 14](#_Toc535344967)

[固定定位 14](#_Toc535344968)

[案例-仿天猫右侧导航 14](#_Toc535344969)

[字体图标Glyphicons 14](#_Toc535344970)

[预定义样式（风格） 14](#_Toc535344971)

[一般元素的预定义风格 15](#_Toc535344972)

[组件的预定义风格 15](#_Toc535344973)

[案例-登录界面的开发 16](#_Toc535344974)

[按钮 16](#_Toc535344975)

[基类样式 16](#_Toc535344976)

[扩展样式 16](#_Toc535344977)

[大小样式 17](#_Toc535344978)

[状态样式 17](#_Toc535344979)

[哪几种元素做按钮 17](#_Toc535344980)

[按钮组 17](#_Toc535344981)

[对齐按钮组（端点 垂直） 18](#_Toc535344982)

[按钮组的大小 18](#_Toc535344983)

[按钮后面加箭头（caret） 19](#_Toc535344984)

[组里加箭头 19](#_Toc535344985)

[改变箭头方向 19](#_Toc535344986)

[案例-github的按钮组 19](#_Toc535344987)

[下拉菜单组件 20](#_Toc535344988)

[下拉菜单依赖 JavaScript 插件 20](#_Toc535344989)

[无障碍的富互联网应用 20](#_Toc535344990)

[不考虑特殊人群 20](#_Toc535344991)

[默认打开弹出菜单 21](#_Toc535344992)

[让菜单整体右边对齐 21](#_Toc535344993)

[让菜单右边对齐 21](#_Toc535344994)

[让菜单和父级一样宽 21](#_Toc535344995)

[菜单的头和分割线 21](#_Toc535344996)

[菜单向上展开 22](#_Toc535344997)

[案例带运动的菜单 22](#_Toc535344998)

[PC机上鼠标滑过显示菜单移动设备上点击弹出菜单 22](#_Toc535344999)

[标签页(选项卡) 22](#_Toc535345000)

[标签页头部导航 23](#_Toc535345001)

[端点对齐 23](#_Toc535345002)

[下边框自适应宽 23](#_Toc535345003)

[胶囊式（药丸） 23](#_Toc535345004)

[垂直式（堆栈式导航） 23](#_Toc535345005)

[带下拉菜单的标签页 24](#_Toc535345006)

[标签页内容部分 24](#_Toc535345007)

[美化 24](#_Toc535345008)

[淡入淡出的切换 25](#_Toc535345009)

[鼠标滑过改变选项卡 25](#_Toc535345010)

[带下拉菜单的选项卡 26](#_Toc535345011)

[导航条组件 27](#_Toc535345012)

[导航条基本结构 27](#_Toc535345013)

[流体布局中的导航条 27](#_Toc535345014)

[开发时用的导航条结构 27](#_Toc535345015)

[消除导航条的圆角 28](#_Toc535345016)

[固定定位在顶部的导航条 28](#_Toc535345017)

[固定定位在底部的导航条 28](#_Toc535345018)

[固定定位的导航条遮挡内容 29](#_Toc535345019)

[复杂的导航条 29](#_Toc535345020)

[导航条添加logo标志 29](#_Toc535345021)

[左右两个导航条 30](#_Toc535345022)

[在导航条中添加按钮、文字、链接 30](#_Toc535345023)

[在导航条中添加搜索框 30](#_Toc535345024)

[在导航条中添加下拉菜单 30](#_Toc535345025)

[响应式菜单导航（超小设备下） 31](#_Toc535345026)

[滚动监听 32](#_Toc535345027)

[bootstrap中JS插件 33](#_Toc535345028)

[插件的使用方式 33](#_Toc535345029)

[如何设置插件的选项 33](#_Toc535345030)

[使用JavaScriptAPI 34](#_Toc535345031)

[自定义事件 34](#_Toc535345032)

[防止插件冲突 34](#_Toc535345033)

[附加导航插件affix 34](#_Toc535345034)

# 什么是库？

库是封装好的一系列方法的集合

以jQurey为例，它就是轻量级的库（用来放置一些定义好的代码），我们可以利用这个封装好的库去完成我们的想要的一些功能，甚至是我们可以去改写一些代码，库的使用几乎是没有什么约束的。

# 什么是框架？

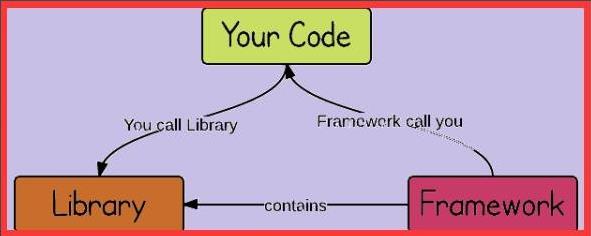
为了**解决某种问题**而设计的**开发结构**（具有一定约束性）

【优点】：

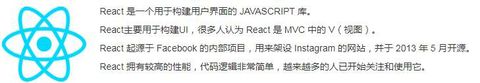
* 更快更好的完成项目任务。
* 维护性好
* 稳定性安全性好

【缺点】：

* 解决某种问题，不是所有问题
* 具有约束性，有自己定义的规则，入门难
* 代码膨胀



React就是一个库，但是呢React和react-router, react-redux三者结合在一起的话那就是一个框架了



# BootStrap框架介绍

BootStrap是Twitter开发的一种Web框架，用于更快地设计响应式网页和应用程序。

计算机引导启动的单词是 boot ，可是，boot原意是靴子，启动与靴子有和关系？

原来，boot是bootstrap的缩写，它来自西方一句 拉鞋带 的谚语  pull oneself up by one's bootstraps  ，翻译为 拽着鞋带把自己拉起来，这和计算机启动很相似

计算机启动是一个很矛盾的过程：必须先运行程序，然后计算机才能启动，但是计算机不启动就无法运行程序——就像鸡生蛋、蛋生鸡一样！

所以，工程师就把这个启动过程叫做”拉鞋带‘’，久而久之就简称为boot了。

1、下载安装BootStrap

2、BootStrap的构成

模板、栅格系统、排版、表格、表单、按钮、图片、组件、JS插件等元素

# 模板

<!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<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 101 Template</title>

<!-- Bootstrap -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css" rel="stylesheet">

<!-- HTML5 shim 和 Respond.js 是为了让 IE8 支持 HTML5 元素和媒体查询（media queries）功能 -->

<!-- 警告：通过 file:// 协议（就是直接将 html 页面拖拽到浏览器中）访问页面时 Respond.js 不起作用 -->

<!--[if lt IE 9]>

<script src="https://cdn.jsdelivr.net/npm/html5shiv@3.7.3/dist/html5shiv.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/respond.js@1.4.2/dest/respond.min.js"></script>

<![endif]-->

</head>

<body>

<h1>你好，世界！</h1>

<!-- jQuery (Bootstrap 的所有 JavaScript 插件都依赖 jQuery，所以必须放在前边) -->

<script src="https://cdn.jsdelivr.net/npm/jquery@1.12.4/dist/jquery.min.js"></script>

<!-- 加载 Bootstrap 的所有 JavaScript 插件。你也可以根据需要只加载单个插件。 -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/js/bootstrap.min.js"></script>

</body>

</html>

用自己写的简洁一点的模板便于演示，项目上线在换成这个模板。

<!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<!-- Bootstrap -->

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

</head>

<body>

<h1>Hello, world!</h1>

<script src="js/jquery.js"></script>

<script src="js/bootstrap.min.js"></script>

</body>

</html>

# 栅格系统的介绍

## 容器：

container-fluid：流体布局

container 固定宽布局

1170px 970px 750px auto

视口的宽>=1200时 container的width=1170 大PC机

视口的宽=（992~1199）时 container的width=970 小PC机

视口的宽=（768~991）时 container的width=750 pad

视口的宽=（0~767）时 container的width=auto 移动设备

## 概念：

将容器分为12列

row 表示行

col 表示列

col-device-number 代表了具体占几格

lg:large 大设备

md:middle 中等设备

sm:small 小型设备

xs:xsmall 超小型设备

number：删格的个数

## 改变视口看看下面代码在什么时候堆叠和水平排列

<div class="container">

<div class="row">

<div class="col-lg-4">col-lg-4</div>

<div class="col-lg-4">col-lg-4</div>

<div class="col-lg-4">col-lg-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-sm-4">col-sm-4</div>

<div class="col-sm-4">col-sm-4</div>

<div class="col-sm-4">col-sm-4</div>

</div>

<div class="row">

<div class="col-xs-4">col-xs-4</div>

<div class="col-xs-4">col-xs-4</div>

<div class="col-xs-4">col-xs-4</div>

</div>

</div>

## 删格的组合使用

<div class="container">

<div class="row">

<div class="col-lg-4 col-md-6">col-lg-4</div>

<div class="col-lg-4 col-md-6">col-lg-4</div>

<div class="col-lg-4 col-md-6">col-lg-4</div>

</div>

</div>

## 组合使用的案例

<div class="container">

<div class="row">

<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

<div class="thumbnail">

<img src="../img/codeguide.png">

<div class="caption">

<h3>abc</h3>

<p>description</p>

</div>

</div>

</div>

<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

<div class="thumbnail">

<img src="../img/codeguide.png">

<div class="caption">

<h3>abc</h3>

<p>description</p>

</div>

</div>

</div>

<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

<div class="thumbnail">

<img src="../img/codeguide.png">

<div class="caption">

<h3>abc</h3>

<p>description</p>

</div>

</div>

</div>

<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

<div class="thumbnail">

<img src="../img/codeguide.png">

<div class="caption">

<h3>abc</h3>

<p>description</p>

</div>

</div>

</div>

</div>

</div>

## 列偏移

<div class="container">

<div class="row">

<div class="col-lg-4">col-lg-4</div>

<div class="col-lg-4">col-lg-4</div>

<div class="col-lg-4">col-lg-4</div>

</div>

<div class="row">

<div class="col-lg-4 col-lg-offset-4">col-lg-4</div>

</div>

</div>

## 组合偏移

<div class="container">

<div class="row">

<div class="col-lg-4">col-lg-4</div>

<div class="col-lg-4">col-lg-4</div>

<div class="col-lg-4">col-lg-4</div>

</div>

<div class="row">

<div class="col-lg-4 col-lg-offset-4 col-md-offset-4">lg</div>

</div>

</div>

<div class="container">

<div class="row">

<div class="col-lg-4">col-lg-4</div>

<div class="col-lg-4">col-lg-4</div>

<div class="col-lg-4">col-lg-4</div>

</div>

<div class="row">

<div class="col-lg-4 col-lg-offset-4 col-md-4 col-md-offset-6">lg</div>

</div>

</div>

## 列排序

<div class="container">

<div class="row">

<div class="col-lg-2">col-lg-4</div>

<div class="col-lg-10">col-lg-4</div>

</div>

<div class="row">

<div class="col-lg-2 col-lg-push-10">col-lg-2</div>

<div class="col-lg-10 col-lg-pull-2">col-lg-10</div>

</div>

</div>

push往后推 pull往前拉

## 列排序和列偏移的区别

<!-- 偏移和排序的区别 -->

<div class="row">

<div class="col-lg-2 col-lg-offset-10">col-lg-offset-10</div>

<div class="col-lg-2 col-lg-push-10">col-lg-push-10</div>

</div>

<!-- 偏移和排序的区别 -->

<div class="row">

<div class="col-lg-2 col-lg-offset-10">col-lg-offset-10</div>

<div class="col-lg-10">col-lg-10</div>

</div>

<div class="row">

<div class="col-lg-2 col-lg-push-10">col-lg-push-10</div>

<div class="col-lg-10">col-lg-10</div>

</div>

<!-- 偏移和排序的区别 -->

<div class="row">

<div class="col-lg-4">col-lg-4</div>

<div class="col-lg-4 col-lg-offset-5">col-lg-4</div>

</div>

<div class="row">

<div class="col-lg-4">col-lg-4</div>

<div class="col-lg-4 col-lg-push-5">col-lg-4</div>

</div>

1. 列偏移只能往右走
2. 多个列存在的情况列偏移至末端放不下会换行，而列排序到末端放不下会超出容器

## 删格嵌套

<div class="row">

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

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

<div class="row">

<div class="col-lg-2">lg-2</div>

<div class="col-lg-2">lg-2</div>

<div class="col-lg-8">lg-2</div>

</div>

</div>

</div>

嵌套在列中的行布局是按照列的宽度进行重新的12列布局

## 清除浮动

<div class="container">

<div class="row">

<div class="col-lg-2">col-lg-2col-lg-2col-lg-2col-lg-2col-lg-2col-lg-2col-lg-2col-lg-2col-lg-2</div>

<div class="col-lg-10">col-lg-10</div>

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

<div class="col-lg-4">col-lg-4</div>

</div>

</div>

# 响应式工具

## 概念

针对不同设备展示或隐藏页面内容,针对打印机显示或隐藏内容的工具类。

【打开】<https://v3.bootcss.com/css/> 改变视口看右侧导航的显示与隐藏

## 可见工具类

visible-\*-block visible-\*-inline visible-\*-inline-block

<div class="row">

<div class="col-lg-2 visible-lg-block">col-lg-2</div>

</div>

## 隐藏工具类

hidden-\*

<div class="row">

<div class="hidden-sm hidden-sx">col-lg-2</div>

</div>

## 打印工具类

.visible-print-block

.visible-print-inline

.visible-print-inline-block  浏览器隐藏  打印机可见

.hidden-print 浏览器可见 打印机隐藏

<div class="row">

<div class="visible-print-block">print</div>

<div class="hidden-print">hidden-print</div>

</div>

# 快速浮动

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

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

# 固定定位

<div class="affix">affix</div>

# 案例-仿天猫右侧导航

.tips1{width: 30px;height: 300px;background:black; }

.tips2{width: 30px;height: 100px;background:rgb(0, 38, 255);}

<div class="container-fluid">

<div class="row">

<div class="tips1 pull-right visible-lg-block affix"></div>

<div class="tips2 pull-right hidden-lg affix"></div>

</div>

</div>

# 字体图标Glyphicons

好处：

1. 减少http请求
2. 容易控制样式

用法：

* font-face
* 字体路径

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

# 预定义样式（风格）

* default(默认)
* primary（首选项）
* success（成功）
* info（一般信息）
* warning（警告）
* danger（危险）

class=”btn btn-primary”

calss=”bg-primary”

class=”text-primary”

## 一般元素的预定义风格

<div class="container">

<br>

<input type="button" value="默认样式" class="btn btn-default">

<input type="button" value="首选项" class="btn btn-primary">

<input type="button" value="成功" class="btn btn-success">

<input type="button" value="一般信息" class="btn btn-info">

<input type="button" value="警告" class="btn btn-warning">

<input type="button" value="危险" class="btn btn-danger">

<br><br>

<p>默认段落</p>

<p class="bg-primary">首选项</p>

<p class="bg-success">成功</p>

<p class="bg-info">一般信息</p>

<p class="bg-warning">警告</p>

<p class="bg-danger">危险</p>

<br><br>

<span>默认文字</span>

<span class="text-primary">首选项</span>

<span class="text-success">成功</span>

<span class="text-info">一般信息</span>

<span class="text-warning">警告</span>

<span class="text-danger">危险</span>

<br><br>

<a href="#" class="btn btn-primary">链接1</a>

<a href="#" class="bg-primary">链接2</a>

<a href="#" class="text-primary">链接3</a>

</div>

## 组件的预定义风格

警告框的风格

<div class="alert alert-warning">我是警告框</div>

<div class="alert alert-danger">我是危险信息</div>

面板的风格

<div class="panel panel-primary">

<div class="panel-heading">我是面板头</div>

<div class="panel-body">我是面板的身体</div>

</div>

## 案例-登录界面的开发

<div id="login" class="panel panel-primary">

<div class="panel-heading">Login</div>

<div class="panel-body">

<form action="">

<div class="form-group">

<label for="">用户名</label>

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

<div class="alert alert-warning">用户名不能为空</div>

<label for="">密码</label>

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

</div>

<a href="#" class="text-info">忘记密码？</a>

<input type="submit" value="登录" class="pull-right btn btn-primary">

</form>

</div>

</div>

# 按钮

* 基类样式

btn

* 扩展样式

btn-default（默认）

btn-link（链接） btn-block(块级)

* 大小样式

btn-\*[lg,sm,xs]

* 状态样式

active

disabled

## 基类样式

<input type="button" value="默认样式">

<input type="button" value="默认样式" class="btn">

## 扩展样式

<input type="button" value="默认样式" class="btn btn-default">

<input type="button" value="链接按钮" class="btn btn-link">

## 大小样式

<input type="button" value="首选项" class="btn btn-primary btn-lg">

<input type="button" value="首选项" class="btn btn-primary">

<input type="button" value="首选项" class="btn btn-primary btn-sm">

<input type="button" value="首选项" class="btn btn-primary btn-xs">

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

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

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

## 状态样式

<input type="button" value="首选项" class="btn btn-primary">

<input type="button" value="首选项" class="btn btn-primary active">

<input type="button" value="首选项" class="btn btn-primary disabled">

## 哪几种元素做按钮

a, input, button

<a href="" class="btn btn-primary">button</a>

<input type="button" class="btn btn-primary" value="button">

<button class="btn btn-primary">button</button>

# 按钮组

<br><!-- 没有组 -->

<input type="button" class="btn btn-primary" value="button1">

<input type="button" class="btn btn-primary" value="button2">

<input type="button" class="btn btn-primary" value="button3">

<br><br>

<!-- 普通按钮组 -->

<div class="btn-group">

<input type="button" class="btn btn-primary" value="button1">

<input type="button" class="btn btn-primary" value="button2">

<input type="button" class="btn btn-primary" value="button3">

</div>

<br><br>

<!-- 链接按钮组 -->

<div class="btn-group">

<a href="#" class="btn btn-primary">button1</a>

<a href="#" class="btn btn-primary">button2</a>

<a href="#" class="btn btn-primary">button3</a>

</div>

## 对齐按钮组（端点 垂直）

<!-- 普通按钮组 端点对齐-->

<div class="btn-group btn-group-justified">

<input type="button" class="btn btn-primary" value="button1">

<input type="button" class="btn btn-primary" value="button2">

<input type="button" class="btn btn-primary" value="button3">

</div>

<!-- 普通按钮组 端点对齐-->

<div class="btn-group btn-group-justified">

<div class="btn-group">

<input type="button" class="btn btn-primary" value="button1">

</div>

<div class="btn-group">

<input type="button" class="btn btn-primary" value="button2">

</div>

<div class="btn-group">

<input type="button" class="btn btn-primary" value="button3">

</div>

</div>

<!-- 链接按钮组 端点对齐 -->

<div class="btn-group btn-group-justified">

<a href="#" class="btn btn-primary">button1</a>

<a href="#" class="btn btn-primary">button2</a>

<a href="#" class="btn btn-primary">button3</a>

</div>

<!-- 链接按钮组 垂直对齐 右上角不圆-->

<div class="btn-group btn-group-vertical">

<a href="#" class="btn btn-primary">button1</a>

<a href="#" class="btn btn-primary">button2</a>

<a href="#" class="btn btn-primary">button3</a>

</div>

<br><br>

<!-- 链接按钮组 垂直对齐 -->

<div class="btn-group-vertical">

<a href="#" class="btn btn-primary">button1</a>

<a href="#" class="btn btn-primary">button2</a>

<a href="#" class="btn btn-primary">button3</a>

</div>

## 按钮组的大小

<div class="btn-group-vertical btn-group-lg">

<a href="#" class="btn btn-primary">button1</a>

<a href="#" class="btn btn-primary">button2</a>

<a href="#" class="btn btn-primary">button3</a>

</div>

## 按钮后面加箭头（caret）

<a href="#" class="btn btn-primary">按钮 <span class="caret"></span></a>

<button class="btn btn-primary">按钮 <span class="caret"></span></button>

<!-- input定义的按钮不适合加箭头 -->

<input type="button" value="按钮" class="btn btn-primary"><span class="caret"></span>

### 组里加箭头

<div class="btn-group">

<a href="" class="btn btn-primary">按钮</a>

<a href="" class="btn btn-primary"><span class="caret"></span></a>

</div>

### 改变箭头方向

<div class="btn-group **dropup**">

<a href="" class="btn btn-primary">按钮</a>

<a href="" class="btn btn-primary"><span class="caret"></span></a>

</div>

**箭头方向默认是dropdown,类样式加在按钮组父类中**

## 案例-github的按钮组

<div class="btn-group btn-group-sm">

<button class="btn btn-default active" style="font-weight: 700">

<span class="glyphicon glyphicon-eye-open"></span>

Watch

</button>

<button class="btn btn-default">3,345</button>

</div>

<div class="btn-group btn-group-sm">

<button class="btn btn-default active" style="font-weight: 700">

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

Star

</button>

<button class="btn btn-default">3,345</button>

</div>

<div class="btn-group btn-group-sm">

<button class="btn btn-default active" style="font-weight: 700">

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

Fork

</button>

<button class="btn btn-default">3,345</button>

</div>

# 下拉菜单组件

## 下拉菜单依赖 JavaScript 插件

<div class="dropdown">

<button class="btn btn-default dropdown-toggle" type="button" id="dropdownMenu1" **data-toggle**="dropdown" **aria-haspopup**="true" **aria-expanded**="true">

Dropdown

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

</button>

<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>

**<script src="./js/jquery-1.11.1.js"></script>**

**<script src="./js/bootstrap.js"></script>**

## 无障碍的富互联网应用

**data-\* aria-\* role这三种属性都是html5新增的自定义属性**

**data-\***：这样的属性是用来和JS交互的(用于存储页面或应用特有的定制化数据)

**role和aria-\*：**这两个属性是用来给辅助设备（屏幕阅读器）识别的

role：属性是用来告诉屏幕阅读器设备，你现在读取到的是什么角色

**aria-\*：**属性是用来告诉屏幕阅读器设备，你现在读取到的角色处于什么状态

WAI-ARIA（Web Accessibility Initiative’s Accessible Rich Internet Applications， 无障碍网页倡议– 无障碍的富互联网应用，也简称ARIA）是一种技术规范，自称“有桥梁作用的技术”。之所以这样说，是因为在HTML提供相应的语义功能之前，它会使用属性来填补一些语义上的空白。（p56 3.13）

## 不考虑特殊人群

<div class="**dropdown**">

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

前端开发

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

</button>

<ul class="**dropdown-menu**">

<li><a href="#">HTML</a></li>

<li><a href="#">CSS</a></li>

<li><a href="#">JavaScrip</a></li>

</ul>

</div>

## 默认打开弹出菜单

<div class="dropdown **open**">…</div>

## 让菜单整体右边对齐

<div class="dropdown open **pull-right**">…</div>

## 让菜单右边对齐

<button class="btn btn-default dropdown-toggle **btn-block**" type="button" data-toggle="dropdown">

前端开发

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

</button>

<ul class="dropdown-menu **dropdown-menu-righ**t">

弹出的菜单绝对定位是不占据文档流的位置的

## 让菜单和父级一样宽

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

前端开发

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

</button>

<ul class="dropdown-menu dropdown-menu-right **btn-block** ">

## 菜单的头和分割线

<li class="**dropdown-header**">基础内容</li>

<li><a href="#">HTML</a></li>

<li><a href="#">CSS</a></li>

<li><a href="#">JS</a></li>

<div class="***divider***"></div><!-- 分割线 -->

<li class="**dropdown-header**">框架内容</li>

<li><a href="#">jQuery</a></li>

<li><a href="#">bootstrap</a></li>

## 菜单向上展开

<div class="dropup">…</div>

## 案例带运动的菜单

.my-btn-style button{width: 84px; height: 50px; border:none; background:#f6f8f8;}

.my-btn-style .dropdown-menu img{margin:5px;}

.my-btn-style .dropdown-menu{animation:fademove 0.5s linear;}

@keyframes fademove{

0%{opacity: 0; transform:translateY(-20px);}

100%{opacity: 1; transform:translateY(0px);}

}

<div class="dropdown **my-btn-style**">…</a>

## PC机上鼠标滑过显示菜单移动设备上点击弹出菜单

if(window.navigator.userAgent.toLowerCase().indexOf("mobile")==-1)

{

var timer=null;

$(".dropdown-toggle").hover(function(){

clearTimeout(timer);

$(".dropdown").addClass("open");

},function(){

timer=setTimeout(function(){

$(".dropdown").removeClass("open");

},1000)

});

$(".dropdown-menu").hover(function(){

clearTimeout(timer)

},function(){

timer=setTimeout(function(){

$(".dropdown").removeClass("open");

},1000)

});

}

# 标签页(选项卡)

.nav 基类

.nav-tabs

.nav-pills

.nav-justified

.nav-tabs-justified

.nav-stacked

## 标签页头部导航

<div class="container">

<br><br>

<ul class="nav nav-tabs">

<li class="active"><a href="">HTML</a></li>

<li><a href="">CSS</a></li>

<li><a href="">BS</a></li>

</ul>

</div>

### 端点对齐

<br><br>

<ul class="nav nav-tabs nav-justified" style="width:400px;">

<li class="active"><a href="">HTML</a></li>

<li><a href="">CSS</a></li>

<li><a href="">BS</a></li>

</ul>

**在xs设备下会堆叠**

### 下边框自适应宽

<br><br>

<ul class="nav nav-tabs nav-tabs-justified">

<li class="active"><a href="">HTML</a></li>

<li><a href="">CSS</a></li>

<li><a href="">BS</a></li>

</ul>

**在xs设备下不会堆叠**

### 胶囊式（药丸）

<br><br>

<ul class="nav nav-pills">

<li class="active"><a href="">HTML</a></li>

<li><a href="">CSS</a></li>

<li><a href="">BS</a></li>

</ul>

<br><br>

<ul class="nav nav-pills nav-justified" style="width:400px;">

<li class="active"><a href="">HTML</a></li>

<li><a href="">CSS</a></li>

<li><a href="">BS</a></li>

</ul>

### 垂直式（堆栈式导航）

<br><br>

<ul class="nav nav-pills nav-stacked" style="width: 400px;">

<li class="active"><a href="">HTML</a></li>

<li><a href="">CSS</a></li>

<li><a href="">BS</a></li>

</ul>

**最好是 nav-pills和nav-stacked配合**

### 带下拉菜单的标签页

<ul class="nav nav-tabs" style="width: 400px;">

<li class="active"><a href="">HTML</a></li>

<li><a href="">CSS</a></li>

<li class="dropdown">

<a href="" class="dropdown-toggle" data-toggle="dropdown">

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

</a>

<ul class="dropdown-menu">

<li><a href="">模板</a></li>

<li><a href="">删格</a></li>

<li><a href="">组件</a></li>

<li><a href="">插件</a></li>

</ul>

</li>

</ul>

## 标签页内容部分

.tab-content

.tab-pane

a标签的交互属性 data-toggle=tab

href对应的id

淡入淡出的切换

鼠标滑过改变选项卡

带下拉菜单的选项卡

<div class="container">

<br><br>

<ul class="nav nav-tabs">

<li class="active"><a href="#a" data-toggle="tab">HTML</a></li>

<li><a href="#b" data-toggle="tab">CSS</a></li>

<li><a href="#c" data-toggle="tab">BS</a></li>

</ul>

<ul class="**tab-content**">

<li id="a" class="**tab-pane active**">aaaa</li>

<li id="b" class="**tab-pane**">bbbb</li>

<li id="c" class="**tab-pane**">cccc</li>

</ul>

</div>

### 美化

<style>

.tab-content{

padding:50px;

border:1px solid #ddd;

border-top:none;

border-radius: 0 0 5px 5px;

}

</style>

<div style="width:400px;">

<ul class="nav nav-tabs">

<li class="active"><a href="#a1" data-toggle="tab">HTML</a></li>

<li><a href="#b1" data-toggle="tab">CSS</a></li>

<li><a href="#c1" data-toggle="tab">BS</a></li>

</ul>

<ul class="tab-content">

<li id="a1" class="tab-pane active">aaaa</li>

<li id="b1" class="tab-pane">bbbb</li>

<li id="c1" class="tab-pane">cccc</li>

</ul>

</div>

### 淡入淡出的切换

<div style="width:400px;">

<ul class="nav nav-tabs">

<li class="active"><a href="#a2" data-toggle="tab">HTML</a></li>

<li><a href="#b2" data-toggle="tab">CSS</a></li>

<li><a href="#c2" data-toggle="tab">BS</a></li>

</ul>

<ul class="tab-content">

<li id="a2" class="tab-pane **fade in** active" >aaaa</li>

<li id="b2" class="tab-pane **fade**">bbbb</li>

<li id="c2" class="tab-pane **fade**">cccc</li>

</ul>

</div>

### 鼠标滑过改变选项卡

<div style="width:400px;">

<ul id="myTabs" class="nav nav-tabs">

<li class="active"><a href="#a3" data-toggle="tab">HTML</a></li>

<li><a href="#b3" data-toggle="tab">CSS</a></li>

<li><a href="#c3" data-toggle="tab">BS</a></li>

</ul>

<ul class="tab-content">

<li id="a3" class="tab-pane fade in active" >aaaa</li>

<li id="b3" class="tab-pane fade">bbbb</li>

<li id="c3" class="tab-pane fade">cccc</li>

</ul>

</div>

</div>

<script src="../js/jquery-1.11.1.js"></script>

<script src="../js/bootstrap.js"></script>

<script>

$('#myTabs a').mouseover(function(){

$(this).tab('show');

})

</script>

### 带下拉菜单的选项卡

<div style="width:400px;">

<ul class="nav nav-tabs">

<li class="active"><a href="#a4" data-toggle="tab">HTML</a></li>

<li><a href="#b4" data-toggle="tab">CSS</a></li>

<li class="dropdown">

<a href="#" data-toggle="dropdown" class="dropdown-toggle">

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

</a>

<ul class="dropdown-menu">

<li><a href="#c4" data-toggle="tab">content1</a></li>

<li><a href="#d4" data-toggle="tab">content2</a></li>

</ul>

</li>

</ul>

<ul class="tab-content">

<li id="a4" class="tab-pane fade in active" >aaaa</li>

<li id="b4" class="tab-pane fade">bbbb</li>

<li id="c4" class="tab-pane fade">cccc</li>

<li id="d4" class="tab-pane fade">dddd</li>

</ul>

</div>

# 导航条组件

navbar navbar-default | navbar-inverse

nav navbar-nav

navbar-static-top(消除圆角)

navbar-fixed-top

navbar-fixed-bottom

遮挡内容的问题

## 导航条基本结构

<div class="navbar navbar-inverse">

<ul class="nav navbar-nav">

<li class="active"><a href="">HTML</a></li>

<li><a href="">CSS</a></li>

<li><a href="">BootStrap</a></li>

</ul>

</div>

## 流体布局中的导航条

<div class="container-fluid">

<div class="navbar navbar-inverse">

<ul class="nav navbar-nav">

<li class="active"><a href="">HTML</a></li>

<li><a href="">CSS</a></li>

<li><a href="">BootStrap</a></li>

</ul>

</div>

</div>

## 开发时用的导航条结构

<div class="navbar navbar-inverse">

<div class="container">

<ul class="nav navbar-nav">

<li class="active"><a href="">HTML</a></li>

<li><a href="">CSS</a></li>

<li><a href="">BootStrap</a></li>

</ul>

</div>

</div>

<div class="navbar navbar-inverse">

<div class="container-fluid">

<ul class="nav navbar-nav">

<li class="active"><a href="">HTML</a></li>

<li><a href="">CSS</a></li>

<li><a href="">BootStrap</a></li>

</ul>

</div>

</div>

## 消除导航条的圆角

<div class="navbar navbar-inverse navbar-static-top">

<ul class="nav navbar-nav">

<li class="active"><a href="">HTML</a></li>

<li><a href="">CSS</a></li>

<li><a href="">BootStrap</a></li>

</ul>

</div>

## 固定定位在顶部的导航条

<div class="navbar navbar-inverse navbar-fixed-top">

<div class="container">

<ul class="nav navbar-nav">

<li class="active"><a href="">HTML0</a></li>

<li><a href="">CSS0</a></li>

<li><a href="">BootStrap0</a></li>

</ul>

</div>

</div>

## 固定定位在底部的导航条

<div class="navbar navbar-inverse navbar-fixed-bottom">

<div class="container">

<ul class="nav navbar-nav">

<li class="active"><a href="">HTML0</a></li>

<li><a href="">CSS0</a></li>

<li><a href="">BootStrap0</a></li>

</ul>

</div>

</div>

## 固定定位的导航条遮挡内容

body{ margin-top:50px; }

# 复杂的导航条

navbar-header navbar-brand

navbar-left navbar-right

navbar-btn

navbar-link

navbar-text

navbar-form

## 导航条添加logo标志

<div class="navbar navbar-inverse navbar-fixed-top">

<div class="container">

**<a href="" class="navbar-brand">logo</a>**

<ul class="nav navbar-nav">

<li class="active"><a href="">HTML0</a></li>

<li><a href="">CSS0</a></li>

<li><a href="">BootStrap0</a></li>

</ul>

</div>

</div>

**加了logo在移动xs设备会出现问题，下面是解决方案：**

<div class="navbar navbar-inverse navbar-fixed-top">

<div class="container">

**<div class="navbar-header">**

**<a href="" class="navbar-brand">logo</a>**

**</div>**

<ul class="nav navbar-nav">

<li class="active"><a href="">HTML0</a></li>

<li><a href="">CSS0</a></li>

<li><a href="">BootStrap0</a></li>

</ul>

</div>

</div>

## 左右两个导航条

<div class="navbar navbar-inverse navbar-fixed-top">

<div class="container">

<div class="navbar-header">

<a href="" class="navbar-brand">logo</a>

</div>

<ul class="nav navbar-nav navbar-left">

<li class="active"><a href="">HTML</a></li>

<li><a href="">CSS</a></li>

<li><a href="">BootStrap</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li><a href="">登录</a></li>

<li><a href="">注册</a></li>

</ul>

</div>

</div>

## 在导航条中添加按钮、文字、链接

<input type="submit" class="btn btn-xs btn-default navbar-btn">

<a class="navbar-link">link</a>

<p class="navbar-left navbar-text">aaaaaaaaaaa</p>

## 在导航条中添加搜索框

<form action="" class="navbar-form navbar-right">

<div class="input-group">

<input type="text" class="form-control" placeholder="请输入关键字">

<span class="btn btn-default input-group-addon">搜索</span>

</div>

</form>

## 在导航条中添加下拉菜单

<div class="navbar navbar-inverse navbar-fixed-top">

    <div class="container">

    <div class="navbar-header">

        <a href="#" class="navbar-brand">logo</a>

</div>

<ul class="nav navbar-nav">

<li class="active"><a href="#">one</a></li>

<li><a href="#">two</a></li>

<li class="dropdown">

    <a href="#" class="dropdown-toggle" data-toggle="dropdown">three <span class="caret"></span></a>

<ul class="dropdown-menu">

    <li><a href="#">child1</a></li>

<li><a href="#">child2</a></li>

</ul>

</li>

</ul>

</div>

</div>

## 响应式菜单导航（超小设备下）

navbar-toggle 小图标按钮

icon-bar 小图标按钮的线

collapse navbar-collapse 折叠展开

<div class="navbar navbar-inverse navbar-fixed-top">

    <div class="container">

    <div class="navbar-header">

    <button class="navbar-toggle" data-toggle="collapse" data-target="#myCollapse">

    <span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

        <a href="#" class="navbar-brand">logo</a>

</div>

<div id="myCollapse" class="collapse navbar-collapse">

<ul class="nav navbar-nav">

<li class="active"><a href="#">one</a></li>

<li><a href="#">two</a></li>

<li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">three <span class="caret"></span></a>

<ul class="dropdown-menu">

<li><a href="#">child1</a></li>

<li><a href="#">child2</a></li>

</ul>

</li>

</ul>

</div>

</div>

</div>

# 滚动监听

data-spy

data-target

data-offset

<style>

.pos{margin-bottom: 500px; padding-top: 50px;}

</style>

</head>

<body style="height:2000px;" **data-spy="scroll" data-target="#mynavbar" data-offset="200"**>

<div class="navbar navbar-inverse navbar-fixed-top" id="mynavbar">

<div class="container">

<div class="navbar-header">

<button class="navbar-toggle" data-toggle="collapse" data-target="#mynav">

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<div class="navbar-brand">logo</div>

</div>

<div id="mynav" class="collapse navbar-collapse">

<ul class="nav navbar-nav navbar-left">

<li class="active"><a href="#a">HTML</a></li>

<li class="dropdown">

<a href="" class="dropdown-toggle" data-toggle="dropdown">

CSS

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

</a>

<ul class="dropdown-menu">

<li><a href="#b">selector</a></li>

<li><a href="">property</a></li>

</ul>

</li>

<li><a href="#c">JS</a></li>

</ul>

<form action="" class="navbar-form navbar-right form-group">

<div class="input-group">

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

<a href="" class="input-group-addon btn btn-default">search</a>

</div>

</form>

</div>

</div>

</div>

<div id="a" class="pos">aaaa</div>

<div id="b" class="pos">bbbb</div>

<div id="c" class="pos">cccc</div>

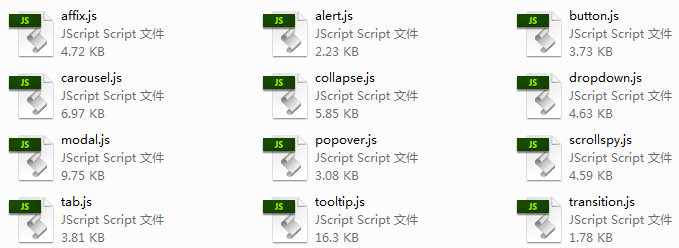
# bootstrap中JS插件

## 插件的使用方式

JavaScript 插件可以单个引入（使用 Bootstrap 提供的单个 \*.js 文件），或者一次性全部引入（使用 bootstrap.js 或压缩版的 bootstrap.min.js）

建议使用压缩版的 JavaScript 文件，因为某些插件和 CSS 组件依赖于其它插件。如果你是单个引入每个插件的，请确保在文档中检查插件之间的依赖关系。

注意，所有插件都依赖 jQuery （也就是说，jQuery必须在所有插件**之前**引入页面）。



## 如何设置插件的选项

**1、在HTML中使用数据属性(data-\*)设置选项**

**你可以仅仅通过 data 属性 API 就能使用所有的 Bootstrap 插件，无需写一行 JavaScript 代码。这是 Bootstrap 中的一等 API，也应该是你的首选方式。**

数据属性设置的缺点：话又说回来，在某些情况下（一个元素触发两种行为/同种多个元素触发相同的行为）可能需要将此功能关闭。因此，我们还提供了关闭 data 属性 API 的方法，即解除以 data-api 为命名空间并绑定在文档上的事件。就像下面这样：

$(document).off('.data-api')

另外，如果是针对某个特定的插件，只需在 data-api 前面添加那个插件的名称作为命名空间，如下：

$(document).off('.alert.data-api')

**2、使用JavaScriptAPI设置插件**

## 使用JavaScriptAPI设置插件

所有BootStrap方法（JQ插件《对象方法》）都接受3种值

**无**——例如，tooltip()。这表明该方法用默认值初始化

**选项对象**——例如，tooltip({html:true}); 选项为一个JSON

**针对特定方法的字符串**——例如，tooltip(‘toggle’)

## 自定义事件

Boostrap为大部分插件提供自定义事件，这些事件的命名采用不定式（show）和过去分词（shown）形式。

**不定式（show）：表示在事件开始时触发。**

**过去分词（shown）：表示在操作完成时触发。**

$("#box").on("show.bs.modal",function(e){

alert('show');

});

$("#box").on("shown.bs.modal",function(e){

alert('shown');

});

## 防止插件冲突

var bsdropdown=$.fn.dropdown.noConflict();

$.fn.bsdropdown=bsdropdown;

$.fn.dropdown=function()

{

alert(1);

}

$("#btn").bsdropdown("toggle");

## 工具提示插件

有些插件（比如工具提示Tooltip）不像之前所讨论的下拉菜单及其他插件那样，它不是纯 CSS 插件。

如需使用该插件，您必须使用 jquery 激活它【$("[data-toggle='tooltip']").tooltip()】

<div class="container">

<br><br>

<button type="button" id="btn1" class="btn btn-default" data-toggle="tooltip" data-html="true" data-trigger="click" data-delay="200" title="看看我怎么出来的1111" >工具提示</button>

<br><br>

<button id="btn2" class="btn btn-default" title="看看我怎么出来的222">工具提示</button>

</div>

<script src="jquery-3.2.1.js"></script>

<script src="bootstrap.min.js"></script>

<script>

$(document).off('.data-api');

//$("[data-toggle='tooltip']").tooltip();

$("#btn1").tooltip();

$('#btn2').tooltip({

html:true,

delay:{"show":200,"hide":1000},

trigger:"hover"

});

</script>

## 附加导航插件affix

数据属性设置：

data-spy=”affix”

data-offset-top=”60”

data-offset-bottom=”120”

<style type="text/css">

/\* Custom Styles \*/

ul.nav-tabs {

width: 140px;

margin-top: 20px;

border-radius: 4px;

border: 1px solid #ddd;

box-shadow: 0 1px 4px rgba(0, 0, 0, 0.067);

}

ul.nav-tabs li {

margin: 0;

border-top: 1px solid #ddd;

}

ul.nav-tabs li:first-child {

border-top: none;

}

ul.nav-tabs li a {

margin: 0;

padding: 8px 16px;

border-radius: 0;

}

ul.nav-tabs li.active a,

ul.nav-tabs li.active a:hover {

color: #fff;

background: #0088cc;

border: 1px solid #0088cc;

}

ul.nav-tabs li:first-child a {

border-radius: 4px 4px 0 0;

}

ul.nav-tabs li:last-child a {

border-radius: 0 0 4px 4px;

}

ul.nav-tabs.affix {

top: 30px;

/\* Set the top position of pinned element \*/

}

</style>

</head>

<body data-spy="scroll" data-target="#myScrollspy">

<div class="container">

<div >

<h1>Bootstrap Affix</h1>

</div>

<div class="row">

<div class="col-xs-3" id="myScrollspy">

<ul class="nav nav-tabs nav-stacked" data-spy="affix" data-offset-top="125" id="myaffix">

<li class="active"><a href="#section-1">第一部分</a></li>

<li><a href="#section-2">第二部分</a></li>

<li><a href="#section-3">第三部分</a></li>

<li><a href="#section-4">第四部分</a></li>

<li><a href="#section-5">第五部分</a></li>

</ul>

</div>

<div class="col-xs-9">

<h2 id="section-1">第一部分</h2>

<p>a</p>

<hr>

<h2 id="section-2">第二部分</h2>

<p>a</p>

<hr>

<h2 id="section-3">第三部分</h2>

<p>a</p>

<hr>

<h2 id="section-4">第四部分</h2>

<p>a</p>

<hr>

<h2 id="section-5">第五部分</h2>

<p>a</p>

<p>a</p>

<p>a</p>

</div>

</div>

</div>

选项设置

$("#myaffix").affix({

offset:{

top:60,

bottom:120

}

})

## 附加导航插件affix事件

$("#myaffix").on("affix.bs.affix",function(){

alert('affix');

});

$("#myaffix").on("affixed.bs.affix",function(){

alert('affixed');

});

$("#myaffix").on("affix-top.bs.affix",function(){

alert('affix-top');

});

$("#myaffix").on("affixed-top.bs.affix",function(){

alert('affixed-top');

});

$("#myaffix").on("affix-bottom.bs.affix",function(){

alert('affix-bottom');

});

$("#myaffix").on("affixed-bottom.bs.affix",function(){

alert('affixed-bottom');

});

## 警告框插件alert

bootstrap中的alert和原本的alert弹框可不太一样，原来我们熟悉的弹框是在执行某个动作的时候，浏览器和我们弹出来的一个提示框，而我们这里的警告框是在html内容之间的提示内容，只是他有着醒目的颜色，以此来达到提醒用户的目的。

**数据属性设置：**

<div class="alert alert-warning alert-dismissible fade in" style="width:400px;">

<button type="button" class="colse pull-right" data-dismiss="alert">

<span>&times;</span>

</button>

警告框插件的使用warning

</div>

**JS选项设置：**

<div class="alert alert-warning alert-dismissible fade in" style="width:400px;">

<button type="button" class="colse pull-right">

<span>&times;</span>

</button>

警告框插件的使用warning

</div>

<script>

$(".colse").click(function(){

alert(123);

$(this).alert('close');

});

</script>

## 按钮插件button

**数据属性设置没用 JS选项设置：**

<button type="button" id="mybtn" class="btn btn-default">

展开

</button>

<script>

$("#mybtn").click(function(){

$(this).button('loading');

});

</script>

## 轮播插件carousel

构建轮播需要3段HTML

轮播指示（轮播底部的显示连续的小点）

幻灯片（每个指示对应的一张幻灯片）

控件（“上一个”“下一个”）

<div class="carousel" id="mycarousel" data-ride="carousel" style="background: lightgreen;height: 500px;">

<!-- 轮播指示（轮播底部的显示连续的小点） -->

<ol class="carousel-indicators">

<li class="active" data-target="#mycarousel" data-slide-to="0"></li>

<li data-target="#mycarousel" data-slide-to="1"></li>

<li data-target="#mycarousel" data-slide-to="2"></li>

</ol>

<!-- 幻灯片（每个指示对应的一张幻灯片） -->

<div class="carousel-inner">

<div class="item active">

<h1>slide1</h1>

</div>

<div class="item">

<h1>slide2</h1>

</div>

<div class="item">

<h1>slide3</h1>

</div>

</div>

<!-- 控件（“上一个”“下一个”） -->

<a href="#mycarousel" class="left carousel-control" data-slide="prev">

<span class="glyphicon glyphicon-chevron-left"></span>

</a>

<a href="#mycarousel" class="right carousel-control" data-slide="next">

<span class="glyphicon glyphicon-chevron-right"></span>

</a>

</div>

## 基本轮播插件(包含标题的轮播)

<div class="carousel" id="mycarousel" data-ride="carousel" style="background: lightgreen;">

<!-- 轮播指示（轮播底部的显示连续的小点） -->

<ol class="carousel-indicators">

<li class="active" data-target="#mycarousel" data-slide-to="0"></li>

<li data-target="#mycarousel" data-slide-to="1"></li>

<li data-target="#mycarousel" data-slide-to="2"></li>

</ol>

<!-- 幻灯片（每个指示对应的一张幻灯片） -->

<div class="carousel-inner">

<figure class="item active">

<img src="images/chrome-big.jpg" alt="">

<figcaption class="carousel-caption">chrome</figcaption>

</figure>

<figure class="item">

<img src="images/safari-big.jpg" alt="">

<figcaption class="carousel-caption">safari</figcaption>

</figure>

<figure class="item">

<img src="images/firefox-big.jpg" alt="">

<figcaption class="carousel-caption">firefox</figcaption>

</figure>

</div>

<!-- 控件（“上一个”“下一个”） -->

<a href="#mycarousel" class="left carousel-control" data-slide="prev">

<span class="glyphicon glyphicon-chevron-left"></span>

</a>

<a href="#mycarousel" class="right carousel-control" data-slide="next">

<span class="glyphicon glyphicon-chevron-right"></span>

</a>

</div>

## 轮播插件选项控制

$('.carousel').carousel({

interval:2000,//循环的时间

keyboard:true//键盘是否可以控制

})

## 折叠插件collapse

依赖过渡插件

<a href="#mycollapse" class="btn btn-default" data-toggle="collapse">折叠插件</a>

<div id="mycollapse">

被折叠的内容被折叠的内容被折叠的内容

</div>

不能折叠段落、表格单元格（td/th）

## 标签页选项卡插件nav-tabs nav-pills

数据属性设置前面讲过啦！

JS选项参数设置

$("#mynav a").click(function(e){

$(this).tab('show');

});

$("#mynav a").click(function(e){

e.preventDefault();//页面地址栏地址不会改变

$(this).tab('show');

});