**BOOTSTRAP第二天笔记**

**全局 CSS 样式**

**1. 列表**

**1）无序列表**

<div class="container">

<div class="row">

<ul>

<li>苹果</li>

<li>香蕉</li>

<li>西瓜

<ul>

<li>大兴西瓜</li>

<li>东北西瓜</li>

</ul>

</li>

<li>芒果</li>

<li>樱桃</li>

</ul>

</div>

</div>

**2）有序列表**

<div class="container">

<div class="row">

<ol>

<li>苹果</li>

<li>香蕉</li>

<li>西瓜

<ol>

<li>大兴西瓜</li>

<li>东北西瓜</li>

</ol>

</li>

<li>芒果</li>

<li>樱桃</li>

</ol>

</div>

</div>

**3）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（只针对直接子元素）。

<div class="container">

<div class="row">

<ul class="list-unstyled">

<li>苹果</li>

<li>香蕉</li>

<li>西瓜

<ul>

<li>大兴西瓜</li>

<li>东北西瓜</li>

</ul>

</li>

<li>芒果</li>

<li>樱桃</li>

</ul>

</div>

</div>

**4）内联列表**

通过设置 display: inline-block; 并添加少量的内补（padding），将所有元素放置于同一行。

<div class="container">

<div class="row">

<ul class="list-inline">

<li>苹果</li>

<li>香蕉</li>

<li>西瓜

<ul>

<li>大兴西瓜</li>

<li>东北西瓜</li>

</ul>

</li>

<li>芒果</li>

<li>樱桃</li>

</ul>

</div>

</div>

**5）描述**

**a. 垂直排列的描述**

带有描述的短语列表。

<div class="container">

<div class="row">

<dl>

<dt>Description lists</dt>

<dd>A description list is perfect for defining terms.</dd>

<dt>Euismod</dt>

<dd>Vestibulum id ligula porta felis euismod semper eget lacinia odio sem nec elit.<br>

Donec id elit non mi porta gravida at eget metus.</dd>

<dt>Malesuada porta</dt>

<dd>Etiam porta sem malesuada magna mollis euismod.</dd>

</dl>

</div>

</div>

**b. 水平排列的描述**

.dl-horizontal 可以让

内的短语及其描述排在一行。

<div class="container">

<div class="row">

<dl class="dl-horizontal">

<dt>Description lists</dt>

<dd>A description list is perfect for defining terms.</dd>

<dt>Euismod</dt>

<dd>Vestibulum id ligula porta felis euismod semper eget lacinia odio sem nec elit.<br>

Donec id elit non mi porta gravida at eget metus.</dd>

<dt>Malesuada porta</dt>

<dd>Etiam porta sem malesuada magna mollis euismod.</dd>

</dl>

</div>

</div>

**2. 表格**

**1）基本表格**

为任意 <table> 标签添加 .table 类可以为其赋予基本的样式。

<div class="container">

<div class="row">

<table class="table">

<thead>

<tr>

<th>ID</th>

<th>名称</th>

<th>价格</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>书籍</td>

<td>12.50</td>

</tr>

<tr>

<td>2</td>

<td>电视</td>

<td>1000</td>

</tr>

<tr>

<td>3</td>

<td>笔记本</td>

<td>2000</td>

</tr>

<tr>

<td>3</td>

<td>笔记本</td>

<td>2000</td>

</tr>

</tbody>

</table>

</div>

</div>

**2）条纹状表格**

通过 .table-striped 类可以给 <tbody> 之内的每一行增加斑马条纹样式。

<div class="container">

<div class="row">

<table class="table table-striped">

<thead>

<tr>

<th>ID</th>

<th>名称</th>

<th>价格</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>书籍</td>

<td>12.50</td>

</tr>

<tr>

<td>2</td>

<td>电视</td>

<td>1000</td>

</tr>

<tr>

<td>3</td>

<td>笔记本</td>

<td>2000</td>

</tr>

<tr>

<td>3</td>

<td>笔记本</td>

<td>2000</td>

</tr>

</tbody>

</table>

</div>

</div>

条纹状表格是依赖 :nth-child CSS 选择器实现的，而这一功能不被 IE 8 支持。

**3）带边框表格**

添加 .table-bordered 类为表格和其中的每个单元格增加边框。

<div class="container">

<div class="row">

<table class="table table-bordered">

<thead>

<tr>

<th>ID</th>

<th>名称</th>

<th>价格</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>书籍</td>

<td>12.50</td>

</tr>

<tr>

<td>2</td>

<td>电视</td>

<td>1000</td>

</tr>

<tr>

<td>3</td>

<td>笔记本</td>

<td>2000</td>

</tr>

<tr>

<td>3</td>

<td>笔记本</td>

<td>2000</td>

</tr>

</tbody>

</table>

</div>

</div>

**4）鼠标悬停**

通过添加 .table-hover 类可以让 <tbody> 中的每一行对鼠标悬停状态作出响应。

<div class="container">

<div class="row">

<table class="table table-hover">

<thead>

<tr>

<th>ID</th>

<th>名称</th>

<th>价格</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>书籍</td>

<td>12.50</td>

</tr>

<tr>

<td>2</td>

<td>电视</td>

<td>1000</td>

</tr>

<tr>

<td>3</td>

<td>笔记本</td>

<td>2000</td>

</tr>

<tr>

<td>3</td>

<td>笔记本</td>

<td>2000</td>

</tr>

</tbody>

</table>

</div>

</div>

**5）紧缩表格**

通过添加 .table-condensed 类可以让表格更加紧凑，单元格中的 padding 均会减半。

<div class="container">

<div class="row">

<table class="table table-condensed">

<thead>

<tr>

<th>ID</th>

<th>名称</th>

<th>价格</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>书籍</td>

<td>12.50</td>

</tr>

<tr>

<td>2</td>

<td>电视</td>

<td>1000</td>

</tr>

<tr>

<td>3</td>

<td>笔记本</td>

<td>2000</td>

</tr>

<tr>

<td>3</td>

<td>笔记本</td>

<td>2000</td>

</tr>

</tbody>

</table>

</div>

</div>

**6）状态类**

通过这些状态类可以为行或单元格设置颜色。

| 类名 | 描述 | | active | 鼠标悬停在行或单元格上时所设置的颜色 | | success | 标识成功或积极的动作 | | info | 标识普通的提示信息或动作 | | warning | 标识警告或需要用户注意 | | danger | 标识危险或潜在的带来负面影响的动作 |

<div class="container">

<div class="row">

<table class="table">

<thead>

<tr>

<th>ID</th>

<th>名称</th>

<th>价格</th>

</tr>

</thead>

<tbody>

<tr class="active">

<td>1</td>

<td>书籍</td>

<td>12.50</td>

</tr>

<tr class="success">

<td>2</td>

<td>电视</td>

<td>1000</td>

</tr>

<tr>

<td class="info">3</td>

<td>笔记本</td>

<td class="warning">2000</td>

</tr>

<tr>

<td>3</td>

<td class="danger">笔记本</td>

<td>2000</td>

</tr>

</tbody>

</table>

</div>

</div>

**值得注意的是：**

这些状态类不仅可以作用于<tr>元素，还可以作用于<td>元素。

**7）响应式表格**

将任何 .table 元素包裹在 .table-responsive 元素内，即可创建响应式表格。其会在小屏幕设备上（小于768px）水平滚动。当屏幕大于 768px 宽度时，水平滚动条消失。

**3. 辅助样式**

**1）背景颜色**

和文本颜色一样，Bootstrap 允许添加不同的背景颜色表示不同含义，具体如下：

| **类名** | **描述** |
| --- | --- |
| bg-primary | 首选项，使用蓝色 |
| bg-success | 成功，使用浅绿色 |
| bg-info | 信息，使用浅蓝色 |
| bg-warning | 警告，使用黄色 |
| bg-danger | 危险，使用褐色 |

<div class="container">

<div class="row">

<p class="bg-muted">提示信息</p>

<p class="bg-primary">首选项</p>

<p class="bg-success">成功</p>

<p class="bg-info">信息</p>

<p class="bg-warning">警告</p>

<p class="bg-danger">危险</p>

</div>

</div>

**2）浮动**

**a. 浮动**

通过添加 .pull-left 或 .pull-right 类，可以将任意元素向左或向右浮动。

* CSS 代码

.block1 {

width : 300px;

height : 200px;

border : 1px solid black;

background : dodgerblue;

}

.block2 {

width : 500px;

height : 300px;

border : 1px solid black;

background : yellowgreen;

}

* HTML 代码

<div class="container">

<div>

<div class="block1 pull-left"></div>

<div class="block1 pull-right"></div>

</div>

<div class="block2"></div>

</div>

**b. 清除浮动**

通过为父元素添加 .clearfix 类可以很容易地清除浮动。

* CSS 代码

.block1 {

width : 300px;

height : 200px;

border : 1px solid black;

background : dodgerblue;

}

.block2 {

width : 500px;

height : 300px;

border : 1px solid black;

background : yellowgreen;

}

* HTML 代码

<div class="container">

<div class="clearfix">

<div class="block1 pull-left"></div>

<div class="block1 pull-right"></div>

</div>

<div class="block2"></div>

</div>

**3）居中**

为任意元素设置 .center-block 类让其中的内容居中。

<div class="center-block"></div>

**4）显示或隐藏**

.show 和 .hidden 类可以强制任意元素显示或隐藏(对于屏幕阅读器也能起效)，这些类只对块级元素起作用。

另外，.invisible 类可以被用来仅仅影响元素的可见性，也就是说，元素的 display 属性不被改变，并且这个元素仍然能够影响布局。

<div class="show"></div>

<div class="hidden"></div>

<div class="invisible"></div>

<div></div>

**4. 栅格系统**

**1）什么是栅格系统**

栅格系统用于通过一系列的行（row）与列（column）的组合来创建页面布局，内容就可以放入这些创建好的布局中。

Bootstrap 提供的栅格系统，主要特点如下：

* 行的宽度，可以是固定宽度，也可以是相对宽度（100%）。
* 一行中最多可以包含 12 个列。如果一行中包含的列大于 12，多余的列则被作为一个整体另起一行排列。
* Bootstrap 根据 4 种分辨率提供不同的栅格系统预定义样式。

**2）栅格系统工作原理**

Bootstrap 的栅格系统的工作原理如下：

* 栅格系统必须指定一个容器元素，该元素的 class 必须为 .container（固定宽度）或 .container-fluid（100%宽度）。
* 作为“行”的元素必须作为容器元素的直接子元素，并且 class 设置为 .row。
* 作为“列”的元素必须是“行”元素的直接子元素（**“行”元素的子元素不能直接包含内容，内容应该被包含在“列”元素中。**），并且一行最多允许创建 12 列。
* “列”元素的预定义样式，具体内容请参考 *“3）栅格系统的参数”* 内容。

**3）栅格系统的参数**

Bootstrap 根据 4 种不同的分辨率提供了不同的预定义样式（栅格类）。栅格类适用于与屏幕宽度大于或等于分界点大小的设备 ， 并且针对小屏幕设备覆盖栅格类。

|  | **超小屏幕 手机（<768px）** | **小屏幕 平板（>=768px）** | **中等屏幕 桌面显示器（>=992px）** | **大屏幕 大桌面显示器（>=1200px）** |
| --- | --- | --- | --- | --- |
| 栅格系统行为 | 总是水平排列 | 以折叠开始，大于断点成水平方式 | 以折叠开始，大于断点成水平方式 | 以折叠开始，大于断点成水平方式 |
| 容器宽度 | None（自动） | 750px | 970px | 1170px |
| class 前缀 | .col-xs- | .col-sm- | .col-md- | .col-lg- |
| 列数 | 12 | 12 | 12 | 12 |
| 列宽 | 自动 | 60px | 78px | 95px |
| 槽宽 | 30px（每列左右均为 15px） | 30px（每列左右均为 15px） | 30px（每列左右均为 15px） | 30px（每列左右均为 15px） |
| 可嵌套 | 是 | 是 | 是 | 是 |
| 偏移 | 是 | 是 | 是 | 是 |
| 排序 | 是 | 是 | 是 | 是 |

**4）栅格系统示例**

* CSS 代码

body {

margin-top : 100px;

}

[class\*=col-] {

border : 1px solid black;

padding-top: 15px;

padding-bottom: 15px;

}

* HTML 代码

<div class="container">

<h3>三等分列</h3>

<div class="row">

<div class="col-md-4">.col-md-4</div>

<div class="col-md-4">.col-md-4</div>

<div class="col-md-4">.col-md-4</div>

</div>

<h3>三不等分列</h3>

<div class="row">

<div class="col-md-3">.col-md-3</div>

<div class="col-md-6">.col-md-6</div>

<div class="col-md-3">.col-md-3</div>

</div>

<h3>手机、平板电脑和PC桌面不同布局</h3>

<div class="row">

<div class="col-md-3 col-sm-6 col-xs-12">col-md-3 col-sm-6 col-xs-12</div>

<div class="col-md-3 col-sm-6 col-xs-12">col-md-3 col-sm-6 col-xs-12</div>

<div class="col-md-3 col-sm-6 col-xs-12">col-md-3 col-sm-6 col-xs-12</div>

<div class="col-md-3 col-sm-6 col-xs-12">col-md-3 col-sm-6 col-xs-12</div>

</div>

</div>

**5）调整列的顺序**

**a. 列偏移**

使用 .col-md-offset-\* 类可以将列向右侧偏移。（\* 表示向右偏移的列数）

<div class="container">

<h3>列偏移</h3>

<div class="row">

<div class="col-md-4">.col-md-4</div>

<div class="col-md-4 col-md-offset-4">.col-md-4 .col-md-offset-4</div>

</div>

<div class="row">

<div class="col-md-3 col-md-offset-3">.col-md-3 .col-md-offset-3</div>

<div class="col-md-3 col-md-offset-3">.col-md-3 .col-md-offset-3</div>

</div>

<div class="row">

<div class="col-md-6 col-md-offset-3">.col-md-6 .col-md-offset-3</div>

</div>

</div>

**b. 嵌套列**

在 Bootstrap 的栅格系统允许进行嵌套。所谓**栅格系统嵌套**，就是指在 Bootstrap 栅格系统中的每列中允许包含一个栅格系统。

被嵌套的“行”中所包含的列的个数依旧不能超过 12 列。

<div class="container">

<h3>嵌套列</h3>

<div class="row">

<div class="col-sm-9">

Level 1: .col-sm-9

<div class="row">

<div class="col-xs-8 col-sm-6">Level 2: .col-xs-8 .col-sm-6</div>

<div class="col-xs-4 col-sm-6">Level 2: .col-xs-4 .col-sm-6</div>

</div>

</div>

</div>

</div>

**c. 列排序**

通过将栅格系统中的列向右移动（push）或向左移动（pull）来改变列的顺序。

* .col-md-push-\*表示向右移动
* .col-md-pull-\*表示向左移动

<div class="container">

<h3>列排序</h3>

<div class="row">

<div class="col-md-9 col-md-push-3">.col-md-9 .col-md-push-3</div>

<div class="col-md-3 col-md-pull-9">.col-md-3 .col-md-pull-9</div>

</div>

</div>

**6）响应式工具**

通过单独或联合使用以下列出的 class，可以针对不同屏幕尺寸隐藏或显示页面内容。

|  | **超小屏幕 手机（<768px）** | **小屏幕 平板（>=768px）** | **中等屏幕 桌面（>=992px）** | **大屏幕 大桌面（>=1200px）** |
| --- | --- | --- | --- | --- |
| .hidden-xs | 隐藏 | 可见 | 可见 | 可见 |
| .hidden-sm | 可见 | 隐藏 | 可见 | 可见 |
| .hidden-md | 可见 | 可见 | 隐藏 | 可见 |
| .hidden-lg | 可见 | 可见 | 可见 | 隐藏 |

<div class="container">

<h3>响应式工具</h3>

<div class="row">

<div class="col-md-3 col-sm-4 hidden-xs">.col-md-3</div>

<div class="col-md-6 col-sm-8 col-xs-12">.col-md-6</div>

<div class="col-md-3 hidden-sm hidden-xs">.col-md-3</div>

</div>

</div>

**7）响应式栅格系统示例**

<!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="utf-8">

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

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

<title>Bootstrap栅格系统案例</title>

<!-- Bootstrap -->

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

<!--[if lt IE 9]>

<script src="bootstrap/js/html5shiv.min.js"></script>

<script src="bootstrap/js/respond.min.js"></script>

<![endif]-->

</head>

<body>

<div class="jumbotron">

<div class="container">

<h1>Hello, world!</h1>

<p>This is a template for a simple marketing or informational website. It includes a large callout called a

jumbotron and three supporting pieces of content. Use it as a starting point to create something more

unique.</p>

</div>

</div>

<div class="container">

<!-- Example row of columns -->

<div class="row">

<div class="col-md-4">

<h2>Heading</h2>

<p>Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris

condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis

euismod. Donec sed odio dui. </p>

<p><a class="btn btn-default" href="#" role="button">View details &raquo;</a></p>

</div>

<div class="col-md-4">

<h2>Heading</h2>

<p>Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris

condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis

euismod. Donec sed odio dui. </p>

<p><a class="btn btn-default" href="#" role="button">View details &raquo;</a></p>

</div>

<div class="col-md-4">

<h2>Heading</h2>

<p>Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula

porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut

fermentum massa justo sit amet risus.</p>

<p><a class="btn btn-default" href="#" role="button">View details &raquo;</a></p>

</div>

</div>

<hr>

<footer>

<p>&copy; 2015 Company, Inc.</p>

</footer>

</div>

<script src="bootstrap/js/jquery-1.11.3.js"></script>

<script src="bootstrap/js/bootstrap.min.js"></script>

</body>

</html>

**5. 表单**

**1）基本表单**

Bootstrap 为单独的表单元素预定义了全局样式，表单内元素的具体使用如下：

* 设置了 .form-control 类的 <input>、<textarea> 和 <select> 元素都将被默认设置宽度属性为 width: 100%;。
* 将 label 元素和前面提到的控件包裹在 .form-group 中可以获得最好的排列。

<div class="container">

<div class="row">

<form>

<div class="form-group">

<label for="exampleInputEmail1">Email address</label>

<input type="email" class="form-control" id="exampleInputEmail1" placeholder="Email">

</div>

<div class="form-group">

<label for="exampleInputPassword1">Password</label>

<input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">

</div>

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

</form>

</div>

</div>

**2）内联表单**

为 <form> 元素添加 .form-inline 类可使其内容左对齐并且表现为 inline-block 级别的控件。

只适用于 viewport 至少在 768px 宽度时（视口宽度再小的话就会使表单折叠）。

<div class="container">

<div class="row">

<form class="form-inline">

<div class="form-group">

<label for="exampleInputName2">Name</label>

<input type="text" class="form-control" id="exampleInputName2" placeholder="Jane Doe">

</div>

<div class="form-group">

<label for="exampleInputEmail2">Email</label>

<input type="email" class="form-control" id="exampleInputEmail2" placeholder="jane.doe@example.com">

</div>

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

</form>

</div>

</div>

**需要注意的是：**

在内联表单，我们将这些元素的宽度设置为 width: auto;，因此，多个控件可以排列在同一行。

如果你没有为每个输入控件设置 label 标签，屏幕阅读器将无法正确识别。对于这些内联表单，你可以通过为 label 设置 .sr-only 类将其隐藏。

<div class="container">

<div class="row">

<form class="form-inline">

<div class="form-group">

<label class="sr-only" for="exampleInputEmail3">Email address</label>

<input type="email" class="form-control" id="exampleInputEmail3" placeholder="Email">

</div>

<div class="form-group">

<label class="sr-only" for="exampleInputPassword3">Password</label>

<input type="password" class="form-control" id="exampleInputPassword3" placeholder="Password">

</div>

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

</form>

</div>

</div>

**3）水平排列的表单**

通过为表单添加 .form-horizontal 类，并联合使用 Bootstrap 预置的栅格类，可以将 label 标签和控件组水平并排布局。

这样做将改变 .form-group 的行为，使其表现为栅格系统中的行（row），因此就无需再额外添加 .row 了。

<div class="container">

<div class="row">

<form class="form-horizontal">

<div class="form-group">

<label for="inputEmail3" class="col-sm-2 control-label">Email</label>

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

<input type="email" class="form-control" id="inputEmail3" placeholder="Email">

</div>

</div>

<div class="form-group">

<label for="inputPassword3" class="col-sm-2 control-label">Password</label>

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

<input type="password" class="form-control" id="inputPassword3" placeholder="Password">

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

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

</div>

</div>

</form>

</div>

</div>

**4）单选框和多选框**

**a. 默认外观**

* 单选框使用 class 为 .radio
* 多选框使用 class 为 .checkbox

单选框或多选框的样式要定义在 div 元素中（该元素作为容器元素，包含文本和单选或多选框元素）。

<div class="container">

<div class="row">

<form>

<div class="checkbox">

<label>

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

Option one is this and that&mdash;be sure to include why it's great

</label>

</div>

<div class="checkbox disabled">

<label>

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

Option two is disabled

</label>

</div>

<div class="radio">

<label>

<input type="radio" name="optionsRadios" id="optionsRadios1" value="option1" checked>

Option one is this and that&mdash;be sure to include why it's great

</label>

</div>

<div class="radio">

<label>

<input type="radio" name="optionsRadios" id="optionsRadios2" value="option2">

Option two can be something else and selecting it will deselect option one

</label>

</div>

<div class="radio disabled">

<label>

<input type="radio" name="optionsRadios" id="optionsRadios3" value="option3" disabled>

Option three is disabled

</label>

</div>

</form>

</div>

</div>

**b. 内联单选和多选框**

通过将 .checkbox-inline 或 .radio-inline 类应用到一系列的多选框（checkbox）或单选框（radio）控件上，可以使这些控件排列在一行。

<div class="container">

<div class="row">

<form>

<label class="checkbox-inline">

<input type="checkbox" id="inlineCheckbox1" value="option1"> 1

</label>

<label class="checkbox-inline">

<input type="checkbox" id="inlineCheckbox2" value="option2"> 2

</label>

<label class="checkbox-inline">

<input type="checkbox" id="inlineCheckbox3" value="option3"> 3

</label>

<label class="radio-inline">

<input type="radio" name="inlineRadioOptions" id="inlineRadio1" value="option1"> 1

</label>

<label class="radio-inline">

<input type="radio" name="inlineRadioOptions" id="inlineRadio2" value="option2"> 2

</label>

<label class="radio-inline">

<input type="radio" name="inlineRadioOptions" id="inlineRadio3" value="option3"> 3

</label>

</form>

</div>

</div>

**组件**

**1. 字体图标**

Bootstrap 使用的字体图标是来自 Glyphicions Halflings 提供的。**在使用的时候，需要遵守以下几点规则：**

* 图标使用的 class 不能与其他组件的 class 混合使用。换言之，图标的 class 必须被定义在 span 元素上。
* 图标使用的 class 具有一个基类和一个修饰类。基类（class）统一为 .glyphicon，修饰类（class）为 .glyphicon-\*-\*或.glyphicon-\*。
* 只对内容为空的元素起作用（图标使用的 class 只能应用在不包含任何文本内容或子元素的元素上）。

<div class="container">

<p class="text-danger">

<span class="glyphicon glyphicon-warning-sign"></span>

<span class="glyphicon glyphicon-alert"></span>

警告：您的浏览器版本太低了!

</p>

<button class="btn btn-success" type="button">

<span class="glyphicon glyphicon-apple"></span>

删除

</button>

<button class="btn btn-danger" type="button">

<span class="glyphicon glyphicon-arrow-left"></span>

</button>

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

<span class="glyphicon glyphicon-star"></span>

</button>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-play"></span>

</a>

</div>

**2. 下拉菜单**

**1）基本样式**

Bootstrap 提供的下拉菜单并不是 HTML 默认的 <select> 元素，而且利用一组 HTML 元素组合而成。

* <div>元素作为容器元素
* <button>或<a>元素作为下拉菜单的提示项。

作为下拉菜单的提示项，Bootstrap 默认显示的是按钮样式（也就是需要设置 class 为 btn btn-\*）。

按钮在被点击后，会提供不同的样式效果。如果想改变这样的样式，需要添加样式dropdown-toggle。

通过为下拉菜单的提示项设置属性 data-toggle=dropdown ，实现下拉菜单的动态显示。（此内容会在 **组件** 详细讲解）

* 无序列表作为下拉菜单的列表项。

**a. 向下弹出**

为作为下拉菜单的容器元素设置样式dropdown即可。

<div class="container">

<div class="dropdown">

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

Dropdown

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

</button>

<ul class="dropdown-menu">

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

<li><a href="#">Another action</a></li>

<li><a href="#">Something else here</a></li>

<li><a href="#">Separated link</a></li>

</ul>

</div>

</div>

**b. 向上弹出**

为作为下拉菜单的容器元素设置样式dropup即可。

<div class="container">

<div class="dropup">

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

Dropdown

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

</button>

<ul class="dropdown-menu">

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

<li><a href="#">Another action</a></li>

<li><a href="#">Something else here</a></li>

<li><a href="#">Separated link</a></li>

</ul>

</div>

</div>

**2）标题**

可以为作为下拉菜单的列表项中，添加 class 为 dropdown-header 的 <li> 元素作为标题，表示一组动作。

<div class="container">

<div class="dropdown">

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

Dropdown

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

</button>

<ul class="dropdown-menu">

<li class="dropdown-header">Dropdown header</li>

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

<li><a href="#">Another action</a></li>

<li><a href="#">Something else here</a></li>

<li class="dropdown-header">Dropdown header</li>

<li><a href="#">Separated link</a></li>

</ul>

</div>

</div>

**3）分割线**

可以为作为下来菜单的列表项中，添加 class 为 divider 的 <li> 元素作为分割线，表示将多个选项进行分组。

<div class="container">

<div class="dropdown">

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

Dropdown

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

</button>

<ul class="dropdown-menu">

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

<li><a href="#">Another action</a></li>

<li><a href="#">Something else here</a></li>

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

<li><a href="#">Separated link</a></li>

</ul>

</div>

</div>

**4）禁用项**

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

<div class="container">

<div class="dropdown">

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

Dropdown

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

</button>

<ul class="dropdown-menu">

<li><a href="#">Regular link</a></li>

<li class="disabled"><a href="#">Disabled link</a></li>

<li><a href="#">Another link</a></li>

</ul>

</div>

</div>

**3. 按钮组**

**1）基本按钮组**

通过按钮组容器把一组按钮放在同一组里，为按钮组容器设置 class 为 btn-group。

<div class="container">

<div class="btn-group">

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-ok-sign"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-remove-sign"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-pencil"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-check"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-question-sign"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-edit"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-search"></span>

</a>

</div>

</div>

**2）不同布局**

**a. 两端对齐**

让一组按钮拉长为相同的尺寸，填满父元素的宽度。为按钮组容器添加样式 btn-group-justified。

<div class="container">

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

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-ok-sign"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-remove-sign"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-pencil"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-check"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-question-sign"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-edit"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-search"></span>

</a>

</div>

</div>

**b. 垂直排列**

让一组按钮垂直堆叠排列显示而不是水平排列，为按钮组容器设置 class 为 btn-group-vertical。

<div class="container">

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

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-ok-sign"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-remove-sign"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-pencil"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-check"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-question-sign"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-edit"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-search"></span>

</a>

</div>

</div>

**3）尺寸**

只要给 .btn-group 加上 .btn-group-\* 类，就省去为按钮组中的每个按钮都赋予尺寸类。

Bootstrap 提供了以下几种尺寸：

| **class 名称** | **描述** |
| --- | --- |
| btn-group-lg | 大 |
| btn-group-sm | 小 |
| btn-group-xs | 很小 |

除上述尺寸外，Bootstrap 还提供一种默认尺寸（无需设置）。

<div class="container">

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

<a class="btn btn-info" href="#">Left</a>

<a class="btn btn-info" href="#">Middle</a>

<a class="btn btn-info" href="#">Right</a>

</div>

<div class="btn-group">

<a class="btn btn-info" href="#">Left</a>

<a class="btn btn-info" href="#">Middle</a>

<a class="btn btn-info" href="#">Right</a>

</div>

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

<a class="btn btn-info" href="#">Left</a>

<a class="btn btn-info" href="#">Middle</a>

<a class="btn btn-info" href="#">Right</a>

</div>

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

<a class="btn btn-info" href="#">Left</a>

<a class="btn btn-info" href="#">Middle</a>

<a class="btn btn-info" href="#">Right</a>

</div>

</div>

**4）按钮工具栏**

把一组 <div class="btn-group"> 组合包裹在一个 <div class="btn-toolbar"> 中就可以做成更复杂的组件。

<div class="container">

<div class="btn-toolbar">

<div class="btn-group">

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-ok-sign"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-remove-sign"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-pencil"></span>

</a>

</div>

<div class="btn-group">

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-check"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-question-sign"></span>

</a>

</div>

<div class="btn-group">

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-edit"></span>

</a>

<a class="btn btn-info" href="#">

<span class="glyphicon glyphicon-search"></span>

</a>

</div>

</div>

</div>

**5）嵌套下拉菜单**

想要把下拉菜单混合到一系列按钮中，只须把 .btn-group 放入另一个 .btn-group 中。

<div class="container">

<div class="btn-group">

<button type="button" class="btn btn-default">1</button>

<button type="button" class="btn btn-default">2</button>

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

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

Dropdown

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

</button>

<ul class="dropdown-menu">

<li><a href="#">Dropdown link</a></li>

<li><a href="#">Dropdown link</a></li>

</ul>

</div>

</div>

</div>

**4. 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。

**1）基本警告框**

Bootstrap 的警告框样式由 .alert 和 4 个修饰class组成。4 个修饰class表示不同的警告信息。

| **class 名称** | **描述** |
| --- | --- |
| alert-success | 成功 |
| alert-info | 信息 |
| alert-warning | 警告 |
| alert-danger | 危险 |

<div class="container">

<div class="alert alert-danger">

<span class="glyphicon glyphicon-alert"></span>

您的浏览器禁用了Cookie！本站的部分功能无法启用！

</div>

<div class="alert alert-info">

<span class="glyphicon glyphicon-alert"></span>

没有更多数据可供加载了！

</div>

<div class="alert alert-warning">

<span class="glyphicon glyphicon-alert"></span>

您的浏览器OUT啦！请使用更新版本的浏览器!

</div>

<div class="alert alert-success">

<span class="glyphicon glyphicon-alert"></span>

您已经登录了！无需再次登录！

</div>

</div>

**2）可关闭的警告框**

为警告框添加一个可选的 .alert-dismissible 类和一个关闭按钮。

<div class="container">

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

<!--.close元素必须是alert的第一个子元素-->

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

<span class="glyphicon glyphicon-alert"></span>

您已经登录了！无需再次登录！

</div>

</div>

**3）警告框中的链接**

用 .alert-link 工具类，可以为链接设置与当前警告框相符的颜色。

<div class="container">

<div class="alert alert-warning">

<span class="glyphicon glyphicon-alert"></span>

您的浏览器OUT啦！请使用更新版本的<a href="#" class="alert-link">浏览器</a>!

</div>

</div>

**本笔记由 *金云龙* 编写**

作者博客地址：[龙哥有话说](http://www.longestory.com)

如转载或发布，请注明作者名称