

## 1.bootstrap3插件1

### 1.0.bootstrap插件基础

- (1) 组件和插件的区别
- (2) bootstrap插件: 先jQuery, 再bootstrap插件
- (3) bootstrap插件的引用方式
- (4) bootstrap插件的不写js代码的基本原理【data属性】
- (5) bootstrap的data属性【利用data属性可以不要写JS代码实现插件功能】

### 1.1.动画过渡效果【各插件在界面效果变更时的需要使用】

### 1.2.模态框

#### 1.2.0.对话框的种类

#### 1.2.1.模态框的组成

要点1a之嵌套层次名称: 【最外层】==》【对话框】==》【内容层】

要点1b之嵌套层次选择器: 【.modal】==》【.modal-dialog】==》【.modal-content】

要点2a之内容层名称: 【内容层】由三部分组成: 头部、体部、尾部

要点2b之内容层选择器【.modal-content】= .modal-header + .modal-body + modal-footer

要点3: 关闭对话框使用data的API: data-dismiss="modal", 表示单击后, 关闭模态框

#### 1.2.2.模态框的三种激活方式

- (1) 代码激活: `$('.modal').modal();`
- (2) 链接激活: `data-toggle="modal"`配合 `href="#id"`
- (3) 插件激活【data属性激活】: `data-toggle="modal"`配合 `data-target="#id"`

#### 1.2.3.模态框布局和样式

#### 1.2.4.模态框尺寸——大小模态框

#### 1.2.5.扩展1: 模态框的data属性

#### 1.2.6.扩展2: JS操作模态框

- (1) JS操作模态框的属性
- (2) JS操作模态框的方法
- (3) JS操作模态框的事件

## 9.b04\_a01\_百度登录框

# 1.bootstrap3插件1

## 1.0.bootstrap插件基础

### (1) 组件和插件的区别

侧重点不同:

组件侧重于UI界面效果,

插件侧重于js交互效果。

基础不同:

组件交互效果, 只是简单地调用jquery的API;

插件交互效果, 需要特别编写基于jquery的插件才能使用。

## (2) bootstrap插件：先jQuery，再bootstrap插件

- 引用JavaScript插件后可以套用通用的HTML格式就能实现酷炫的交互特效
- JavaScript插件是依托在jQuery10+的基础上,所以要使用它前提是必须先引入jQuery库
- JavaScript插件总共有12种, 过渡效果、模态框、选项卡、轮播图、滚动监听、下拉菜单、按钮等

## (3) bootstrap插件的引用方式

- 一次性全部引入 (使用 bootstrap.js 或压缩版的 bootstrap.min.js)
- JavaScript 插件可以单个引入

一次性全部引入bootstrap.min.js

单个引入【不建议】

## (4) bootstrap插件的不写js代码的基本原理【data属性】

前端页面的标准交互的组成元素 = H5标签结构 + CSS3样式 + JS代码

bootstrap插件的组成元素 = H5标签结构 + 放入写好的类选择器 + 可以不写JS代码

为什么bootstrap插件交互时可以不写JS代码？

JS代码肯定是有，都放在对应的插件里【bootstrap.min.js有全部插件】。

怎么找到具体的插件代码呢？data打头的属性——【data属性】

data-xxx，这类属性，写在H5标签结构中的相关标记中。

以data-打头，主要是避免跟正常属性冲突，bootstrap插件认的关键字data-后面的值。

即：

写在H5标签结构，使用的是data-XXX；

用JS代码时，则直接使用XXX。

#### (5) bootstrap的data属性【利用data属性可以不要写JS代码实现插件功能】

- **data属性API**是Bootstrap中的一等API，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**。这是首选的使用方式
- **关闭 data 属性 API 的方法**
  - ◆ `$(document).off('.data-api')`

不同插件，data属性也不一样。

### 1.1.动画过渡效果【各插件在界面效果变更时的需要使用】

---

- 可以单独引入**transition.js**
- 过渡效果全部使用了**css3**语法，IE6-IE8不支持动画过渡效果
- 下述插件使用动画过渡效果
  - ◆ 模态框 (Modal) 的滑动和渐变效果
  - ◆ 选项卡 (Tab) 的渐变效果
  - ◆ 警告框 (Alert) 的渐变效果
  - ◆ 旋转轮播 (Carousel) 的滑动效果

transition.js插件，引入了transitionEnd事件，在动画过渡效果结束后执行。

## 1.2.模态框

---

### 1.2.0.对话框的种类

普通框：弹出来之后，可以继续处理非对话框中的内容；

模态框：弹出来之后，不能继续处理非对话框中的内容，只能先处理好框中的内容；

普通框：类似JS的window.open()打开的浏览器窗口。

模态框：类似JS的alert()函数打开的对话框。

### 1.2.1.模态框的组成

要点1a之嵌套层次名称：【最外层】==》【对话框】==》【内容层】

要点1b之嵌套层次选择器：【.modal】==》【.modal-dialog】==》【.modal-content】

要点2a之内容层名称：【内容层】由三部分组成：头部、体部、尾部

要点2b之内容层选择器【.modal-content】= .modal-header + .modal-body + modal-footer

要点3：关闭对话框使用data的API：data-dismiss="modal"，表示单击后，关闭模态框

```

<!--弹出的模态框-->
<!-- 【模态框的最外层】modal固定布局，表示上下左右都是0表示充满全屏，支持移动设备上触摸方式进行滚动。-->
<!-- fade表示淡入淡出的效果 -->
<div id="myModal" class="modal fade">

  <!-- 【模态框的对话框：定位与大小】modal-dialog默认相对定位，自适应宽度，大于768px时宽度600，居中-->
  <div class="modal-dialog">
    <!-- 【模态框的内容层：边框边距】modal-content模态框的边框、边距、背景色、阴影效果。-->
    <div class="modal-content">
      <!-- 【模态框的头部】 -->
      <div class="modal-header">
        <button class="close" data-dismiss="modal">&times;</button>
        <h4 class="modal-title">我是标题</h4>
      </div>
      <!-- 【模态框的体部】 -->
      <div class="modal-body">
        <p> 我是内容。。。 </p>
      </div>
      <!-- 【模态框的尾部】 -->
      <div class="modal-footer">
        <button class="btn btn-default" data-dismiss="modal">关闭</button>
        <button class="btn btn-primary">保存</button>
      </div>
    </div>
  </div>
</div>

```

## 1.2.2.模态框的三种激活方式

### (1) 代码激活: `$('#modal').modal();`

```

<script>
  // 模态框02A：代码激活
  $('#modal').modal();
</script>

```

### (2) 链接激活：data-toggle="modal"配合 href="#id"

```

<a href="#myModal" class="btn btn-primary" data-toggle="modal">点击按钮触发弹框2</a>

```

2.显示的元素为#myModal

1.显示的样式为modal模态框

```

<!--弹出的模态框-->
<!-- 【模态框的最外层】modal固定布局，表示上下左右都是0表示充满全屏，支持移动设备上触摸方式进行滚动。-->
<!-- fade表示淡入淡出的效果 -->
<div id="myModal" class="modal fade">

```

### (3) 插件激活【data属性激活】：data-toggle="modal"配合 data-target="#id"

```

<button class="btn btn-primary" data-toggle="modal" data-target="#myModal">点击按钮触发弹框2</button>

```

1.显示的样式为modal模态框 2.显示的元素为#myModal

```

<!--弹出的模态框-->
<!-- 【模态框的最外层】modal固定布局，表示上下左右都是0表示充满全屏，支持移动设备上触摸方式进行滚动。-->
<!-- fade表示淡入淡出的效果 -->
<div id="myModal" class="modal fade">

```

### 1.2.3.模态框布局和样式

#### ! 注意

- **不要**在一个模态框上**重叠**另外一个模态框，要想同时支持多个模态框，需要**自己写**额外的代码来实现
- 一定要将**模态框的HTML代码**放在文档的**最高层**内（尽量**作为body标签的直接子元素**），以避免其他组件影响模态框的展现和功能

### 1.2.4.模态框尺寸——大小模态框

【大模态框：.bs-example-modal-lg > .modal-lg】

【小模态框：.bs-example-modal-sm > .modal-sm】

<!--弹出的大模态框-->

```
<div id="mymodal1" class="modal fade bs-example-modal-lg">  
  <div class="modal-dialog modal-lg">
```



```
<div id="mymodal2" class="modal fade bs-example-modal-sm">
  <div class="modal-dialog modal-sm">
```



### 1.2.5.扩展1：模态框的data属性

- 使用**data属性API**来实现一些交互功能，除了data-toggle外，模态框还可以设置**其他的data属性**来完成别的功能

属性名称	类型	默认值	描述
<b>data-backdrop</b>	布尔型	true	是否包含一个背景div，如果为true，则单击该背景（不是弹窗本身）时关闭弹窗。如果值为static，则单击背景div元素时不关闭弹窗
<b>data-keyboard</b>	布尔型	true	按Esc键时关闭弹窗，如果设置为false，则不会关闭弹窗
<b>data-show</b>	布尔型	true	初始化时是否显示
<u>href</u>	URL路径	空值	如果提供的是URL，将利用 <u>jQuery</u> 的 load 方法从此 URL 地址加载要展示的内容（只加载一次）并插入 .modal-content 内。如果使用的是 data 属性 API，还可以利用 <u>href</u> 属性指定内容来源地址

### 1.2.6.扩展2：JS操作模态框

#### (1) JS操作模态框的属性

## ■ 模态框属性

属性名称	类型	默认值	描述
<b>backdrop</b>	布尔型	true	是否包含一个背景div，如果为true，则单击该背景（不是弹窗本身）时关闭弹窗。如果值为static，则单击背景div元素时不关闭弹窗
<b>keyboard</b>	布尔型	true	按Esc键时关闭弹窗，如果设置为false，则不会关闭弹窗
<b>show</b>	布尔型	true	初始化时是否显示

### (2) JS操作模态框的方法

## ■ 模态框方法

参数名称	使用方法	描述
<b>toggle</b>	<code>\$('#myModal').modal('toggle');</code>	触发时，反转弹窗的状态，如果当时是开启，则关闭。反之则开启
<b>show</b>	<code>\$('#myModal').modal('show');</code>	触发时，显示弹窗
<b>hide</b>	<code>\$('#myModal').modal('hide');</code>	触发时，关闭弹窗

### (3) JS操作模态框的事件

## ■ 模态框事件

事件类型	描述
<b><u>show.bs.modal</u></b>	show 方法调用之后立即触发该事件。如果是通过点击某个作为触发器的元素，则此元素可以通过事件的 <u>relatedTarget</u> 属性进行访问
<b><u>shown.bs.modal</u></b>	此事件在模态框已经显示出来（并且同时在 CSS 过渡效果完成）之后被触发。如果是通过点击某个作为触发器的元素，则此元素可以通过事件的 <u>relatedTarget</u> 属性进行访问
<b><u>hide.bs.modal</u></b>	hide 方法调用之后立即触发该事件
<b><u>hidden.bs.modal</u></b>	此事件在模态框被隐藏（并且同时在 CSS 过渡效果完成）之后被触发



## 9.b04\_a01\_百度登录框

---