# ：起步

## 1，初识Bootstrap



Bootstrap，来自 Twitter，是目前很受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。 它由Twitter的设计师Mark Otto和Jacob Thornton合作开发，是一个CSS/HTML框架。Bootstrap提供了优雅的HTML和CSS规范，它即是由动态CSS语言Less写成。Bootstrap一经推出后颇受欢迎，一直是GitHub上的热门开源项目，包括NASA的MSNBC（微软全国广播公司）的Breaking News都使用了该项目。 国内一些移动开发者较为熟悉的框架，如WeX5前端开源框架等，也是基于Bootstrap源码进行性能优化而来。

**说的直观一点，bootstrap就是一个web框架。**

官网是**：<http://getbootstrap.com>**

中文网**：<http://www.bootcss.com/>**

**特点：**

1，响应式，移动设备优先

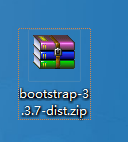
2，跨平台，跨浏览器

3，提供提供了全面组件

4，支持HTML5， CSS3

## 2，如何使用Bootstrap

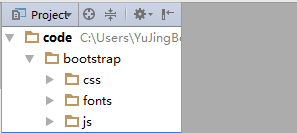
1，Bootstrap是个框架，要使用这个框架，首先去下载这个框架，官网。下载完之后是一个压缩包。



2，解压这个压缩包， 它的文件结构如下：



3，将解压后的bootstrap放到我们的项目中



4，如果我们用到了Bootstrap中的样式，那么我们就引入bootstrap.css, 如果我们用到了里面的插件，那么我们就需要引入bootstrap.js， 需要注意问题： bootstrap是基于jq, 所以当你想引入bootstrap.js时，你需要在bootstrap前面引入jquery。



# ：全局CSS样式

## 1，栅格系统

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。

说的直观一点，栅格系统就是一系列的列和行构建的一种布局，它是用来布局的。随着屏幕的放大，最多可以显示12列。

**容量：**

bootstrap提供两种容器，这两种容器都是用来布局的

* 固定容器 class = “container”
* 流式容器 class = “container-fluid”

固定容器使内容居中了一点，两侧留出了空白，流式容器是铺满整个屏幕。

**栅格参数：**

col-lg-\* large(大屏幕>1200px) PC

col-md-\* middle(中等屏幕 992px~1200px) 大平板

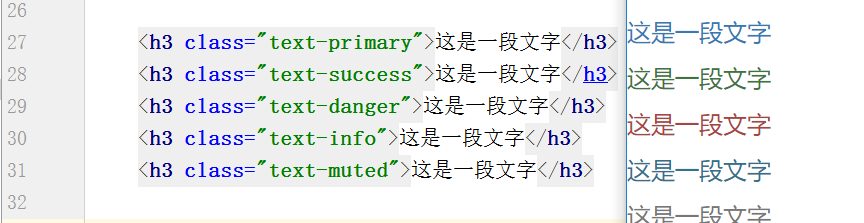
col-sm-\* small(小屏幕 768px~992px) 小平板

col-xs-\* 小于768px 手机

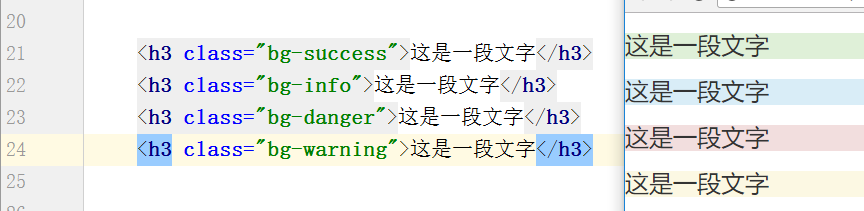


## 2，辅助类

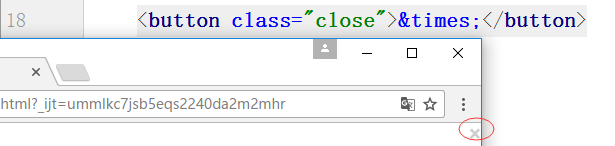
**文本颜色： text-**



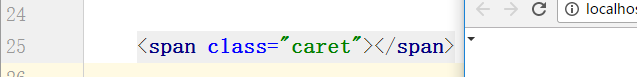
**背景样式：bg-**



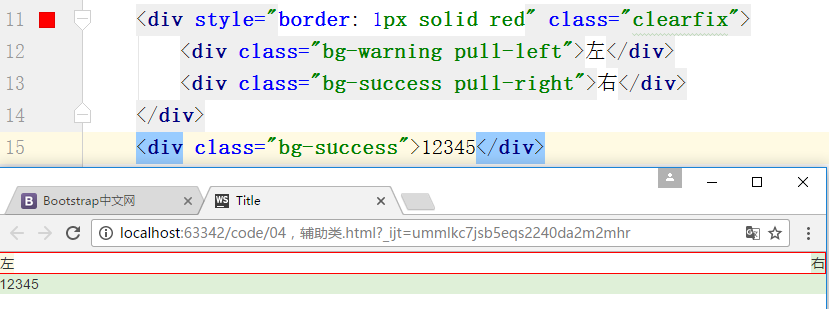
**关闭按钮：close**



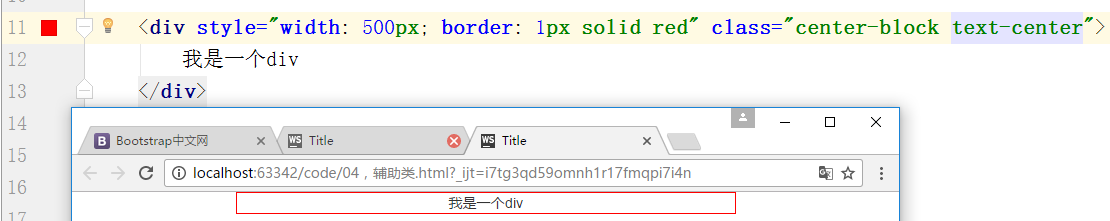
**三角图标：caret**



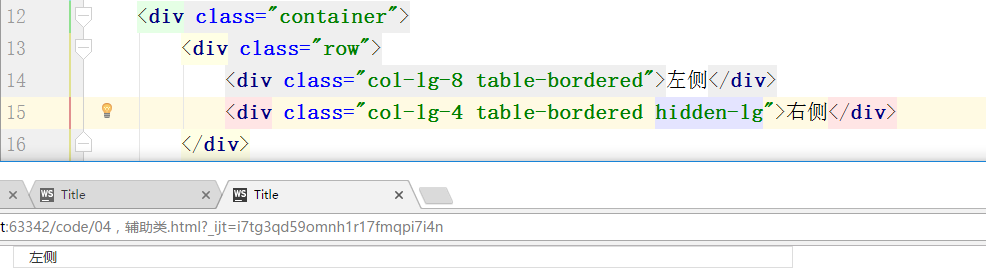
**快速浮动和清除浮动： pull-left pull-rigth clearfix**



**内容块居中：center-block 文字居中：text-center**



**显示，隐藏：visible-lg 只有大屏时才显示， hidden-lg 只有大屏时才隐藏**



## 3，表格

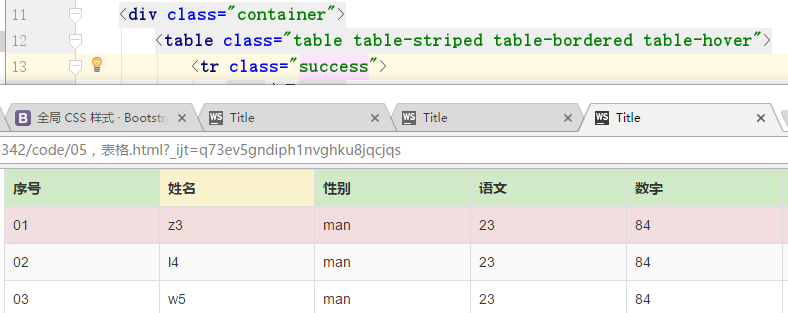
只需要给table标签加上table类就可以了

条纹表格：table-striped

带边框的表格：table-bordered

鼠标悬停效果：table-hover

表格的颜色：active, success, warning, danger,info.....



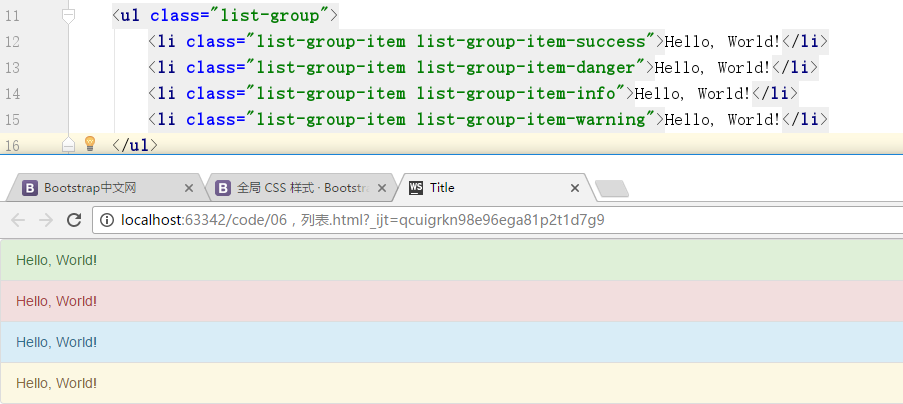
## 4，列表

使用步骤：

1，给ul添加 list-group

2，给li添加list-group-item

3，给li添加不同的颜色 list-group-item-颜色



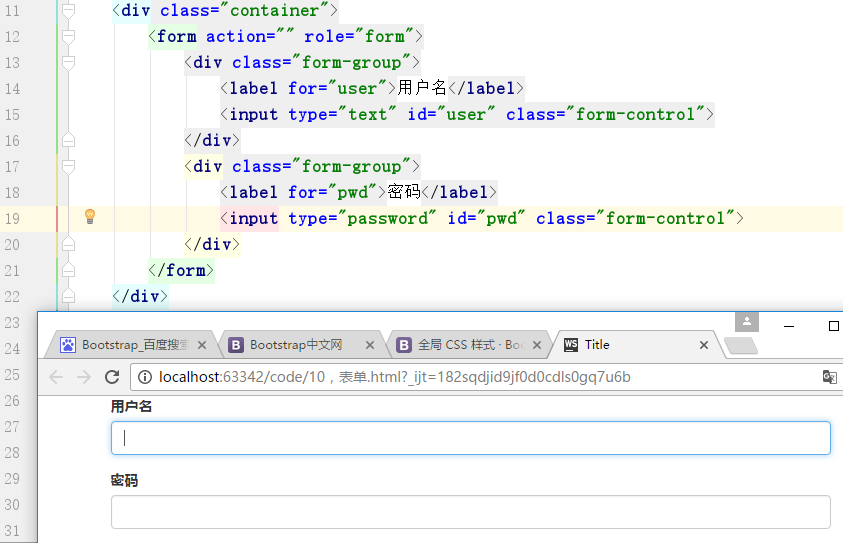
## 5，表单

**基本表单的使用步骤：**

1，给form添加角色 role=”form”

2，给每个表单项添加一个form-group

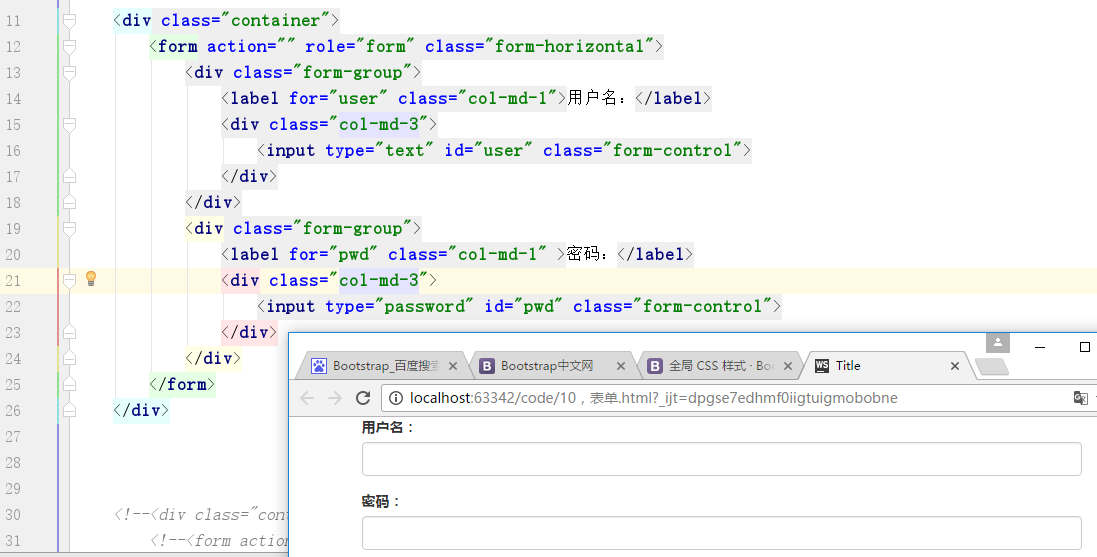
3，给每一个表单项内部再添加一个form-control



**水平表单：**

1，给form添加一个form-horizontal类

2，给label添加栅格参数， 给input单独添加一个div，给这个div也需要设置栅格参数



**案例：**

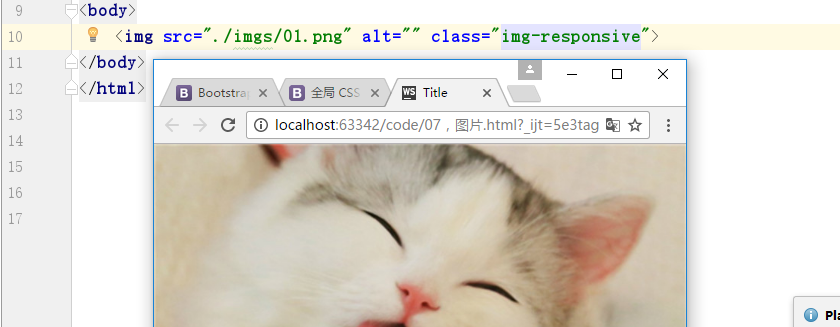


**效果：**



## 6，图片

**响应式图片：**



**图片的形状：**



## 7，按钮

生成按钮有三种方式：

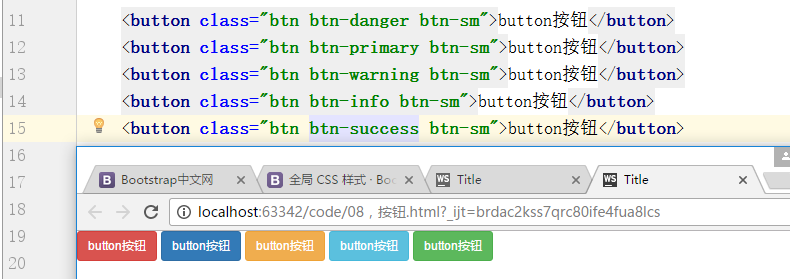
* 1，button
* 2，input
* 3，a



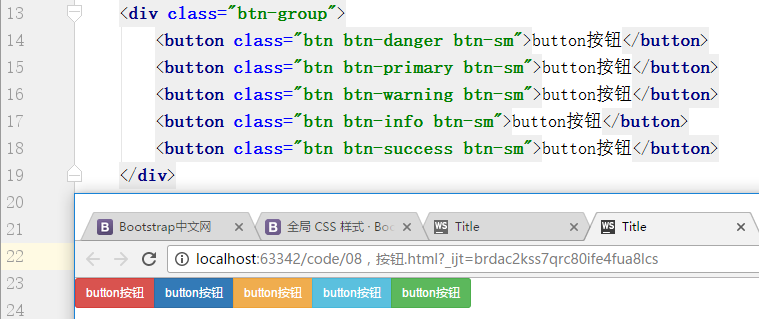
**按钮大小：btn-xs btn-sm btn-lg btn-block**



**按钮的颜色：btn-primary btn-success btn-warning btn-info btn-danger**



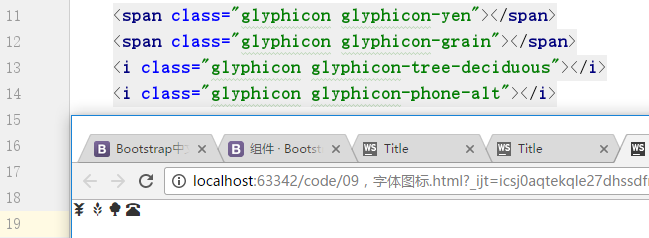
**按钮组：**



# 第三章：组件

## 1，字体图标

使用字体图标很简单，直接把官网上的copy过来就只可以了，常见对应的标签span i



## 2，下拉菜单

下拉菜单在网页中用的比较多，下拉菜单需要使用到bootstrap.Js， 所以我们要引入bootstrap.JS，前提是也需要引入JQeury

步骤：

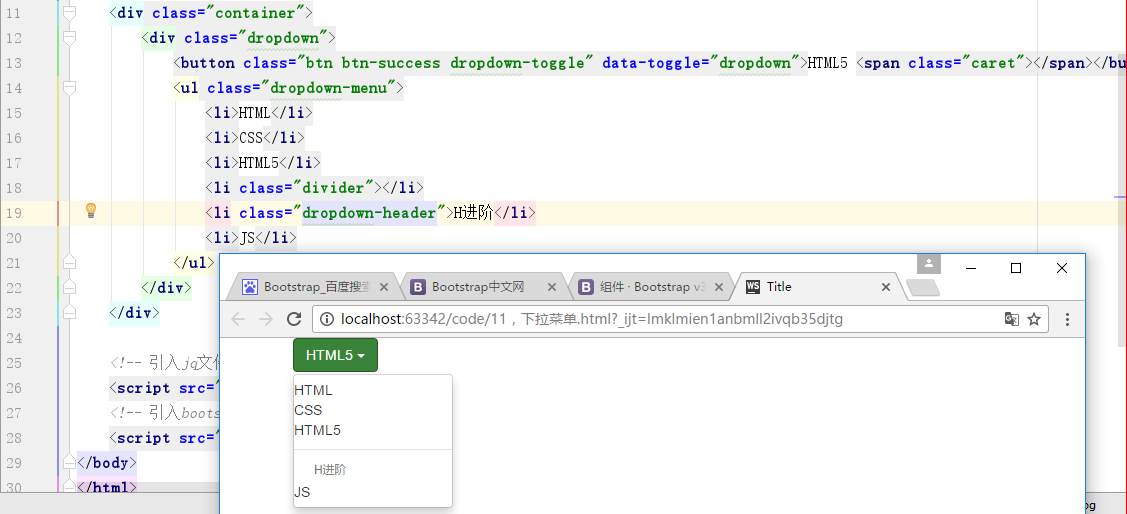
1，使用class=”dropdown”的容器包裹整个下拉菜单

2，使用button作为触发器，class=”dropdown-toggle” 还需要给button添加个自定义属性data-toggle=”dropdwon”, 我们可以把这个自定义属性看作实现下拉菜单的具体功能。

3，使用ul+li实现下拉菜单，需要给ul加上class=”dropdown-menu”

4，需要使用分割线上，再添加一个li 加上class=”divider”

4，需要使用标题上，给li 加上class=”dropdown-header”

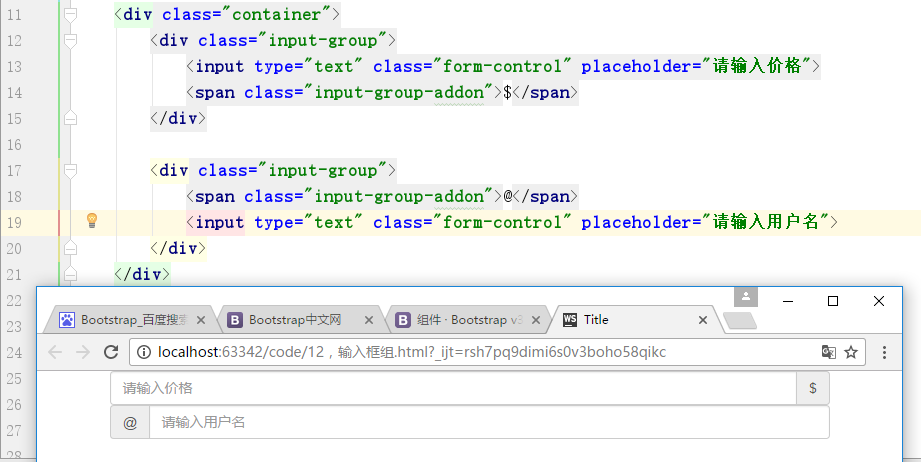


## 3，输入框组

所谓的输入框组就是在输入框的前面或后面加上修饰：



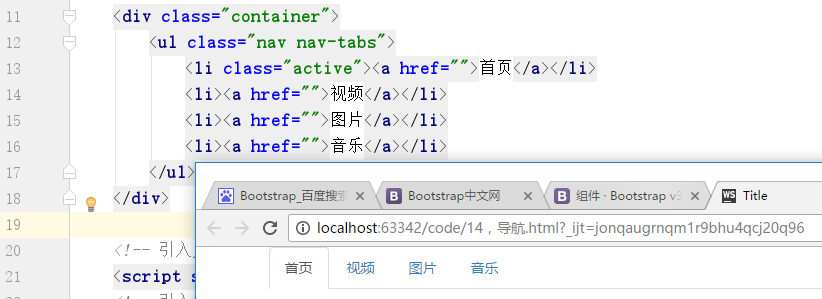
实例如下：



## 4，导航

Bootstrap中的导航必须依赖于一个基类 class=”nav” 但是nav并不提供导航的样式，我们需要去添加额外的样式。

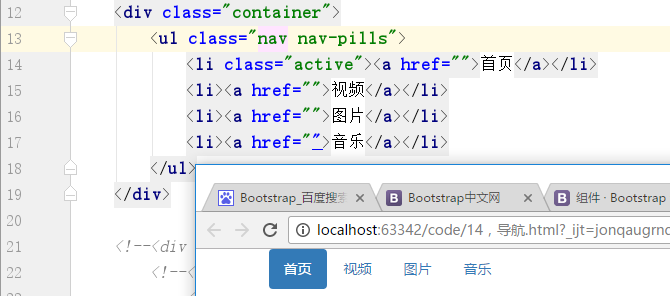
### Tab选项卡式的导航



注意问题：nav和nav-tabs结合后才能显示出效果。

### **Pills导航**

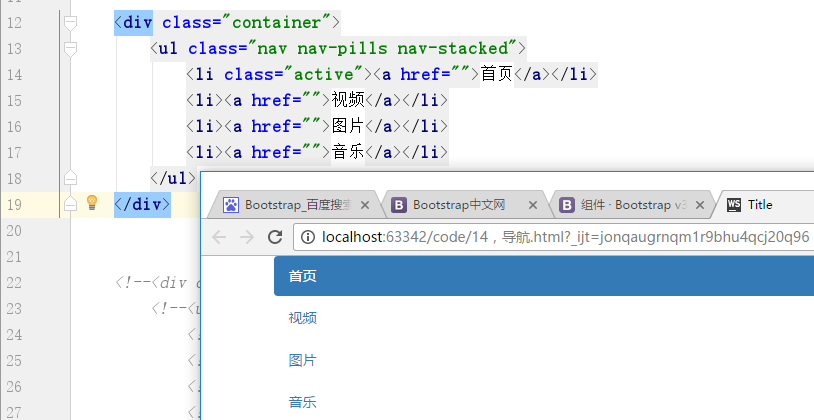
pills导航也叫胶囊式导航，因为它长的类似胶囊



### **垂直导航**

前面我们学习的导航是水平的，垂直的需要加上nav-stacked

如下：



### **自适应导航**

自适应导航是指导航栏占容器的全部宽度，并且可以根据这个宽度进行自适应。

nav-justified

如下：



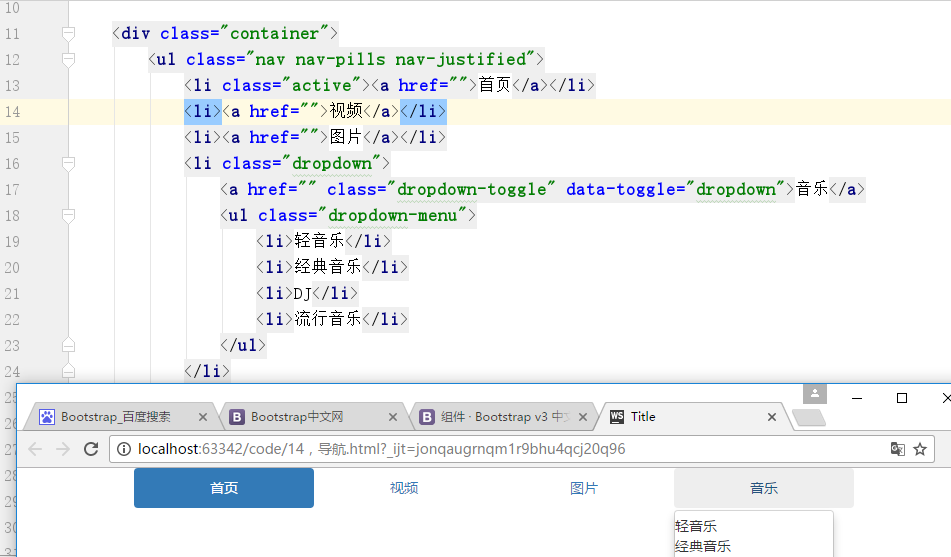
### **导航的状态**

active 活动状态 disabled 禁用状态



### **导航二级菜单**

导航里面添加下拉菜单



### **面包屑导航（故事）**

用来指示当前页面所处的位置，class=”breadcrumb”



## 5，导航条

导航和导航条两者的区别：导航条可以包含导航，也可以包含表单元素， 链接，可以有背景色。

### 导航条的基本用法

导航用：nav 导航条用：navbar 也是基本 navbar-default



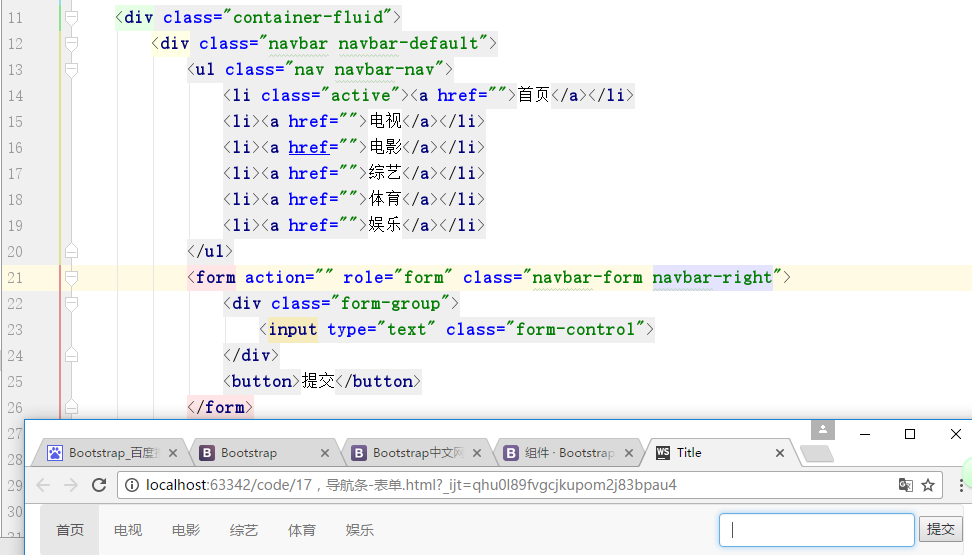
### 品牌图标

意思是在导航条最前面加上logo



### 导航条-表单

在导航条中，我们经常会用到搜索框，让导航条具有搜索功能



### 导航条-按钮



### 导航条-文本和链接



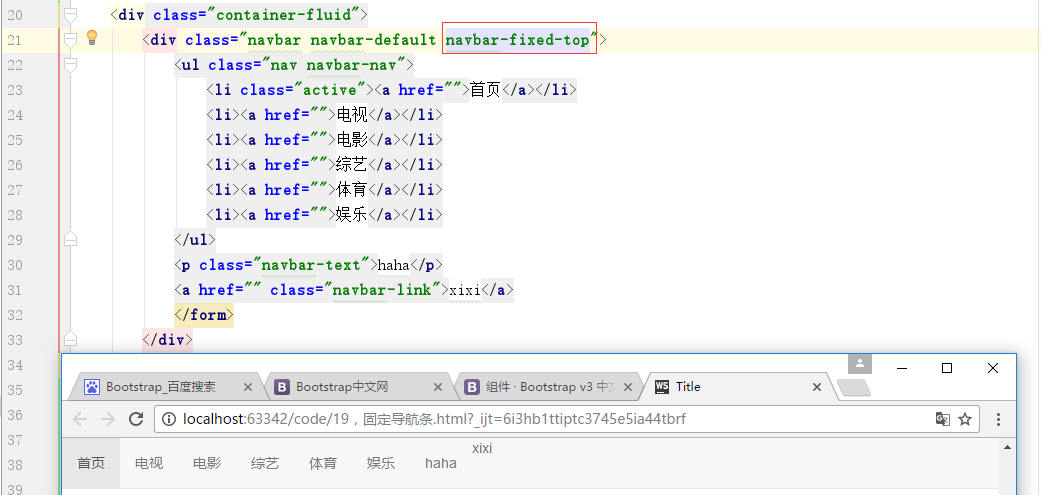
### 导航条中的二级菜单

导航条中的二级菜单和导航中的二级菜单一样，就是给某个导航添加下拉菜单。略过。

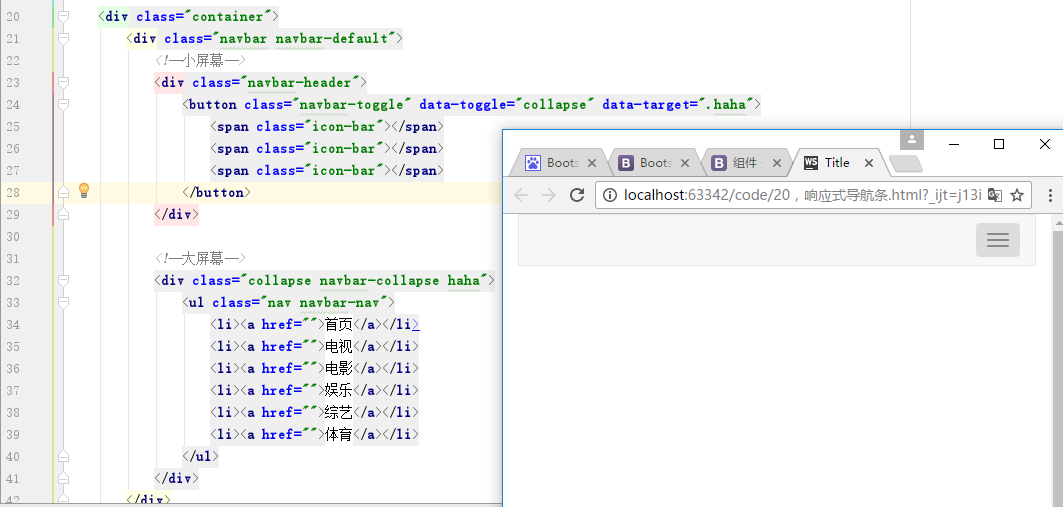
### 固定导航条

1，navbar-fixed-top 将导航条固定在顶部

2，navbar-fixed-bottom 将导航条固定在底部



### 响应式导航条



### 反色导航条

只需要把navbar-default 换成　navbar-inverse就只可以了



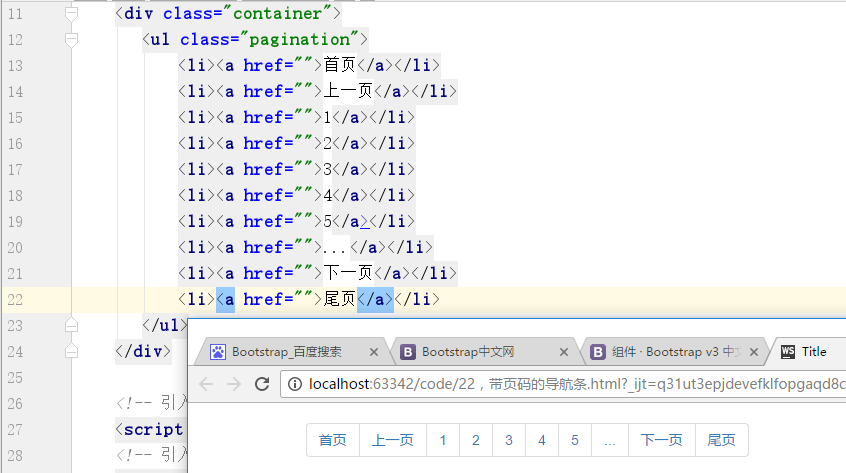
### 分页导航

分布导航条有两种：

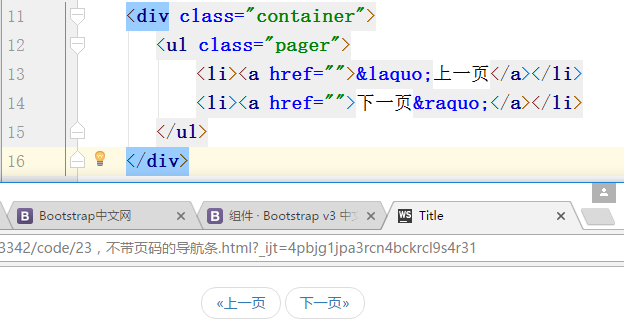
1，带页码的导航条

2，带翻页的导航条

**带页码的导航条**



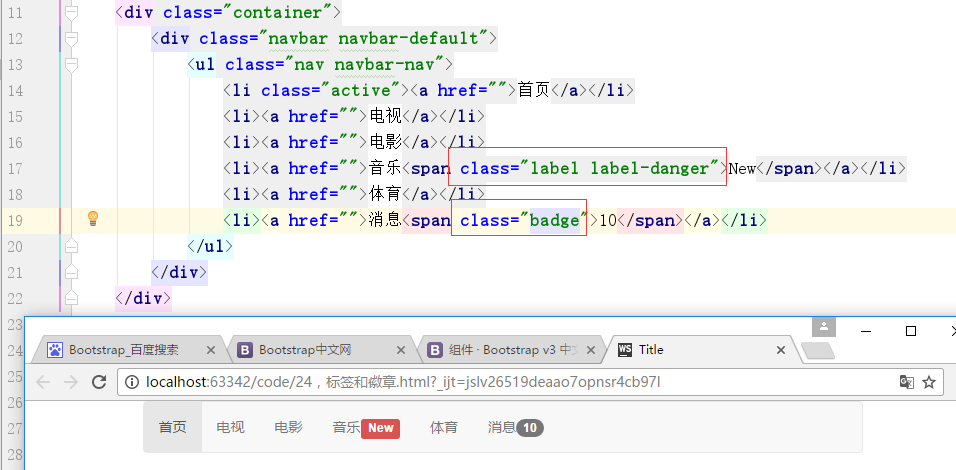
**带翻页的导航条**



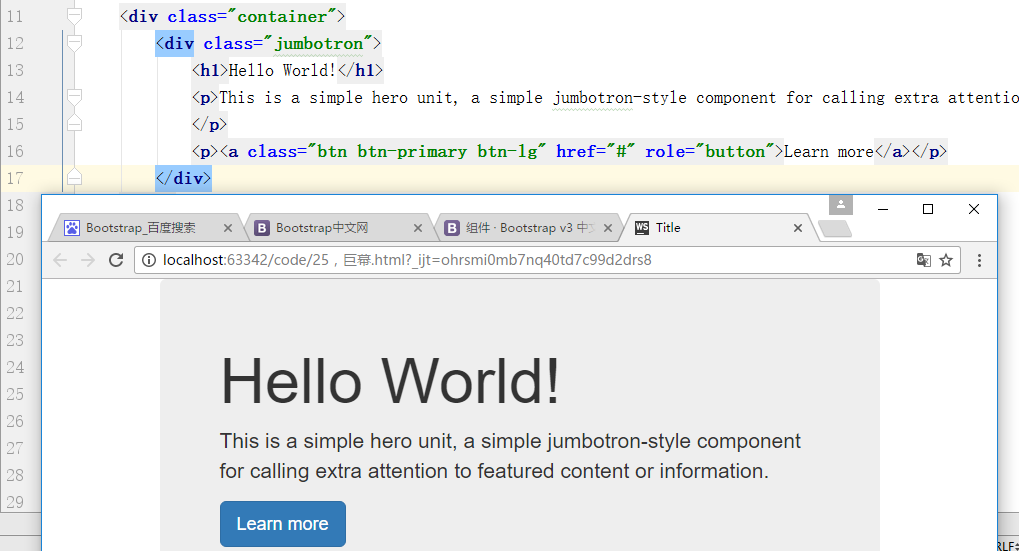
## 6，标签和徽章

标签：常用的就是人导航条中的某个导航添加额外的样式，比如给音乐添加New

徽章：和标签类似



## 7，巨幕

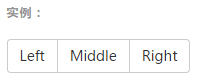


## 8，警告框，进度条等其它组件内容

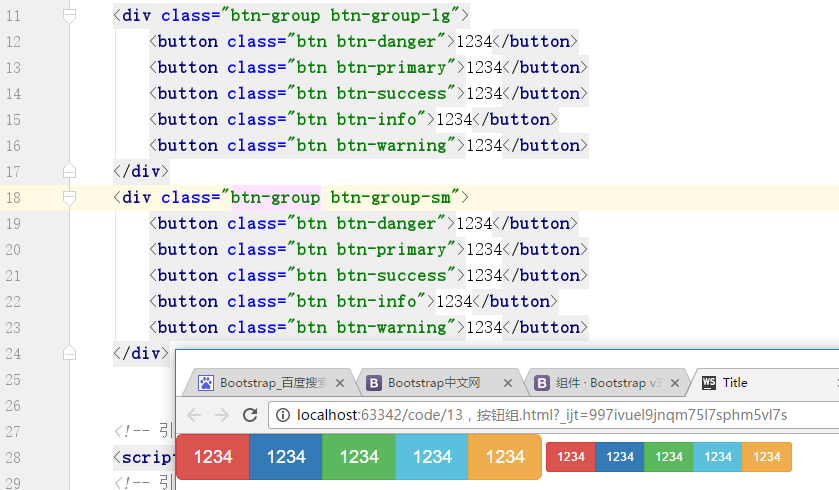
**其它内容比较简单，自己看管网学习一下，用的不多， 用到时，再学习！**

## 9，按钮组

所谓按钮组就是多个按钮挨在一起，如下：



代码如下：



# ：JavaScript插件

## Tab选项卡



## 轮播图

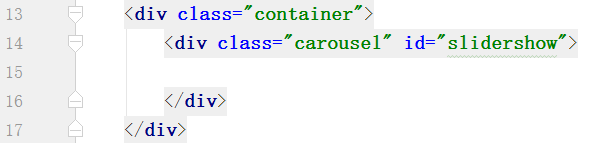
**一个轮播图片主要包括三个部分：**

　　☑ 轮播的图片

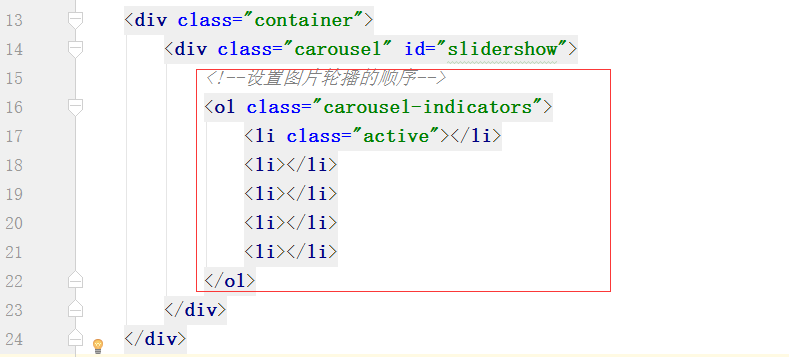
　　☑ 轮播图片的计数器

☑ 轮播图片的控制器

**第一步：**设计轮播图片的容器。在 Bootstrap 框架中采用 carousel 样式，并且给这个容器定义一个 ID 值，方便后面采用 data 属性来声明触发。



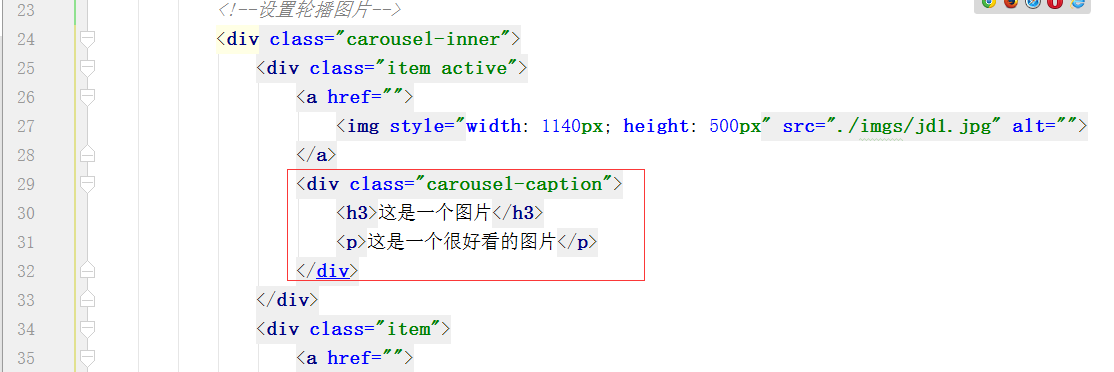
**第二步：**设计轮播图片计数器。在容器 div.carousel 的内部添加轮播图片计算器，采用 carousel-indicators 样式，其主要功能是显示当前图片的播放顺序(有几张图片就放置几个li)，一般采用有顺列表来制作：



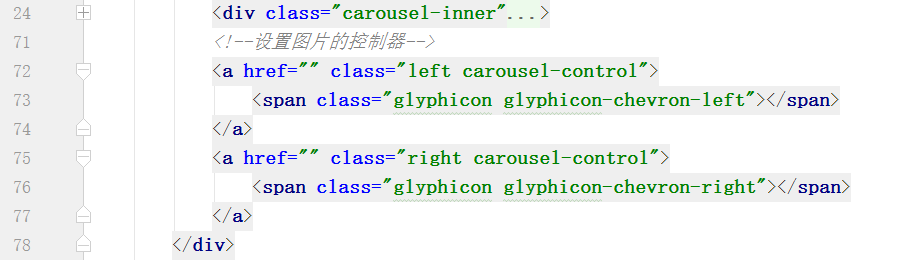
**第三步：**设计轮播图片播放区。轮播图整个效果中，播放区是最关键的一个区域，这个区域主要用来放置需要轮播的图片。这个区域使用 carousel-inner 样式来控制，而且其同样放置在 carousel 容器内，并且通过 item 容器来放置每张轮播的图片：



**第四步:**设置轮播图片描述.很多轮播图片效果，在每个图片上还对应有自己的标题和描述内容。其实 Bootstrap 框架中的 Carousel 也提供类似的效果。只需要在 item 中图片底部添加对应的代码.



**第五步：**设计轮播图片控制器。很多时候轮播图片还具有一个向前播放和向后播放的控制器。在 Carousel 中通过 carousel-control样式配合 left 和 right 来实现。其中left表示向前播放，right表示向后播放。其同样放在carousel容器内：



**第六步:**声明式触轮播图的播放(无需JS)

声明式方法是通过定义 data 属性来实现，data 属性可以很容易地控制轮播的位置。其主要包括以下几种：

* 1 . data-ride 属性：取值 carousel，并且将其定义在 carousel 上。
* 2 . data-target 属性：取值 carousel 定义的 ID 名或者其他样式识别符，如前面示例示，取值为“#slidershow”，并且将其定义在轮播图计数器的每个 li 上。
* 3 . data-slide 属性：取值包括 prev，next，prev表示向后滚动，next 表示向前滚动。该属性值同样定义在轮播图控制器的 a 链接上，同时设置控制器 href 值为容器
* 4 . data-slide-to 属性：用来传递某个帧的下标，比如 data-slide-to="2"，可以直接跳转到这个指定的帧（下标从0开始计），同样定义在轮播图计数器的每个 li 上。
* 5，data-inerval 等待时间
* 6，data-pause 鼠标放到图片上，是否停止播放

文章地址：<http://www.jb51.net/article/75806.htm>

## 模态框

## 滚动监听