目录

[Bootstrap学习笔记 2](#_Toc21140)

[表格 table 3](#_Toc15942)

[表单 4](#_Toc16196)

[表单控件 4](#_Toc9723)

[按钮 5](#_Toc21233)

[Input 8](#_Toc3632)

[文本框控件 8](#_Toc9667)

[图标 9](#_Toc5130)

[下拉菜单 10](#_Toc14175)

[图片 11](#_Toc6990)

[导航栏 11](#_Toc31847)

[导航条 12](#_Toc3674)

[分页导航 15](#_Toc25849)

[固定导航条 16](#_Toc30501)

[响应式导航条 17](#_Toc3130)

[导航超文本连接 18](#_Toc286)

[列表组 18](#_Toc308)

[缩略图 19](#_Toc13088)

[进度条 20](#_Toc23429)

[面板 21](#_Toc27086)

[媒体对象 22](#_Toc8887)

[过渡动画 24](#_Toc7708)

# Bootstrap学习笔记

1：导入bootstrap css样式文件

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

或者在线css文件样式

<linkrel="stylesheet"href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">

2：导入 jquery 和 bootstrapjs文件

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

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

加入控制头：

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

<!--控制 视口的 大小-->

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

<i>斜体</i> <em> 倾斜</em> 这两个标签实现文字倾斜

**文字位置**

.text-left 文字居左 .text-right 居右 .text-center居中 .text-justify 两端对齐

<p class="text-left">我居左</p>

**文字颜色**

.text-muted：提示，使用浅灰色（#999）

.text-primary：主要，使用蓝色（#428bca）

.text-success：成功，使用浅绿色(#3c763d)

.text-info：通知信息，使用浅蓝色（#31708f）

.text-warning：警告，使用黄色（#8a6d3b）

.text-danger：危险，使用褐色（#a94442）

<div class="text-success">.text-success效果</div>

list-unstyled ：实现去点列表

list-inline ：列表放到一行（内联列表），同时去点

<ul class="list-unstyled">

### 表格 table

<table class="table">

.table：基础表格

.table-striped：斑马线表格

.table-bordered：带边框的表格

.table-hover：鼠标悬停高亮的表格

.table-condensed：紧凑型表格

.table-responsive：响应式表格

几个值可以同时使用

用在tr 上的 class 值（ 表格行的类）

<tr class="active">

class 可以 同时引用 多个样式 中间用 空格隔开

控制的是 颜色等样式的信息

.active 表示当前活动的信息

.success 表示成功或者正确的行为

.info 表示中立的信息或行为

.warning 表示警告，需要特别注意

.danger 表示危险或者可能是错误的行为

### 表单

Bootstrap框架中的**“form-control”**样式的表单控件宽度为100%或auto。

宽度占全屏

<form role="form">

Form ：基础表单

<form class="form-horizontal" role="form">

form-horizontal：水平表单

form-inline：内联表单

form-group: 控制 样式大小

class="col-sm-10" ： 属性 限制div大小

Placeholder：占位符 ，背景文字

<div class="form-group">

<div class="col-sm-10" >

<input type="email" class="form-control" placeholder="请输入您的邮箱地址">

</div>

</div>

### 表单控件

**下拉框**：

select 添加 属性 multiple （值可以省略）变为多行显示

<div class="form-group" style="width: 100px;">

<select class="form-control">

<option>1</option>

<option>5</option>

</select>

</div>

**文本域textarea ：**rows跨行 设置高度：cols：设置宽度

<div class="form-group">

<textarea class="form-control" rows="1" cols="1"></textarea>

</div>

**复选框checkbox，单选框：radio**

1、不管是checkbox还是radio都使用label包起来

2、checkbox连同label标签放置在一个名为“.checkbox”的容器内

3、radio连同label标签放置在一个名为“.radio”的容器内

在Bootstrap框架中，主要借助“.checkbox”和“.radio”样式，来处理复选框、单选按钮与标签的对齐方式。

class="checkbox-inline" 加到 label 标签上 复选框会在 同一行

//复选框

<div class="checkbox">

<!--复选框-->

<label class="checkbox-inline">

<input type="checkbox" value="">

记住密码

</label>

</div>

//单选框

<div class="radio">

<label class="radio-inline">

<input type="radio" value="love" checked>

喜欢

</label>

<label class="radio-inline">

<input type="radio" >

不喜欢

</label>

</div>

### 按钮

在Bootstrap框架中的按钮都是采用<button>来实现。

btn 这个类名是不可缺少的

这些 按钮的 class属性值 也可以 用到 input a span div等标签

btn-lg 变大 btn-sm 变小 btn-xs 超小 可以控制 按钮大小

通过 .disabled 来禁用按钮

按钮也是表单重要控件之一,制作按钮通常使用下面代码来实现：

input[type=“submit”]

input[type=“button”]

input[type=“reset”]

<button>

通过 class 属性的值 btn btn-success ,btn-warning, btn-danger

...等 控制 他的样式

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

#### 按钮垂直分组

前面看到的示例，按钮组都是水平显示的。但在实际运用当中，

总会碰到垂直显示的效果。在Bootstrap框架中也提供了这样的风格。

我们只需要把水平分组的“btn-group”类名换成“btn-group-vertical”即可。如下所示：

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

vertical： 垂直

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

<button class="btn btn-default" type="button">案例分析</button>

<button class="btn btn-default" type="button">联系我们</button>

<div class="btn-group">

<button class="btn btn-default dropdown-toggle" data-toggle="dropdown" type="button">关于我们<span class="caret"></span></button>

<ul class="dropdown-menu">

<li><a href="#">企业文化</a></li>

<li><a href="#">客服服务</a></li>

</ul>

</div>

</div>

#### 按钮工具栏

那么Bootstrap框架按钮工具栏也提供了这样的制作方法，

你只需要将按钮组“btn-group”**按组放在一个大的容器“btn-toolbar”**

.btn-group-lg:大按钮组

.btn-group-sm:小按钮组

.btn-group-xs:超小按钮组

<div class=**"btn-toolbar"**>

<div class="btn-group">

<button type="button" class="btn btn-default"><span class="glyphicon glyphicon-align-right"></span></button>

<button type="button" class="btn btn-default"><span class="glyphicon glyphicon-align-justify"></span></button>

</div>

<div class="btn-group">

<button type="button" class="btn btn-default"><span class="glyphicon glyphicon-indent-left"></span></button>

<button type="button" class="btn btn-default"><span class="glyphicon glyphicon-indent-right"></span></button>

</div>

<div class="btn-group">

<button type="button" class="btn btn-default"><span class="glyphicon glyphicon-bold"></span></button>

<button type="button" class="btn btn-default"><span class="glyphicon glyphicon-italic"></span></button>

</div>

<div class="btn-group">

<button type="button" class="btn btn-default"><span class="glyphicon glyphicon-text-height"></span></button>

<button type="button" class="btn btn-default"><span class="glyphicon glyphicon-text-width"></span></button>

</div>

</div>

#### 等分按钮

等分按钮的效果在移动端上特别的实用。整个按钮组宽度是容器的100%，

而按钮组里面的每个按钮平分整个容器宽度。例如，如果你按钮组里面有五个按钮，

那么每个按钮是20%的宽度，如果有四个按钮，那么每个按钮是25%宽度，以此类推。

等分按钮也常被称为是自适应分组按钮，其实现方法也非常的简单，

只需要在按钮组“btn-group”上追加一个“btn-group-justified”类名，如下所示：

<div class="btn-wrap">

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

<a class="btn btn-default" href="#">首页</a>

<a class="btn btn-default" href="#">联系我们</a>

</div>

</div>

### Input

.form-control : 会让 input 宽度占全屏

input-lg 能使控件变大

input-sm 能使 input 变小 高度变小

这两个类适用于表单中的input，textarea和select控件

col-xs-1：控制 input的宽度，直接在 input class中使用无效

<div class="col-xs-1">

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

</div>

### 文本框控件

平常在制作表单验证时，要提供不同的提示信息。在Bootstrap框架中也提供了这样的效果。

使用了一个"help-block"样式，将提示信息以块状显示，并且显示在控件底部。

<span class="help-block">你输入的信息是正确的</span>

<div class="form-group has-success has-feedback">

<label class="control-label" for="inputSuccess1">成功状态</label>

<input type="text" class="form-control" id="is" placeholder="成功状态" >

<span class="help-block">你输入的信息是正确的</span>

<span class="glyphicon glyphicon-ok form-control-feedback"></span>

</div>

1、.has-warning:警告状态（黄色）

2、.has-error：错误状态（红色）

3、.has-success：成功状态（绿色）

使用的时候只需要在form-group容器上对应添加状态类名

<div class="form-group has-success">

<label class="control-label" for="inputSuccess1">成功状态</label>

<input type="text" class="form-control" id="is" placeholder="成功状态" >

</div>

### 图标

很多时候，在表单验证的时候，不同的状态会提供不同的 icon，

比如成功是一个对号（√），错误是一个叉号（×）等。在Bootstrap框

中也提供了这样的效果。如果你想让表单在对应的状态下显示 icon 出来

，只需要在对应的状态下添加类名“has-feedback”。请注意，此类名要与

“has-error”、“has-warning”和“has-success”在一起：

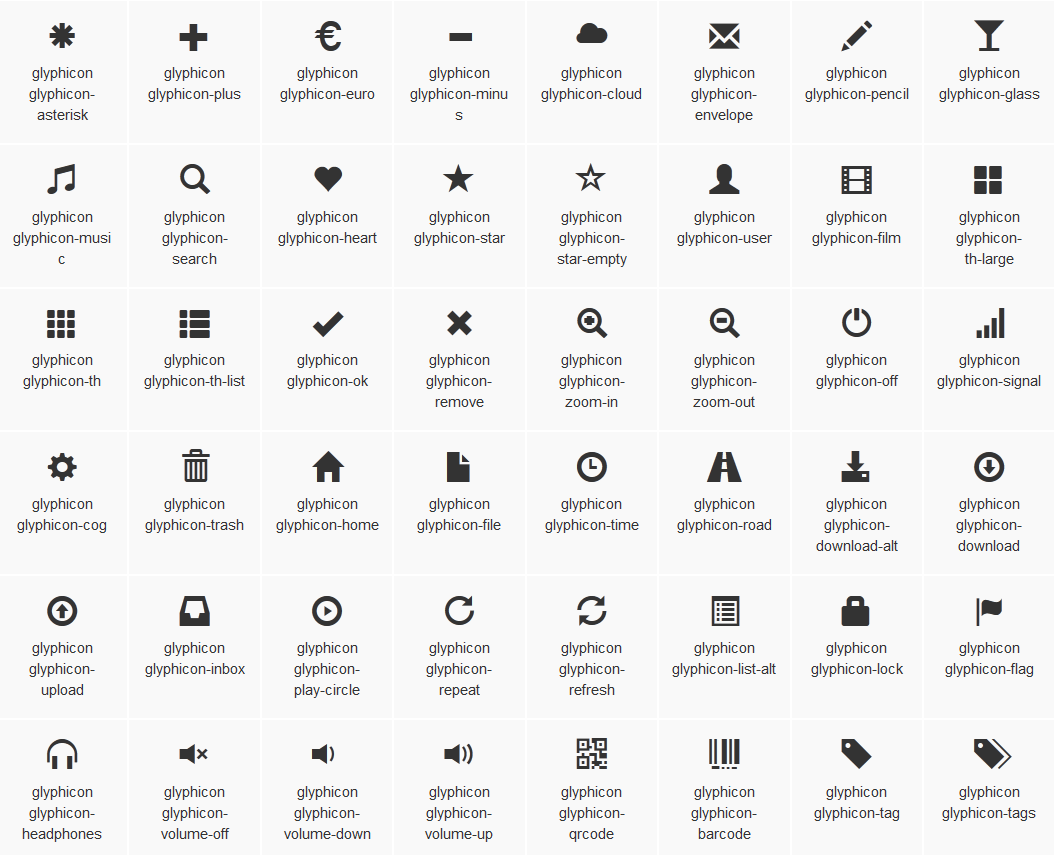
<div class="form-group has-success has-feedback">

<label class="control-label" >成功状态</label>

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

<span class="glyphicon glyphicon-ok form-control-feedback"></span>

</div>



### 下拉菜单

是在使用Bootstrap框架中的下拉菜单组件时，其结构运用的正确与否非常的重要如果结构和类名未使用正确，直接影响组件是否能正常运用

1. 使用一个名为“dropdown”的容器包裹了整个下拉菜单元素

2、使用了一个<button>按钮做为父菜单，并且定义类名“dropdown-toggle” (下拉菜单切换 之意)

和自定义“data-toggle”属性 : 这个有什么用

且值必须和最外容器类名一致，此示例为:

data-toggle="dropdown"

3、下拉菜单项使用一个ul列表，并且定义一个类名为“dropdown-menu”

divider: 给下拉菜单设置分割线 起分割菜单的作用

dropdown-header ： 给下拉菜单设置 标题 提示 作用

<li role="presentation" class="dropdown-header">第一部分菜单头部</li>

使用 pull-right 使下拉菜单和 父容器 右对齐和 dropdown-menu-right 属性值 作用一样 右对齐

dropdown-menu-left 左对齐

<ur class=”dropdown-menupull-right” role=”menu”>

下拉菜单的状态： .active ：当前状态 默认被选中的状态

.disabled 禁用状态

dropdown-menu 该样式 默认将 下拉菜单隐藏

<div class="dropdown" style="border: 1px solid;">

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

下拉菜单

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

</button>

**<!--dropdown-menu 该样式 默认将 下拉菜单隐藏 -->**

<ul class="dropdown-menu dropdown-menu-right" role="menu" aria-labelledby="dropdownMenu1">

<!--头部-->

<li role="presentation" class="dropdown-header">第一部分菜单头部</li>

<li role="presentation" ><a role="menuitem" tabindex="-1" href="#">下拉菜单项</a></li>

<li role="presentation" class="active"><a role="menuitem" tabindex="-1" href="#">当前下拉状态</a></li>

<li role="presentation" class="disabled"><a role="menuitem" tabindex="-1" href="#">禁用下拉菜单</a></li>

<!--分割线-->

<li role="presentation" class="divider"><a role="menuitem" tabindex="-1" href="#">下拉菜单项</a></li>

<li role="presentation" class="dropdown-header">第二部分菜单头部</li>

<li role="presentation" ><a role="menuitem" tabindex="-1" href="#">下拉菜单项</a></li>

</ul>

</div>

Breadcrumb :改变样式

<ol class="breadcrumb">

<li><a href="#">首页</a></li>

<li><a href="#">我的书</a></li>

<li class="active">《图解CSS3》</li>

</ol>

### 图片

1、img-responsive：响应式图片，主要针对于响应式设计

2、img-rounded：圆角图片

3、img-circle：圆形图片

4、img-thumbnail：缩略图片

<img class="img-circle" alt="140x140" src="../">

### 导航栏

Bootstrap框架中制作导航条主要通过 ".nav" 样式。默认的 ".nav" 样式不提供默认的导航样式

必须附加另外一个样式才会有效，比如 "nav-tabs"、"nav-pills" 之类。比如右侧代码编辑器中

就有一个tab导航条的例子，他的实现方法就是为ul标签加入.nav和nav-tabs两个类样式。

active 默认被选中 disabled 禁用

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

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

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

<li class="disabled"><a href="#">Responsive</a></li>

</ul>

导航下拉菜单

nav-pills 胶囊导航

nav-pills nav-stacked 组合使用形成垂直堆叠的导航栏

nav-justified 再加个 这 自适应导航栏

<ul class="nav nav-pills">

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

<li class="dropdown">

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

<ul class="dropdown-menu">

<li><a href="##">CSS3</a></li>

<li><a href="##">CSS3</a></li>

</ul>

</li>

<li><a href="##">关于我们</a></li>

</ul>

### 导航条

导航条（navbar）和上一节介绍的导航（nav），

就相差一个字，多了一个“条”字。其实在Bootstrap框架中

他们还是明显的区别。**在导航条(navbar)中有一个背景色、而且**

**导航条可以是纯链接（类似导航），也可以是表单，还有就是表单和**

**导航一起结合等多种形式。**role 属性 有什么用？？

导航条

<div class="navbar navbar-default" role="navigation">

<ul class="nav navbar-nav">

<li class="active"><a href="#">网站首页</a></li>

<li><a href="##">系列教程</a></li>

<li><a href="##">关于我们</a></li>

</ul>

</div

导航条

<div class="navbar navbar-default" role="navigation">

　<div class="navbar-header">

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

　</div>

<!--导航栏-->

<ul class="nav navbar-nav">

<li class="active"><a href="##">网站首页</a></li>

<!--

class="dropdown" 和 class="dropdown-toggle" 有什么用

-->

<li class="dropdown">

<a href="##" data-toggle="dropdown" class="dropdown-toggle" >系列教程<span class="caret"></span></a>

<ul class="dropdown-menu">

<li><a href="##">CSS3</a></li>

<li class="disabled"><a href="##">PHP</a></li>

</ul>

</li>

<li><a href="##">名师介绍</a></li>

</ul>

<form action="##" class="navbar-form navbar-left" role="search">

<div class="form-group">

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

</div>

<button type="submit" class="btn btn-default">搜索</button>

</form>

</div>

很多时候，我们常把下拉菜单和普通的按钮组排列在一起，实现类似于导航菜单的效果。如下所示：

<span class="caret"></span>:向下的三角型

<div class="btn-group dropup"> 在 class 属性值上 加 dropup 就变成向上的三角型，下拉菜单也变到上面

<div class="btn-group" >

<button class="btn btn-default" type="button">首页</button>

<div class="btn-group dropup">

<button class="btn btn-default dropdown-toggle" data-toggle="dropdown" type="button">产品展示<span class="caret"></span></button>

<ul class="dropdown-menu">

<li><a href="#">拖拉机</a></li>

<li><a href="#">收割机</a></li>

</ul>

</div>

<button class="btn btn-default" type="button">案例分析</button>

<!--下拉菜单-->

<div class="btn-group">

<button class="btn btn-default dropdown-toggle" data-toggle="dropdown" type="button">关于我们<span class="caret"></span></button>

<ul class="dropdown-menu">

<li><a href="#">公司简介</a></li>

<li><a href="#">企业文化</a></li>

</ul>

</div>

</div>

### 分页导航

在Bootstrap框架中提供了**两种分页导航**：

1 带页码的分页导航

2 带翻页的分页导航

pagination : 页码，标记页数

平时很多同学喜欢用div>a和div>span结构来制作带页码的分页导航。

不过，在Bootstrap框架中使用的是ul>li>a这样的结构，在ul标签上加入pagination方法：

disabled ：禁用状态

pagination-lg：分页导航变大

pagination-sm ： 分页导航变小

#### 分页导航条

<ul class="pagination pagination-lg">

<li><a href="#">&laquo;第一页</a></li>

<li><a href="#">11</a></li>

<li class="active"><a href="#">13</a></li>

<li class="disabled"><a href="#">最后一页&raquo;</a></li>

</ul>

#### 翻页导航条 pager

<ul class="pager">

<li><a href="#">&laquo;上一页</a></li>

<li><a href="#">下一页&raquo;</a></li>

</ul>

<!--左右对齐-->

<ul class="pager">

<li class="previous"><a href="#">&laquo;上一页</a></li>

<li class="next"><a href="#">下一页&raquo;</a></li>

</ul>

### 固定导航条

.navbar-fixp：导航条固定在浏览器窗口顶部

.navbar-fixed-bottom：导航条固定在浏览器窗口底部

<div class="navbar navbar-default navbar-fixed-top" role="navigation">

　<div class="navbar-header">

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

　</div>

<ul class="nav navbar-nav">

<li class="active"><a href="##">网站首页</a></li>

<li><a href="##">关于我们</a></li>

</ul>

</div>

### 响应式导航条

响应式导航条 （适应于各种屏幕）

1、保证在窄屏时需要折叠的内容必须包裹在带一个div内，

并且为这个div加入collapse、navbar-collapse两个类名。最后为这个div添加一个class类名或者id名。

2、保证在窄屏时要显示的图标样式（固定写法）

**navbar-inverse 替换掉 navbar-default 导航条背景变为黑色（反色导航条）**

<div class="navbar navbar-inverse " role="navigation">

<div class="navbar-header">

　<!-- .navbar-toggle样式用于toggle收缩的内容，即nav-collapse collapse样式所在元素 -->

<button class="navbar-toggle" type="button" data-toggle="collapse" data-target=".navbar-responsive-collapse">

<span class="sr-only">Toggle Navigation</span>

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

</button>

<!-- 确保无论是宽屏还是窄屏，navbar-brand都显示 -->

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

</div>

<!-- 屏幕宽度小于768px时，div.navbar-responsive-collapse容器里的内容都会隐藏，显示icon-bar图标，当点击icon-bar图标时，再展开。屏幕大于768px时，默认显示。 -->

<div class="collapse navbar-collapse navbar-responsive-collapse">

<ul class="nav navbar-nav">

<li class="active"><a href="##">网站首页</a></li>

<li><a href="##">系列教程</a></li>

</ul>

</div>

</div>

### 导航超文本连接

1、导航条中的按钮navbar-btn

2、导航条中的文本navbar-text

3、导航条中的普通链接navbar-link

<div class="navbar navbar-default" role="navigation">

　<div class="navbar-header"> <!--导航条标题-->

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

　</div>

<!--超链接 外面加一层 li 标签 超链接的 下划线就没了-->

<ul class="nav navbar-nav">

<li><a href="##" class="navbar-text">Navbar Text</a></li>

</ul>

</div>

### 列表组

用来做列表清单垂直导航等效果

list-group：列表组容器，常用的是ul元素，当然也可以是ol或者div元素

list-group-item：列表项，常用的是li元素，当然也可以是div元素

//多彩列表组

<div class="list-group">

<a href="##" class="list-group-item active"><span class="badge">5902</span>图解CSS3</a>

<a href="##" class="list-group-item list-group-item-success"><span class="badge"> 15902</span>W3cplus</a>

</div>

//普通列表组

<ul class="list-group">

<li class="list-group-item">

<span class="badge">13</span>揭开CSS3的面

</li>

</ul>

//自定义列表组

<div class="list-group">

<a href="##" class="list-group-item">

<h4 class="list-group-item-heading">图解CSS3</h4>

<p class="list-group-item-text">详细讲解了选择器、边框、背景、文本、颜色、盒模型、伸缩布局盒模型、多列布局、渐变、过渡、动画、媒体、响应Web设计、Web字体等主题下涵盖的所有CSS3新特性...</p>

</a>

</div>

**警示框 alert 类可以追加 alert-success 等样式**

//普通警示框

<div class="alert alert-info" role="alert">请输入正确的密码</div>

//可关闭的警示框

<div class="alert alert-success alert-dismissable" role="alert">

<button class="close" type="button" data-dismiss="alert">&times;</button>

恭喜您操作成功！

</div>

//可链接的警示框

<div class="alert alert-success" role="alert">

<strong>Well done!</strong>

You successfully read

<a href="#" class="alert-link">this important alert message</a>

</div>

### 缩略图

缩略图在网站中最常用的地方就是产品列表页面，一行显示几张图片，

有的在图片底下（左侧或右侧）带有标题、描述等信息。Bootstrap框架

将这一部独立成一个模块组件。并通过“thumbnail”样式配合bootstrap的网格系统来实现。可以将产品列表页变得更好看

<div class="container">

<div class="row">

//一个图

<div class="col-xs-6 col-md-3">

<a href="#" class="thumbnail">

<img alt="100%x180" src="" display: block;">

</a>

<div class="caption">

<h3>Bootstrap框架系列教程</h3>

<p>优美的网站...</p>

<p>

<a href="##" class="btn btn-primary">开始学习</a>

<a href="##" class="btn btn-info">正在学习</a>

</p>

</div>

</div>

</div>

</div>

### 进度条

外容器使用“progress”样式，子容器使用“progress-bar”样式。其中progress用来设置进度条的容器样式，而progress-bar用于限制进度条的进度

progress-bar-success: 等 属性 设置进度条颜色

progress-striped： 在progress 后加 ：条纹进度条

active 要和 progress-striped 一起使用 ：动态进度条

//基本进度条

<div class="progress">

<div class="progress-bar progress-bar-success" style="width:40%">

</div>

</div>

//条纹进度条

<div class="progress progress-striped">

<div class="progress-bar progress-bar-success" style="width:40%"></div>

</div>

//动态条纹进度条

<div class="progress progress-striped active">

<div class="progress-bar progress-bar-success" style="width:40%"></div>

</div>

// 带进度的进度条

<div class="progress">

<div class="progress-bar progress-bar-success" role="progressbar" aria-valuenow="20" aria-valuemin="0" aria-valuemax="100" style="width:20%">20%</div>

</div>

### 面板

面板容器用 .panel

面板样式： panel-default panel-primary panel-success 等等

//基础面板

<div class="panel panel-default">

<div class="panel-body">我是一个基础面板，带有默认主题样式风格</div>

</div>

//有头有尾的面板

//panel-primary 等值控制 面板的颜色样式

<div class="panel panel-primary">

<div class="panel-heading">图解CSS3</div>

<div class="panel-body">新特性</div>

<div class="panel-footer">作者：大漠</div>

</div>

//面板的嵌套

<div class="panel panel-default">

<div class="panel-heading">图解CSS3</div>

<div class="panel-body">

<p>详细讲解了选择器、边框、背景、文本、颜色、盒模型、伸缩布局盒模型、多列布局、渐变、过渡、动画、媒体、响应Web设计、Web字体等主题下涵盖的所有CSS3新特性

</p>

<table class="table table-bordered">

<thead>

<tr>

<th>＃</th>

<th>发布时间</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>2014-07-10</td>

</tr>

</tbody>

</table>

</div>

<div class="panel-footer">作者：大漠</div>

</div>

### 媒体对象

图像居左；文字描述居右 或者相反

容器 一般是 div 也可以是别的

媒体对像的容器：常使用“media”类名表示，用来容纳媒体对象的所有内容

媒体对像的对象：常使用“media-object”表示，就是媒体对象中的对象，常常是图片

媒体对象的主体：常使用“media-body”表示，就是媒体对像中的主体内容，可以是任何元素，常常是图片侧边内容

媒体对象的标题：常使用“media-heading”表示，就是用来描述对象的一个标题，此部分可选

媒体嵌套 ，一个媒体的 media-body嵌套另一个 媒体

媒体列表，见下方 ：media-list ul的 class 属性值

<div class="media">

<a class="pull-left" href="#">

<img class="media-object" src="" alt="...">

</a>

<div class="media-body">

<h4 class="media-heading">我是大漠</h4>

<div>我是W3cplus站长大漠，我在写Bootstrap框中的媒体对象测试用例</div>

<div class="media">

<a class="pull-left" href="#">

<img class="media-object" src="" alt="...">

</a>

<div class="media-body">

<h4 class="media-heading">慕课网</h4>

<div class="media">

<a class="pull-left" href="#">

<img class="media-object" src="" alt="...">

</a>

<div class="media-body">

<h4 class="media-heading">W3cplus</h4>

</div>

</div>

</div>

</div>

</div>

</div>

//媒体列表

<ul class="media-list">

<li class="media">

<a class="pull-left" href="#">

<img class="media-object" src="http://tp2.sinaimg.cn/3306361973/50/22875318196/0" alt="...">

</a>

<div class="media-body">

<h4 class="media-heading">慕课网</h4>

<div>大漠写的《玩转Bootstrap》系列教程即将会在慕课网上发布</div>

</div>

</li>

<li class="media">

<a class="pull-left" href="#">

<img class="media-object" src="http://tp4.sinaimg.cn/1167075935/50/22838101204/1" alt="...">

</a>

<div class="media-body">

<h4 class="media-heading">W3cplus</h4>

<div>W3cplus站上还有很多教程....</div>

</div>

</li>

</ul>

### 过渡动画

默认情况之下，Bootstrap框架中以下组件使用了过渡动画效果：

模态弹出窗（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；

图片轮播（Carousel）的滑动效果。

Bootstrap框架中的模态弹出框，

分别运用了“modal”、“modal-dialog”和“modal-content”样式，

而弹出窗真正的内容都放置在“modal-content”中，其主要又包括三个部分：

弹出框头部，一般使用“modal-header”表示，主要包括标题和关闭按钮

弹出框主体，一般使用“modal-body”表示，弹出框的主要内容

弹出框脚部，一般使用“modal-footer”表示，主要放置操作按钮

实现原理分析：

在窗口 class 加上 fade 淡入淡出

show.bs.modal

在show方法调用时立即触发（尚未显示之前）；如果单击了一个元素，那么该元素将作为事件的relatedTarget属性

shown.bs.modal

该事件在模态弹出窗完全显示给用户之后（并且等CSS动画完成之后）触发；如果单击了一个元素，那么该元素将作为事件的relatedTarget事件

hide.bs.modal

在hide方法调用时（但还未关闭隐藏）立即触发

hidden.bs.modal

该事件在模态弹出窗完全隐藏之后（并且CSS动画漂完成之后）触发

#### 弹窗之js方式触发：

<button class="btn btn-primary" type="button" id="b1">点击我</button>

<div class="modal fade" id="mymodal">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal"><span aria-hidden="true">&times;</span><span class="sr-only">Close</span></button>

<h4 class="modal-title">模态弹出窗标题</h4>

</div>

<div class="modal-body">

<p>模态弹出窗主体内容</p>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">关闭</button>

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

</div>

</div><!-- /.modal-content -->

</div><!-- /.modal-dialog -->

</div><!-- /.modal -->

Js代码 toggle: 点击按钮（获其他触发方式）弹窗隐藏时，显示，显示时，隐藏

$(function(){

$("#b1").click(function(){

$("#mymodal").modal("toggle");

$("#mymodal").on(‘show.bs.modal’,function(event){

});

});

});

#### 弹窗之 data-target触发方式

data-toggle="modal" ：必要的属性

Data-target:必要的属性 指定弹窗（id class 都可以）

Data-whatever: 可选属性，可以通过它 在触发弹窗时可做些操作

Fade:该类 实现 淡入淡出效果

Modal-lg: 大弹窗 modal-sm: 小弹窗

<button data-toggle="modal" class="btn btn-primary"

type="button"data-whatever="helloworld" data-target="#mymodal">点击我</button>

<div class="modal fade" id="mymodal">

<div class="modal-dialog modal-lg">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal"><span aria-hidden="true">&times;</span><span class="sr-only">Close</span></button>

<h4 class="modal-title">模态弹出窗标题</h4>

</div>

<div class="modal-body">

<p>模态弹出窗主体内容</p>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">关闭</button>

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

</div>

</div><!-- /.modal-content -->

</div><!-- /.modal-dialog -->

</div><!-- /.modal -->

//js 代码

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

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

var button = $(e.relatedTarget)

var recipient = button.data('whatever')

alert(button.attr("data-whatever"))

})

//可以通过超链接 触发弹窗

<a class="btn btn-primary" href="#d1" data-toggle="modal">通过链接href属性触发</a>