[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/DEVWEB/DAY02/CASE/01/index.html" \l "page_top_case)

# NSD Devweb DAY02

1. [案例1：CSS 基础选择器](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/DEVWEB/DAY02/CASE/01/index.html" \l "case1)
2. [案例2：编写Bootstrap模板](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/DEVWEB/DAY02/CASE/01/index.html" \l "case2)
3. [案例3：使用全局 CSS 样式](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/DEVWEB/DAY02/CASE/01/index.html" \l "case3)
4. [案例4：使用常用组件](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/DEVWEB/DAY02/CASE/01/index.html" \l "case4)
5. [例4：使用常用插件](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/DEVWEB/DAY02/CASE/01/index.html" \l "case5)

## 1 案例1：CSS 基础选择器

### 1.1 问题

【参见COOKBOOK】

1. 使用 CSS 基础选择器为页面元素设置样式

### 1.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：通用选择器

通用选择器，显示为一个星号（\*）

可以与任何元素匹配

常用于设置一些默认样式，比如设置整个文档的文本的默认字体和大小、颜色等

1. <html>
2. <head>
3. <style>
4. \* {color : red;font-size : 9pt;font-family : "Times New Roman";}
5. </style>
6. </head>
7. <body>
8. <h1>我的第一个标题</h1>
9. <p>我的第一个段落</p>
10. <h2>我的第二个标题</h2>
11. </body>
12. </html>

实例演示如图-1所示：



图-1

步骤二：元素选择器

html 文档的元素就是选择器， 比如 <p>、<h1> 等

1. <html>
2. <head>
3. <style>
4. h1 {color : blue ;}
5. h2 {color : silver ;}
6. </style>
7. </head>
8. <body>
9. <h1>我的第一个标题</h1>
10. <p>我的第一个段落</p>
11. <h2>我的第二个标题</h2>
12. </body>
13. </html>

实例演示如图-2所示：



图-2

步骤三：类选择器

1)语法为：.className { color:red;}

类名称不能以数字开头

2)所有能够附带class属性的元素都可以使用此样式声明

将元素的 class属性的值设置为样式类名

1. <html>
2. <head>
3. <style>
4. #样式表中声明一个样式类
5. .myClass
6. {
7. background-color : Pink;
8. font-size : 35pt;
9. }
10. </style>
11. </head>
12. <body>
13. #将元素的 class 属性的值设置为样式类的名称
14. <h1 class="myClass">我的第一个标题</h1>
15. <p class="myClass">我的第一个段落</p>
16. <h2>我的第二个标题</h2>
17. </body>
18. </html>

实例演示如图-3所示：

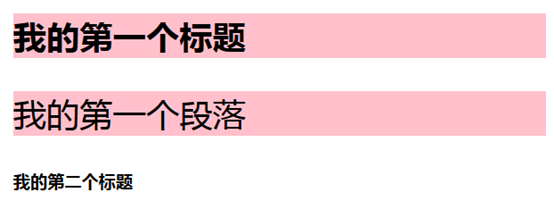


图-3

3)多类选择器

可以将多个类选择器应用于同一个元素

HTML 元素的 class 属性的值中可能包含一个词列表

各个词之间用空格分隔，每个词都是一个类选择器

1. <html>
2. <head>
3. <style>
4. #样式表中声明一个样式类
5. .important {
6. font-weight : bold;
7. }
8. .warning {
9. color : red;
10. }
11. </style>
12. </head>
13. <body>
14. #将元素的 class 属性的值设置为样式类的名称
15. <h1 class="important">我的第一个标题</h1>
16. <p class="warning">我的第一个段落</p>
17. <h2>我的第二个标题</h2>
18. </body>
19. </html>

实例演示如图-4所示：

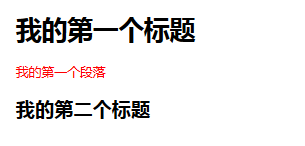


图-4

4)分类选择器

可以将类选择器和元素选择器结合起来使用，实现对某种元素中不同样式的细分控制

5)语法为：元素选择器 .className { }

先声明一个元素选择器

然后使用一个点号（.）代表将使用类选择器

然后声明一个类的名称

最后使用一对大括号声明样式规则

6)将样式规则与附带 class 属性的某种元素匹配

元素的 class 属性的值为分类选择器中指定的样式类名

1. <html>
2. <head>
3. <style>
4. #样式表中声明一个样式类
5. p.important
6. {
7. color : red ;
8. font-size : 20pt;
9. border:1px solid black;
10. }
11. </style>
12. </head>
13. <body>
14. #将元素的 class 属性的值设置为样式类的名称
15. <h1 class="important">我的第一个标题</h1>
16. <p class="important">我的第一个段落</p>
17. <h2>我的第二个标题</h2>
18. </body>
19. </html>

注意：只有 <p> 元素使用样式，实例演示如图-5所示：

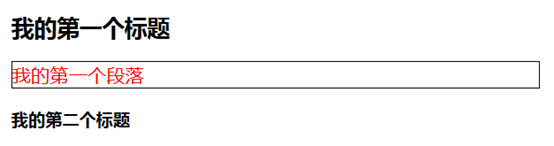


图-5

步骤四：id选择器

1)id 选择器以一种独立于文档元素的方式来指定样式

它仅作用于 id 属性的值

2)语法为：

选择器前面需要有一个 # 号

选择器本身则为文档中某个元素的 id 属性的值

1. <html>
2. <head>
3. <style>
4. #定义一个 id 选择器
5. #mostImportant
6. {
7. color:red;
8. background:yellow;
9. }
10. </style>
11. </head>
12. <body>
13. 将元素的 id 属性的值设置为选择器的名称
14. <h1 id="mostImportant">我的第一个标题</h1>
15. <p>我的第一个段落</p>
16. <h2>我的第二个标题</h2>
17. </body>
18. </html>

实例演示如图-6所示：

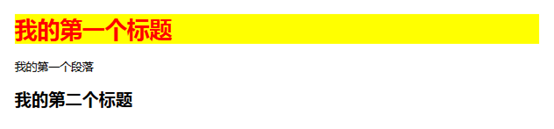


图-6

步骤五：群组选择器

选择器声明为以逗号隔开的选择器列表，将一些相同的规则作用于多个元素

1. <html>
2. <head>
3. <style>
4. h2,p.important
5. {
6. color:green;
7. font-size:20pt;
8. border:1px solid red;
9. }
10. </style>
11. </head>
12. <body>
13. <h1>我的第一个标题</h1>
14. <p class="important">我的第一个段落</p>
15. <h2>我的第二个标题</h2>
16. </body>
17. </html>

实例演示如图-7所示：

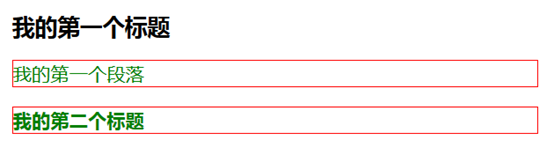


图-7

步骤六：伪类选择器

1)伪类用于向某些选择器添加特殊的效果

2)使用冒号(:)作为结合符，结合符左边是其他选择器，右边是伪类

选择器 : 伪类选择器

3)链接伪类

: link ，适用于尚未访问的链接

: visited ，适用于访问过的链接

4)动态伪类，用于呈现用户操作

:hover，适用于鼠标悬停在 HTML 元素时

:active，适用于 HTML 元素被激活时

:focus，适用于 HTML 元素获取焦点时

1. #定义伪类选择器
2. a:link {
3. color: black;
4. font-size:15pt;
5. }
6. a:visited {
7. color: pink;
8. font-size:15pt;
9. }
10. a:hover {
11. font-size : 20pt;
12. }
13. a:active {
14. color : red;
15. }

实例演示如图-8、图-9所示：

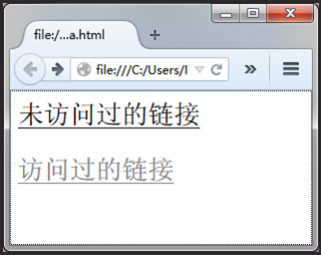


图-8



图-9

## 2 案例2：编写Bootstrap模板

### 2.1 问题

【参见COOKBOOK】

1. 使用 BootStrap 模版创建页面

### 2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：使用内联样式为<p>元素设置样式

目录说明：

1)css目录用于存放Bootstrap框架使用的样式文件：

bootstrap.css文件：Bootstrap框架的样式文件

bootstrap.min.css文件：Bootstrap框架的样式压缩文件

bootstrap-theme.css文件：Bootstrap框架的主题文件

2)fonts目录用于存放Bootstrap框架使用的字体文件：

3)js目录用于存放Bootstrap框架使用的核心javascript文件：

bootstrap.js文件：Bootstrap框架的核心javascript文件

bootstrap.min.js文件：Bootstrap框架的核心javascript压缩文件

1. <!—声明 -->
2. <!DOCTYPE html>
3. <html lang="en">
4. <head>
5. <meta charset="UTF-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <!-- 上述meta标签\*必须\*放在最前面，任何其他内容都\*必须\*跟随其后！ 元数据-->
8. <title>my bootstrap test</title>
9. <!-- Bootstrap 样式-->
10. <link rel="stylesheet" href="static/css/bootstrap.min.css">
11. </head>
12. <body>
13. <!—页面内容-->
14. <div class="container">
15. <p>
16. press <kbd>ctrl + C</kbd> to break.
17. </p>
18. <a href="http://www.bootcss.com">bootstrap中文网</a>
19. </div>
20. <!-- jQuery (Bootstrap 的所有 JavaScript 插件都依赖 jQuery)，想要使用Bootstrap框架，必须要先引入jQuery文 件 -->
21. <script src="static/js/jquery.min.js"></script>
22. <!-- 加载 Bootstrap 的所有 JavaScript 插件-->
23. <script src="static/js/bootstrap.min.js"></script>
24. </body>
25. </html>

步骤二：实例演示如图-10所示：

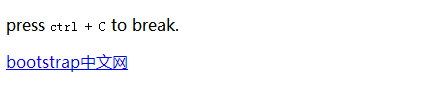


图-10

## 3 案例3：使用全局 CSS 样式

### 3.1 问题

【参见COOKBOOK】

1. 使用全局 CSS 样式定义页面

### 3.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：HTML5 文档类型

Bootstrap 使用到的某些 HTML 元素和 CSS 属性需要将页面设置为 HTML5 文档类型。在你项目中的每个页面都要参照下面的格式进行设置

1. <!DOCTYPE html>
2. <html lang="zh-CN">
3. ...
4. </html>

步骤二：移动设备优先

为了确保适当的绘制和触屏缩放，需要在<head>之中添加viewport元数据元素

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

步骤三：将Bootstrap模板补充完整

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="UTF-8">
5. <meta name="viewport" content="width=device-width, initial-scale=1">
6. <title>my bootstrap test</title>
7. <link rel="stylesheet" href="static/css/bootstrap.min.css">
8. </head>
9. <body>
10. ...
11. ...
12. <script src="static/js/jquery.min.js"></script>
13. <script src="static/js/bootstrap.min.js"></script>
14. </body>
15. </html>

步骤四：了解CSS全局样式的设置

1)为 body 元素设置 background-color: #fff;body元素背景色设置为白色

2)使用 @font-family-base、@font-size-base 和 @line-height-base 变量作为排版的基本参数

3)为所有链接设置了基本颜色 @link-color ，并且当链接处于 :hover 状态时才添加下划线

步骤五：布局容器

.container 类用于固定宽度并支持响应式布局的容器

1. <div class="container">
2. ...
3. </div>

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

1. <div class="container-fluid">
2. ...
3. </div>

步骤五：按钮

1)按钮预定义样式

使用下面列出的类可以快速创建一个带有预定义样式的按钮：

1. <button type="button" class="btn btn-default">（默认样式）Default</button>
2. <button type="button" class="btn btn-primary">（首选项）Primary</button>
3. <button type="button" class="btn btn-success">（成功）Success</button>
4. <button type="button" class="btn btn-info">（一般信息）Info</button>
5. <button type="button" class="btn btn-warning">（警告）Warning</button>
6. <button type="button" class="btn btn-danger">（危险）Danger</button>
7. <!--链接按钮鼠标悬停才有下划线-->
8. <button type="button" class="btn btn-link">（链接）Link</button>

实例演示如图-11所示：



图-11

2)按钮尺寸

使用 .btn-lg、.btn-sm 或 .btn-xs 可以获得不同尺寸的按钮

1. <p>
2. <button type="button" class="btn btn-primary btn-lg">（大按钮）Large button</button>
3. <button type="button" class="btn btn-default btn-lg">（大按钮）Large button</button>
4. </p>
5. <p>
6. <button type="button" class="btn btn-primary">（默认尺寸）Default button</button>
7. <button type="button" class="btn btn-default">（默认尺寸）Default button</button>
8. </p>
9. <p>
10. <button type="button" class="btn btn-primary btn-sm">（小按钮）Small button</button>
11. <button type="button" class="btn btn-default btn-sm">（小按钮）Small button</button>
12. </p>
13. <p>
14. <button type="button" class="btn btn-primary btn-xs">（超小尺寸）Extra small button</button>
15. <button type="button" class="btn btn-default btn-xs">（超小尺寸）Extra small button</button>
16. </p>

实例演示如图-12所示：



图-12

通过给按钮添加 .btn-block 类可以将其拉伸至父元素100%的宽度，而且按钮也变为了块级（block）元素。

1. <button type="button" class="btn btn-primary btn-lg btn-block">（块级元素）Block level button</button>
2. <button type="button" class="btn btn-default btn-lg btn-block">（块级元素）Block level button</button>

实例演示如图-13所示：



图-13

步骤六：文本

1)对齐

1. <!--设定文本左对齐-->
2. <p class="text-left">Left aligned text.</p>
3. <!--设定文本右对齐-->
4. <p class="text-center">Center aligned text.</p>
5. <!--设定文本居中对齐-->
6. <p class="text-right">Right aligned text.</p>
7. <!--设定文本对齐，段落超出屏幕部分文字自动换行-->
8. <p class="text-justify">Justified text.</p>
9. <!--段落超出屏幕部分不换行-->
10. <p class="text-nowrap">No wrap text.</p>

2)大小写

1. <!--小写（仅英文）-->
2. <p class="text-lowercase">Lowercased text.</p>
3. <!--文字大写（仅英文）-->
4. <p class="text-uppercase">Uppercased text.</p>
5. <!--首字母大写-->
6. <p class="text-capitalize">Capitalized text.</p>

实例演示如图-14所示：

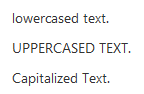


图-14

3)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可以还可以加深，就像默认的链接一样

1. <p class="text-muted">...</p>
2. <p class="text-primary">...</p>
3. <p class="text-success">...</p>
4. <p class="text-info">...</p>
5. <p class="text-warning">...</p>
6. <p class="text-danger">...</p>

实例演示如图-15所示：

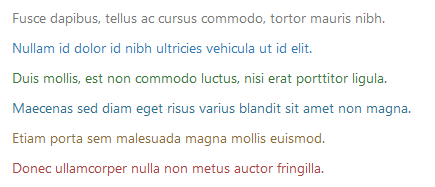


图-15

步骤七：列表

1)无样式列表

移除了默认的 list-style 样式和左侧外边距的一组元素

1. <ul class="list-unstyled">
2. <li>...</li>
3. </ul>

实例演示如图-16所示：

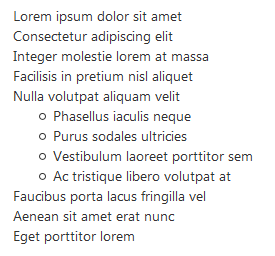


图-16

2)内联列表

通过设置 display: inline-block，将所有元素放置于同一行。

1. <ul class="list-inline">
2. <li>...</li>
3. </ul>

实例演示如图-17所示：

IMG_272

图-17

3)描述

带有描述的短语列表

1. <dl>
2. <dt>...</dt>
3. <dd>...</dd>
4. </dl>

实例演示如图-18所示：

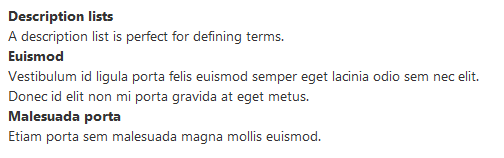


图-18

水平排列的描述

.dl-horizontal 可以让 <dl> 内的短语及其描述排在一行。开始是像 <dl> 的默认样式堆叠在一起，随着导航条逐渐展开而排列在一行。

1. <dl class="dl-horizontal">
2. <dt>...</dt>
3. <dd>...</dd>
4. </dl>

实例演示如图-19所示：

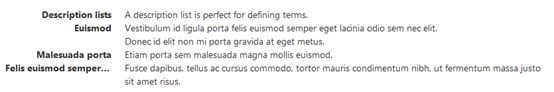


图-19

步骤八：表单

1)基本表单

实现基本的表单样式

1. <form action=“” role=“form”>
2. <div class="form-group">
3. <label>电子邮件：</label>
4. <input type="email" class="form-control" placeholder="email地址">
5. </div>
6. <div class="form-group">
7. <label>密码</label>
8. <input type="password" class="form-control" placeholder="输入密码">
9. </div>
10. <input class=“btn btn-primary” type=“submit”value=“提交”>
11. </form>

实例演示如图-20所示：

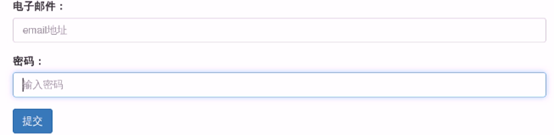


图-20

2)内联表单

让表单左对齐浮动，并表现为inline-block 内联块结构，为上面代码form元素增加类

1. <form class="form-inline">

实例演示如图-21所示：

IMG_276

图-21

3)表单合组

前后增加片段

1. <form action=“” role=“form”>
2. <div class="input-group">
3. <div class="input-group-addon">￥</div>
4. <input type="text" class="form-control">
5. <div class="input-group-addon">.00</div>
6. </div>
7. </form>

实例演示如图-22所示：

IMG_277

图-22

4)水平排列

让表单内的元素保持水平排列

1. <form class="form-horizontal">
2. <div class="form-group">
3. <label class="col-sm-2 control-label">电子邮件</label>
4. <div class="col-sm-10">
5. <input type="email" class="form-control" placeholder="请输入您的电子邮件">
6. </div>
7. </div>
8. </form>

注：这里用到了col-sm 栅格系统，后面章节会重点讲解，而control-label 表示和

父元素样式同步。

5)复选框和单选框

设置复选框，在一行

1. <div class="checkbox">
2. <label>
3. <input type="checkbox">体育
4. </label>
5. </div>
6. <div class="checkbox">
7. <label>
8. <input type="checkbox">音乐
9. </label>
10. </div>

设置禁用的复选框

1. <div class="checkbox disabled">
2. <label>
3. <input type="checkbox" disabled>音乐
4. </label>
5. </div>

设置内联一行显示的复选框

1. <label class="checkbox-inline">
2. <input type="checkbox">体育
3. </label>
4. <label class="checkbox-inline disabled">
5. <input type="checkbox" disabled>音乐
6. </label>

设置单选框

1. <div class="radio disabled">
2. <label>
3. <input type="radio" name="sex" disabled>男
4. </label>

6)下拉列表

设置下拉列表

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

## 4 案例4：使用常用组件

### 4.1 问题

【参见COOKBOOK】

1. 使用常用组件为页面添加元素

### 4.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：字体图标

1)Bootstrap提供了一套专用于Web开发/移动开发常用的

一套图标字体——Glyphicons Halflings

基本样式类 glyphicon

其他修饰类

2)注意事项

引入 fonts 文件

不要和其他组件混合使用

只对内容为空的元素起作用

1. <button type="button" class="btn btn-default" aria-label="Left Align">
2. <span class="glyphicon glyphicon-align-left" aria-hidden="true"></span>
3. </button>
4. <button type="button" class="btn btn-default btn-lg">
5. <span class="glyphicon glyphicon-star" aria-hidden="true"></span> Star
6. </button>

实例演示如图-23所示：

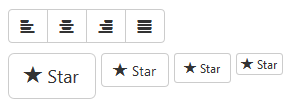


图-23

步骤二：下拉菜单

1)基本格式

将下拉菜单触发器和下拉菜单都包裹在 .dropdown 里，或者另一个声明了 position: relative; 的元素。然后加入组成菜单的 HTML 代码

1. <div class="dropdown">
2. <button class="btn btn-default dropdown-toggle" type="button" id="dropdownMenu1" data-toggle="dropdown" aria-haspopup="true" aria-expanded="true">
3. Dropdown
4. <span class="caret"></span>
5. </button>
6. <ul class="dropdown-menu" aria-labelledby="dropdownMenu1">
7. <li><a href="#">Action</a></li>
8. <li><a href="#">Another action</a></li>
9. <li><a href="#">Something else here</a></li>
10. <li role="separator" class="divider"></li>
11. <li><a href="#">Separated link</a></li>
12. </ul>
13. </div>

实例演示如图-24所示：

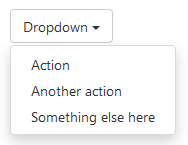


图-24

2)标题

1. <ul class="dropdown-menu" aria-labelledby="dropdownMenu3">
2. ...
3. <li class="dropdown-header">Dropdown header</li>
4. ...
5. </ul>

实例演示如图-25所示：

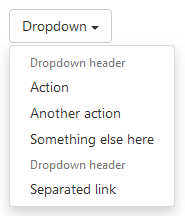


图-25

3)分割线

1. <ul class="dropdown-menu" aria-labelledby="dropdownMenuDivider">
2. ...
3. <li role="separator" class="divider"></li>
4. ...
5. </ul>

实例演示如图-26所示：

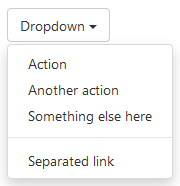


图-26

4)禁用的菜单项

为下拉菜单中的 <li> 元素添加 .disabled 类，从而禁用相应的菜单项

1. <ul class="dropdown-menu" aria-labelledby="dropdownMenu4">
2. <li><a href="#">Regular link</a></li>
3. <li class="disabled"><a href="#">Disabled link</a></li>
4. <li><a href="#">Another link</a></li>
5. </ul>

实例演示如图-27所示：

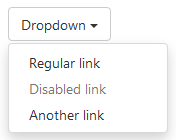


图-27

步骤三：按钮组

1)基本按钮组

1. <div class="btn-group" role="group" aria-label="...">
2. <button type="button" class="btn btn-default">Left</button>
3. <button type="button" class="btn btn-default">Middle</button>
4. <button type="button" class="btn btn-default">Right</button>
5. </div>

实例演示如图-28所示：

IMG_283

图-28

2)按钮工具栏

1. <div class="btn-toolbar" role="toolbar" aria-label="...">
2. <div class="btn-group" role="group" aria-label="...">...</div>
3. <div class="btn-group" role="group" aria-label="...">...</div>
4. <div class="btn-group" role="group" aria-label="...">...</div>
5. </div>

实例演示如图-29所示：

IMG_284

图-29

3)尺寸

1. <div class="btn-group btn-group-lg" role="group" aria-label="...">...</div>
2. <div class="btn-group" role="group" aria-label="...">...</div>
3. <div class="btn-group btn-group-sm" role="group" aria-label="...">...</div>
4. <div class="btn-group btn-group-xs" role="group" aria-label="...">...</div>

实例演示如图-30所示：



图-30

4)垂直排列

1. <div class="btn-group-vertical" role="group" aria-label="...">
2. ...
3. </div>

实例演示如图-31所示：

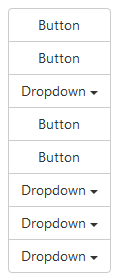


图-31

步骤四：警告框

警告框组件通过提供一些灵活的预定义消息，为常见的用户动作提供反馈消息

将任意文本和一个可选的关闭按钮组合在一起就能组成一个警告框，.alert 类是必须要设置的，另外我们还提供了有特殊意义的4个类（例如，.alert-success），代表不同的警告信息。

1. <div class="alert alert-success" role="alert">...</div>
2. <div class="alert alert-info" role="alert">...</div>
3. <div class="alert alert-warning" role="alert">...</div>
4. <div class="alert alert-danger" role="alert">...</div>

实例演示如图-32所示：

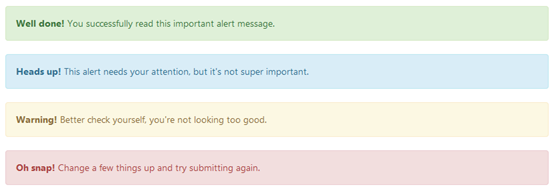


图-32

## 5 例4：使用常用插件

### 5.1 问题

【参见COOKBOOK】

1. 使用常用插件为页面添加元素

### 5.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：标签页

1)标签页

通过data属性:添加data-toggle="tab"或data-toggle="pill"到锚文本链接中

添加nav 和nav-tabs 类到ul中，将会应用Bootstrap标签样式，添加nav和 nav-pills类到ul中，将会应用Bootstrap胶囊式样式

1. <ul class="nav nav-tabs">
2. <li class="active"><a href="#home" data-toggle="tab">Home</a></li>
3. <li><a href="#Pro" data-toggle="tab">Profile</a></li>
4. <li><a href="#Pro" data-toggle="tab">Dropdown</a></li>
5. </ul>

实例演示如图-33所示：

IMG_288

图-33

2)淡入淡出效果

如果需要为标签页设置淡入淡出效果，请添加 .fade 到每个 .tab-pane 后面。第一个标签页必须添加 .in 类，以便淡入显示初始内容

1. <div class="tab-content">
2. <div class="tab-pane fade in active" id="home">...</div>
3. <div class="tab-pane fade" id="svn">...</div>
4. <div class="tab-pane fade" id="ios">...</div>
5. <div class="tab-pane fade" id="java">...</div>
6. </div>

3)折叠框

data-toggle="collapse"添加到您想要展开或折叠的组件的链接上。

href或data-target属性添加到父组件，它的值是子组件的id。

data-parent属性把折叠面板（accordion）的id添加到要展开或折叠的组件链接上。

1. <div class="panel-group" id="accordion">
2. <div class="panel panel-default">
3. <div class="panel-heading">
4. <h4 class="panel-title">
5. <a data-toggle="collapse" data-parent="#accordion"
6. href="#collapseOne">
7. 点击我进行展开，再次点击我进行折叠。第 1 部分
8. </a>
9. </h4>
10. </div>
11. <div id="collapseOne" class="panel-collapse collapse in">
12. <div class="panel-body">
13. Nihil anim keffiyeh helvetica, craft beer labore wes anderson
14. cred nesciunt sapiente ea proident. Ad vegan excepteur butcher
15. vice lomo.
16. </div>
17. </div>
18. </div>
19. <div class="panel panel-default">
20. <div class="panel-heading">
21. <h4 class="panel-title">
22. <a data-toggle="collapse" data-parent="#accordion"
23. href="#collapseTwo">
24. 点击我进行展开，再次点击我进行折叠。第 2 部分
25. </a>
26. </h4>
27. </div>
28. <div id="collapseTwo" class="panel-collapse collapse">
29. <div class="panel-body">
30. Nihil anim keffiyeh helvetica, craft beer labore wes anderson
31. cred nesciunt sapiente ea proident. Ad vegan excepteur butcher
32. vice lomo.
33. </div>
34. </div>
35. </div>
36. <div class="panel panel-default">
37. <div class="panel-heading">
38. <h4 class="panel-title">
39. <a data-toggle="collapse" data-parent="#accordion"
40. href="#collapseThree">
41. 点击我进行展开，再次点击我进行折叠。第 3 部分
42. </a>
43. </h4>
44. </div>
45. <div id="collapseThree" class="panel-collapse collapse">
46. <div class="panel-body">
47. Nihil anim keffiyeh helvetica, craft beer labore wes anderson
48. cred nesciunt sapiente ea proident. Ad vegan excepteur butcher
49. vice lomo.
50. </div>
51. </div>
52. </div>
53. </div>

实例演示如图-34所示：

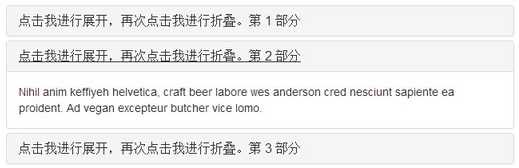


图-34

3)工具提示

提示工具（Tooltip）插件根据需求生成内容和标记，默认情况下是把提示工具（tooltip）放在它们的触发元素后面。

通过 data 属性：如需添加一个提示工具（tooltip），只需向一个标签添加 data-toggle="tooltip" 即可。标签的 title 即为提示工具（tooltip）的文本。默认情况下，插件把提示工具（tooltip）设置在顶部。

1. <button data-toggle="tooltip" title="默认的 Tooltip"> 默认的 Tooltip</button>
2. <button data-toggle="tooltip" data-placement="left" title="左侧的 Tooltip">左侧的 Tooltip</button>
3. <button data-toggle="tooltip" data-placement="top" title="顶部的 Tooltip">顶部的 Tooltip</button>
4. <button data-toggle="tooltip" data-placement="bottom" title="底部的 Tooltip">底部的 Tooltip</button>
5. <button data-toggle="tooltip" data-placement="right" title="右侧的 Tooltip">右侧的 Tooltip</button>

提示工具（Tooltip）插件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（读取 javascript）。使用下面的脚本来启用页面中的所有的提示工具（tooltip）：

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

实例演示如图-35、图-36所示：

IMG_290

图-35

IMG_291

图-36

4)弹出框

弹出框（Popover）插件根据需求生成内容和标记，默认情况下是把弹出框（popover）放在它们的触发元素后面。

通过 data 属性：如需添加一个弹出框（popover），只需向一个锚/按钮标签添加 data-toggle="popover" 即可。锚的 title 即为弹出框（popover）的文本。默认情况下，插件把弹出框（popover）设置在顶部。。

1. <a href="#" data-toggle="popover" title="请悬停在我的上面">
2. 请悬停在我的上面
3. </a>

弹出框（Popover）插件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（读取 javascript）。使用下面的脚本来启用页面中的所有的弹出框（popover）：

1. <script>
2. $(function () { $("[data-toggle='popover']").popover(); });
3. </script>

实例演示如图-37所示：

IMG_292

图-37