[5-1\*下拉菜单（基本用法） 1](#_Toc409645255)

[5-2下拉菜单（原理分析） 2](#_Toc409645256)

[5-3下拉菜单（下拉分隔线）空的<li>，并且给这个<li>添加类名“divider” 4](#_Toc409645257)

[<li role="presentation" class="divider"></li> 4](#_Toc409645258)

[5-4下拉菜单（菜单标题）class="dropdown-header" 5](#_Toc409645259)

[<li role="presentation" class="dropdown-header">第一部分菜单头部</li>，颜色是灰的，head点不动。 5](#_Toc409645260)

[5-5下拉菜单（对齐方式）， 6](#_Toc409645261)

[默认左对齐，想要右对齐：<ul class="dropdown-menu pull-right" role="menu" aria-labelledby="dropdownMenu1">或者“dropdown-menu-right 6](#_Toc409645262)

[5-6\*下拉菜单（菜单项状态） 6](#_Toc409645263)

[默认的状态（不用设置）（:hover）（:focus）： 6](#_Toc409645264)

[除了上面两种状态，还有当前状态（.active）和禁用状态（.disabled）需要在对应的菜单项上添加对应的类名 6](#_Toc409645265)

# 5-1\*下拉菜单（基本用法）

统一调用压缩好的“bootstrap.min.js”文件：

<script src="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>

特别声明：因为Bootstrap的组件交互效果都是依赖于jQuery库写的插件，所以在使用bootstrap.min.js之前一定要先加载jquery.min.js才会生效果。

我们先来看官网上一个简单的示例：

<div class="**dropdown**">//“**dropdown**”的容器包裹了整个下拉菜单元素

<**button** class="btn btn-default **dropdown-toggle**" type="button" id="dropdownMenu1" **data-toggle="dropdown"**>//<button>**按钮做为父菜单，并且定义类名“**dropdown-toggle**”和自定义data-toggle="dropdown"**

下拉菜单

<span class="caret"></span> //最右边的小图标

</**button**>//下拉菜单项使用一个**ul**列表，并且定义一个类名为“**dropdown-menu**”，此示例为:

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

<**ul** class="**dropdown-menu**" role="menu" aria-labelledby="dropdownMenu1">//role标签的目的**WAI-ARIA无障碍网页**

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

…

<li role="presentation" class="divider"></li>

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

</**ul**>

</div>

**使用方法：**

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。我们来简单的看看：

1、使用一个名为“**dropdown**”的容器包裹了整个下拉菜单元素，示例中为:

<div class="dropdown"></div>

2、使用了一个**<button>**按钮做为父菜单，并且定义类名“**dropdown-toggle**”和自定义“**data-toggle**”属性，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**”，此示例为:

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

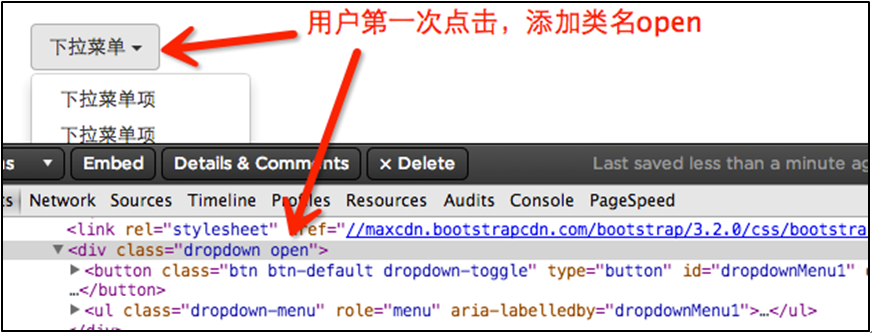
# 5-2下拉菜单（原理分析）

Bootstrap框架中的下拉菜单组件，其下拉菜单项默认是隐藏的。

默认情况：

[](http://img.mukewang.com/53e314020001537208700352.jpg)

用户第一次点击：

[](http://img.mukewang.com/53e314360001a81808720333.jpg)

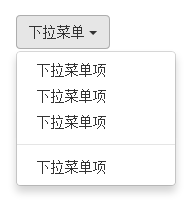
用户再次点击：

[](http://img.mukewang.com/53e31461000180e608710333.jpg)

# 5-3下拉菜单（下拉分隔线）空的<li>，并且给这个<li>添加类名“divider”

## <li role="presentation" class="divider"></li>

在Bootstrap框架中的下拉菜单还提供了下拉分隔线，假设下拉菜单有两个组，那么组与组之间可以通过添加一个空的<li>，并且给这个<li>添加类名“divider”来实现添加下拉分隔线的功能。



# 5-4下拉菜单（菜单标题）****class="dropdown-header"****

## <li role="presentation" **class="dropdown-header"**>第一部分菜单头部</li>，颜色是灰的，head点不动。



# 5-5下拉菜单（对齐方式），

### 默认左对齐，想要右对齐：<ul class="dropdown-menu **pull-right**" role="menu" aria-labelledby="dropdownMenu1">或者“dropdown-menu-right

# 5-6\*下拉菜单（菜单项状态）

### 默认的状态（不用设置）（:hover）（:focus）：

### 除了上面两种状态，还有当前状态（.active）和禁用状态（.disabled）需要在对应的菜单项上添加对应的类名

<div class="dropdown">

<button class="btn btn-default dropdown-toggle" type="button" id="dropdownMenu1" data-toggle="dropdown">

下拉菜单

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

</button>

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

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

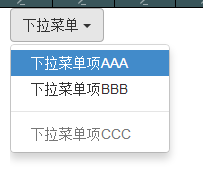
<li role="presentation"><a role="menuitem" tabindex="-1" href="#">下拉菜单项BBB</a></li>

<li role="presentation" class="divider"></li>

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

</ul>

</div>

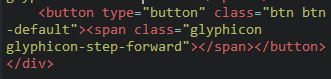
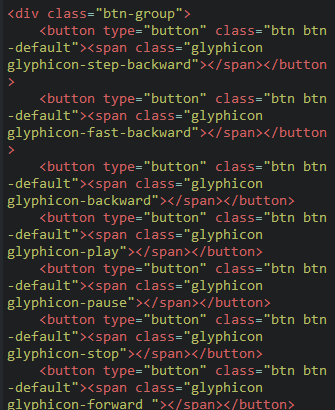


# 5-7按钮（小图标按钮组）btn容器bootstrap.js

**用方法：**

按钮组和下拉菜单组件一样，需要依赖于button.js插件才能正常运行。不过我们同样可以直接只调用bootstrap.js文件。因为这个文件已集成了button.js插件功能。

对于结构方面，非常的简单。使用一个名为“btn-group”的容器，把多个按钮放到这个容器中。如下所示：





# 5-8按钮（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”中，如下所示：

<div class="btn-toolbar">

<div class="btn-group">

…

</div>

<div class="btn-group">

…

</div>

<div class="btn-group">

…

</div>

<div class="btn-group">

…

</div>

</div>

## ****5-8-1按钮组大小设置**** <div class="btn-group btn-group-sm">

## btn-lg、btn-sm和btn-xs调整单个按钮大小

## .btn-group-lg:大按钮组.btn-group-sm:小按钮组.btn-group-xs:超小按钮组

只需要在“.btn-group”类名上追加对应的类名，就可以得到不同大小的按钮组。如下所示：

<div class="btn-toolbar">

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

…

</div>

<div class="btn-group">

…

</div>

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

…

</div>

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

…

</div>

</div>

运行效果如下所示：

[](http://img.mukewang.com/53e4632b0001bb2808230100.jpg)

# 5-9按钮（下拉菜单和普通的按钮组排列）类似导航

使用的时候，只需要把当初制作下拉菜单的“dropdown”的容器换成“btn-group”，并且和普通的按钮放在同一级。如下所示：

<div class="btn-group">

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

<button class="btnbtn-default" type="button">产品展示</button>

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

<button class="btnbtn-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="##">公司简介</a></li>

<li><a href="##">企业文化</a></li>

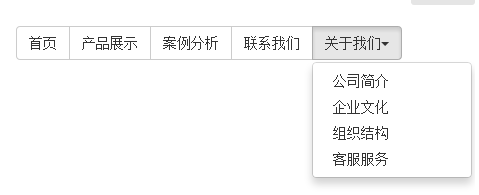
<li><a href="##">组织结构</a></li>

<li><a href="##">客服服务</a></li>

</ul>

</div>

</div>



# 5-10按钮（垂直分组）btn-group”类名换成“btn-group-vertical”

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。只需要把水平分组的“btn-group”类名换成“btn-group-vertical”即可。



# 5-11按钮（等分按钮）适用于移动端“****btn-group-justified****”

充满容器

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

# 5-12向上弹起的下拉菜单<div class="btn-group dropup">

有些菜单是需要向上弹出的，比如说你的菜单在页面最底部，而这个菜单正好有一个下拉菜单，为了让用户有更好的体验，不得不让下拉菜单向上弹出。在Bootstrap框架中专门为这种效果提代了一个类名“dropup”。使用方法正如前面所示，只需要在“btn-group”上添加这个类名（当然，如果是普通向上弹出下拉菜单，你只需要在“dropdown”类名基础上追加“**dropup**”类名即可）。

<div class="btn-group dropup">

<button class="btn btn-default dropdown-toggle" data-toggle="dropdown" type="button">按钮下拉菜单<span class="caret"></span></button>

<ul class="dropdown-menu">

<li><a href="##">按钮下拉菜单项</a></li>

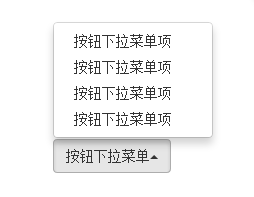
<li><a href="##">按钮下拉菜单项</a></li>

<li><a href="##">按钮下拉菜单项</a></li>

<li><a href="##">按钮下拉菜单项</a></li>

</ul>

</div>



# 5-13导航（基础样式）<ul class="nav nav-tabs">

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。

<body>

<ul class="nav nav-tabs">

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

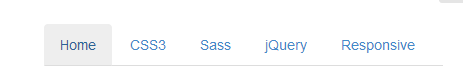
<li><a href="##">CSS3</a></li>

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

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

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

</ul>



# 5-14导航（标签形tab导航**class="active"**）

标签形导航，也称为选项卡导航。特别是在很多内容分块显示的时，使用这种选项卡来分组十分适合。

一般情况之下，选项卡教会有一个当前选中项。其实在Bootstrap框架也相应提供了。假设我们想让“Home”项为当前选中项，只需要在其标签上添加类名“class="active"”即可：

<ul class="nav nav-tabs">

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

…

</ul>

除了当前项之外，有的选项卡还带有禁用状态，实现这样的效果，只需要在标签项上添加“class="disabled"”即可：

<ul class="nav nav-tabs">

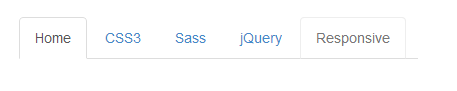
<li class="active"><a href="##">Home</a></li>

…

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

</ul>

运行效果如下：



# 5-15 (pills)导航<ul class="****nav nav-pills****">

当前项高亮显示，并带有圆角效果。其实现方法和“nav-tabs”类似，同样的结构，只需要把类名“nav-tabs”换成“**nav-pills**”即可：

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

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

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

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

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

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

</ul>

[](http://img.mukewang.com/53e86ee60001711e08160307.jpg)

# 5-16导航（垂直堆叠的导航）

## 5-16-1<ul class="nav nav-pills ****nav-stacked****">

## 5-16-2 导航下拉列表分割线<li class=”nav-divider”></li>

在实际运用当中，除了水平导航之外，还有垂直导航，就类似前面介绍的垂直排列按钮一样。制作垂直堆叠导航只需要在“nav-pills”的基础上添加一个“nav-stacked”类名即可：

<ul class="nav nav-pills **nav-stacked**">

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

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

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

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

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

</ul>

[](http://img.mukewang.com/53e871a2000102b707240444.jpg)

5-17自适应导航（使用）“**nav-justified**”

自适应导航指的是导航占据容器全部宽度，而且菜单项可以像表格的单元格一样自适应宽度。自适应导航和前面使用“btn-group-justified”制作的自适应按钮组是一样的。只不过在制作自适应导航时更换了另一个类名“**nav-justified**”。当然他需要和“nav-tabs”或者“nav-pills”配合在一起使用。

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

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

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

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

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

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

</ul>

<ul class="nav nav-pills **nav-justified**">

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

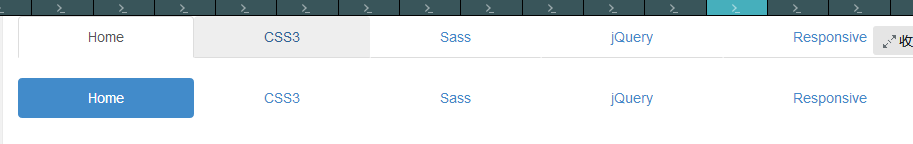
<li><a href="##">CSS3</a></li>

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

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

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

</ul>



# 5-18导航加下拉菜单（二级导航）

## 5-18-1使用类名“dropdown”，同时在该li中嵌套另一个列表ul，li

在Bootstrap框架中制作二级导航就更容易了。只需要将li当作父容器，使用类名“dropdown”，同时在li中嵌套另一个列表ul，使用前面介绍下拉菜单的方法就可以：

<ul class="nav nav-tabs"> //或者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="##">Sass</a></li>

<li class="nav-divider"></li>

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

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

</ul>

</li>

<li><a href="##">关于我们</a></li>

</ul>

# 5-19面包屑式导航

面包屑(Breadcrumb)一般用于导航，主要是起的作用是告诉用户现在所处页面的位置（当前位置）。

**使用方法：**

使用方式就很简单，为ol加入breadcrumb类：

<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>

