

Bootstrap4 教程



Bootstrap 是全球最受欢迎的前端组件库，用于开发响应式布局、移动设备优先的 WEB 项目。

Bootstrap4 目前是 Bootstrap 的最新版本，是一套用于 HTML、CSS 和 JS 开发的开源工具集。利用我们提供的 Sass 变量和大量 mixin、响应式栅格系统、可扩展的预制组件、基于 jQuery 的强大的插件系统，能够快速为你的想法开发出原型或者构建整个 app。

谁适合阅读本教程？

只要您具备 HTML 和 CSS 的基础知识，您就可以阅读本教程，进而开发出自己的网站。在您学习完本教程后，您即可达到使用 Bootstrap 开发 Web 项目的中等水平。

阅读本教程前，您需要了解的知识：

在您开始阅读本教程之前，您必须具备 HTML、CSS 和 JavaScript 的基础知识。如果您还不了解这些概念，那么建议您先阅读我们的这些教程：

[HTML 教程](#)

[CSS 教程](#)

[JavaScript 教程](#)

Bootstrap4 实例

Bootstrap4 实例

```
<div class="jumbotron text-center">
<h1>我的第一个 Bootstrap 页面</h1>
<p>重置浏览器大小查看效果!</p>
</div>
<div class="container">
<div class="row">
<div class="col-sm-4">
<h3>第一列</h3>
<p>菜鸟教程</p>
<p>学的不仅是技术，更是梦想!!! </p>
</div>
<div class="col-sm-4">
<h3>第二列</h3>
<p>菜鸟教程..</p>
<p>学的不仅是技术，更是梦想!!! </p>
</div>
<div class="col-sm-4">
<h3>第三列</h3>
<p>菜鸟教程..</p>
<p>学的不仅是技术，更是梦想!!! </p>
</div>
</div>
</div>
```

[尝试一下 »](#)

Bootstrap4 与 Bootstrap3

Bootstrap4 是 Bootstrap 的最新版本，与 Bootstrap3 相比拥有了更多的具体的类以及把一些有关的部分变成了相关的组件。同时 Bootstrap.min.css 的体积减少了40%以上。

Bootstrap4 放弃了对 IE8 以及 iOS 6 的支持，现在仅仅支持 IE9 以上 以及 iOS 7 以上版本的浏览器。如果对于其中需要用到以前的浏览器，那么请

使用 [Bootstrap3](#)。

Bootstrap4 安装使用 ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号：闽ICP备15012807号-1

☐

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

☐ Bootstrap4 教程

Bootstrap4 网格系统 ☐

Bootstrap4 安装使用

我们可以通过以下两种方式来安装 Bootstrap4:

使用 Bootstrap 4 CDN。

从官网 [getbootstrap.com](#) 下载 Bootstrap 4。

Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v4.1.0** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see [CSS files comparison](#))
- Compiled and minified JavaScript plugins

This doesn't include documentation, source files, or any optional JavaScript dependencies (jQuery and Popper.js).

Download

Bootstrap 4 CDN

国内推荐使用 BootCDN 上的库:

Bootstrap4 CDN

```
<!-- 新 Bootstrap4 核心 CSS 文件 -->
<link rel="stylesheet" href="https://cdn.bootcss.com/bootstrap/4.1.0/css/bootstrap.min.css">
<!-- jQuery文件。务必在bootstrap.min.js 之前引入 -->
<script src="https://cdn.bootcss.com/jquery/3.2.1/jquery.min.js"></script>
<!-- popper.min.js 用于弹窗、提示、下拉菜单 -->
<script src="https://cdn.bootcss.com/popper.js/1.12.5/umd/popper.min.js"></script>
<!-- 最新的 Bootstrap4 核心 JavaScript 文件 -->
<script src="https://cdn.bootcss.com/bootstrap/4.1.0/js/bootstrap.min.js"></script>
```

此外, 你还可以使用以下的 CDN 服务:

国内推荐使用 : <https://www.staticfile.org/>

国际推荐使用：<https://cdnjs.com/>

下载 Bootstrap 4

你可以去官网 <https://getbootstrap.com/> 下载 Bootstrap4 资源库。

注：此外你还可以通过包的管理工具 *npm*、*gem*、*composer* 等来安装：

```
npm install bootstrap@4.0.0-beta.2

gem 'bootstrap', '~> 4.0.0.beta2'

composer require twbs/bootstrap:4.0.0-beta.2
```

创建第一个 Bootstrap 4 页面

1、添加 HTML5 doctype

Bootstrap 要求使用 HTML5 文件类型，所以需要添加 HTML5 doctype 声明。

HTML5 doctype 在文档头部声明，并设置对应编码：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
</html>
```

移动设备优先

为了让 Bootstrap 开发的网站对移动设备友好，确保适当的绘制和触屏缩放，需要在网页的 head 之中添加 viewport meta 标签，如下所示：

```
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
```

width=device-width 表示宽度是设备屏幕的宽度。

initial-scale=1 表示初始的缩放比例。

shrink-to-fit=no 自动适应手机屏幕的宽度。

容器类

Bootstrap 4 需要一个容器元素来包裹网站的内容。

我们可以使用以下两个容器类：

`.container` 类用于固定宽度并支持响应式布局的容器。

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

`.container`

`.container-fluid`

两个 Bootstrap 4 页面

Bootstrap4 .container 实例

```
<div class="container">
<h1>我的第一个 Bootstrap 页面</h1>
<p>这是一些文本。</p>
</div>
```

尝试一下 »

以下实例展示了占据全部视口（viewport）的容器。

Bootstrap4 .container-fluid 实例

```
<div class="container-fluid">
<h1>我的第一个 Bootstrap 页面</h1>
<p>使用了 .container-fluid, 100% 宽度, 占据全部视口（viewport）的容器。</p>
</div>
```

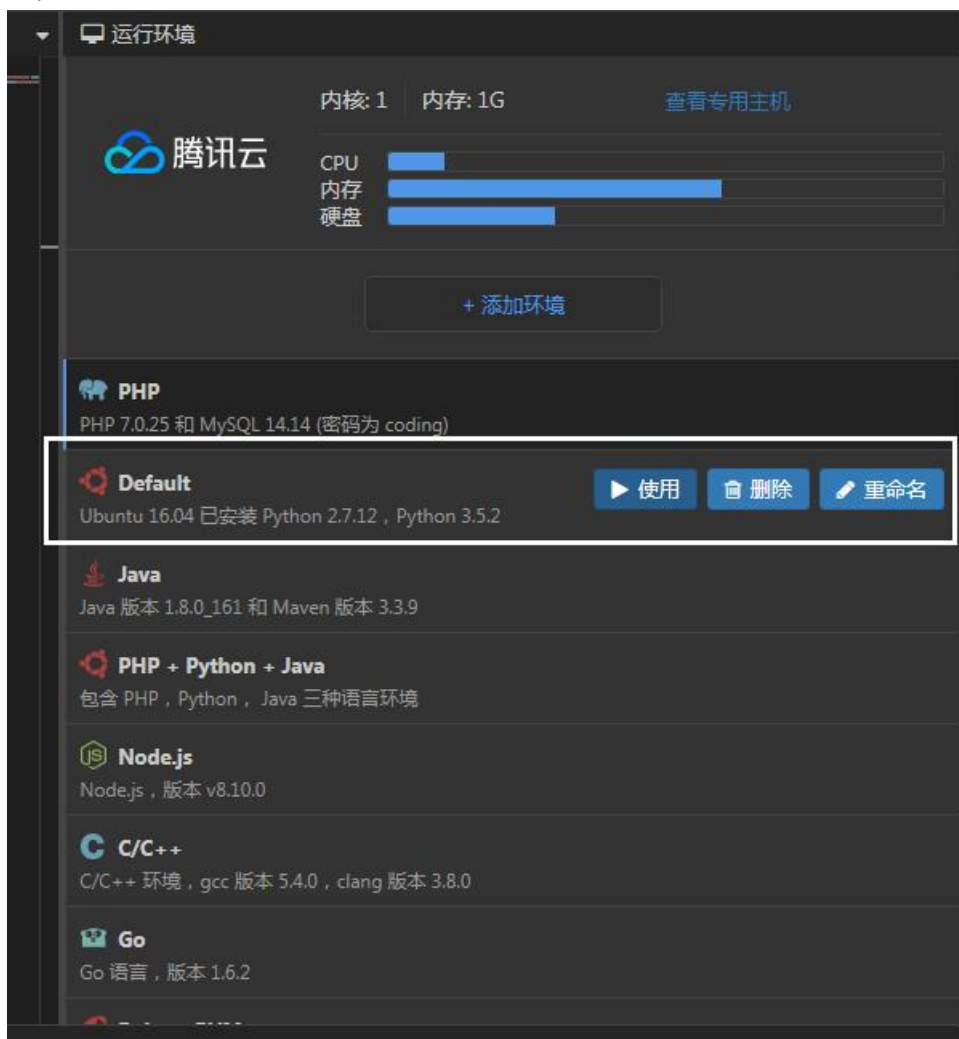
尝试一下 »

在 Cloud Studio 中使用 Bootstrap

下面我们介绍如何在 Cloud Studio 中安装、使用 Bootstrap4:

step1: 访问 [Cloud Studio](#), 注册/登录账户。

step2: 在右侧的运行环境菜单选择: "ubuntu"



step3: 在左侧代码目录中新建 html 目录, 编写你的 HTML 代码, 例如 index.html

step4: 在 index.html 文件的中粘贴如下代码、来引入 Bootstrap4 (也可以从官网 <https://getbootstrap.com/> 下载到 Bootstrap4 资源库):

```
<link rel="stylesheet" <!-- 新 Bootstrap4 核心 CSS 文件 -->

<link rel="stylesheet" href="https://cdn.bootcss.com/bootstrap/4.1.0/css/bootstrap.min.css">
```

```

<!-- jQuery文件。务必在bootstrap.min.js 之前引入 -->

<script src="https://cdn.bootcss.com/jquery/3.2.1/jquery.min.js"></script>

<!-- popper.min.js 用于弹窗、提示、下拉菜单 -->

<script src="https://cdn.bootcss.com/popper.js/1.12.5/umd/popper.min.js"></script>

<!-- 最新的 Bootstrap4 核心 JavaScript 文件 -->

<script src="https://cdn.bootcss.com/bootstrap/4.1.0/js/bootstrap.min.js"></script>

```

step5: 在终端中输入命令 `sudo vim /etc/nginx/site-enabled/default`。将配置文件红框部分修改为如下图所示，然后输入命令：`sudo nginx restart` 重启 `nginx` 服务（`nginx` 安装完成并启动后默认会监听 80 端口。我们需要将 `nginx` 的站点目录以及监听的端口号改为我们需要的）

```

server {
    listen 8080 default_server;
    listen [::]:8080 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /home/coding/workspace/html;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;

    server_name _;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }
}

```

step6: 点击最右侧的【访问链接】选项卡，在访问链接面板中填写端口号为：8080（和刚才 `nginx` 配置文件中的端口号一致），点击创建链接，即可点击生成的链接访问我们刚刚编写的代码，查看 `Bootstrap4` 组件的效果。



试试Bootstrap的模态框吧！

在CloudStudio中运行HTML代码

我是运行在CodingStudio中的Bootstrap组件！

关闭

[Bootstrap4 教程](#)

[Bootstrap4 网格系统](#)

[点我分享笔记](#)

[反馈/建议](#)

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号：闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[Bootstrap4 安装使用](#)

[Bootstrap4 文字排版](#)

Bootstrap4 网格系统

Bootstrap 提供了一套响应式、移动设备优先的流式网格系统，随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多 12 列。

我们也可以根据自己的需要，定义列数：

| | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | | | | 4 | | | | 4 | | | |
| 4 | | | | 8 | | | | | | | |
| 6 | | | | | | 6 | | | | | |
| 12 | | | | | | | | | | | |

Bootstrap 4 的网格系统是响应式的，列会根据屏幕大小自动重新排列。

网格类

Bootstrap 4 网格系统有以下 5 个类：

.col- 针对所有设备

.col-sm- 平板 - 屏幕宽度等于或大于 576px

.col-md- 桌面显示器 - 屏幕宽度等于或大于 768px)

.col-lg- 大桌面显示器 - 屏幕宽度等于或大于 992px)

`.col-xl-` 超大桌面显示器 - 屏幕宽度等于或大于 1200px)

网格系统规则

Bootstrap4 网格系统规则:

网格每一行需要放在设置了 `.container` (固定宽度) 或 `.container-fluid` (全屏宽度) 类的容器中, 这样就可以自动设置一些外边距与内边距。

使用行来创建水平的列组。

内容需要放置在列中, 并且只有列可以是行的直接子节点。

预定义的种类如 `.row` 和 `.col-sm-4` 可用于快速制作网格布局。

列通过填充创建列内容之间的间隙。这个间隙是通过 `.rows` 类上的负边距设置第一行和最后一列的偏移。

网格列是通过跨越指定的 12 个可用列来创建。例如, 设置三个相等的列, 需要使用三个 `.col-sm-4` 来设置。

Bootstrap 3 和 Bootstrap 4 最大的区别在于 Bootstrap 4 现在使用 **flexbox** (弹性盒子) 而不是浮动。Flexbox 的一大优势是, 没有指定宽度的网格列将自动设置为等宽与等高列。如果您想了解有关Flexbox的更多信息, 可以阅读我们的CSS Flexbox教程。

下表总结了 Bootstrap 网格系统如何在不同设备上工作的:

| | 超小设备 ≤576px | 平板 ≥576px | 桌面显示器 ≥768px | 大桌面显示器 ≥992px | 超大桌面显示器 ≥1200px |
|--------|----------------------|--------------|-----------------|------------------|--------------------|
| 容器最大宽度 | None (auto) | 540px | 720px | 960px | 1140px |
| 类前缀 | .col- | .col-sm- | .col-md- | .col-lg- | .col-xl- |
| 列数量和 | 12 | | | | |
| 间隙宽度 | 30px (一个列的每边分别 15px) | | | | |
| 可嵌套 | Yes | | | | |
| 列排序 | Yes | | | | |

以下各个类可以一起使用, 从而创建更灵活的页面布局。

Bootstrap 4 网格的基本结构

以下代码为 Bootstrap 4 网格的基本结构:

Bootstrap4 网格基本结构

```
<!-- 第一个例子: 控制列的宽度及在不同的设备上如何显示 -->
<div class="row">
<div class="col-*-*"></div>
</div>
<div class="row">
<div class="col-*-*"></div>
<div class="col-*-*"></div>
<div class="col-*-*"></div>
</div>
<!-- 第二个例子: 或让 Bootstrap 者自动处理布局 -->
<div class="row">
<div class="col"></div>
<div class="col"></div>
<div class="col"></div>
</div>
```

第一个例子: 创建一行(`<div class="row">`)。然后, 添加是需要的列(`.col-*-*` 类中设置)。第一个星号 (*) 表示响应的设备: sm, md, lg 或 xl, 第二个星号 (*) 表示一个数字, 同一行的数字相加为 12。

第二个例子: 不在每个 `col` 上添加数字, 让 bootstrap 自动处理布局, 同一行的每个列宽度相等: 两个 `"col"`, 每个就为 50% 的宽度。三个 `"col"` 每个就为 33.33% 的宽度, 四个 `"col"` 每个就为 25% 的宽度, 以此类推。同样, 你可以使用 `.col-sm|md|lg|xl` 来设置列的响应规则。

接下来我们可以看看实例。

创建相等宽度的列, Bootstrap 自动布局

实例

```
<div class="row">
<div class="col">.col</div>
<div class="col">.col</div>
<div class="col">.col</div>
</div>
```

尝试一下 »

等宽响应式列

以下实例演示了如何在平板及更大屏幕上创建等宽度的响应式列。在移动设备上，即屏幕宽度小于 **576px** 时，四个列将会上下堆叠排版：

实例

```
<div class="col-sm-3">.col-sm-3</div>
<div class="col-sm-3">.col-sm-3</div>
<div class="col-sm-3">.col-sm-3</div>
<div class="col-sm-3">.col-sm-3</div>
```

尝试一下 »

不等宽响应式列

以下实例演示了在平板及更大屏幕上创建不等宽度的响应式列。在移动设备上，即屏幕宽度小于 **576px** 时，四个列将会上下堆叠排版：

实例

```
<div class="row">
<div class="col-sm-4">.col-sm-4</div>
<div class="col-sm-8">.col-sm-8</div>
</div>
```

尝试一下 »

平板和桌面端

以下实例演示了在桌面设备的显示器上两个列的宽度各占 **50%**，如果在平板端则左边的宽度为 **25%**，右边的宽度为 **75%**，在移动手机等小型设备上会堆叠显示。

实例

```
<div class="container-fluid">
<div class="row">
<div class="col-sm-3 col-md-6">
<p>RUNOOB</p>
</div>
<div class="col-sm-9 col-md-6">
<p>菜鸟教程</p>
</div>
</div>
```

尝试一下 »

平板、桌面、大桌面显示器、超大桌面显示器

以下实例在平板、桌面、大桌面显示器、超大桌面显示器的宽度比例为分别为：**25%/75%**、**50%/50%**、**33.33%/66.67%**、**16.67/83.33%**，在移动手机等小型设备上会堆叠显示。

实例

```
<div class="container-fluid">
<div class="row">
<div class="col-sm-3 col-md-6 col-lg-4 col-xl-2">
<p>RUNOOB</p>
```



```
</div>
<div class="col-sm-9 col-md-6 col-lg-8 col-xl-10">
<p>菜鸟教程</p>
</div>
</div>
</div>
```

尝试一下 »

偏移列

偏移列通过 `offset-*-*` 类来设置。第一个星号(*)可以是 **sm**、**md**、**lg**、**xl**，表示屏幕设备类型，第二个星号(*)可以是 **1** 到 **11** 的数字。

为了在大屏幕显示器上使用偏移，请使用 `.offset-md-*` 类。这些类会把一个列的左外边距（margin）增加 * 列，其中 * 范围是从 **1** 到 **11**。

例如：`.offset-md-4` 是把.col-md-4 往右移了四列格。

实例

```
<div class="row">
<div class="col-md-4">.col-md-4</div>
<div class="col-md-4 offset-md-4">.col-md-4 .offset-md-4</div>
</div>
<div class="row">
<div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>
<div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>
</div>
<div class="row">
<div class="col-md-6 offset-md-3">.col-md-6 .offset-md-3</div>
</div>
```

尝试一下 »

☐ Bootstrap4 安装使用

Bootstrap4 文字排版 ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

☐ Bootstrap4 网格系统

Bootstrap4 颜色 ☐

Bootstrap4 文字排版

Bootstrap 4 默认设置

Bootstrap 4 默认的 `font-size` 为 16px，`line-height` 为 1.5。

默认的 `font-family` 为 "Helvetica Neue", Helvetica, Arial, sans-serif。

此外，所有的 `<p>` 元素 `margin-top: 0`、`margin-bottom: 1rem` (16px)。

<h1> - <h6>

Bootstrap 中定义了所有的 HTML 标题（h1 到 h6）的样式。请看下面的实例：

实例

```
<div class="container">
<h1>h1 Bootstrap 标题 (2.5rem = 40px)</h1>
<h2>h2 Bootstrap 标题 (2rem = 32px)</h2>
<h3>h3 Bootstrap 标题 (1.75rem = 28px)</h3>
<h4>h4 Bootstrap 标题 (1.5rem = 24px)</h4>
<h5>h5 Bootstrap 标题 (1.25rem = 20px)</h5>
<h6>h6 Bootstrap 标题 (1rem = 16px)</h6>
</div>
```

尝试一下 »

Display 标题类

Bootstrap 还提供了四个 Display 类来控制标题的样式: .display-1, .display-2, .display-3, .display-4。

实例

```
<div class="container">
<h1>Display 标题</h1>
<p>Display 标题可以输出更大更粗的字体样式。</p>
<h1 class="display-1">Display 1</h1>
<h1 class="display-2">Display 2</h1>
<h1 class="display-3">Display 3</h1>
<h1 class="display-4">Display 4</h1>
</div>
```

尝试一下 »

<small>

在 Bootstrap 4 中 HTML <small> 元素用于创建字号更小的颜色更浅的文本：

实例

```
<div class="container">
<h1>更小文本标题</h1>
<p>small 元素用于字号更小的颜色更浅的文本:</p>
<h1>h1 标题 <small>副标题</small></h1>
<h2>h2 标题 <small>副标题</small></h2>
<h3>h3 标题 <small>副标题</small></h3>
<h4>h4 标题 <small>副标题</small></h4>
<h5>h5 标题 <small>副标题</small></h5>
<h6>h6 标题 <small>副标题</small></h6>
</div>
```

尝试一下 »

<mark>

Bootstrap 4 定义 <mark> 为黄色背景及有一定的内边距：

实例

```
<div class="container">
<h1>高亮文本</h1>
<p>使用 mark 元素来 <mark>高亮</mark> 文本。</p>
</div>
```

尝试一下 »

<abbr>

Bootstrap 4 定义 HTML <abbr> 元素的样式为显示在文本底部的一条虚线边框：

实例

```
<div class="container">
<h1>Abbreviations</h1>
<p>The abbr element is used to mark up an abbreviation or acronym:</p>
<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>
</div>
```

尝试一下 »

<blockquote>

对于引用的内容可以在 `<blockquote>` 上添加 `.blockquote` 类：

实例

```
<div class="container">
<h1>Blockquotes</h1>
<p>The blockquote element is used to present content from another source:</p>
<blockquote class="blockquote">
<p>For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.</p>
<footer class="blockquote-footer">From WWF's website</footer>
</blockquote>
</div>
```

尝试一下 »

<dl>

Bootstrap 4 定义 HTML `<dl>` 元素的样式如下：

实例

```
<div class="container">
<h1>Description Lists</h1>
<p>The dl element indicates a description list:</p>
<dl>
<dt>Coffee</dt>
<dd>- black hot drink</dd>
<dt>Milk</dt>
<dd>- white cold drink</dd>
</dl>
</div>
```

尝试一下 »

<code>

Bootstrap 4 定义 HTML `<code>` 元素的样式如下：

实例

```
<div class="container">
<h1>代码片段</h1>
<p>可以将一些代码元素放到 code 元素里面:</p>
<p>以下 HTML 元素：<code>span</code>，<code>section</code>，和 <code>div</code> 用于定义部分文档内容。</p>
</div>
```

尝试一下 »

<kbd>

Bootstrap 4 定义 HTML `<kbd>` 元素的样式如下：

实例

```
<div class="container">
<h1>Keyboard Inputs</h1>
<p>To indicate input that is typically entered via the keyboard, use the kbd element:</p>
<p>Use <kbd>ctrl + p</kbd> to open the Print dialog box.</p>
</div>
```

尝试一下 »

<pre>

Bootstrap 4 定义 HTML `<pre>` 元素的样式如下:

实例

```
<div class="container">
<h1>Multiple Code Lines</h1>
<p>For multiple lines of code, use the pre element:</p>
<pre>
Text in a pre element
is displayed in a fixed-width
font, and it preserves
both spaces and
line breaks.
</pre>
</div>
```

尝试一下 »

更多排版类

下表提供了 Bootstrap4 更多排版类的实例:

| 类名 | 描述 | 实例 |
|----------------------------------|------------------------|------|
| <code>.font-weight-bold</code> | 加粗文本 | 尝试一下 |
| <code>.font-weight-normal</code> | 普通文本 | 尝试一下 |
| <code>.font-weight-light</code> | 更细的文本 | 尝试一下 |
| <code>.font-italic</code> | 斜体文本 | 尝试一下 |
| <code>.lead</code> | 让段落更突出 | 尝试一下 |
| <code>.small</code> | 指定更小文本 (为父元素的 85%) | 尝试一下 |
| <code>.text-left</code> | 左对齐 | 尝试一下 |
| <code>.text-center</code> | 居中 | 尝试一下 |
| <code>.text-right</code> | 右对齐 | 尝试一下 |
| <code>.text-justify</code> | 设定文本对齐,段落中超出屏幕部分文字自动换行 | 尝试一下 |
| <code>.text-nowrap</code> | 段落中超出屏幕部分不换行 | 尝试一下 |
| <code>.text-lowercase</code> | 设定文本小写 | 尝试一下 |
| <code>.text-uppercase</code> | 设定文本大写 | 尝试一下 |
| <code>.text-capitalize</code> | 设定单词首字母大写 | 尝试一下 |

| | | |
|------------------------------|---|------|
| <code>.initialism</code> | 显示在 <code><abbr></code> 元素中的文本以小号字体展示，且可以将小写字母转换为大写字母 | 尝试一下 |
| <code>.list-unstyled</code> | 移除默认的列表样式，列表项中左对齐 (<code></code> 和 <code></code> 中)。这个类仅适用于直接子列表项 (如果需要移除嵌套的列表项，你需要在嵌套的列表中使用该样式) | 尝试一下 |
| <code>.list-inline</code> | 将所有列表项放置同一行 | 尝试一下 |
| <code>.pre-scrollable</code> | 使 <code><pre></code> 元素可滚动，代码块区域最大高度为340px,一旦超出这个高度,就会在Y轴出现滚动条 | 尝试一下 |

[Bootstrap4 网格系统](#)

[Bootstrap4 颜色](#)

[点我分享笔记](#)

反馈/建议

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号：闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[Bootstrap4 文字排版](#)

[Bootstrap4 表格](#)

Bootstrap4 颜色

Bootstrap 4 提供了一些有代表意义的颜色类：`.text-muted`，`.text-primary`，`.text-success`，`.text-info`，`.text-warning`，`.text-danger`，`.text-secondary`，`.text-white`，`.text-dark` and `.text-light`：

实例

```
<div class="container">
<h2>代表指定意义的文本颜色</h2>
<p class="text-muted">柔和的文本。</p>
<p class="text-primary">重要的文本。</p>
<p class="text-success">执行成功的文本。</p>
<p class="text-info">代表一些提示信息的文本。</p>
<p class="text-warning">警告文本。</p>
<p class="text-danger">危险操作文本。</p>
<p class="text-secondary">副标题。</p>
<p class="text-dark">深灰色文字。</p>
<p class="text-light">浅灰色文本（白色背景上看不清楚）。</p>
<p class="text-white">白色文本（白色背景上看不清楚）。</p>
</div>
```

尝试一下 »

在链接中使用

实例

```
<div class="container">
<h2>文本颜色</h2>
<p>鼠标移动到链接。</p>
<a href="#" class="text-muted">柔和的链接。</a>
<a href="#" class="text-primary">主要链接。</a>
```

```
<a href="#" class="text-success">成功链接。</a>
<a href="#" class="text-info">信息文本链接。</a>
<a href="#" class="text-warning">警告链接。</a>
<a href="#" class="text-danger">危险链接。</a>
<a href="#" class="text-secondary">副标题链接。</a>
<a href="#" class="text-dark">深灰色链接。</a>
<a href="#" class="text-light">浅灰色链接。</a>
</div>
```

尝试一下 »

背景颜色

提供背景颜色的类有: `.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, `.bg-danger`, `.bg-secondary`, `.bg-dark` 和 `.bg-light`。

注意背景颜色不会设置文本的颜色, 在一些实例中你需要与 `.text-*` 类一起使用。

实例

```
<div class="container">
<h2>背景颜色</h2>
<p class="bg-primary text-white">重要的背景颜色。</p>
<p class="bg-success text-white">执行成功背景颜色。</p>
<p class="bg-info text-white">信息提示背景颜色。</p>
<p class="bg-warning text-white">警告背景颜色</p>
<p class="bg-danger text-white">危险背景颜色。</p>
<p class="bg-secondary text-white">副标题背景颜色。</p>
<p class="bg-dark text-white">深灰背景颜色。</p>
<p class="bg-light text-dark">浅灰背景颜色。</p>
</div>
```

尝试一下 »

[Bootstrap4 文字排版](#)

[Bootstrap4 表格](#)

[点我分享笔记](#)

反馈/建议

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号: 闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[Bootstrap4 颜色](#)

[Bootstrap4 图像形状](#)

Bootstrap4 表格

Bootstrap4 基础表格

Bootstrap4 通过 `.table` 类来设置基础表格的样式, 实例如下:

实例

```
<table class="table">
<thead>
<tr>
```

```
<th>Firstname</th>
<th>Lastname</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr>
<td>Mary</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr>
<td>July</td>
<td>Dooley</td>
<td>july@example.com</td>
</tr>
</tbody>
</table>
```

尝试一下 »

条纹表格

通过添加 `.table-striped` 类，您将在 `<tbody>` 内的行上看到条纹，如下面的实例所示：

实例

```
<table class="table table-striped">
<thead>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr>
<td>Mary</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr>
<td>July</td>
<td>Dooley</td>
<td>july@example.com</td>
</tr>
</tbody>
</table>
```

尝试一下 »

带边框表格

`.table-bordered` 类可以为表格添加边框

实例

```
<table class="table table-bordered">
<thead>
<tr>
```

```
<th>Firstname</th>
<th>Lastname</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr>
<td>Mary</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr>
<td>July</td>
<td>Dooley</td>
<td>july@example.com</td>
</tr>
</tbody>
</table>
```

尝试一下 »

鼠标悬停状态表格

.table-hover 类可以为表格的每一行添加鼠标悬停效果（灰色背景）：

实例

```
<table class="table table-hover">
<thead>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr>
<td>Mary</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr>
<td>July</td>
<td>Dooley</td>
<td>july@example.com</td>
</tr>
</tbody>
</table>
```

尝试一下 »

黑色背景表格

.table-dark 类可以为表格添加黑色背景：

实例

```
<table class="table table-dark">
<thead>
<tr>
```



```
<th>Firstname</th>
<th>Lastname</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr>
<td>Mary</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr>
<td>July</td>
<td>Dooley</td>
<td>july@example.com</td>
</tr>
</tbody>
</table>
```

尝试一下 »

黑色条纹表格

联合使用 `.table-dark` 和 `.table-striped` 类可以创建黑色的条纹表格：

实例

```
<table class="table table-dark table-striped">
<thead>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr>
<td>Mary</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr>
<td>July</td>
<td>Dooley</td>
<td>july@example.com</td>
</tr>
</tbody>
</table>
```

尝试一下 »

鼠标悬停效果 - 黑色背景表格

联合使用 `.table-dark` 和 `.table-hover` 类可以设置黑色背景表格的鼠标悬停效果：

实例

```
<table class="table table-dark table-hover">
<thead>
<tr>
```

```
<th>Firstname</th>
<th>Lastname</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr>
<td>Mary</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr>
<td>July</td>
<td>Dooley</td>
<td>july@example.com</td>
</tr>
</tbody>
</table>
```

尝试一下 »

指定意义的颜色类

通过指定意义的颜色类可以为表格的行或者单元格设置颜色：

实例

```
<table class="table">
<thead>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Defaultson</td>
<td>def@somemail.com</td>
</tr>
<tr class="table-primary">
<td>Primary</td>
<td>Joe</td>
<td>joe@example.com</td>
</tr>
<tr class="table-success">
<td>Success</td>
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr class="table-danger">
<td>Danger</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr class="table-info">
<td>Info</td>
<td>Dooley</td>
<td>july@example.com</td>
</tr>
<tr class="table-warning">
<td>Warning</td>
<td>Refs</td>
<td>bo@example.com</td>
</tr>
```

```
<tr class="table-active">
<td>Active</td>
<td>Activeson</td>
<td>act@example.com</td>
</tr>
<tr class="table-secondary">
<td>Secondary</td>
<td>Secondson</td>
<td>sec@example.com</td>
</tr>
<tr class="table-light">
<td>Light</td>
<td>Angie</td>
<td>angie@example.com</td>
</tr>
<tr class="table-dark text-dark">
<td>Dark</td>
<td>Bo</td>
<td>bo@example.com</td>
</tr>
</tbody>
</table>
```

尝试一下 »

下表列出了表格颜色类的说明：

| 类名 | 描述 |
|-------------------------------|-------------------|
| <code>.table-primary</code> | 蓝色: 指定这是一个重要的操作 |
| <code>.table-success</code> | 绿色: 指定这是一个允许执行的操作 |
| <code>.table-danger</code> | 红色: 指定这是可以危险的操作 |
| <code>.table-info</code> | 浅蓝色: 表示内容已变更 |
| <code>.table-warning</code> | 橘色: 表示需要注意的操作 |
| <code>.table-active</code> | 灰色: 用于鼠标悬停效果 |
| <code>.table-secondary</code> | 灰色: 表示内容不怎么重要 |
| <code>.table-light</code> | 浅灰色, 可以是表格行的背景 |
| <code>.table-dark</code> | 深灰色, 可以是表格行的背景 |

表头颜色

在 Bootstrap v4.0.0-beta.2 中 `.thead-dark` 类用于给表头添加黑色背景, `.thead-light` 类用于给表头添加灰色背景:

在 Bootstrap v4.0.0-beta 这个版本中, `.thead-inverse` 类用于给表头添加黑色背景, `.thead-default` 类用于给表头添加灰色背景。

实例

```
<table class="table">
<thead class="thead-dark">
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
```

```
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr>
<td>Mary</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr>
<td>July</td>
<td>Dooley</td>
<td>july@example.com</td>
</tr>
</tbody>
</table>
<table class="table">
<thead class="thead-light">
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr>
<td>Mary</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr>
<td>July</td>
<td>Dooley</td>
<td>july@example.com</td>
</tr>
</tbody>
</table>
```

尝试一下 »

较小的表格

.table-sm 类用于通过减少内边距来设置较小的表格:

实例

```
<table class="table table-bordered table-sm">
<thead>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr>
<td>Mary</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr>
<td>July</td>
```

```
<td>Dooley</td>
<td>july@example.com</td>
</tr>
</tbody>
</table>
```

尝试一下 »

响应式表格

.table-responsive 类用于创建响应式表格：在屏幕宽度小于 992px 时会创建水平滚动条，如果可视区域宽度大于 992px 则显示不同效果（没有滚动条）：

实例

```
<div class="table-responsive">
<table class="table">
<thead>
<tr>
<th>#</th>
<th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
<th>City</th>
<th>Country</th>
<th>Sex</th>
<th>Example</th>
<th>Example</th>
<th>Example</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Anna</td>
<td>Pitt</td>
<td>35</td>
<td>New York</td>
<td>USA</td>
<td>Female</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
</div>
```

尝试一下 »

你可以通过以下类设定在指定屏幕宽度下显示滚动条：

| 类名 | 屏幕宽度 |
|-----------------------------------|----------|
| <code>.table-responsive-sm</code> | < 576px |
| <code>.table-responsive-md</code> | < 768px |
| <code>.table-responsive-lg</code> | < 992px |
| <code>.table-responsive-xl</code> | < 1200px |

实例

```
<div class="table-responsive-sm">
<table class="table">
...

```

```
</table>
</div>
```

尝试一下 »

[Bootstrap4 颜色](#)

[Bootstrap4 图像形状](#)

[点我分享笔记](#)

[反馈/建议](#)

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号: 闽ICP备15012807号-1

□

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[Bootstrap4 表格](#)

[Bootstrap4 Jumbotron](#)

Bootstrap4 图像形状

圆角图片

.rounded 类可以让图片显示圆角效果:

实例

```

```

尝试一下 »

椭圆图片

.rounded-circle 类可以设置椭圆形图片:

实例

```

```

尝试一下 »

缩略图

.img-thumbnail 类用于设置图片缩略图(图片有边框):

实例

```

```

尝试一下 »

图片对齐方式

使用 **.float-right** 类来设置图片右对齐, 使用 **.float-left** 类设置图片左对齐:

实例

```


```

尝试一下 »

响应式图片

图像有各种各样的尺寸，我们需要根据屏幕的大小自动适应。

我们可以通过在 `` 标签中添加 `.img-fluid` 类来设置响应式图片。

`.img-fluid` 类设置了 `max-width: 100%;` 、 `height: auto;` :

实例

```

```

尝试一下 »

☐ Bootstrap4 表格

Bootstrap4 Jumbotron ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号: 闽ICP备15012807号-1

□

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

☐ Bootstrap4 图像形状

Bootstrap4 信息提示框 ☐

Bootstrap4 Jumbotron

Jumbotron（超大屏幕）会创建一个大的灰色背景框，里面可以设置一些特殊的内容和信息。

提示: Jumbotron 里头可以放一些 HTML 标签，也可以是 Bootstrap 的元素。

菜鸟教程

学的不仅是技术，更是梦想!!!

我们可以通过在 `<div>` 元素 中添加 `.jumbotron` 类来创建 jumbotron:

实例

```
<div class="jumbotron">
<h1>菜鸟教程</h1>
<p>学的不仅是技术，更是梦想!!! </p>
</div>
```

尝试一下 »

全屏幕的 Jumbotron

如果你想创建一个没有圆角的全屏幕，可以在 `.jumbotron-fluid` 类里头的 `div` 添加 `.container` 或 `.container-fluid` 类来实现：

实例

```
<div class="jumbotron jumbotron-fluid">
<div class="container">
<h1>菜鸟教程</h1>
<p>学的不仅是技术，更是梦想！！</p>
</div>
</div>
```

尝试一下 »

☐ Bootstrap4 图像形状

Bootstrap4 信息提示框 ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

☐ Bootstrap4 Jumbotron

Bootstrap4 按钮 ☐

Bootstrap4 信息提示框

Bootstrap 4 可以很容易实现信息提示框。

成功! 指定操作成功提示信息。

信息! 请注意这个信息。

警告! 设置警告信息。

错误! 失败的操作

首选! 这是一个重要的操作信息。

次要的! 显示一些不重要的信息。

深灰色! 深灰色提示框。

提示框可以使用 `.alert` 类, 后面加上 `.alert-success`, `.alert-info`, `.alert-warning`, `.alert-danger`, `.alert-primary`, `.alert-secondary`, `.alert-light` 或 `.alert-dark` 类来实现:

实例

```
<div class="alert alert-success">
<strong>成功!</strong> 指定操作成功提示信息。
</div>
```

尝试一下 »

提示框添加链接

提示框中在链接的标签上添加 `alert-link` 类来设置匹配提示框颜色的链接:

实例

```
<div class="alert alert-success">
<strong>成功!</strong> 你应该认真阅读 <a href="#" class="alert-link">这条信息</a>。
</div>
```

尝试一下 »

关闭提示框

我们可以在提示框中的 `div` 中添加 `.alert-dismissible` 类, 然后在关闭按钮的链接上添加 `class="close"` 和 `data-dismiss="alert"` 类来设置提示框的关闭操作。

实例

```
<div class="alert alert-success alert-dismissible">
<button type="button" class="close" data-dismiss="alert">&times;</button>
<strong>成功!</strong> 指定操作成功提示信息。
</div>
```

尝试一下 »

提示: `×` (×) 是 HTML 实体字符, 来表示关闭操作, 而不是字母 "x"。

提示框动画

`.fade` 和 `.show` 类用于设置提示框在关闭时的淡出和淡入效果：

实例

```
<div class="alert alert-danger alert-dismissible fade show">
```

尝试一下 »

[Bootstrap4 Jumbotron](#)

Bootstrap4 按钮 [□](#)

[□](#) 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号：闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[□](#) Bootstrap4 信息提示框

Bootstrap4 按钮组 [□](#)

Bootstrap4 按钮

Bootstrap 4 提供了不同样式的按钮。

实例

```
<button type="button" class="btn">基本按钮</button>
<button type="button" class="btn btn-primary">主要按钮</button>
<button type="button" class="btn btn-secondary">次要按钮</button>
<button type="button" class="btn btn-success">成功</button>
<button type="button" class="btn btn-info">信息</button>
<button type="button" class="btn btn-warning">警告</button>
<button type="button" class="btn btn-danger">危险</button>
<button type="button" class="btn btn-dark">黑色</button>
<button type="button" class="btn btn-light">浅色</button>
<button type="button" class="btn btn-link">链接</button>
```

尝试一下 »

按钮类可用于 `<a>`，`<button>`，或 `<input>` 元素上：

实例

```
<a href="#" class="btn btn-info" role="button">链接按钮</a>
<button type="button" class="btn btn-info">按钮</button>
<input type="button" class="btn btn-info" value="输入框按钮">
<input type="submit" class="btn btn-info" value="提交按钮">
```

尝试一下 »

按钮设置边框

实例

```
<button type="button" class="btn btn-outline-primary">主要按钮</button>
```

```
<button type="button" class="btn btn-outline-secondary">次要按钮</button>
<button type="button" class="btn btn-outline-success">成功</button>
<button type="button" class="btn btn-outline-info">信息</button>
<button type="button" class="btn btn-outline-warning">警告</button>
<button type="button" class="btn btn-outline-danger">危险</button>
<button type="button" class="btn btn-outline-dark">黑色</button>
<button type="button" class="btn btn-outline-light text-dark">浅色</button>
```

尝试一下 »

不同大小的按钮

Bootstrap 4 可以设置按钮的大小：

实例

```
<button type="button" class="btn btn-primary btn-lg">大号按钮</button>
<button type="button" class="btn btn-primary">默认按钮</button>
<button type="button" class="btn btn-primary btn-sm">小号按钮</button>
```

尝试一下 »

块级按钮

通过添加 `.btn-block` 类可以设置块级按钮：

实例

```
<button type="button" class="btn btn-primary btn-block">按钮 1</button>
```

尝试一下 »

激活和禁用的按钮

按钮可设置为激活或者禁止点击的状态。

`.active` 类可以设置按钮是可用的，`disabled` 属性可以设置按钮是不可点击的。注意 `<a>` 元素不支持 `disabled` 属性，你可以通过添加 `.disabled` 类来禁止链接的点击。

实例

```
<button type="button" class="btn btn-primary active">点击后的按钮</button>
<button type="button" class="btn btn-primary disabled">禁止点击的按钮</button>
<a href="#" class="btn btn-primary disabled">禁止点击的链接</a>
```

尝试一下 »

☐ Bootstrap4 信息提示框

Bootstrap4 按钮组 ☐

☐ 点我分享笔记

反馈/建议

Bootstrap4 按钮组

Bootstrap 4 中允许我们将按钮放在同一行上。

可以在 `<div>` 元素上添加 `.btn-group` 类来创建按钮组。

实例

```
<div class="btn-group">
<button type="button" class="btn btn-primary">Apple</button>
<button type="button" class="btn btn-primary">Samsung</button>
<button type="button" class="btn btn-primary">Sony</button>
</div>
```

[尝试一下 »](#)

提示: 我们可以使用 `.btn-group-lg|sm` 类来设置按钮组的大小。

实例

```
<div class="btn-group btn-group-lg">
<button type="button" class="btn btn-primary">Apple</button>
<button type="button" class="btn btn-primary">Samsung</button>
<button type="button" class="btn btn-primary">Sony</button>
</div>
```

[尝试一下 »](#)

垂直按钮组

可以使用 `.btn-group-vertical` 类来创建垂直的按钮组:

实例

```
<div class="btn-group-vertical">
<button type="button" class="btn btn-primary">Apple</button>
<button type="button" class="btn btn-primary">Samsung</button>
<button type="button" class="btn btn-primary">Sony</button>
</div>
```

[尝试一下 »](#)

内嵌按钮组及下拉菜单

我们可以在按钮组内设置下拉菜单:

实例

```
<div class="btn-group">
<button type="button" class="btn btn-primary">Apple</button>
<button type="button" class="btn btn-primary">Samsung</button>
<div class="btn-group">
<button type="button" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">
Sony
</button>
<div class="dropdown-menu">
<a class="dropdown-item" href="#">Tablet</a>
<a class="dropdown-item" href="#">Smartphone</a>
</div>
</div>
</div>
```

尝试一下 »

拆分按钮下拉菜单

实例

```
<div class="btn-group">
<button type="button" class="btn btn-primary">Sony</button>
<button type="button" class="btn btn-primary dropdown-toggle dropdown-toggle-split" data-toggle="dropdown">
<span class="caret"></span>
</button>
<div class="dropdown-menu">
<a class="dropdown-item" href="#">Tablet</a>
<a class="dropdown-item" href="#">Smartphone</a>
</div>
</div>
```

尝试一下 »

垂直按钮组及下拉菜单

实例

```
<div class="btn-group-vertical">
<button type="button" class="btn btn-primary">Apple</button>
<button type="button" class="btn btn-primary">Samsung</button>
<div class="btn-group">
<button type="button" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">
Sony
</button>
<div class="dropdown-menu">
<a class="dropdown-item" href="#">Tablet</a>
<a class="dropdown-item" href="#">Smartphone</a>
</div>
</div>
</div>
```

尝试一下 »

☐ Bootstrap4 按钮

Bootstrap4 徽章 (Badges) ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号: 闽ICP备15012807号-1

□

首页 HTML CSS JS 本地书签

☐ Bootstrap4 按钮组

Bootstrap4 进度条 ☐

Bootstrap4 徽章 (Badges)

徽章

测试标题 New

测试标题 New

测试标题 New

测试标题 New

测试标题 New

测试标题 New

徽章（Badges）主要用于突出显示新的或未读的项。如需使用徽章，只需要将 `.badge` 类加上带有指定意义的颜色类 (如 `.badge-secondary`) 添加到 `` 元素上即可。徽章可以根据父元素的大小的变化而变化：

实例

```
<h1>测试标题 <span class="badge badge-secondary">New</span></h1>
<h2>测试标题 <span class="badge badge-secondary">New</span></h2>
<h3>测试标题 <span class="badge badge-secondary">New</span></h3>
<h4>测试标题 <span class="badge badge-secondary">New</span></h4>
<h5>测试标题 <span class="badge badge-secondary">New</span></h5>
<h6>测试标题 <span class="badge badge-secondary">New</span></h6>
```

尝试一下 »

各种颜色类型的徽章

以下列出了所有颜色类型的徽章：

实例

```
<span class="badge badge-primary">主要</span>
<span class="badge badge-secondary">次要</span>
<span class="badge badge-success">成功</span>
<span class="badge badge-danger">危险</span>
<span class="badge badge-warning">警告</span>
<span class="badge badge-info">信息</span>
<span class="badge badge-light">浅色</span>
<span class="badge badge-dark">深色</span>
```

尝试一下 »

药丸形状徽章

使用 `.badge-pill` 类来设置药丸形状徽章：

实例

```
<span class="badge badge-pill badge-default">默认</span>
<span class="badge badge-pill badge-primary">主要</span>
<span class="badge badge-pill badge-success">成功</span>
<span class="badge badge-pill badge-info">信息</span>
<span class="badge badge-pill badge-warning">警告</span>
<span class="badge badge-pill badge-danger">危险</span>
```

尝试一下 »

徽章插入到元素内

以下实例将徽章嵌入到按钮内：

实例

```
<button type="button" class="btn btn-primary">
Messages <span class="badge badge-light">4</span>
</button>
```

尝试一下 »

☐ Bootstrap4 按钮组

Bootstrap4 进度条 ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

☐

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

☐ Bootstrap4 徽章（Badges）

Bootstrap4 分页 ☐

Bootstrap4 进度条



进度条可以显示用户任务的完成过程。

创建一个基本的进度条的步骤如下：

添加一个带有 `.progress` 类的 `<div>`。

接着，在上面的 `<div>` 内，添加一个带有 `class .progress-bar` 的空的 `<div>`。

添加一个带有百分比表示的宽度的 `style` 属性，例如 `style="width:70%"` 表示进度条在 **70%** 的位置。

实例

```
<div class="progress">
<div class="progress-bar" style="width:70%"></div>
</div>
```

尝试一下 »

进度条高度

进度条高度默认为 **16px**。我们可以使用 `CSS` 的 `height` 属性来修改他：

实例

```
<div class="progress" style="height:20px;">
<div class="progress-bar" style="width:40%"></div>
</div>
```

尝试一下 »

进度条标签

可以在进度条内添加文本，如进度的百分比：

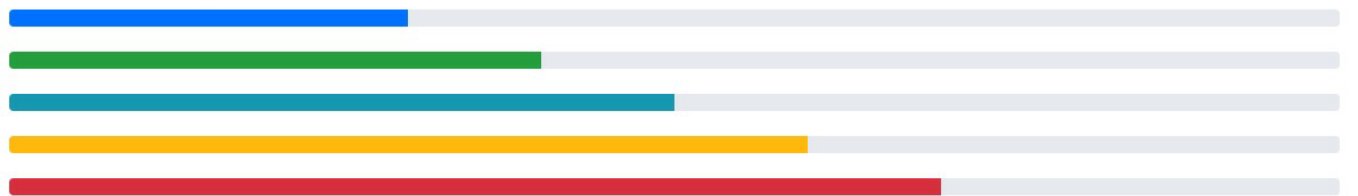
实例

```
<div class="progress">
<div class="progress-bar" style="width:70%">70%</div>
</div>
```

尝试一下 »

不同颜色的进度条

默认情况下进度条为蓝色，Bootstrap4 还提供了以下颜色的进度条：



实例

```
<div class="progress">
<div class="progress-bar bg-success" style="width:40%"></div>
</div>
<div class="progress">
<div class="progress-bar bg-info" style="width:50%"></div>
</div>
<div class="progress">
<div class="progress-bar bg-warning" style="width:60%"></div>
</div>
<div class="progress">
<div class="progress-bar bg-danger" style="width:70%"></div>
</div>
```

尝试一下 »

条纹的进度条

可以使用 .progress-bar-striped 类来设置条纹进度条：

实例

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

尝试一下 »

动画进度条

使用 .progress-bar-animated 类可以为进度条添加动画：

实例

```
<div class="progress-bar progress-bar-striped progress-bar-animated" style="width: 40%"></div>
```

尝试一下 »

混合色彩进度条

进度条可以设置多种颜色：

实例


```
<div class="progress">
<div class="progress-bar bg-success" style="width:40%">
Free Space
</div>
<div class="progress-bar bg-warning" style="width:10%">
Warning
</div>
<div class="progress-bar bg-danger" style="width:20%">
Danger
</div>
</div>
```

尝试一下 »

Bootstrap4 徽章 (Badges)

Bootstrap4 分页

点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号: 闽ICP备15012807号-1

首页 HTML CSS JS 本地书签

Bootstrap4 进度条

Bootstrap4 列表组

Bootstrap4 分页

网页开发过程，如果碰到内容过多，一般都会做分页处理。

Bootstrap 4 可以很简单的实现分页效果。

要创建一个基本的分页可以在 `` 元素上添加 `.pagination` 类。然后在 `` 元素上添加 `.page-item` 类：

实例

```
<ul class="pagination">
<li class="page-item"><a class="page-link" href="#">Previous</a></li>
<li class="page-item"><a class="page-link" href="#">1</a></li>
<li class="page-item"><a class="page-link" href="#">2</a></li>
<li class="page-item"><a class="page-link" href="#">3</a></li>
<li class="page-item"><a class="page-link" href="#">Next</a></li>
</ul>
```

尝试一下 »

当前页页码状态

当前页可以使用 `.active` 类来高亮显示：

实例

```
<ul class="pagination">
<li class="page-item"><a class="page-link" href="#">Previous</a></li>
<li class="page-item"><a class="page-link" href="#">1</a></li>
<li class="page-item active"><a class="page-link" href="#">2</a></li>
<li class="page-item"><a class="page-link" href="#">3</a></li>
```

```
<li class="page-item"><a class="page-link" href="#">Next</a></li>
</ul>
```

尝试一下 »

不可点击的分页链接

`.disabled` 类可以设置分页链接不可点击:

实例

```
<ul class="pagination">
<li class="page-item disabled"><a class="page-link" href="#">Previous</a></li>
<li class="page-item"><a class="page-link" href="#">1</a></li>
<li class="page-item"><a class="page-link" href="#">2</a></li>
<li class="page-item"><a class="page-link" href="#">3</a></li>
<li class="page-item"><a class="page-link" href="#">Next</a></li>
</ul>
```

尝试一下 »

分页显示大小

可以将分页条目设置为不同的大小。

`.pagination-lg` 类设置大字体的分页条目, `.pagination-sm` 类设置小字体的分页条目:

实例

```
<ul class="pagination pagination-lg">
<li class="page-item"><a class="page-link" href="#">Previous</a></li>
<li class="page-item"><a class="page-link" href="#">1</a></li>
<li class="page-item"><a class="page-link" href="#">2</a></li>
<li class="page-item"><a class="page-link" href="#">3</a></li>
<li class="page-item"><a class="page-link" href="#">Next</a></li>
</ul>
<ul class="pagination pagination-sm">
<li class="page-item"><a class="page-link" href="#">Previous</a></li>
<li class="page-item"><a class="page-link" href="#">1</a></li>
<li class="page-item"><a class="page-link" href="#">2</a></li>
<li class="page-item"><a class="page-link" href="#">3</a></li>
<li class="page-item"><a class="page-link" href="#">Next</a></li>
</ul>
```

尝试一下 »

面包屑导航

`.breadcrumb` 和 `.breadcrumb-item` 类用于设置面包屑导航:

实例

```
<ul class="breadcrumb">
<li class="breadcrumb-item"><a href="#">Photos</a></li>
<li class="breadcrumb-item"><a href="#">Summer 2017</a></li>
<li class="breadcrumb-item"><a href="#">Italy</a></li>
<li class="breadcrumb-item active">Rome</li>
</ul>
```

尝试一下 »



Bootstrap4 列表组

大部分基础列表组都是无序的。

First item

Second item

Third item

要创建列表组，可以在 `` 元素上添加 `.list-group` 类, 在 `` 元素上添加 `.list-group-item` 类:

实例

```
<ul class="list-group">
<li class="list-group-item">First item</li>
<li class="list-group-item">Second item</li>
<li class="list-group-item">Third item</li>
</ul>
```

尝试一下 »

激活状态的列表项

通过添加 `.active` 类来设置激活状态的列表项:

实例

```
<ul class="list-group">
<li class="list-group-item active">Active item</li>
<li class="list-group-item">Second item</li>
<li class="list-group-item">Third item</li>
</ul>
```

尝试一下 »

禁用的列表项

`.disabled` 类用于设置禁用的列表项:

实例

```
<ul class="list-group">
<li class="list-group-item disabled">Disabled item</li>
<li class="list-group-item">Second item</li>
<li class="list-group-item">Third item</li>
</ul>
```

尝试一下 »

链接列表项

要创建一个链接的列表项，可以将 `` 替换 `<div>`，`<a>` 替换 ``。如果你想鼠标悬停显示灰色背景就添加 `.list-group-item-action` 类：

实例

```
<div class="list-group">
<a href="#" class="list-group-item list-group-item-action">First item</a>
<a href="#" class="list-group-item list-group-item-action">Second item</a>
<a href="#" class="list-group-item list-group-item-action">Third item</a>
</div>
```

尝试一下 »

多种颜色列表项

| |
|--------|
| 激活列表项 |
| 成功列表项 |
| 次要列表项 |
| 信息列表项 |
| 警告列表项 |
| 危险列表项 |
| 主要列表项 |
| 深灰色列表项 |
| 浅色列表项 |

列表项目的颜色可以通过以下列来设置：`.list-group-item-success`，`list-group-item-secondary`，`list-group-item-info`，`list-group-item-warning`，`.list-group-item-danger`，`list-group-item-dark` 和 `list-group-item-light`：

实例

```
<ul class="list-group">
<li class="list-group-item list-group-item-success">成功列表项</li>
<li class="list-group-item list-group-item-secondary">次要列表项</li>
<li class="list-group-item list-group-item-info">信息列表项</li>
<li class="list-group-item list-group-item-warning">警告列表项</li>
<li class="list-group-item list-group-item-danger">危险列表项</li>
<li class="list-group-item list-group-item-primary">主要列表项</li>
<li class="list-group-item list-group-item-dark">深灰色列表项</li>
<li class="list-group-item list-group-item-light">浅色列表项</li>
</ul>
```

尝试一下 »

链接的多种颜色列表项

实例

```
<div class="list-group">
<a href="#" class="list-group-item list-group-item-action">激活列表项</a>
<a href="#" class="list-group-item list-group-item-success">成功列表项</a>
<a href="#" class="list-group-item list-group-item-secondary">次要列表项</a>
```

```
<a href="#" class="list-group-item list-group-item-info">信息列表项</a>
<a href="#" class="list-group-item list-group-item-warning">警告列表项</a>
<a href="#" class="list-group-item list-group-item-danger">危险列表项</a>
<a href="#" class="list-group-item list-group-item-primary">主要列表项</a>
<a href="#" class="list-group-item list-group-item-dark">深灰色列表项</a>
<a href="#" class="list-group-item list-group-item-light">浅色列表项</a>
</div>
```

尝试一下 »

[Bootstrap4 分页](#)

Bootstrap4 卡片 [Bootstrap4 卡片](#)

[点我分享笔记](#)

反馈/建议

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号: 闽ICP备15012807号-1

[Bootstrap4 卡片](#)

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[Bootstrap4 列表组](#)

Bootstrap4 下拉菜单 [Bootstrap4 下拉菜单](#)

Bootstrap4 卡片

简单的卡片

我们可以通过 Bootstrap4 的 `.card` 与 `.card-body` 类来创建一个简单的卡片，实例如下：

实例

```
<div class="card">
<div class="card-body">简单的卡片</div>
</div>
```

尝试一下 »

Bootstrap 4 的卡片类似 Bootstrap 3 中的面板、图片缩略图、well。

头部和底部

`.card-header` 类用于创建卡片的头部样式，`.card-footer` 类用于创建卡片的底部样式：

实例

```
<div class="card">
<div class="card-header">头部</div>
<div class="card-body">内容</div>
<div class="card-footer">底部</div>
</div>
```

尝试一下 »

多种颜色卡片

Bootstrap 4 提供了多种卡片的背景颜色类: `.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, `.bg-danger`, `.bg-secondary`, `.bg-dark` 和 `.bg-light`。

实例

```
<div class="container">
<h2>多种颜色卡片</h2>
<div class="card">
<div class="card-body">Basic card</div>
</div>
<br>
<div class="card bg-primary text-white">
<div class="card-body">Primary card</div>
</div>
<br>
<div class="card bg-success text-white">
<div class="card-body">Success card</div>
</div>
<br>
<div class="card bg-info text-white">
<div class="card-body">Info card</div>
</div>
<br>
<div class="card bg-warning text-white">
<div class="card-body">Warning card</div>
</div>
<br>
<div class="card bg-danger text-white">
<div class="card-body">Danger card</div>
</div>
<br>
<div class="card bg-secondary text-white">
<div class="card-body">Secondary card</div>
</div>
<br>
<div class="card bg-dark text-white">
<div class="card-body">Dark card</div>
</div>
<br>
<div class="card bg-light text-dark">
<div class="card-body">Light card</div>
</div>
</div>
```

尝试一下 »

标题、文本和链接

我们可以在头部元素上使用 `.card-title` 类来设置卡片的标题。`.card-text` 类用于设置卡片正文的内容。`.card-link` 类用于给链接设置颜色。

实例

```
<div class="card">
<div class="card-body">
<h4 class="card-title">Card title</h4>
<p class="card-text">Some example text. Some example text.</p>
<a href="#" class="card-link">Card link</a>
<a href="#" class="card-link">Another link</a>
</div>
</div>
```

尝试一下 »

图片卡片

我们可以给 `` 添加 `.card-img-top` (图片在文字上方) 或 `.card-img-bottom` (图片在文字下方) 来设置图片卡片:

实例

```
<div class="card" style="width:400px">

<div class="card-body">
<h4 class="card-title">John Doe</h4>
<p class="card-text">Some example text.</p>
<a href="#" class="btn btn-primary">See Profile</a>
</div>
</div>
```

尝试一下 »

如果图片要设置为背景，可以使用 `.card-img-overlay` 类：

实例

```
<div class="card" style="width:500px">

<div class="card-img-overlay">
<h4 class="card-title">John Doe</h4>
<p class="card-text">Some example text.</p>
<a href="#" class="btn btn-primary">See Profile</a>
</div>
</div>
```

尝试一下 »

[Bootstrap4 列表组](#)

[Bootstrap4 下拉菜单](#)

[点我分享笔记](#)

反馈/建议

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号：闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[Bootstrap4 卡片](#)

[Bootstrap4 折叠](#)

Bootstrap4 下拉菜单

下拉菜单是可切换的，是以列表格式显示链接的上下文菜单。

实例

```
<div class="dropdown">
<button type="button" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">
Dropdown button
</button>
<div class="dropdown-menu">
<a class="dropdown-item" href="#">Link 1</a>
<a class="dropdown-item" href="#">Link 2</a>
<a class="dropdown-item" href="#">Link 3</a>
</div>
</div>
```

尝试一下 »

实例解析

`.dropdown` 类用来指定一个下拉菜单。

我们可以使用一个按钮或链接来打开下拉菜单， 按钮或链接需要添加 `.dropdown-toggle` 和 `data-toggle="dropdown"` 属性。

`<div>` 元素上添加 `.dropdown-menu` 类来设置实际下拉菜单，然后在下拉菜单的选项中添加 `.dropdown-item` 类。

下拉菜单中的分割线

`.dropdown-divider` 类用于在下拉菜单中创建一个水平的分割线：

实例

```
<div class="dropdown-divider"></div>
```

尝试一下 »

下拉菜单中的标题

`.dropdown-header` 类用于在下拉菜单中添加标题：

实例

```
<div class="dropdown-header">Dropdown header 1</div>
```

尝试一下 »

下拉菜单中的可用项与禁用项

`.active` 类会让下拉菜单的选项高亮显示 (添加蓝色背景)。

如果要禁用下拉菜单的选项，可以使用 `.disabled` 类。

实例

```
<a class="dropdown-item active" href="#">Active</a>
<a class="dropdown-item disabled" href="#">Disabled</a>
```

尝试一下 »

下拉菜单的定位

如果我们想让下拉菜单右对齐，可以在元素上的 `.dropdown-menu` 类后添加 `.dropdown-menu-right` 类。

实例

```
<div class="dropdown-menu dropdown-menu-right">
```

尝试一下 »

指定向上弹出的下拉菜单

如果你希望下拉菜单向上弹出，可以将 `<div>` 元素的 `class="dropdown"` 替换为 `"dropup"`：

实例

```
<div class="dropup">
```

尝试一下 »

按钮中设置下拉菜单

我们可以在按钮中添加下拉菜单：

实例

```
<div class="btn-group">
<button type="button" class="btn btn-primary">Sony</button>
<button type="button" class="btn btn-primary dropdown-toggle dropdown-toggle-split" data-toggle="dropdown">
<span class="caret"></span>
</button>
<div class="dropdown-menu">
<a class="dropdown-item" href="#">Tablet</a>
<a class="dropdown-item" href="#">Smartphone</a>
</div>
</div>
```

尝试一下 »

☐ 点我分享笔记

反馈/建议



Bootstrap4 折叠

Bootstrap4 折叠可以很容易的实现内容的显示与隐藏。

实例

```
<button data-toggle="collapse" data-target="#demo">折叠</button>
<div id="demo" class="collapse">
  Lorem ipsum dolor text....
</div>
```

尝试一下 »

实例解析

.collapse 类用于指定一个折叠元素 (实例中的 `<div>`); 点击按钮后会在隐藏与显示之间切换。

控制内容的隐藏与显示，需要在 `<a>` 或 `<button>` 元素上添加 `data-toggle="collapse"` 属性。`data-target="#id"` 属性是对应折叠的内容 (`<div id="demo">`)。

注意: `<a>` 元素上你可以使用 `href` 属性来代替 `data-target` 属性：

实例

```
<a href="#demo" data-toggle="collapse">Collapsible</a>
<div id="demo" class="collapse">
  Lorem ipsum dolor text....
</div>
```

尝试一下 »

默认情况下折叠的内容是隐藏的，你可以添加 `.show` 类让内容默认显示：

实例

```
<div id="demo" class="collapse show">
  Lorem ipsum dolor text...
</div>
```

尝试一下 »

以下实例通过扩展卡片组件来显示简单的手风琴。

注意：使用 `data-parent` 属性来确保所有的折叠元素在指定的父元素下，这样就能实现在一个折叠选项显示时其他选项就隐藏。

实例

```
<div id="accordion">
  <div class="card">
    <div class="card-header">
      <a class="card-link" data-toggle="collapse" href="#collapseOne">
        选项一
      </a>
    </div>
    <div id="collapseOne" class="collapse show" data-parent="#accordion">
      <div class="card-body">
        #1 内容：菜鸟教程 -- 学的不仅是技术，更是梦想！！
      </div>
    </div>
  </div>
  <div class="card">
    <div class="card-header">
      <a class="collapsed card-link" data-toggle="collapse" href="#collapseTwo">
        选项二
      </a>
    </div>
    <div id="collapseTwo" class="collapse" data-parent="#accordion">
      <div class="card-body">
        #2 内容：菜鸟教程 -- 学的不仅是技术，更是梦想！！
      </div>
    </div>
  </div>
  <div class="card">
    <div class="card-header">
      <a class="collapsed card-link" data-toggle="collapse" href="#collapseThree">
        选项三
      </a>
    </div>
    <div id="collapseThree" class="collapse" data-parent="#accordion">
      <div class="card-body">
        #3 内容：菜鸟教程 -- 学的不仅是技术，更是梦想！！
      </div>
    </div>
  </div>
</div>
```

尝试一下 »

☐ Bootstrap4 下拉菜单

Bootstrap4 导航 ☐

☐ 点我分享笔记

反馈/建议

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)[Bootstrap4 折叠](#)[Bootstrap4 导航栏](#)

Bootstrap4 导航

如果你想创建一个简单的水平导航栏，可以在 `` 元素上添加 `.nav` 类，在每个 `` 选项上添加 `.nav-item` 类，在每个链接上添加 `.nav-link` 类：

实例

```
<ul class="nav">
<li class="nav-item">
<a class="nav-link" href="#">Link</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link</a>
</li>
<li class="nav-item">
<a class="nav-link disabled" href="#">Disabled</a>
</li>
</ul>
```

[尝试一下 »](#)

导航对齐方式

`.justify-content-center` 类设置导航居中显示，`.justify-content-end` 类设置导航右对齐。

实例

```
<!-- 导航居中 -->
<ul class="nav justify-content-center">
<!-- 导航右对齐 -->
<ul class="nav justify-content-end">
</div>
```

[尝试一下 »](#)

垂直导航

`.flex-column` 类用于创建垂直导航：

实例

```
<ul class="nav flex-column">
```

[尝试一下 »](#)

选项卡

使用 `.nav-tabs` 类可以将导航转化为选项卡。然后对于选中的选项使用 `.active` 类来标记。

实例

```
<ul class="nav nav-tabs">
<li class="nav-item">
<a class="nav-link active" href="#">Active</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link</a>
</li>
<li class="nav-item">
<a class="nav-link disabled" href="#">Disabled</a>
</li>
</ul>
```

尝试一下 »

胶囊导航

.nav-pills 类可以将导航项设置成胶囊形状。

实例

```
<ul class="nav nav-pills">
<li class="nav-item">
<a class="nav-link active" href="#">Active</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link</a>
</li>
<li class="nav-item">
<a class="nav-link disabled" href="#">Disabled</a>
</li>
</ul>
```

尝试一下 »

导航等宽

.nav-justified 类可以设置导航项齐行等宽显示。

实例

```
<ul class="nav nav-pills nav-justified">..</ul>
<ul class="nav nav-tabs nav-justified">..</ul>
```

尝试一下 »

胶囊下拉菜单

实例

```
<ul class="nav nav-pills">
<li class="nav-item">
<a class="nav-link active" href="#">Active</a>
</li>
<li class="nav-item dropdown">
<a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#">Dropdown</a>
<div class="dropdown-menu">
<a class="dropdown-item" href="#">Link 1</a>
<a class="dropdown-item" href="#">Link 2</a>
<a class="dropdown-item" href="#">Link 3</a>
</div>
</li>
<li class="nav-item">
```

```
<a class="nav-link" href="#">Link</a>
</li>
<li class="nav-item">
<a class="nav-link disabled" href="#">Disabled</a>
</li>
</ul>
```

尝试一下 »

选项卡下拉菜单

实例

```
<ul class="nav nav-tabs">
<li class="nav-item">
<a class="nav-link active" href="#">Active</a>
</li>
<li class="nav-item dropdown">
<a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#">Dropdown</a>
<div class="dropdown-menu">
<a class="dropdown-item" href="#">Link 1</a>
<a class="dropdown-item" href="#">Link 2</a>
<a class="dropdown-item" href="#">Link 3</a>
</div>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link</a>
</li>
<li class="nav-item">
<a class="nav-link disabled" href="#">Disabled</a>
</li>
</ul>
```

尝试一下 »

动态选项卡

如果你要设置选项卡是动态可切换的，可以在每个链接上添加 `data-toggle="tab"` 属性。然后在每个选项对应的内容的上添加 `.tab-pane` 类。

如果你希望有淡入效果可以在 `.tab-pane` 后添加 `.fade` 类：

实例

```
<!-- Nav tabs -->
<ul class="nav nav-tabs">
<li class="nav-item">
<a class="nav-link active" data-toggle="tab" href="#home">Home</a>
</li>
<li class="nav-item">
<a class="nav-link" data-toggle="tab" href="#menu1">Menu 1</a>
</li>
<li class="nav-item">
<a class="nav-link" data-toggle="tab" href="#menu2">Menu 2</a>
</li>
</ul>
<!-- Tab panes -->
<div class="tab-content">
<div class="tab-pane active container" id="home">...</div>
<div class="tab-pane container" id="menu1">...</div>
<div class="tab-pane container" id="menu2">...</div>
</div>
```

尝试一下 »

胶囊状动态选项卡

胶囊状动态选项卡只需要将以上实例的代码中 `data-toggle` 属性设置为 `data-toggle="pill"`：

实例

```
<!-- Nav pills -->
<ul class="nav nav-pills">
<li class="nav-item">
<a class="nav-link active" data-toggle="pill" href="#home">Home</a>
</li>
<li class="nav-item">
<a class="nav-link" data-toggle="pill" href="#menu1">Menu 1</a>
</li>
<li class="nav-item">
<a class="nav-link" data-toggle="pill" href="#menu2">Menu 2</a>
</li>
</ul>
<!-- Tab panes -->
<div class="tab-content">
<div class="tab-pane active container" id="home">...</div>
<div class="tab-pane container" id="menu1">...</div>
<div class="tab-pane container" id="menu2">...</div>
</div>
```

尝试一下 »

Bootstrap4 折叠

Bootstrap4 导航栏

点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号：闽ICP备15012807号-1



首页 HTML CSS JS 本地书签

Bootstrap4 导航

Bootstrap4 表单

Bootstrap4 导航栏

导航栏一般放在页面的顶部。

我们可以使用 `.navbar` 类来创建一个标准的导航栏，后面紧跟：`.navbar-expand-xl|lg|md|sm` 类来创建响应式的导航栏 (大屏幕水平铺开，小屏幕垂直堆叠)。

导航栏上的选项可以使用 `` 元素并添加 `class="navbar-nav"` 类。然后在 `` 元素上添加 `.nav-item` 类，`<a>` 元素上使用 `.nav-link` 类：

实例

```
<!-- 小屏幕上水平导航栏会切换为垂直的 -->
<nav class="navbar navbar-expand-sm bg-light">
<!-- Links -->
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" href="#">Link 1</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link 2</a>
</li>
<li class="nav-item">
```

```
<a class="nav-link" href="#">Link 3</a>
</li>
</ul>
</nav>
```

尝试一下 »

垂直导航栏

通过删除 `.navbar-expand-xl|lg|md|sm` 类来创建垂直导航栏：

实例

```
<!-- 垂直导航栏 -->
<nav class="navbar bg-light">
<!-- Links -->
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" href="#">Link 1</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link 2</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link 3</a>
</li>
</ul>
</nav>
```

尝试一下 »

不同颜色导航栏

可以使用以下类来创建不同颜色导航栏：`.bg-primary`，`.bg-success`，`.bg-info`，`.bg-warning`，`.bg-danger`，`.bg-secondary`，`.bg-dark` 和 `.bg-light`）。

提示: 对于暗色背景需要设置文本颜色为浅色的，对于浅色背景需要设置文本颜色为深色的。

实例

```
<!-- 灰底黑字 -->
<nav class="navbar navbar-expand-sm bg-light navbar-light">
<ul class="navbar-nav">
<li class="nav-item active">
<a class="nav-link" href="#">Active</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link</a>
</li>
<li class="nav-item">
<a class="nav-link disabled" href="#">Disabled</a>
</li>
</ul>
</nav>
<!-- 黑底白字 -->
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">...</nav>
<!-- 蓝底白字 -->
<nav class="navbar navbar-expand-sm bg-primary navbar-dark">...</nav>
```

尝试一下 »

激活和禁用状态: 可以在 `<a>` 元素上添加 `.active` 类来高亮显示选中的选项。`.disabled` 类用于设置该链接是不可点击的。

品牌/Logo

`.navbar-brand` 类用于高亮显示品牌/Logo:

实例

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
<a class="navbar-brand" href="#">Logo</a>
...
</nav>
```

尝试一下 »

如果使用图片，可以使用 `.navbar-brand` 类来设置图片自适应导航栏。

实例

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
<a class="navbar-brand" href="#">

</a>
...
</nav>
```

尝试一下 »

折叠导航栏

通常，小屏幕上我们都会折叠导航栏，通过点击来显示导航选项。

要创建折叠导航栏，可以在按钮上添加 `class="navbar-toggler"`，`data-toggle="collapse"` 与 `data-target="#thetarget"` 类。然后在设置 `class="collapse navbar-collapse"` 类的 `div` 上包裹导航内容（链接），`div` 元素上的 `id` 匹配按钮 `data-target` 的上指定的 `id`:

实例

```
<nav class="navbar navbar-expand-md bg-dark navbar-dark">
<!-- Brand -->
<a class="navbar-brand" href="#">Navbar</a>
<!-- Toggler/collapsibe Button -->
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">
<span class="navbar-toggler-icon"></span>
</button>
<!-- Navbar links -->
<div class="collapse navbar-collapse" id="collapsibleNavbar">
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" href="#">Link</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link</a>
</li>
</ul>
</div>
</nav>
```

尝试一下 »

导航栏使用下拉菜单

导航栏上可以设置下拉菜单：

实例

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
<!-- Brand -->
<a class="navbar-brand" href="#">Logo</a>
<!-- Links -->
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" href="#">Link 1</a>
```



```
</li>
<li class="nav-item">
  <a class="nav-link" href="#">Link 2</a>
</li>
<!-- Dropdown -->
<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle" href="#" id="navbardrop" data-toggle="dropdown">
    Dropdown link
  </a>
  <div class="dropdown-menu">
    <a class="dropdown-item" href="#">Link 1</a>
    <a class="dropdown-item" href="#">Link 2</a>
    <a class="dropdown-item" href="#">Link 3</a>
  </div>
</li>
</ul>
</nav>
```

尝试一下 »

导航栏的表单与按钮

导航栏的表单 `<form>` 元素使用 `class="form-inline"` 类来排版输入框与按钮：

实例

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <form class="form-inline">
    <input class="form-control" type="text" placeholder="Search">
    <button class="btn btn-success" type="submit">Search</button>
  </form>
</nav>
```

尝试一下 »

你也可以使用其他的输入框类，如 `.input-group-addon` 类用于在输入框前添加小标签。

实例

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <form class="form-inline">
    <div class="input-group">
      <span class="input-group-addon">@</span>
      <input type="text" class="form-control" placeholder="Username">
    </div>
  </form>
</nav>
```

尝试一下 »

导航栏文本

使用 `.navbar-text` 类来设置导航栏上非链接文本，可以保证水平对齐，颜色与内边距一样。

实例

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <!-- Links -->
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link" href="#">Link 1</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link 2</a>
    </li>
  </ul>
  <!-- Navbar text-->
  <span class="navbar-text">
    Navbar text
```

```
</span>
</nav>
```

尝试一下 »

固定导航栏

导航栏可以固定在头部或者底部。

我们使用 `.fixed-top` 类来实现导航栏的固定：

实例

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-top">
...
</nav>
```

尝试一下 »

`.fixed-bottom` 类用于设置导航栏固定在底部：

实例

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-bottom">
...
</nav>
```

尝试一下 »

☐ Bootstrap4 导航

Bootstrap4 表单 ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

☐

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

☐ Bootstrap4 自定义表单

Bootstrap4 面包屑导航（Breadcrumb）

面包屑导航是一种基于网站层次信息的显示方式。以博客为例，面包屑导航可以显示发布日期、类别或标签。它们表示当前页面在导航层次结构内的位置，是在用户界面中的一种导航辅助。

Bootstrap 中的面包屑导航是一个简单的带有 `.breadcrumb` class 的无序列表。分隔符会通过 CSS（`bootstrap.min.css`）中的 `::before` 和 `content` 来添加，下面所示的 class 自动被添加：

```
.breadcrumb-item + .breadcrumb-item::before {

    display: inline-block;

    padding-right: 0.5rem;
```

```
color: #6c757d;

content: "/";

}
```

Bootstrap4 面包屑导航实例

```
<ol class="breadcrumb">
<li class="breadcrumb-item active">Home</li>
</ol>
<ol class="breadcrumb">
<li class="breadcrumb-item"><a href="#">Home</a></li>
<li class="breadcrumb-item active">Library</li>
</ol>
<ol class="breadcrumb">
<li class="breadcrumb-item"><a href="#">Home</a></li>
<li class="breadcrumb-item"><a href="#">Library</a></li>
<li class="breadcrumb-item active">Data</li>
</ol>
```

尝试一下 »

我们也可以不用列表形式:

Bootstrap4 面包屑导航实例

```
<nav class="breadcrumb">
<a class="breadcrumb-item" href="#">Home</a>
<a class="breadcrumb-item" href="#">Library</a>
<a class="breadcrumb-item" href="#">Data</a>
<span class="breadcrumb-item active">Bootstrap</span>
</nav>
```

尝试一下 »

☐ Bootstrap4 自定义表单

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号: 闽ICP备15012807号-1



首页 HTML CSS JS 本地书签

☐ Bootstrap4 导航栏

Bootstrap4 表单控件 ☐

Bootstrap4 表单

在本章中,我们将学习如何使用 Bootstrap 创建表单。Bootstrap 通过一些简单的 HTML 标签和扩展的类即可创建出不同样式的表单。

表单元素 `<input>`, `<textarea>`, 和 `<select>` elements 在使用 `.form-control` 类的情况下,宽度都是设置为 100%。

Bootstrap4 表单布局

堆叠表单 (全屏宽度): 垂直方向

内联表单: 水平方向

Bootstrap 提供了两种类型的表单布局:

堆叠表单

以下实例使用两个输入框，一个复选框，一个提交按钮来创建堆叠表单：

实例

```
<form>
<div class="form-group">
<label for="email">Email address:</label>
<input type="email" class="form-control" id="email">
</div>
<div class="form-group">
<label for="pwd">Password:</label>
<input type="password" class="form-control" id="pwd">
</div>
<div class="form-check">
<label class="form-check-label">
<input class="form-check-input" type="checkbox"> Remember me
</label>
</div>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
```

尝试一下 »

内联表单

所有内联表单中的元素都是左对齐的。

注意：在屏幕宽度小于 **576px** 时为垂直堆叠，如果屏幕宽度大于等于 **576px** 时表单元素才会显示在同一个水平线上。

内联表单需要在 `<form>` 元素上添加 `.form-inline` 类。

以下实例使用两个输入框，一个复选框，一个提交按钮来创建内联表单：

实例

```
<form class="form-inline">
<label for="email">Email address:</label>
<input type="email" class="form-control" id="email">
<label for="pwd">Password:</label>
<input type="password" class="form-control" id="pwd">
<div class="form-check">
<label class="form-check-label">
<input class="form-check-input" type="checkbox"> Remember me
</label>
</div>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
```

尝试一下 »

[Bootstrap4 导航栏](#)

Bootstrap4 表单控件 [Bootstrap4 表格](#)

[点我分享笔记](#)

反馈/建议



Bootstrap4 表单控件

Bootstrap4 支持以下表单控件：

- input
- textarea
- checkbox
- radio
- select

Bootstrap Input

Bootstrap 支持所有的 HTML5 输入类型: text, password, datetime, datetime-local, date, month, time, week, number, email, url, search, tel, 以及 col or。

注意：：如果 input 的 type 属性未正确声明，输入框的样式将不会显示。

以下实例使用两个 input 元素，一个是 text，一个是 password：

实例

```
<div class="form-group">
<label for="usr">用户名:</label>
<input type="text" class="form-control" id="usr">
</div>
<div class="form-group">
<label for="pwd">密码:</label>
<input type="password" class="form-control" id="pwd">
</div>
```

尝试一下 »

Bootstrap textarea

以下实例演示了 textarea 的样式。

实例

```
<div class="form-group">
<label for="comment">评论:</label>
<textarea class="form-control" rows="5" id="comment"></textarea>
</div>
```

尝试一下 »

Bootstrap 复选框(checkbox)

复选框用于让用户从一系列预设置的选项中进行选择，可以选一个或多个。

以下实例包含了三个选项。最后一个是禁用的：

实例

```
<div class="form-check">
<label class="form-check-label">
```

```
<input type="checkbox" class="form-check-input" value="">Option 1
</label>
</div>
<div class="form-check">
<label class="form-check-label">
<input type="checkbox" class="form-check-input" value="">Option 2
</label>
</div>
<div class="form-check disabled">
<label class="form-check-label">
<input type="checkbox" class="form-check-input" value="" disabled>Option 3
</label>
</div>
```

尝试一下 »

使用 `.form-check-inline` 类可以让选项显示在同一行上:

实例

```
<div class="form-check form-check-inline">
<label class="form-check-label">
<input type="checkbox" class="form-check-input" value="">Option 1
</label>
</div>
<div class="form-check form-check-inline">
<label class="form-check-label">
<input type="checkbox" class="form-check-input" value="">Option 2
</label>
</div>
<div class="form-check form-check-inline disabled">
<label class="form-check-label">
<input type="checkbox" class="form-check-input" value="" disabled>Option 3
</label>
</div>
```

尝试一下 »

Bootstrap 单选框(Radio)

单选框用于让用户从一系列预设置的选项中进行选择，只能选一个。

以下实例包含了三个选项。最后一个是禁用的：

实例

```
<div class="radio">
<label><input type="radio" name="optradio">Option 1</label>
</div>
<div class="radio">
<label><input type="radio" name="optradio">Option 2</label>
</div>
<div class="radio disabled">
<label><input type="radio" name="optradio" disabled>Option 3</label>
</div>
```

尝试一下 »

使用 `.radio-inline` 类可以让选项显示在同一行上:

实例

```
<label class="radio-inline"><input type="radio" name="optradio">Option 1</label>
<label class="radio-inline"><input type="radio" name="optradio">Option 2</label>
<label class="radio-inline"><input type="radio" name="optradio">Option 3</label>
```

尝试一下 »

Bootstrap select 下拉菜单

当您想让用户从多个选项中进行选择，但是默认情况下只能选择一个选项时，则使用选择框。

以下实例包含了两个下拉菜单：

实例

```
<div class="form-group">
<label for="sel1">下拉菜单:</label>
<select class="form-control" id="sel1">
<option>1</option>
<option>2</option>
<option>3</option>
<option>4</option>
</select>
</div>
```

尝试一下 »

[点我分享笔记](#)

反馈/建议



Bootstrap4 输入框组

我们可以使用 `.input-group` 类来向表单输入框中添加更多的样式，如图标、文本或者按钮。

使用 `.input-group-prepend` 类可以在输入框的的前面添加文本信息，`.input-group-append` 类添加在输入框的后面。

最后，我们还需要使用 `.input-group-text` 类来设置文本的样式。

Bootstrap4 实例

```
<form>
<div class="input-group mb-3">
<div class="input-group-prepend">
<span class="input-group-text">@</span>
</div>
<input type="text" class="form-control" placeholder="Username">
</div>
<div class="input-group mb-3">
<input type="text" class="form-control" placeholder="Your Email">
<div class="input-group-append">
<span class="input-group-text">@runoob.com</span>
</div>
</div>
</form>
```

尝试一下 »

输入框大小

使用 `.input-group-sm` 类来设置小的输入框，`.input-group-lg` 类设置大的输入框：

Bootstrap4 实例

```
<form>
<div class="input-group mb-3 input-group-sm">
<div class="input-group-prepend">
<span class="input-group-text">Small</span>
</div>
<input type="text" class="form-control">
</div>
</form>
<form>
<div class="input-group mb-3">
<div class="input-group-prepend">
<span class="input-group-text">Default</span>
</div>
<input type="text" class="form-control">
</div>
</form>
<form>
<div class="input-group mb-3 input-group-lg">
<div class="input-group-prepend">
<span class="input-group-text">Large</span>
</div>
<input type="text" class="form-control">
</div>
</form>
```

尝试一下 »

多个输入框和文本

Bootstrap4 实例

```
<!-- 多个输入框 -->
<form>
<div class="input-group mb-3">
<div class="input-group-prepend">
<span class="input-group-text">Person</span>
</div>
<input type="text" class="form-control" placeholder="First Name">
<input type="text" class="form-control" placeholder="Last Name">
</div>
</form>
<!-- 多个文本信息 -->
<form>
<div class="input-group mb-3">
<div class="input-group-prepend">
<span class="input-group-text">One</span>
<span class="input-group-text">Two</span>
<span class="input-group-text">Three</span>
</div>
<input type="text" class="form-control">
</div>
</form>
```

尝试一下 »

复选框与单选框

文本信息可以使用复选框与单选框替代：

□

Bootstrap4 实例


```
<div class="input-group mb-3">
<div class="input-group-prepend">
<div class="input-group-text">
<input type="checkbox">
</div>
</div>
<input type="text" class="form-control" placeholder="RUNOOB">
</div>
<div class="input-group mb-3">
<div class="input-group-prepend">
<div class="input-group-text">
<input type="radio">
</div>
</div>
<input type="text" class="form-control" placeholder="GOOGLE">
</div>
```

尝试一下 »

输入框添加按钮组

Bootstrap4 实例

```
<div class="input-group mb-3">
<div class="input-group-prepend">
<button class="btn btn-outline-secondary" type="button">Basic Button</button>
</div>
<input type="text" class="form-control" placeholder="Some text">
</div>
<div class="input-group mb-3">
<input type="text" class="form-control" placeholder="Search">
<div class="input-group-append">
<button class="btn btn-success" type="submit">Go</button>
</div>
</div>
<div class="input-group mb-3">
<input type="text" class="form-control" placeholder="Something clever..">
<div class="input-group-append">
<button class="btn btn-primary" type="button">OK</button>
<button class="btn btn-danger" type="button">Cancel</button>
</div>
</div>
```

尝试一下 »

设置下拉菜单

输入框中添加下拉菜单不需要使用 .dropdown 类。

□

Bootstrap4 实例

```
<div class="input-group mt-3 mb-3">
<div class="input-group-prepend">
<button type="button" class="btn btn-outline-secondary dropdown-toggle" data-toggle="dropdown">
选择网站
</button>
<div class="dropdown-menu">
<a class="dropdown-item" href="https://www.google.com">GOOGLE</a>
<a class="dropdown-item" href="https://www.runoob.com">RUNOOB</a>
<a class="dropdown-item" href="https://www.taobao.com">TAOBAO</a>
</div>
</div>
<input type="text" class="form-control" placeholder="网站地址">
</div>
```

尝试一下 »

输入框组标签

在输入框组通过在输入框组外围的 `label` 来设置标签，标签的 `for` 属性需要与输入框组的 `id` 对应，点击标签后可以聚焦输入框：

Bootstrap4 实例

```
<label for="demo">Write your email here:</label>
<div class="input-group mb-3">
<input type="text" class="form-control" placeholder="Email" id="demo" name="email">
<div class="input-group-append">
<span class="input-group-text">@runoob.com</span>
</div>
</div>
```

尝试一下 »

[点我分享笔记](#)

反馈/建议

Bootstrap4 自定义表单

Bootstrap4 可以自定义一些表单的样式来替换浏览器默认的风格。

自定义复选框

如果要自定义一个复选框，可以设置 `<div>` 为父元素，类为 `.custom-control` 和 `.custom-checkbox`，复选框作为子元素放在该 `<div>` 里头，然后复选框设置为 `type="checkbox"`，类为 `.custom-control-input`。

复选框的文本使用 `label` 标签，标签使用 `.custom-control-label` 类，`label` 的 `for` 属性值需要匹配复选框的 `id`。

Bootstrap4 实例

```
<form>
<div class="custom-control custom-checkbox">
<input type="checkbox" class="custom-control-input" id="customCheck" name="example1">
<label class="custom-control-label" for="customCheck">自定义复选框</label>
</div>
</form>
```

尝试一下 »

自定义单选框

如果要自定义一个单选框，可以设置 `<div>` 为父元素，类为 `.custom-control` 和 `.custom-radio`，复选框作为子元素放在该 `<div>` 里头，然后单选框设置为 `type="radio"`，类为 `.custom-control-input`。

单选框的文本使用 `label` 标签，标签使用 `.custom-control-label` 类，`label` 的 `for` 属性值需要匹配单选框的 `id`。

Bootstrap4 实例

```
<form>
<div class="custom-control custom-radio">
<input type="radio" class="custom-control-input" id="customRadio" name="example1" value="customEx">
<label class="custom-control-label" for="customRadio">自定义单选框</label>
</div>
</form>
```

尝试一下 »

自定义控件显示在同一行

我们可以在外部元素上使用 `.custom-control-inline` 类来包裹自定义表单控件，这样自定义表单控件就能显示在同一行：

Bootstrap4 实例

```
<form>
<div class="custom-control custom-radio custom-control-inline">
<input type="radio" class="custom-control-input" id="customRadio" name="example" value="customEx">
<label class="custom-control-label" for="customRadio">自定义单选框 1</label>
</div>
<div class="custom-control custom-radio custom-control-inline">
<input type="radio" class="custom-control-input" id="customRadio2" name="example" value="customEx">
<label class="custom-control-label" for="customRadio2">自定义单选框 2</label>
</div>
</form>
```

尝试一下 »

自定义选择菜单

创建自定义选择菜单可以在 `<select>` 元素上添加 `.custom-select` 类：

Bootstrap4 实例

```
<form>
<select name="cars" class="custom-select-sm">
<option selected>自定义选择菜单</option>
<option value="Google">Google</option>
<option value="Runoob">Runoob</option>
<option value="Taobao">Taobao</option>
</select>
</form>
```

尝试一下 »

如果我们要设置自定义选择菜单大小，可以使用 `.custom-select-sm`、`.custom-select-lg` 来设置它们的大小：

Bootstrap4 实例

```
<form>
<!-- 小 -->
<select name="cars" class="custom-select-sm">
<option selected>比较小的自定义选择菜单</option>
<option value="Google">Google</option>
<option value="Runoob">Runoob</option>
<option value="Taobao">Taobao</option>
</select>
<!-- 大 -->
<select name="cars" class="custom-select-lg">
<option selected>比较大的自定义选择菜单</option>
<option value="Google">Google</option>
<option value="Runoob">Runoob</option>
<option value="Taobao">Taobao</option>
</select>
</form>
```

尝试一下 »

自定义滑块控件

我们可以在 **input** 为 **type="range"** 的输入框中添加 `.custom-range` 类来设置自定义滑块控件：

Bootstrap4 实例

```
<form>
<label for="customRange">自定义滑块控件</label>
<input type="range" class="custom-range" id="customRange" name="points1">
</form>
```

尝试一下 »

自定义文件上传控件

我们可以在父元素添加 `.custom-file` 类，然后在 **input** 设置为 **type="file"** 并添加 `.custom-file-label`：

上传控件的文本使用 **label** 标签，标签使用 `.custom-file-label` 类，**label** 的 **for** 属性值需要匹配复选框的 **id**。

Bootstrap4 实例

```
<form>
<div class="custom-file">
<input type="file" class="custom-file-input" id="customFile">
<label class="custom-file-label" for="customFile">选择文件</label>
</div>
</form>
```

尝试一下 »

☐ Bootstrap4 输入框组

Bootstrap4 面包屑导航（Breadcrumb） ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

□

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

☐ Bootstrap4 表单控件

Bootstrap4 模态框 ☐

Bootstrap4 轮播

轮播是一个循环的幻灯片：

-
-
-



如何创建轮播

以下实例创建了一个简单的图片轮播效果：

实例

```
<div id="demo" class="carousel slide" data-ride="carousel">
  <!-- 指示符 -->
  <ul class="carousel-indicators">
    <li data-target="#demo" data-slide-to="0" class="active"></li>
    <li data-target="#demo" data-slide-to="1"></li>
    <li data-target="#demo" data-slide-to="2"></li>
  </ul>
  <!-- 轮播图片 -->
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <!-- 左右切换按钮 -->
  <a class="carousel-control-prev" href="#demo" data-slide="prev">
    <span class="carousel-control-prev-icon"></span>
  </a>
  <a class="carousel-control-next" href="#demo" data-slide="next">
    <span class="carousel-control-next-icon"></span>
  </a>
</div>
```

尝试一下 »

轮播图片上添加描述

在每个 `<div class="carousel-item">` 内添加 `<div class="carousel-caption">` 来设置轮播图片的描述文本：

实例

```
<div class="carousel-item">
  
  <div class="carousel-caption">
    <h3>第一张图片描述标题</h3>
    <p>描述文字!</p>
  </div>
</div>
```

尝试一下 »

以上实例中使用的类说明

| 类 | 描述 |
|-----------------------------|--|
| .carousel | 创建一个轮播 |
| .carousel-indicators | 为轮播添加一个指示符，就是轮播图底下的一个个小点，轮播的过程可以显示目前是第几张图。 |
| .carousel-inner | 添加要切换的图片 |
| .carousel-item | 指定每个图片的内容 |
| .carousel-control-prev | 添加左侧的按钮，点击会返回上一张。 |
| .carousel-control-next | 添加右侧按钮，点击会切换到下一张。 |
| .carousel-control-prev-icon | 与 .carousel-control-prev 一起使用，设置左侧的按钮 |
| .carousel-control-next-icon | 与 .carousel-control-next 一起使用，设置右侧的按钮 |
| .slide | 切换图片的过渡和动画效果，如果你不需要这样的效果，可以删除这个类。 |

[Bootstrap4 表单控件](#)

[Bootstrap4 模态框](#)

[点我分享笔记](#)

[反馈/建议](#)

Copyright © 2013-2018 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号: 闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

[Bootstrap4 轮播](#)

[Bootstrap4 提示框](#)

Bootstrap4 模态框

模态框（Modal）是覆盖在父窗体上的子窗体。通常，目的是显示来自一个单独的源的内容，可以在不离开父窗体的情况下有一些互动。子窗体可提供信息交互等。

如何创建模态框

以下实例创建了一个简单的模态框效果：

实例

```
<!-- 按钮：用于打开模态框 -->
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#myModal">
打开模态框
</button>
<!-- 模态框 -->
<div class="modal fade" id="myModal">
<div class="modal-dialog">
<div class="modal-content">
<!-- 模态框头部 -->
<div class="modal-header">
<h4 class="modal-title">模态框头部</h4>
```

```
<button type="button" class="close" data-dismiss="modal">&times;</button>
</div>
<!-- 模态框主体 -->
<div class="modal-body">
模态框内容..
</div>
<!-- 模态框底部 -->
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-dismiss="modal">关闭</button>
</div>
</div>
</div>
</div>
```

尝试一下 »

模态框尺寸

我们可以通过添加 `.modal-sm` 类来创建一个小模态框，`.modal-lg` 类可以创建一个模态框。

尺寸类放在 `<div>` 元素的 `.modal-dialog` 类后：

实例 - 小模态框

```
<div class="modal-dialog modal-sm">
```

尝试一下 »

实例 - 大模态框

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

尝试一下 »

☐ Bootstrap4 轮播

Bootstrap4 提示框 ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

□

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

☐ Bootstrap4 模态框

Bootstrap4 弹出框 ☐

Bootstrap4 提示框

提示框是一个小小的弹窗，在鼠标移动到元素上显示，鼠标移到元素外就消失。

如何创建提示框

通过向元素添加 `data-toggle="tooltip"` 来创建提示框。

`title` 属性的内容为提示框显示的内容：

```
<a href="#" data-toggle="tooltip" title="我是提示内容!">鼠标移动到我这</a>
```

注意: 提示框要写在 jQuery 的初始化代码里: 然后在指定的元素上调用 `tooltip()` 方法。

以下实例可以在文档的任何地方使用提示框:

实例

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

尝试一下 »

指定提示框的位置

默认情况下提示框显示在元素上方。

可以使用 `data-placement` 属性来设定提示框显示的方向: `top`, `bottom`, `left` 或 `right`:

实例

```
<a href="#" data-toggle="tooltip" data-placement="top" title="我是提示内容!">鼠标移动到我这</a>
<a href="#" data-toggle="tooltip" data-placement="bottom" title="我是提示内容!">鼠标移动到我这</a>
<a href="#" data-toggle="tooltip" data-placement="left" title="我是提示内容!">鼠标移动到我这</a>
<a href="#" data-toggle="tooltip" data-placement="right" title="我是提示内容!">鼠标移动到我这</a>
```

尝试一下 »

[Bootstrap4 模态框](#)

[Bootstrap4 弹出框](#)

[点我分享笔记](#)

反馈/建议



[Bootstrap4 提示框](#)

[Bootstrap4 滚动监听](#)

Bootstrap4 弹出框

弹出框控件类似于提示框，它在鼠标点击到元素后显示，与提示框不同的是它可以显示更多的内容。

如何创建弹出框

通过向元素添加 `data-toggle="popover"` 来创建弹出框。

`title` 属性的内容为弹出框的标题，`data-content` 属性显示了弹出框的文本内容:

```
<a href="#" data-toggle="popover" title="弹出框标题" data-content="弹出框内容">多次点我</a>
```

注意: 弹出框要写在 jQuery 的初始化代码里: 然后在指定的元素上调用 `popover()` 方法。

以下实例可以在文档的任何地方使用弹出框:

实例

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

尝试一下 »

指定弹出框的位置

默认情况下弹出框显示在元素右侧。

可以使用 `data-placement` 属性来设定弹出框显示的方向: top, bottom, left 或 right:

实例

```
<a href="#" title="Header" data-toggle="popover" data-placement="top" data-content="Content">点我</a>
<a href="#" title="Header" data-toggle="popover" data-placement="bottom" data-content="Content">点我</a>
<a href="#" title="Header" data-toggle="popover" data-placement="left" data-content="Content">点我</a>
<a href="#" title="Header" data-toggle="popover" data-placement="right" data-content="Content">点我</a>
```

尝试一下 »

关闭弹出框

默认情况下，弹出框在再次点击指定元素后就会关闭，你可以使用 `data-trigger="focus"` 属性来设置在鼠标点击元素外部区域来关闭弹出框：

实例

```
<a href="#" title="取消弹出框" data-toggle="popover" data-trigger="focus" data-content="点击文档的其他地方关闭我">点我</a>
```

尝试一下 »

提示:如果你想实现在鼠标移动到元素上显示，移除后消失的效果，可以使用 `data-trigger="hover"` 属性，并设置值为 "hover":

实例

```
<a href="#" title="Header" data-toggle="popover" data-trigger="hover" data-content="一些内容">鼠标移动到我这</a>
```

尝试一下 »

Bootstrap 滚动监听(Scrollspy)

滚动监听（Scrollspy）插件，即自动更新导航插件，会根据滚动条的位置自动更新对应的导航目标。其基本的实现是随着您的滚动。

- [Section 1](#)
 - [Section 2](#)
 - [Section 3](#)
 - [Section 4](#)
- [Link 1](#) [Link 2](#)

Section 1

Try to scroll this section and look at the navigation bar while scrolling! Try to scroll this section and look at the navigation bar while scrolling!

Try to scroll this section and look at the navigation bar while scrolling! Try to scroll this section and look at the navigation bar while scrolling!

如何创建滚动监听

以下实例演示了如何创建滚动监听：

实例

```
<!-- 可滚动区域 -->
<body data-spy="scroll" data-target=".navbar" data-offset="50">
<!-- The navbar - The <a> elements are used to jump to a section in the scrollable area -->
<nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-top">
...
<ul class="navbar-nav">
<li><a href="#section1">Section 1</a></li>
...
</nav>
<!-- 第一部分内容 -->
<div id="section1">
<h1>Section 1</h1>
<p>Try to scroll this page and look at the navigation bar while scrolling!</p>
</div>
...
</body>
```

尝试一下 »

实例解析

向您想要监听的元素（通常是 `body`）添加 `data-spy="scroll"` 。

然后添加 `data-target` 属性，它的值为导航栏的 `id` 或 `class`（`.navbar`）。这样就可以联系上可滚动区域。

注意可滚动项元素上的 `id`（`<div id="section1">`）必须匹配导航栏上的链接选项（``）。

可选项 `data-offset` 属性用于计算滚动位置时，距离顶部的偏移像素。默认为 10 px。

设置相对定位：使用 `data-spy="scroll"` 的元素需要将其 CSS `position` 属性设置为 `"relative"` 才能起作用。

以下实例设置了垂直滚动监听：

实例

```
<body data-spy="scroll" data-target="#myScrollspy" data-offset="1">
<div class="container-fluid">
<div class="row">
<nav class="col-sm-3 col-4" id="myScrollspy">
<ul class="nav nav-pills flex-column">
<li class="nav-item">
<a class="nav-link active" href="#section1">Section 1</a>
```

```
</li>
...
</ul>
</nav>
<div class="col-sm-9 col-8">
<div id="section1">
<h1>Section 1</h1>
<p>Try to scroll this page and look at the navigation list while scrolling!</p>
</div>
...
</div>
</div>
</div>
</body>
```

尝试一下 »

☐ Bootstrap4 弹出框

Bootstrap4 小工具 ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号: 闽ICP备15012807号-1



首页 HTML CSS JS 本地书签

☐ Bootstrap4 滚动监听

Bootstrap 4 Flex（弹性）布局 ☐

Bootstrap4 小工具

Bootstrap4 提供了一些小工具，可以让我们不用写 CSS 代码就能实现想要的效果。

边框

使用 `border` 类可以添加或移除边框：

实例

```
<span class="border"></span>
<span class="border border-0"></span>
<span class="border border-top-0"></span>
<span class="border border-right-0"></span>
<span class="border border-bottom-0"></span>
<span class="border border-left-0"></span>
```

尝试一下 »

边框颜色

Bootstrap4 提供了一些类来设置边框颜色：

实例

```
<span class="border border-primary"></span>
<span class="border border-secondary"></span>
```

```
<span class="border border-success"></span>
<span class="border border-danger"></span>
<span class="border border-warning"></span>
<span class="border border-info"></span>
<span class="border border-light"></span>
<span class="border border-dark"></span>
<span class="border border-white"></span>
```

尝试一下 »

边框圆角设置

使用 `rounded` 类可以添加圆角边框：

实例

```
<span class="rounded"></span>
<span class="rounded-top"></span>
<span class="rounded-right"></span>
<span class="rounded-bottom"></span>
<span class="rounded-left"></span>
<span class="rounded-circle"></span>
<span class="rounded-0"></span>
```

尝试一下 »

浮动

`.float-right` 类用于设置元素右浮动，`.float-left` 设置元素左浮动，`.clearfix` 类用于清除浮动：

实例

```
<div class="clearfix">
<span class="float-left">左浮动</span>
<span class="float-right">右浮动</span>
</div>
```

尝试一下 »

响应式浮动

我们可以设置浮动 (`.float-*-left|right` - * 为 sm, md, lg 或 xl) 的方向依赖于屏幕的大小：

实例

```
<div class="float-sm-right">在大于小屏幕尺寸上右浮动</div><br>
<div class="float-md-right">在大于中等屏幕尺寸上右浮动</div><br>
<div class="float-lg-right">在大于大屏幕尺寸上右浮动</div><br>
<div class="float-xl-right">在大于超大屏幕尺寸上右浮动</div><br>
<div class="float-none">没有浮动</div>
```

尝试一下 »

居中对齐

使用 `.mx-auto` 类来设置居中对齐：

实例

```
<div class="mx-auto bg-warning" style="width:150px">居中显示</div>
```

尝试一下 »

宽度

元素上使用 `w*` 类 (`.w-25`, `.w-50`, `.w-75`, `.w-100`, `.mw-100`) 来设置宽度：

实例

```
<div class="w-25 bg-warning">宽度 25%</div>
<div class="w-50 bg-warning">宽度 50%</div>
<div class="w-75 bg-warning">宽度 75%</div>
<div class="w-100 bg-warning">宽度 100%</div>
<div class="mw-100 bg-warning">最大宽度 100%</div>
```

尝试一下 »

高度

元素上使用 `h-*` 类 (`.h-25`, `.h-50`, `.h-75`, `.h-100`, `.mh-100`) 来设置高度:

实例

```
<div style="height:200px;background-color:#ddd">
<div class="h-25 bg-warning">高度 25%</div>
<div class="h-50 bg-warning">高度 50%</div>
<div class="h-75 bg-warning">高度 75%</div>
<div class="h-100 bg-warning">高度 100%</div>
<div class="mh-100 bg-warning" style="height:500px">最大高度 100%</div>
</div>
```

尝试一下 »

☐ Bootstrap4 滚动监听

Bootstrap 4 Flex（弹性）布局 ☐

☐ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号: 闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

☐ Bootstrap4 小工具

Bootstrap 4 多媒体对象 ☐

Bootstrap 4 Flex（弹性）布局

Bootstrap 4 通过 `flex` 类来控制页面的布局。

弹性盒子(flexbox)

Bootstrap 3 与 Bootstrap 4 最大的区别就是 Bootstrap 4 使用弹性盒子来布局，而不是使用浮动来布局。

弹性盒子是 CSS3 的一种新的布局模式，更适合响应式的设计，如果你还不了解 `flex`，可以阅读我们的 [CSS3 弹性盒子\(Flex Box\) 教程](#)

注意: IE9 及其以下版本不支持弹性盒子，所以如果你需要兼容 IE8-9，请使用 Bootstrap 3。

以下实例使用 `d-flex` 类创建一个弹性盒子容器，并设置三个弹性子元素：

实例

```
<div class="d-flex p-3 bg-secondary text-white">
```

```
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary">Flex item 3</div>
</div>
```

尝试一下 »

创建显示在同一行上的弹性盒子容器可以使用 **d-inline-flex** 类:

实例

```
<div class="d-inline-flex p-3 bg-secondary text-white">
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary">Flex item 3</div>
</div>
```

尝试一下 »

水平方向

.flex-row 可以设置弹性子元素水平显示, 这是默认的。

使用 **.flex-row-reverse** 类用于设置右对齐显示, 即与 **.flex-row** 方向相反。

实例

```
<div class="d-flex flex-row bg-secondary">
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary">Flex item 3</div>
</div>
<div class="d-flex flex-row-reverse bg-secondary">
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary">Flex item 3</div>
</div>
```

尝试一下 »

垂直方向

.flex-column 类用于设置弹性子元素垂直方向显示, **.flex-column-reverse** 用于翻转子元素:

实例

```
<div class="d-flex flex-column">
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary">Flex item 3</div>
</div>
<div class="d-flex flex-column-reverse">
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary">Flex item 3</div>
</div>
```

尝试一下 »

内容排列

.justify-content-* 类用于修改弹性子元素的排列方式, * 号允许的值有: **start** (默认), **end**, **center**, **between** 或 **around**:

实例

```
<div class="d-flex justify-content-start">...</div>
<div class="d-flex justify-content-end">...</div>
<div class="d-flex justify-content-center">...</div>
<div class="d-flex justify-content-between">...</div>
```

```
<div class="d-flex justify-content-around">...</div>
```

尝试一下 »

等宽

`.flex-fill` 类强制设置各个弹性子元素的宽度是一样的:

实例

```
<div class="d-flex">
<div class="p-2 bg-info flex-fill">Flex item 1</div>
<div class="p-2 bg-warning flex-fill">Flex item 2</div>
<div class="p-2 bg-primary flex-fill">Flex item 3</div>
</div>
```

尝试一下 »

扩展

`.flex-grow-1` 用于设置子元素使用剩下的空间。以下实例中前面两个子元素只设置了它们所需要的空间，最后一个获取剩余空间。：

实例

```
<div class="d-flex">
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary flex-grow-1">Flex item 3</div>
</div>
```

尝试一下 »

提示: 使用 `.flex-shrink-1` 用于设置子元素的收缩规则。

排序

`.order` 类可以设置弹性子元素的排序，从 `.order-1` 到 `.order-12`，数字越低权重越高(`.order-1` 排在 `.order-2` 之前)：

实例

```
<div class="d-flex bg-secondary">
<div class="p-2 bg-info order-3">Flex item 1</div>
<div class="p-2 bg-warning order-2">Flex item 2</div>
<div class="p-2 bg-primary order-1">Flex item 3</div>
</div>
```

尝试一下 »

外边距

`.mr-auto` 类可以设置子元素右边边距为 `auto`，即 `margin-right: auto!important;`，`.ml-auto` 类可以设置子元素左外边距为 `auto`，即 `margin-left: auto!important;`：

实例

```
<div class="d-flex bg-secondary">
<div class="p-2 mr-auto bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary">Flex item 3</div>
</div>
<div class="d-flex bg-secondary">
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 ml-auto bg-primary">Flex item 3</div>
</div>
```

尝试一下 »

包裹

弹性容器中包裹子元素可以使用以下三个类：`.flex-nowrap` (默认), `.flex-wrap` 或 `.flex-wrap-reverse`。设置 `flex` 容器是单行或者多行。

实例

```
<div class="d-flex flex-wrap">..<div class="d-flex flex-wrap-reverse">..<div class="d-flex flex-nowrap">..
```

尝试一下 »

内容对齐

我们可以使用 `.align-content-*` 来控制在垂直方向上如何去堆叠子元素，包含的值有：`.align-content-start` (默认), `.align-content-end`, `.align-content-center`, `.align-content-between`, `.align-content-around` 和 `.align-content-stretch`。

这些类在只有一行的弹性子元素中是无效的。

实例

```
<div class="d-flex flex-wrap align-content-start">..<div class="d-flex flex-wrap align-content-end">..<div class="d-flex flex-wrap align-content-center">..<div class="d-flex flex-wrap align-content-around">..<div class="d-flex flex-wrap align-content-stretch">..
```

尝试一下 »

子元素对齐

如果要设置单行的子元素对齐可以使用 `.align-items-*` 类来控制，包含的值有：`.align-items-start`, `.align-items-end`, `.align-items-center`, `.align-items-baseline`, 和 `.align-items-stretch` (默认)。

实例

```
<div class="d-flex flex-wrap align-content-start">..<div class="d-flex flex-wrap align-content-end">..<div class="d-flex flex-wrap align-content-center">..<div class="d-flex flex-wrap align-content-around">..<div class="d-flex flex-wrap align-content-stretch">..
```

尝试一下 »

指定子元素对齐

如果要设置指定子元素对齐对齐可以使用 `.align-self-*` 类来控制，包含的值有：`.align-self-start`, `.align-self-end`, `.align-self-center`, `.align-self-baseline`, 和 `.align-self-stretch` (默认)。

实例

```
<div class="d-flex bg-light" style="height:150px">  
<div class="p-2 border">Flex item 1</div>  
<div class="p-2 border align-self-start">Flex item 2</div>  
<div class="p-2 border">Flex item 3</div>  
</div>
```

尝试一下 »

响应式 flex 类

我们可以根据不同的设备，设置 `flex` 类，从而实现页面响应式布局，以下表格中的 * 号可以的值有：`sm`, `md`, `lg` 或 `xl`, 对应的是小型设备、中型设备，大型设备，超大型设备。

| 类 | 实例 | 实现 |
|---|----|----|
|---|----|----|

| | | |
|----------------------------|---------------------------------------|------|
| 弹性容器 | | |
| .d-*-flex | 根据不同的屏幕设备创建弹性盒子容器 | 尝试一下 |
| .d-*-inline-flex | 根据不同的屏幕设备创建行内弹性盒子容器 | 尝试一下 |
| 方向 | | |
| .flex-*-row | 根据不同的屏幕设备在水平方向显示弹性子元素 | 尝试一下 |
| .flex-*-row-reverse | 根据不同的屏幕设备在水平方向显示弹性子元素，且右对齐 | 尝试一下 |
| .flex-*-column | 根据不同的屏幕设备在垂直方向显示弹性子元素 | 尝试一下 |
| .flex-*-column-reverse | 根据不同的屏幕设备在垂直方向显示弹性子元素，且方向相反 | 尝试一下 |
| 内容对齐 | | |
| .justify-content-*-start | 根据不同屏幕设备在开始位置显示弹性子元素 (左对齐) | 尝试一下 |
| .justify-content-*-end | 根据不同屏幕设备在尾部显示弹性子元素 (右对齐) | 尝试一下 |
| .justify-content-*-center | 根据不同屏幕设备在 flex 容器中居中显示子元素 | 尝试一下 |
| .justify-content-*-between | 根据不同屏幕设备使用 " between " 显示弹性子元素 | 尝试一下 |
| .justify-content-*-around | 根据不同屏幕设备使用 " around " 显示弹性子元素 | 尝试一下 |
| 等宽 | | |
| .flex-*-fill | 根据不同的屏幕设备强制等宽 | 尝试一下 |
| 扩展 | | |
| .flex-*-grow-0 | 不同的屏幕设备不设置扩展 | |
| .flex-*-grow-1 | 不同的屏幕设备设置扩展 | |
| 收缩 | | |
| .flex-*-shrink-0 | 不同的屏幕设备不设置收缩 | |
| .flex-*-shrink-1 | 不同的屏幕设备设置收缩 | |
| 包裹 | | |
| .flex-*-nowrap | 不同的屏幕设备不设置包裹元素 | 尝试一下 |
| .flex-*-wrap | 不同的屏幕设备设置包裹元素 | 尝试一下 |
| .flex-*-wrap-reverse | 不同的屏幕设备反转包裹元素 | 尝试一下 |

内容排列

| | | |
|--|---------------------------|------|
| <code>.align-content-*--start</code> | 根据不同屏幕设备在起始位置堆叠元素 | 尝试一下 |
| <code>.align-content-*--end</code> | 根据不同屏幕设备在结束位置堆叠元素 | 尝试一下 |
| <code>.align-content-*--center</code> | 根据不同屏幕设备在中间位置堆叠元素 | 尝试一下 |
| <code>.align-content-*--around</code> | 根据不同屏幕设备，使用 "around" 堆叠元素 | 尝试一下 |
| <code>.align-content-*--stretch</code> | 根据不同屏幕设备，通过伸展元素来堆叠 | 尝试一下 |

元素对齐

| | | |
|---------------------------------------|--------------------------|------|
| <code>.align-items-*--start</code> | 根据不同屏幕设备，让元素在头部显示在同一行。 | 尝试一下 |
| <code>.align-items-*--end</code> | 根据不同屏幕设备，让元素在尾部显示在同一行。 | 尝试一下 |
| <code>.align-items-*--center</code> | 根据不同屏幕设备，让元素在中间位置显示在同一行。 | 尝试一下 |
| <code>.align-items-*--baseline</code> | 根据不同屏幕设备，让元素在基线上显示在同一行。 | 尝试一下 |
| <code>.align-items-*--stretch</code> | 根据不同屏幕设备，让元素延展高度并显示在同一行。 | 尝试一下 |

单独一个子元素的对齐方式

| | | |
|--------------------------------------|-------------------------|------|
| <code>.align-self-*--start</code> | 据不同屏幕设备，让单独一个子元素显示在头部。 | 尝试一下 |
| <code>.align-self-*--end</code> | 据不同屏幕设备，让单独一个子元素显示在尾部 | 尝试一下 |
| <code>.align-self-*--center</code> | 据不同屏幕设备，让单独一个子元素显示在居中位置 | 尝试一下 |
| <code>.align-self-*--baseline</code> | 据不同屏幕设备，让单独一个子元素显示在基线位置 | 尝试一下 |
| <code>.align-self-*--stretch</code> | 据不同屏幕设备，延展一个单独子元素 | 尝试一下 |

Bootstrap4 小工具

Bootstrap 4 多媒体对象

点我分享笔记

反馈/建议



Bootstrap 4 多媒体对象

Bootstrap 提供了很好的方式来处理多媒体对象（图片或视频）和内容的布局。应用场景有博客评论、微博等：

基础多媒体对象

要创建一个多媒体对象，可以在容器元素上添加 `.media` 类，然后将多媒体内容放到子容器上，子容器需要添加 `.media-body` 类，然后添加外边距，内边距等效果：

实例

```
<div class="media border p-3">

<div class="media-body">
<h4>菜鸟教程</h4>
<p>学的不仅是技术，更是梦想！！</p>
</div>
</div>
```

[尝试一下 »](#)

多媒体对象嵌套

多媒体对象可以多个嵌套（一个多媒体对象中包含另外一个多媒体对象）

要嵌套多媒体对象，可以把新的 `.media` 容器放到 `.media-body` 容器中：

实例

```
<div class="media border p-3">

<div class="media-body">
<h4>菜鸟教程</h4>
<p>学的不仅是技术，更是梦想！！</p>
<div class="media p-3">

<div class="media-body">
<h4>菜鸟教程</h4>
<p>学的不仅是技术，更是梦想！！</p>
</div>
</div>
</div>
</div>
```

[尝试一下 »](#)

多媒体对象图片显示在右边

如果你想将头像图片显示在右侧，可以在 `.media-body` 容器后添加图片：

实例

```
<div class="media border p-3">
<div class="media-body">
<h4>菜鸟教程</h4>
<p>学的不仅是技术，更是梦想！！</p>
</div>

</div>
```

[尝试一下 »](#)

定位多媒体图片位置

我们可以使用 `align-self*` 相关类来设置多媒体对象的图片显示位置：

实例

```
<!-- 头部 -->
<div class="media">

<div class="media-body">
<h4>头部 -- 菜鸟教程</h4>
<p>学的不仅是技术，更是梦想！！</p>
</div>
</div>
<!-- 居中 -->
<div class="media">

<div class="media-body">
<h4>居中 -- 菜鸟教程</h4>
<p>学的不仅是技术，更是梦想！！</p>
</div>
</div>
<!-- 底部 -->
<div class="media">

<div class="media-body">
<h4>底部 -- 菜鸟教程</h4>
<p>学的不仅是技术，更是梦想！！</p>
</div>
</div>
```

尝试一下 »

☐ 点我分享笔记

反馈/建议



Bootstrap4 创建一个网页

接下来我们通过 Bootstrap4 来创建一个简单的响应式网页。
在学习之前我们可以先看下效果：https://www.runoob.com/try/demo_source/bootstrap3-makewebsite.htm

我的第一个 Bootstrap 4 页面

重置浏览器窗口大小查看效果！

导航

- [链接](#)
- [链接](#)
- [链接](#)

关于我

我的照片：

这边插入图像

关于我的介绍..

HTML 代码

```
<div class="jumbotron text-center" style="margin-bottom:0">
<h1>我的第一个 Bootstrap 4 页面</h1>
<p>重置浏览器窗口大小查看效果！</p>
</div>
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
<a class="navbar-brand" href="#">导航</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="collapsibleNavbar">
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" href="#">链接</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">链接</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">链接</a>
</li>
</ul>
</div>
</nav>
<div class="container" style="margin-top:30px">
<div class="row">
<div class="col-sm-4">
<h2>关于我</h2>
<h3>我的照片:</h3>
<div class="fakeimg">这边插入图像</div>
<p>关于我的介绍..</p>
<h3>一些链接</h3>
<p>说明文本</p>
<ul class="nav nav-pills flex-column">
<li class="nav-item">
<a class="nav-link active" href="#">激活状态</a>
```

```
</li>
<li class="nav-item">
<a class="nav-link" href="#">链接</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">链接</a>
</li>
<li class="nav-item">
<a class="nav-link disabled" href="#">禁用</a>
</li>
</ul>
<hr class="d-sm-none">
</div>
<div class="col-sm-8">
<h2>标题</h2>
<h5>副标题</h5>
<div class="fakeimg">图像</div>
<p>一些文本..</p>
<p>菜鸟教程，学的不仅是技术，更是梦想！！！菜鸟教程，学的不仅是技术，更是梦想！！！菜鸟教程，学的不仅是技术，更是梦想！！</p>
<br>
<h2>标题</h2>
<h5>副标题</h5>
<div class="fakeimg">图像</div>
<p>一些文本..</p>
<p>菜鸟教程，学的不仅是技术，更是梦想！！！菜鸟教程，学的不仅是技术，更是梦想！！</p>
</div>
</div>
<div class="jumbotron text-center" style="margin-bottom:0">
<p>底部内容</p>
</div>
```

尝试一下 »

☐ Bootstrap 4 多媒体对象

Bootstrap4 输入框组 ☐

☐ 点我分享笔记

反馈/建议