

Bootstrap基础知识

Bootstrap

引用

(<http://v3.bootcss.com/css/#buttons>)

```
<link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.1/css/bootstrap.min.css"/>
```

<https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/css/bootstrap.min.css>

```
<script src="http://cdn.bootcss.com/jquery/1.11.1/jquery.min.js"></script>
```

```
<script src="http://cdn.bootcss.com/bootstrap/3.3.0/js/bootstrap.min.js">
</script>
```

简介

jQuery是操作DOM的js库。Bootstrap是css库。Bootstrap是一个css框架。固定布局和响应式布局。一个手机PC通用框架。比较初级，代码嵌套太多，类名太多。是整套的前端解决方案。

插件之间的依赖关系

某些插件和 CSS 组件依赖于其它插件。如果你是单个引入每个插件的，请确保在文档中检查插件之间的依赖关系。注意，所有插件都依赖 jQuery（也就是说，jQuery必须在所有插件之前引入页面）。 [bower.json](#) 文件中列出了 Bootstrap 所支持的 jQuery 版本。

```
<div container-fluid>100%宽度</div>
```

```
<div container >适合固定宽度</div>
```

class=" text-center" （或在css里设置" text-align:center;" ）

```
<button class="btn btn-block">Like</button>
```

颜色

任何带有 class *.btn* 的元素都会继承圆角灰色按钮的默认外观。

class=" btn-primary/default/success/info/warning/danger/link"

class=" text-danger" （也可以通过修改）

背景颜色也是如此：（也可以通过<div>修改）

```
<p class="bg-primary">...</p>
```

```
<p class="bg-success">...</p>
```

```
<p class="bg-info">...</p>
```

```
<p class="bg-warning">...</p>
```

```
<p class="bg-danger">...</p>
```

class=" row"

class=" col-xs-*

(屏幕大于(≥ 992 px) , 使用col-md-* 而不是col-sm-*)

如果屏幕大于(≥ 768 px) , 小雨 ≤ 992 px , 使用col-sm-* 而不是col-md-*)

经验！默认把所在<div>分为12列，*为所占列数。

经验！每一个img/col/row/container 都可以作为调整的变量。

经验！文字参与布局时，连续打一串英文会被识别为一个单词！

列偏移：col-xs-offset-*

往右推：col-xs-push-*

往左拉：col-xs-pull-*

经验！同一row似乎不能设置其中的col的不同高度

表单

单独的表单控件会被自动赋予一些全局样式。所有设置了 .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。（其实是增加标签行间距）

class=" img-responsive" 响应

class=" form-control" width:100%;

```
<select class="form-control">
```

```
<option>1</option>
```

```
<option>2</option>
```

```
<option>3</option>
```

```
<option>4</option>
```

```
<option>5</option>
</select>
```

```
<select multiple class="form-control">
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
  <option>5</option>
</select>
```

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

class=" well" (不能少了.form-group)

<form>中可加入： role="form"这个是语义化属性，是给浏览器和搜索引擎看的

水平放置：

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

如果需要在表单中将一行纯文本和 label 元素放置于同一行，为 <p> 元素添加 .form-control-static 类即可。

添加 .has-warning、.has-error 或 .has-success, 任何包含在此元素之内的 .control-label、.form-control 和 .help-block 元素都将接受这些校验状态的样式。

在框框右尽头加小图标（图标类不能和其它组件直接联合使用。它们不能在同一个元素上与其他类共同存在。应该创建一个嵌套的 标签，并将图标类应用到这个 标签上。）

```
<div class="form-group has-success has-feedback">  
  <label class="control-label" for="inputSuccess2">Input with success</label>  
  <input type="text" class="form-control" id="inputSuccess2">  
  <span class="glyphicon glyphicon-ok form-control-feedback"></span>  
</div>
```

通过 .input-lg/.input-sm 类似的类可以为控件设置高度

通过添加 .form-group-lg 或 .form-group-sm 类，为 .form-horizontal 包裹的 label 元素和表单控件快速设置尺寸。

用栅格系统中的列（column）包裹输入框或其任何父元素，都可很容易的为其设置宽度。（比如:col-xs-5,直接加在上一层<div>中）

辅助文本:

```
<span class="help-block">XXX</span>
```

内联元素

```
<del></del>;<s></s>
```

```
<mark></mark>
```

```
<ins></ins>;<u></u>
```

```
<small></small>;<big></big>
```

```
<b></b>
```

.text-justify:两端对齐

```
<p class="text-lowercase">全部小写</p>
```

```
<p class="text-uppercase">全部大写</p>
```

```
<p class="text-capitalize">单词首字母大写</p>
```

<footer> </footer>

<blockquote class="blockquote-reverse"> ...</blockquote> 右靠齐的引言

<ul class="list-unstyled"> ... 无序列表

<ul class="list-inline"> ...

<dl class="dl-horizontal"> 水平排列的描述

<dt>...</dt>

<dd>...</dd>

</dl>

(自动截断通过 text-overflow 属性，水平排列的描述列表将会截断左侧太长的短语。在较窄的视口 (viewport) 内，列表将变为默认堆叠排列的布局方式。)

<kbd> </kbd>

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。

添加 .table-bordered 类为表格和其中的每个单元格增加边框。

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。

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

通过这些状态类可以为行或单元格设置颜色。(加在table上面)

颜色

Class	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
---------	-------------------

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。

图片形状

```

```

```

```

```

```

```
<div class="center-block">...</div>  给此块状元素居中 ( 文本居中可用text-align-center; )
```

可见性

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

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

下拉菜单

需要先加载插件 (可上官网寻找地址)

```
<script src="http://cdn.bootcss.com/jquery/1.11.1/jquery.min.js"></script>
```

```
<script src="http://cdn.bootcss.com/bootstrap/3.3.0/js/bootstrap.min.js"></script>
```

```
<div class="dropdown/btn-group">
```

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

```
    Dropdown
```

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

```
</button>
```

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

```
  <li role="presentation"><a role="menuitem" tabindex="-1"
href="#">Action</a></li>
```

```
  <li role="presentation"><a role="menuitem" tabindex="-1"
href="#">Another action</a></li>
```

```
  <li role="presentation"><a role="menuitem" tabindex="-1"
href="#">Something else here</a></li>
```

```
  <li role="presentation"><a role="menuitem" tabindex="-1"
```

href="#">Separated link

</div>

tabindex 属性规定元素的 **tab** 键控制次序 (当 **tab** 键用于导航时) 。1~32767

**** 这个是按钮上的小箭头

右对齐

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

...

标题

在任何下拉菜单中均可通过添加标题来标明一组动作。

<li role="presentation" class="dropdown-header">Dropdown header

分割线

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

禁用

<li role="presentation" class="disabled">Disabled link

按钮组

<div class="btn-group">

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

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

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

</div>

按钮工具栏

把一组 <div class="btn-group"> 组合进一个 <div class="btn-toolbar"> 中就可以做成更复杂的组件。

```
<div class="btn-toolbar" role="toolbar">
  <div class="btn-group">...</div>
  <div class="btn-group">...</div>
  <div class="btn-group">...</div>
</div>
```

尺寸

```
<div class="btn-group btn-group-lg">...</div>
```

嵌套

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。
(原来的.dropdown此时可以省略)

垂直排列

让一组按钮垂直堆叠排列显示而不是水平排列。**分列式按钮下拉菜单不支持这种方式。**

```
<div class="btn-group-vertical">
  ...
</div>
```

水平调整

```
<div class="btn-group btn-group-justified">
  <div class="btn-group">
    <button type="button" class="btn btn-default">Left</button>
  </div>
  <div class="btn-group">
    <button type="button" class="btn btn-default">Middle</button>
  </div>
  <div class="btn-group">
    <button type="button" class="btn btn-default">Right</button>
  </div>
</div>
```

分裂式按钮下拉菜单

```
<div class="btn-group">
  <button type="button" class="btn btn-danger">Action</button>
```



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

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

```
<span class="sr-only">Toggle Dropdown</span>
```

```
</button>
```

```
<ul class="dropdown-menu" role="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>
```

全称是 **screen reader only**,意为:(仅供)屏幕阅读器, **.sr-only**是 除了屏幕阅读器外,其他设备上隐藏该元素,这个是为了帮助残障人士阅读的。

尺寸

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

```
Large button <span class="caret"></span>
```

```
</button>
```

向上弹出式菜单

给父元素添加 **.dropup** 类就能使触发的下拉菜单朝上方打开。

输入框组

通过在文本输入框 **<input>** 前面、后面或是两边加上文字或按钮,可以实现对表单控件的扩展。为 **.input-group** 赋予 **.input-group-addon** 类,可以给 **.form-control** 的前面或后面添加额外的元素。

只支持文本输入框 **<input>**

这里请避免使用 **<select>** 元素,因为 WebKit 浏览器不能完全绘制它的样式。

避免使用 `<textarea>` 元素，由于它们的 `rows` 属性在某些情况下不被支持。

输入框组中的工具提示和弹出框需要特别的设置

为 `.input-group` 中所包含的元素应用工具提示 (`tooltip`) 或 `popover` (弹出框) 时，必须设置 `container: 'body'` 参数，为的是避免意外的副作用 (例如，工具提示或弹出框被激活后，可能会让当前元素变得更宽或/和变得失去其圆角)。

不要和其他组件混用

不要将表单组或栅格列 (`column`) 类直接和输入框组混合使用。而是将输入框组嵌套到表单组或栅格相关元素的内部。

在输入框的任意一侧添加额外元素或按钮。你还可以在输入框的两侧同时添加额外元素。

```
<div class="input-group">  
  <span class="input-group-addon">@</span>  
  <input type="text" class="form-control" placeholder="Username">  
</div>
```

```
<div class="input-group">  
  <input type="text" class="form-control">  
  <span class="input-group-addon">.00</span>  
</div>
```

```
<div class="input-group">    ( 如果没有这个类，添加的东西就会变成一条长的 )  
  <span class="input-group-addon">$</span>  
  <input type="text" class="form-control">  
  <span class="input-group-addon">.00</span>  
</div>
```

尺寸

为 `.input-group` 添加相应的尺寸类，其内部包含的元素将自动调整自身的尺寸。不需要为输入框组中的每个元素重复地添加控制尺寸的类。

作为额外元素的多选框和单选框

可以将多选框或单选框作为额外元素添加到输入框组中。

```

<div class="input-group">
  <span class="input-group-addon">
    <input type="checkbox">
  </span>
  <input type="text" class="form-control">
</div>

```

作为额外元素的按钮

为输入框组添加按钮需要额外添加一层嵌套，不是 .input-group-addon，而是添加 .input-group-btn 来包裹按钮元素。由于不同浏览器的默认样式无法被统一的重新赋值，所以才需要这样做。

```

<div class="input-group">
  <span class="input-group-btn">
    <button class="btn btn-default" type="button">Go!</button>
  </span>
  <input type="text" class="form-control">
</div>

```

作为额外元素的按钮式下拉菜单

```

<div class="input-group">

```

<div class="input-group-btn"> 这个类起到的作用类似于dropdown/btn-group,使下拉

```

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

```

```

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

```

```

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

```

```
</div>
```

作为额外元素的分裂式按钮下拉菜单

```
<div class="input-group">  
  <div class="input-group-btn">  
    <!-- Button and dropdown menu -->  
  </div>  
  <input type="text" class="form-control">  
</div>
```

```
<div class="input-group">  
  <input type="text" class="form-control">  
  <div class="input-group-btn">  
    <!-- Button and dropdown menu -->  
  </div>  
</div>
```

导航

Bootstrap 中的导航组件都依赖同一个 .nav 类，状态类也是共用的。改变修饰类可以改变样式。

标签页

注意 .nav-tabs 类依赖 .nav 基类。

```
<ul class="nav nav-tabs" role="tablist">  
  <li role="presentation" class="active"><a href="#" data-toggle="tab">Home</a></li>  
  <li role="presentation"><a href="#" data-toggle="tab">Profile</a></li>  
  <li role="presentation"><a href="#" data-toggle="tab">Messages</a></li>  
</ul>
```

胶囊式标签页

HTML 标记相同，但使用 .nav-pills 类：

胶囊是标签页也是可以垂直方向堆叠排列的。只需添加 .nav-stacked 类。

```
<ul class="nav nav-pills nav-stacked" role="tablist">
```

```
...
</ul>
```

两端对齐的标签页

在大于 768px 的屏幕上，通过 `.nav-justified` 类可以很容易的让标签页或胶囊式标签呈现出同等宽度。在小屏幕上，导航链接呈现堆叠样式。

禁用的链接

对任何导航组件（标签页、胶囊是标签页），都可以添加 `.disabled` 类，从而实现**链接为灰色且没有鼠标悬停效果**。

链接功能不受到影响

这个类只改变 `<a>` 的外观，不改变功能。可以自己写 JavaScript 禁用这里的链接。

添加下拉菜单

用一点点额外 HTML 代码并加入[下拉菜单插件的 JavaScript 插件](#)即可。

```
<ul class="nav nav-tabs" role="tablist">
...
<li role="presentation" class="dropdown">
  <a class="dropdown-toggle" data-toggle="dropdown" href="#">
    Dropdown <span class="caret"></span>
  </a>
  <ul class="dropdown-menu" role="menu">
    ...
  </ul>
</li>
...
</ul>
```

导航条

修改视口的阈值，从而影响导航条的排列模式

当浏览器视口（viewport）的宽度小于 `@grid-float-breakpoint` 值时，导航条内部的元素变为折叠排列，也就是变现为移动设备展现模式；当浏览器视口（viewport）的宽度大于 `@grid-float-breakpoint` 值时，导航条内部的元素变为水平排列，也就是变现为非移动设

备展现模式。通过调整源码中的这个值，就可以控制导航条何时堆叠排列，何时水平排列。默认值是 768px（小屏幕 -- 或者说是平板 -- 的最小值，或者说是平板）。

```
<nav class="navbar navbar-default" role="navigation">
  <div class="container-fluid">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">Brand</a>
    </div>

    <!-- Collect the nav links, forms, and other content for toggling -->
    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Link</a></li>
        <li><a href="#">Link</a></li>
        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-
toggle="dropdown">Dropdown <span class="caret"></span></a>
          <ul class="dropdown-menu" role="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>
            <li class="divider"></li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

```

        <li> <a href="#">One more separated link</a> </li>
    </ul>
</li>
</ul>
<form class="navbar-form navbar-left" role="search">
    <div class="form-group">
        <input type="text" class="form-control" placeholder="Search">
    </div>
    <button type="submit" class="btn btn-default">Submit</button>
</form>
<ul class="nav navbar-nav navbar-right">
    <li> <a href="#">Link</a> </li>
    <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-
toggle="dropdown">Dropdown <span class="caret"></span> </a>
        <ul class="dropdown-menu" role="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>
    </li>
</ul>
</div> <!-- /.navbar-collapse -->
</div> <!-- /.container-fluid -->
</nav>

```

按钮

对于不包含在 `<form>` 中的 `<button>` 元素，加上 `.navbar-btn` 后，可以让它在导航条里垂直居中。

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

Context-specific usage

Like the standard [button classes](#), `.navbar-btn` can be used on `<a>` and `<input>` elements. However, neither `.navbar-btn` nor the standard button classes should be used on `<a>` elements within `.navbar-nav`.

文本

把文本包裹在 `.navbar-text` 中时，为了有正确的行距和颜色，通常使用 `<p>` 标签。

```
<p class="navbar-text">Signed in as Mark Otto</p>
```

非导航的链接

或许你希望在标准的导航组件之外添加标准链接，那么，使用 `.navbar-link` 类可以让链接有正确的默认颜色和反色设置。

```
<p class="navbar-text navbar-right">Signed in as <a href="#" class="navbar-link">Mark Otto</a></p>
```

组件排列

通过添加 `.navbar-left` 和 `.navbar-right` 工具类让导航链接、表单、按钮或文本对齐。两个类都会通过 CSS 设置特定方向的浮动样式。例如，要对齐导航链接，就要把它们放在个分开的、应用了工具类的 `` 标签里。

这些类是 `.pull-left` 和 `.pull-right` 的 `mixin` 版本，但是他们被限定在了媒体查询（`media query`）中，这样可以更容易的在各种尺寸的屏幕上处理导航条组件。

向右侧对齐多个组件Right aligning multiple components

导航条目前不支持多个 `.navbar-right` 类。为了让内容之间有合适的空隙，我们为最后一个 `.navbar-right` 元素使用负边距（`margin`）。如果有多个元素使用这个类，它们的边距（`margin`）将不能按照你的预期正常展现。

我们将在 v4 版本中重写这个组件时重新审视这个功能。

固定在顶部

添加 `.navbar-fixed-top` 类可以让导航条固定在顶部，还可包含一个 `.container` 或 `.container-fluid` 容器，从而让导航条居中，并在两侧添加内部（`padding`）。

```
<nav class="navbar navbar-default navbar-fixed-top" role="navigation">
  <div class="container">
    ...
```



```
</div>
```

```
</nav>
```

需要为 **body** 元素设置内部 (padding)

p>这个固定的导航条会遮住页面上的其它内容，除非你给 `<body>` 元素底部设置了 padding。用你自己的值，或用下面给出的代码都可以。提示：导航条的默认高度是 50px。

```
body { padding-top: 70px; }
```

Make sure to include this after the core Bootstrap CSS.

固定在底部

添加 `.navbar-fixed-bottom` 类可以让导航条固定在底部，并且还可以包含一个 `.container` 或 `.container-fluid` 容器，从而让导航条居中，并在两侧添加内部 (padding)。

需要为 **body** 元素设置内部 (padding)

这个固定的导航条会遮住页面上的其它内容，除非你给 `<body>` 元素底部设置了 padding。用你自己的值，或用下面给出的代码都可以。提示：导航条的默认高度是 50px。

静止在顶部

通过添加 `.navbar-static-top` 类即可创建一个与页面等宽度的导航条，它会随着页面向下滚动而消失。还可以包含一个 `.container` 或 `.container-fluid` 容器，用于将导航条居中对齐并在两侧添加内部 (padding)。

反色的导航条

通过添加 `.navbar-inverse` 类可以改变导航条的外观

可以用css改色: `.navbar a {}` 以及 `.navbar {}`

面包屑导航

在一个带有层次的导航结构中标明当前页面的位置。

各路径间的分隔符已经自动通过 CSS 的 `:before` 和 `content` 属性添加了。

```
<ol class="breadcrumb" >
```

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

```
<li> <a href="#" >Library</a> </li>
```

```
<li class="active" >Data</li>
```

```
</ol>
```

默认分页

受 Rdio 的启发，我们提供了这个简单的分页组件，用在应用或搜索结果中超级棒。组件中的每个部分都很大，有点事容易点击、易缩放、点击区域大。

```
<nav>

  <ul class="pagination">

    <li><a href="#">&laquo;</a></li>

    <li><a href="#">1</a></li>

    <li><a href="#">2</a></li>

    <li><a href="#">3</a></li>

    <li><a href="#">4</a></li>

    <li><a href="#">5</a></li>

    <li><a href="#">&raquo;</a></li>

  </ul>

</nav>
```

禁用和激活状态

链接在不同情况下可以定制。你可以给不能点击的链接添加 .disabled 类、给当前页添加 .active 类。

```
<nav>

  <ul class="pagination">

    <li class="disabled"><a href="#">&laquo;</a></li>

    <li class="active"><a href="#">1 <span class="sr-only">(current)</span>
</a></li>

    ...

  </ul>

</nav>
```

你还可以将 active 或 disabled 状态应用于 标签，即替换 <a> 标签，这样就可以让其保持需要的样式并移除点击功能。

```
<nav>

  <ul class="pagination">

    <li class="disabled"><span>&laquo;</span></li>
```

```
<li class="active"><span>1 <span class="sr-only">(current)</span></span>
</li>

...

</ul>

</nav>
```

尺寸

想要更小或更大的分页？.pagination-lg 或 .pagination-sm 类提供了额外可供选择的尺寸。

```
<nav><ul class="pagination pagination-lg">...</ul></nav>

<nav><ul class="pagination">...</ul></nav>

<nav><ul class="pagination pagination-sm">...</ul></nav>
```

翻页

用简单的标记和样式，就能做个上一页和下一页的简单翻页。用在像博客和杂志这样的简单站点上棒极了。

默认实例

在默认的翻页中，链接居中对齐。

```
<nav>

  <ul class="pager">

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

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

  </ul>

</nav>
```

对齐链接

你还可以把链接向两端对齐：

```
<nav>

  <ul class="pager">

    <li class="previous"><a href="#">&larr; Older</a></li>

    <li class="next"><a href="#">Newer &rarr;</a></li>

  </ul>

</nav>
```

标签

```
<h3>Example heading <span class="label label-default">New</span> </h3>
```

用下面的任何一个类即可改变标签的外观。

```
<span class="label label-default">Default</span>
```

```
<span class="label label-primary">Primary</span>
```

```
<span class="label label-success">Success</span>
```

```
<span class="label label-info">Info</span>
```

```
<span class="label label-warning">Warning</span>
```

```
<span class="label label-danger">Danger</span>
```

如果标签数量很多怎么办？

如果你有大量的设置为 inline 属性的标签全部放在一个较窄的容器元素内，在页面上展示这些标签就会出现問題，每个标签就会有自己的一个 inline-block 元素（就像图标一样）。解决的办法是为每个标签都设置为 display: inline-block; 属性。关于这个问题以及实例，请参考 [#13219](#)。

徽章

给链接、导航等元素嵌套 `` 元素，可以很醒目的展示新的或未读的信息条目。

```
<a href="#">Inbox <span class="badge">42</span> </a>
```

```
<button class="btn btn-primary" type="button">
```

```
  Messages <span class="badge">4</span>
```

```
</button>
```

巨幕

这是一个轻量、灵活的组件，它能延伸至整个浏览器视口来展示网站上的关键内容。

```
<div class="jumbotron">
```

```
  <h1>Hello, world!</h1>
```

```
  <p>...</p>
```

```
  <p><a class="btn btn-primary btn-lg" href="#" role="button">Learn more</a></p>
```

```
</div>
```

如果需要让巨幕组件的宽度与浏览器宽度一致并且没有圆角，请把此组件放在所有 .container 元素的外面，并在组件内部添加一个 .container 元素。

Copy

```
<div class="jumbotron">
  <div class="container">
    ...
  </div>
</div>
```

页头

页头组件能够为 h1 标签增加适当的空间，并且与页面的其他部分形成一定的分隔。它支持 h1 标签内内嵌 small 元素的默认效果，还支持大部分其他组件（需要增加一些额外的样式）。

```
<div class="page-header">
  <h1>Example page header <small>Subtext for header</small> </h1>
</div>
```

缩略图

通过缩略图组件扩展 Bootstrap 的 [栅格系统](#)，可以很容易地展示栅格样式的图像、视频、文本等内容。

默认样式的实例

Bootstrap 缩略图的默认设计仅需最少的标签就能展示带链接的图片。

```
<div class="row">
  <div class="col-xs-6 col-md-3">
    <a href="#" class="thumbnail">
      
    </a>
  </div>
  ...
</div>
```

data-*（也就是是data-src,data-info等等前面带data的），是存放数据的地方，仅仅是存放数据的地方，不对对页面有任何的影响，上面说的什么延迟加载都是屁，只有和js配合起来这个data-*这个功能才有用，而且有大用，可以用js的 `element.getAttribute("data-*")` 方法取到！

这个属性应用广泛，就楼主说的data-src一般用作懒加载和延期加载和图像加载错误重载等，data-src只是个语义化命名而已，用data-x都可以实现上述功能

```
<div class="row">
  <div class="col-sm-6 col-md-4">
    <div class="thumbnail">
      
      <div class="caption">
        <h3>Thumbnail label</h3>
        <p>...</p>
        <p><a href="#" class="btn btn-primary" role="button">Button</a> <a href="#" class="btn btn-default" role="button">Button</a></p>
      </div>
    </div>
  </div>
</div>
```

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。

实例

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。

没有默认类

警告框没有默认类，只有基类和修饰类。默认的灰色警告框并没有多少意义。所以您要使用一种有意义的警告类。目前提供了成功、消息、警告或危险。

```
<div class="alert alert-success" role="alert">...</div>
<div class="alert alert-info" role="alert">...</div>
<div class="alert alert-warning" role="alert">...</div>
<div class="alert alert-danger" role="alert">...</div>
```

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。

依赖警告框 JavaScript 插件

如果需要为警告框组件提供关闭功能，请使用 [jQuery 警告框插件](#)。

```
<div class="alert alert-warning alert-dismissible" role="alert">  
  <button type="button" class="close" data-dismiss="alert"><span aria-  
hidden="true">&times;</span><span class="sr-only">Close</span></button>  
  <strong>Warning!</strong> Better check yourself, you're not looking too  
good.  
</div>
```

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="alert alert-success" role="alert">  
  <a href="#" class="alert-link">...</a>  
</div>  
  
<div class="alert alert-info" role="alert">  
  <a href="#" class="alert-link">...</a>  
</div>  
  
<div class="alert alert-warning" role="alert">  
  <a href="#" class="alert-link">...</a>  
</div>  
  
<div class="alert alert-danger" role="alert">  
  <a href="#" class="alert-link">...</a>  
</div>
```

进度条

通过这些简单、灵活的进度条，为当前工作流程或动作提供实时反馈。

跨浏览器兼容性

进度条组件使用了 CSS3 的 `transition` 和 `animation` 属性来完成一些特效。这些特性在 Internet Explorer 9 或以下版本中、Firefox 的老版本中没有被支持。Opera 12 不支持 `animation` 属性。

```
<div class="progress">
```

```
<div class="progress-bar" role="progressbar" aria-valuenow="60" aria-  
valuemin="0" aria-valuemax="100" style="width: 60%;">  
    <span class="sr-only">60% Complete</span>  
</div>  
</div>
```

带有提示标签的进度条

从进度条组件中移除 .sr-only 类，从而让当前进度显示出来。考虑到当前进度比例很低的情况，建议为进度标签添加 min-width 属性，以保证进度标签完全显示出来。

进度数值较低的情况

在当前进度数值较低的情况下，进度条组件显示的是带有一个数字的进度值，也包括 0% 的情况，为了清晰显示出来，建议为进度标签添加一个 min-width: 20px; 属性。

根据情境变化效果

进度条组件使用与按钮和警告框相同的类，根据不同情境展现相应的效果。

```
<div class="progress">  
    <div class="progress-bar progress-bar-success" role="progressbar" aria-  
valuenow="40" aria-valuemin="0" aria-valuemax="100" style="width: 40%">  
        <span class="sr-only">40% Complete (success)</span>  
    </div>  
</div>
```

条纹效果

通过渐变可以为进度条创建条纹效果，IE8 不支持。

```
<div class="progress">  
    <div class="progress-bar progress-bar-success progress-bar-striped"  
role="progressbar" aria-valuenow="40" aria-valuemin="0" aria-valuemax="100"  
style="width: 40%">  
        <span class="sr-only">40% Complete (success)</span>  
    </div>  
</div>
```

动画效果

为 .progress-bar-striped 添加 .active 类，使其呈现出由右向左运动的动画效果。IE9 及更低版本的浏览器不支持。


```
<div class="progress">
```

```
  <div class="progress-bar progress-bar-striped active" role="progressbar" aria-  
valuenow="45" aria-valuemin="0" aria-valuemax="100" style="width: 45%">
```

```
    <span class="sr-only">45% Complete</span>
```

```
  </div>
```

```
</div>
```

堆叠效果

把多个进度条放入同一个 .progress 中，使它们呈现堆叠的效果。

```
<div class="progress">
```

```
  <div class="progress-bar progress-bar-success" style="width: 35%">
```

```
    <span class="sr-only">35% Complete (success)</span>
```

```
  </div>
```

```
  <div class="progress-bar progress-bar-warning progress-bar-striped"  
style="width: 20%">
```

```
    <span class="sr-only">20% Complete (warning)</span>
```

```
  </div>
```

```
  <div class="progress-bar progress-bar-danger" style="width: 10%">
```

```
    <span class="sr-only">10% Complete (danger)</span>
```

```
  </div>
```

```
</div>
```

媒体对象

这是一个抽象的样式，用以构建不同类型的组件，这些组件都具有在文本内容的左或右侧对齐的图片（就像博客评论或 Twitter 消息等）。

默认样式

默认样式的媒体对象组件允许在一个内容块的左边或右边展示一个多媒体内容（图像、视频、音频）。

```
<div class="media">
```

```
  <a class="media-left" href="#">
```

```
    
```


<div class="media-body" >

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

...

</div>

</div>

Media alignment

The images or other media can be aligned top, middle, or bottom. The default is top aligned.

<div class="media" >

<div class="media-body" >

<h4 class="media-heading">Middle aligned media</h4>

...

</div>

</div>

媒体对象列表

用一点点额外的标记，就能在列表内使用媒体对象组件（对评论或文章列表很有用）。

<ul class="media-list" >

<li class="media" >

<div class="media-body" >

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

...

</div>


```
</ul>
```

列表组

基本实例

最简单的列表组仅仅是一个带有多个列表条目的无序列表，另外还需要设置适当的类。我们提供了一些预定义的样式，你可以根据自身的需求通过 CSS 自己定制。

```
<ul class="list-group">
  <li class="list-group-item">Cras justo odio </li>
  <li class="list-group-item">Dapibus ac facilisis in</li>
  <li class="list-group-item">Morbi leo risus</li>
  <li class="list-group-item">Porta ac consectetur ac</li>
  <li class="list-group-item">Vestibulum at eros</li>
</ul>
```

徽章

给列表组加入徽章组件，它会自动被放在右边。

```
<ul class="list-group">
  <li class="list-group-item">
    <span class="badge">14</span>
    Cras justo odio
  </li>
</ul>
```

链接

用 `<a>` 标签代替 `` 标签可以组成一个全部是链接的列表组（还要注意的是，我们需要将 `` 标签替换为 `<div>` 标签）。没必要给列表组中的每个元素都加一个父元素。

```
<div class="list-group">
  <a href="#" class="list-group-item active">
    Cras justo odio
  </a>
  <a href="#" class="list-group-item">Dapibus ac facilisis in</a>
  <a href="#" class="list-group-item">Morbi leo risus</a>
```

```
<a href="#" class="list-group-item">Porta ac consectetur ac</a>
<a href="#" class="list-group-item">Vestibulum at eros</a>
</div>
```

被禁用的条目

为 .list-group-item 添加 .disabled 类可以让单个条目显示为灰色，表现出被禁用的效果。

情境类

为列表中的条目添加情境类，默认样式或链接列表都可以。还可以为列表中的条目设置 .active 状态。

```
<ul class="list-group">
  <li class="list-group-item list-group-item-success">Dapibus ac facilisis in</li>
  <li class="list-group-item list-group-item-info">Cras sit amet nibh libero</li>
  <li class="list-group-item list-group-item-warning">Porta ac consectetur
ac</li>
  <li class="list-group-item list-group-item-danger">Vestibulum at eros</li>
</ul>
```

定制内容

列表组中的每个元素都可以是任何 HTML 内容，甚至是像下面的带链接的列表组。

```
<div class="list-group">
  <a href="#" class="list-group-item active">
    <h4 class="list-group-item-heading">List group item heading</h4>
    <p class="list-group-item-text">...</p>
  </a>
</div>
```

面板

基本实例

默认的 .panel 组件所做的只是设置基本的边框（border）和内补（padding）来包含内容。

```
<div class="panel panel-default">
  <div class="panel-body">
    Basic panel example
```

```
</div>
```

```
</div>
```

带标题的面版

通过 `.panel-heading` 可以很简单地为面板加入一个标题容器。你也可以通过添加设置了 `.panel-title` 类的 `<h1>-<h6>` 标签，添加一个预定义样式的标题。

为了给链接设置合适的颜色，务必将链接放到带有 `.panel-title` 类的标题标签内。

```
<div class="panel panel-default">
```

```
<div class="panel-heading">Panel heading without title</div>
```

```
<div class="panel-body">
```

```
Panel content
```

```
</div>
```

```
</div>
```

```
<div class="panel panel-default">
```

```
<div class="panel-heading">
```

```
<h3 class="panel-title">Panel title</h3>
```

```
</div>
```

```
<div class="panel-body">
```

```
Panel content
```

```
</div>
```

```
</div>
```

带脚注的面版

把按钮或次要的文本放入 `.panel-footer` 容器内。注意面版的脚注**不会**从情境效果中继承颜色，因为他们并不是主要内容。

```
<div class="panel panel-default">
```

```
<div class="panel-body">
```

```
Panel content
```

```
</div>
```

```
<div class="panel-footer">Panel footer</div>
```

```
</div>
```

情境效果

```
<div class="panel panel-primary">...</div>
```

```
<div class="panel panel-success">...</div>
```

```
<div class="panel panel-info">...</div>
```

```
<div class="panel panel-warning">...</div>
```

```
<div class="panel panel-danger">...</div>
```

带表格的面版

为面板中不需要边框的表格添加 `.table` 类，是整个面板看上去更像是一个整体设计。如果是带有 `.panel-body` 的面板，我们为表格的上方添加一个边框，看上去有分隔效果。

```
<div class="panel panel-default">
```

```
  <!-- Default panel contents -->
```

```
  <div class="panel-heading">Panel heading</div>
```

```
  <div class="panel-body">
```

```
    <p>...</p>
```

```
  </div>
```

```
  <!-- Table -->
```

```
  <table class="table">
```

```
    ...
```

```
  </table>
```

```
</div>
```

带列表组的面版

可以简单地在任何面版中加入具有最大宽度的[列表组](#)。

具有响应式特性的嵌入内容

根据被嵌入内容的外部容器的宽度，自动创建一个固定的比例，从而让浏览器自动确定视频或 `slideshow` 的尺寸，能够在各种设备上缩放。

这些规则被直接应用在 `<iframe>`、`<embed>`、`<video>` 和 `<object>` 元素上。如果你希望让最终样式与其他属性相匹配，还可以明确地使用一个派生出来的 `.embed-responsive-item` 类。

超级提示：不需要为 `<iframe>` 元素设置 `frameborder="0"` 属性，因为我们已经替你这样

做了！

```
<!-- 16:9 aspect ratio -->
```

```
<div class="embed-responsive embed-responsive-16by9">  
  <iframe class="embed-responsive-item" src="..."></iframe>  
</div>
```

```
<!-- 4:3 aspect ratio -->
```

```
<div class="embed-responsive embed-responsive-4by3">  
  <iframe class="embed-responsive-item" src="..."></iframe>  
</div>
```

Well

默认效果

把 Well 用在元素上，能有嵌入（inset）的简单效果。

```
<div class="well">...</div>
```

可选类

通过这两种可选修饰类，可以控制此组件的内补（padding）和圆角的设置。

```
<div class="well well-lg">...</div>
```

模态框

<http://www.runoob.com/bootstrap/bootstrap-v2-modal-plugin.html>

不支持模态框重叠

千万不要在一个模态框上重叠另一个模态框。要想同时支持多个模态框，需要自己写额外的代码来实现。

模态框的 HTML 代码放置的位置

务必将模态框的 HTML 代码放在文档的最高层级内（也就是说，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/或功能。

Overflow and scrolling

<body> 元素在 iOS 和 Android 上对 overflow: hidden 的支持很有限。结果就是，在这两种设备上的浏览器中，当你滚动屏幕超过模态框的顶部或底部时，<body> 中的内容将开始随着滚动。

标签页

```
<!-- Nav tabs -->
```

```
<ul class="nav nav-tabs" role="tablist">
```

```
  <li role="presentation" class="active"><a href="#home" role="tab" data-  
toggle="tab">Home</a></li>
```

```
  <li role="presentation"><a href="#profile" role="tab" data-  
toggle="tab">Profile</a></li>
```

```
  <li role="presentation"><a href="#messages" role="tab" data-  
toggle="tab">Messages</a></li>
```

```
  <li role="presentation"><a href="#settings" role="tab" data-  
toggle="tab">Settings</a></li>
```

```
</ul>
```

```
<!-- Tab panes -->
```

```
<div class="tab-content">
```

```
  <div role="tabpanel" class="tab-pane active" id="home">...</div>
```

```
  <div role="tabpanel" class="tab-pane" id="profile">...</div>
```

```
  <div role="tabpanel" class="tab-pane" id="messages">...</div>
```

```
  <div role="tabpanel" class="tab-pane" id="settings">...</div>
```

```
</div>
```

Fade effect

To make tabs fade in, add `.fade` to each `.tab-pane`. The first tab pane must also have `.in` to properly fade in initial content.

```
<div class="tab-content">
```

```
  <div role="tabpanel" class="tab-pane fade in active" id="home">...</div>
```

```
  <div role="tabpanel" class="tab-pane fade" id="profile">...</div>
```

```
  <div role="tabpanel" class="tab-pane fade" id="messages">...</div>
```

```
  <div role="tabpanel" class="tab-pane fade" id="settings">...</div>
```

```
</div>
```

Methods

`$.tab`

Activates a tab element and content container. Tab should have either a data-target or an href targeting a container node in the DOM.

Tooltips

```
<button type="button" class="btn btn-default" data-toggle="tooltip" data-placement="left" title="Tooltip on left">Tooltip on left</button>
```

```
<button type="button" class="btn btn-default" data-toggle="tooltip" data-placement="top" title="Tooltip on top">Tooltip on top</button>
```

```
<button type="button" class="btn btn-default" data-toggle="tooltip" data-placement="bottom" title="Tooltip on bottom">Tooltip on bottom</button>
```

```
<button type="button" class="btn btn-default" data-toggle="tooltip" data-placement="right" title="Tooltip on right">Tooltip on right</button>
```

这是一个 `默认的 Tooltip`.

这是一个 `左侧的 Tooltip`.

这是一个 `顶部的 Tooltip`.

这是一个 `底部的 Tooltip`.

这是一个 `右侧的 Tooltip`

```
<script>
```

```
    $(function () { $('[data-toggle="tooltip"]').tooltip(); });
```

```
</script>
```

```
<!-- HTML to write -->
```

```
<a href="#" data-toggle="tooltip" title="Some tooltip text!">Hover over me</a>
```

```
<!-- Generated markup by the plugin -->
```

```
<div class="tooltip top" role="tooltip">
```

```
<div class="tooltip-arrow"></div>
<div class="tooltip-inner">
  Some tooltip text!
</div>
</div>
```

Popovers弹出框

```
$(function () { $('[data-toggle="popover"]').popover(); });
```

Popovers in button groups and input groups require special setting

When using popovers on elements within a `.btn-group` or an `.input-group`, you'll have to specify the option `container: 'body'` (documented below) to avoid unwanted side effects (such as the element growing wider and/or losing its rounded corners when the popover is triggered).

Don't try to show popovers on hidden elements

Invoking `$(...).popover('show')` when the target element is `display: none;` will cause the popover to be incorrectly positioned.

Popovers on disabled elements require wrapper elements

To add a popover to a disabled or `.disabled` element, put the element inside of a `<div>` and apply the popover to that `<div>` instead.

Multiple-line links

Sometimes you want to add a popover to a hyperlink that wraps multiple lines. The default behavior of the popover plugin is to center it horizontally and vertically. Add `white-space: nowrap;` to your anchors to avoid this.

```
<button type="button" class="btn btn-lg btn-danger" data-toggle="popover"
data-placement="right" title="Popover title" data-content="And here's some
amazing content. It's very engaging. Right?">Click to toggle popover</button>
```

Dismiss on next click

Use the focus trigger to dismiss popovers on the next click that the user makes.

Specific markup required for dismiss-on-next-click

For proper cross-browser and cross-platform behavior, you must use the `<a>` tag, *not* the `<button>` tag, and you also must include a `tabindex` attribute.

```
<a href="#" tabindex="0" class="btn btn-lg btn-danger" role="button" data-  
toggle="popover" data-trigger="focus" title="Dismissible popover" data-  
content="And here's some amazing content. It's very engaging.  
Right?">Dismissible popover</a>
```

警告框

Add dismiss functionality to all alert messages with this plugin.

When using a .close button, it must be the first child of the .alert-dismissible and no text content may come before it in the markup.

```
<div class="alert alert-warning">  
  
<a href="#" class="close" data-dismiss="alert">      &times;  
  
</a>  
  
    <strong>警告！</strong>您的网络连接有问题。  
</div>
```

.close与× ; 是按钮 data-dismiss是取消功能 alert是粗条 alert-warning是黄色

按钮-其他

加载状态

```
<button id="fat-btn" class="btn btn-primary" data-loading-text="Loading..."  
    type="button"> 加载状态  
</button>  
  
<script>  
    $(function() {  
        $(".btn").click(function(){  
            $(this).button('loading').delay(1000).queue(function() {  
                // $(this).button('reset');  
                // $(this).dequeue();  
            });  
        });  
    });
```

```
});
```

```
</script>
```

单个切换

如需激活单个按钮的切换（即改变按钮的正常状态为按压状态，反之亦然），只需向 `button` 元素添加 **`data-toggle="button"`** 作为其属性即可，如下面实例所示：

复选框 (Checkbox)

您可以创建复选框组，并通过向 **`btn-group`** 添加 `data` 属性 **`data-toggle="buttons"`** 来添加复选框组的切换。

单选按钮 (Radio)

类似地，您可以创建单选按钮组，并通过向 **`btn-group`** 添加 `data` 属性 **`data-toggle="buttons"`** 来添加单选按钮组的切换。

折叠

<http://www.runoob.com/bootstrap/bootstrap-collapse-plugin.html>

Bootstrap 轮播 (Carousel) 插件

<http://www.runoob.com/bootstrap/bootstrap-carousel-plugin.html>

Bootstrap 轮播 (Carousel) 插件是一种灵活的响应式的向站点添加滑块的方式。除此之外，内容也是足够灵活的，可以是图像、内嵌框架、视频或者其他您想要放置的任何类型的内容。

如果您想要单独引用该插件的功能，那么您需要引用 **`carousel.js`**。或者，正如 [Bootstrap 插件概览](#) 一章中所提到，您可以引用 *`bootstrap.js`* 或压缩版的 *`bootstrap.min.js`*。

Bootstrap 附加导航 (Affix) 插件

<http://www.runoob.com/bootstrap/bootstrap-affix-plugin.html>

附加导航 (Affix) 插件允许某个 `<div>` 固定在页面的某个位置。您也可以在打开或关闭使用该插件之间进行切换。一个常见的例子是社交图标。它们将在某个位置开始，但当页面点击某个标记，该 `<div>` 会锁定在某个位置，不会随着页面其他部分一起滚动。

如果您想要单独引用该插件的功能，那么您需要引用 **`affix.js`**。或者，正如 [Bootstrap 插件概览](#) 一章中所提到，您可以引用 *`bootstrap.js`* 或压缩版的 *`bootstrap.min.js`*。



Leanote
Upgrade Account