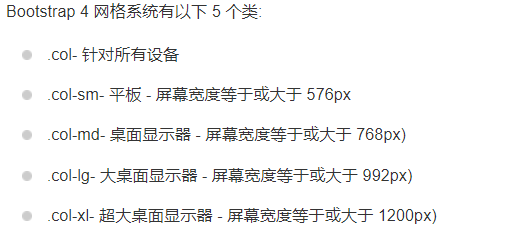
Bootstrap4学习笔记

## [容器](https://www.runoob.com/bootstrap4/bootstrap4-install.html)

* div.container：用于固定宽度并支持响应式布局的容器；
* div.container-fluid：用于 100% 宽度，占据全部视口（viewport）的容器；

## [网格系统（栅格布局）](https://www.runoob.com/bootstrap4/bootstrap4-grid-basic.html#content)

### 不同设备的应用区别





### 偏移

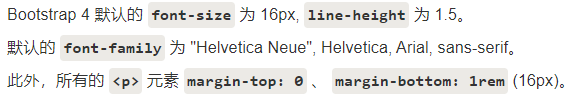
偏移列通过 **offset-\*-\*** 类来设置。第一个星号( \* )可以是 **sm、md、lg、xl**，表示屏幕设备类型，第二个星号( \* )可以是 **1** 到 **11** 的数字。为了在大屏幕显示器上使用偏移，请使用 **.offset-md-\*** 类。这些类会把一个列的左外边距（margin）增加 **\*** 列，其中 **\*** 范围是从 **1** 到 **11**。例如：.offset-md-4 是把.col-md-4 往右移了四列格；

### 使用注意事项

* 网格每一行（div.row）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；
* 对于不同的设备，可以添加多个class列宽选择器，这样，当屏幕宽度发生改变时，就会做出不同的响应；

## [Bootstrap4 文字排版](https://www.runoob.com/bootstrap4/bootstrap4-typography.html#content)

### 默认排版



### 标题和普通文本

一般放置在div.container容器中；

* 更改了所有标题（h1-h6）的默认样式；
* 针对标题定义了.display-1、.display-2、.display-3、.display-4，共四个字体样式，可以输出更大更粗的字体样式；
* 在标题中嵌套small标签（父元素的85%）用于创建副标题；
* 在普通文本中，嵌套mark标签用于表示高亮文本；
* 在普通文本中，嵌套abbr标签表示缩略语，设置title属性可提示详细信息；
* <blockquote class=”blockquote”><p>content</p><footer class=”blockquote-footer”>source</footer></blockquote>这个标签结构用于引用内容并附上出处；
* 更改了自定义列表dl+dd+dt的默认样式；
* 在普通文本中，嵌套code标签用于表示部分代码片段；
* 在普通文本中，嵌套kbd标签用于表示某些系统命令；
* <pre>code</pre>标签用于插入代码；
* 在标题、普通文本中，设置class=”font-weight-bold / font-weight-normal / font-weight-light | font-italic”，可对文本进行加粗、正常化、字体变细、让字体变成斜体；
* 在段落中，添加class = “lead”可以使段落更加突出；
* 添加class = ”text-left / text-center / text-right”，可以设置段落的靠左、居中、靠右对齐；
* 添加class = “text-justify”可使文本在超出屏幕范围时，自动换行；添加class = “text-nowrap”可使文本在超出屏幕范围时，不换行；
* 添加class = “text-lowercase / text-uppercase / text-capitalize”，分别实现所有字母小写、所有字母大写、所有单词首字母大写；
* 在abbr标签中（缩略语），添加class = ”initialism”，可以设置缩略语字体大小更小（font-size:90%），所有字母大写；
* 在列表（ul、ol）中，添加class = “unstyled”，可以清楚列表的默认样式，但是不能清楚列表项中嵌套的列表样式；
* 在列表（ul、ol）中，添加class = “list-inline”，在其子项li中，添加class = “list-inline-item”，可将所有列表项放在一行；
* 在代码标签（pre）中，添加class = “pre-scrollable”，使 <pre> 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（貌似一直会有显示滚动条的地方）；

## [Bootstrap4 颜色](https://www.runoob.com/bootstrap4/bootstrap4-colors.html#content)

### 文本颜色

* 添加class = “text-muted”，设置文本颜色呈“柔和的文本”；
* 添加class = “text-primary”，设置文本颜色呈“重要的文本”；
* 添加class = “text-success”，设置文本颜色呈“执行成功的文本”；
* 添加class = “text-info”，设置文本颜色呈“代表一些提示信息的文本”；
* 添加class = “text-warning”，设置文本颜色呈“警告的文本”；
* 添加class = “text-danger”，设置文本颜色呈“危险操作的文本”；
* 添加class = “text-secondary”，设置副标题的文本颜色；
* 添加class = “text-dark”，设置文本颜色呈“深灰色文本”；
* 添加class = “text-light”，设置文本颜色呈“浅灰色文本”；
* 添加class = “text-white”，设置文本颜色呈“白色文本”；

对于链接设置相应文本颜色对鼠标有相应的响应；

### 背景颜色

通常需要与text-\*一起配合使用（常用的是text-white）;

* 添加class = “bg-primary”，设置重要信息的背景颜色
* 添加class = “bg-success”，设置执行成功提示信息的背景颜色；
* 添加class = “bg-info”，设置提示信息的背景颜色；
* 添加class = “bg-warning”，设置警告信息的背景颜色；
* 添加class = “bg-danger”，设置危险信息的背景颜色；
* 添加class = “bg-secondary”，设置副标题的背景颜色；
* 添加class = “bg-dark”，设置深灰色的背景颜色；
* 添加class = “bg-light”，设置浅灰色的背景颜色；

## [表格](https://www.runoob.com/bootstrap4/bootstrap4-tables.html#content)

table的机构为table>thead(>tr>th)+tbody(>tr>td)；

* table标签添加class = “table”，设置基础表格样式；
* table标签添加class = “table-striped”，可在<tbody>内的行上看见条纹；
* table标签添加class = “table-bordered”，可为表格添加边框；
* table标签添加class = “table-hover”，可为<tbody>添加鼠标悬停效果（灰色背景）；
* table标签添加class = “table-dark”，可为表格添加黑色背景，同时可在此基础上添加table-hover、table-striped、table-hover会有对应的效果；
* tr标签添加class = “table-primary”，该行背景呈蓝色，指定这是一个重要的操作；
* tr标签添加class = “table-success”，该行背景呈绿色，指定这是一个允许执行的操作；
* tr标签添加class = “table-danger”，该行背景呈红色，指定这是一个危险的操作；
* tr标签添加class = “table-info”，该行背景呈浅蓝色，表示内容已变更；
* tr标签添加class = “table-warning”，该行背景呈橘色，表示需要注意的操作；
* tr标签添加class = “table-active”，该行背景呈灰色，用于鼠标悬停效果；
* tr标签添加class = “table-secondary”，该行背景呈灰色，表示内容不怎么重要；
* tr标签添加class = “table-light”，该行背景呈浅灰色，可以是表格行的背景；
* tr标签添加class = “table-dark”，该行背景呈深灰色，可以是表格行的背景；
* thead标签添加class = “thead-dark”，给表头添加黑色背景；
* thead标签添加class = “thead-light”，给表头添加灰色背景；
* table标签添加class = “table-sm”，用于通过减少内边距来设置较小的表格；
* table标签父元素添加class = “table-responsive”，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）；针对不同的设置可以做不同的设置：



## [图像形状](https://www.runoob.com/bootstrap4/bootstrap4-images.html#content)

* img标签添加class = “rounded”，可使图片出现小圆角（本身img就可以设置border-radius）；
* img标签添加class = “rounded-circle”，可使图片呈（椭）圆形（取决于图片的长宽是否相等）展示；
* img标签添加class = “img-thumbnail”，用于设置图片缩略图(图片有边框)；
* img标签添加class = “float-left/float-right”，用于设置图片左/右对齐；
* img标签添加class = “img-fluid”，用于设置响应式图片；

## [jumbotron（超大屏幕）](https://www.runoob.com/bootstrap4/bootstrap4-jumbotron.html#content)

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；

* 在容器标签中添加class = “jumbotron”，会创建一个大的灰色背景框（圆角）；
* 全屏幕jumbotron：在已添加class = ”jumbotron”的容器中再添加class = “jumbotron-fluid”； 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；

## [信息提示框](https://www.runoob.com/bootstrap4/bootstrap4-alerts.html#content)

信息提示框，其实此处表现形式大概如此：



* 在容器（div）中，添加class = “alert alert-success”，指定操作成功的提示信息；
* 在容器（div）中，添加class = “alert alert-info”，指定值得注意的信息；
* 在容器（div）中，添加class = “alert alert-warning”，设置警告信息；
* 在容器（div）中，添加class = “alert alert-danger”，指定操作失败的提示信息；
* 在容器（div）中，添加class = “alert alert-primary”，指定非常重要的信息；
* 在容器（div）中，添加class = “alert alert-secondary”，指定一些不重要的信息；
* 在容器（div）中，添加class = “alert alert-dark”，深灰色提示框；
* 在容器（div）中，添加class = “alert alert-light”，浅灰色提示框；

对于信息提示框中的链接，表现形式如下：



需要在容器中的a标签添加class = “alert-link”;

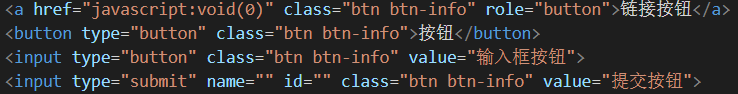
* 设置可关闭的提示框：在容器（div）中添加.alert-dismissible类，然后在关闭按钮的链接上添加class = “close”和data-dismiss = “alert”类来设置提示框的关闭操作，关闭符号可用&times;；
* 对关闭按钮，在父容器中添加.fade和.show类用于设置提示框在关闭时的淡出和淡入的效果；

## [按钮](https://www.runoob.com/bootstrap4/bootstrap4-buttons.html#content)

以下按钮类都是在<button type=”button”>的基础之上的：

* 添加class = “btn”，表示基本按钮；
* 添加class = “btn btn-primary”，表示主要按钮；
* 添加class = “btn btn-secondary”，表示次要按钮；
* 添加class = “btn btn-success”，表示“成功”；
* 添加class = “btn btn-info”，表示“信息”；
* 添加class = “btn btn-warning”，表示“警告”；
* 添加class = “btn btn-danger”，表示“危险”；
* 添加class = “btn btn-dark”，表示深色按钮；
* 添加class = “btn btn-light”，表示浅色按钮；
* 添加class = “btn btn-link”，表示链接按钮；

**按钮类可以用于<a>、<button>或input元素上；**



### 按钮设置边框

* 添加class = “btn btn-outline-primary”，设置主要边框按钮；
* 添加class = “btn btn-outline-secondary”，设置次要边框按钮；
* 添加class = “btn btn-outline-success”，设置“成功”边框按钮；
* 添加class = “btn btn-outline-info”，设置“信息”边框按钮;
* 添加class = “btn btn-outline-warning”，设置“警告”边框按钮；
* 添加class = “btn btn-outline-danger”，设置“危险”边框按钮；
* 添加class = “btn btn-outline-dark”，设置深色边框按钮；
* 添加class = “btn btn-outline-light text-dark”，设置浅色边框按钮（添加字体颜色，按钮应用场景可能看不见）；

### 设置按钮大小

在已设置按钮样式后(什么都不添加表示默认大小)：

* 添加class = “btn-lg”，大号按钮(比默认大小大)；
* 添加class = “btn-sm”，小号按钮(比默认大小小)；

### 块级按钮

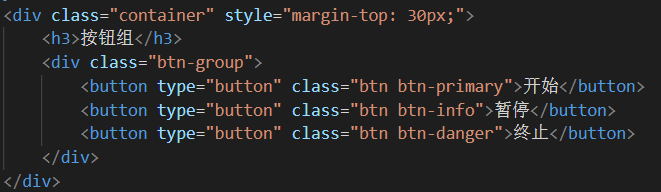
在已设置按钮样式后，添加class = “btn-block”，按钮会转换成块级元素；

### 激活和禁用的按钮

* 添加class = “active”，激活“按钮”；
* button、input添加disabled属性，禁用按钮，a标签不支持这种方法；
* a添加class = “disabled”，a标签特有的禁用按钮方式；

## [按钮组](https://www.runoob.com/bootstrap4/bootstrap4-button-groups.html#content)

bootstrap4 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，例如：





一下是一些设置：

* 在按钮组的容器中，我们可以使用 .btn-group-lg | sm 类来设置按钮组的大小（不写则是默认大小）；
* 在按钮组的容器中，用.btn-group-vertical替换.btn-group即可设置垂直按钮组；

### 内置按钮组及下拉菜单

不知道什么原因，必须引用在线CDN资源文件才可以出现下拉菜单；

<script src="https://cdn.staticfile.org/popper.js/1.15.0/umd/popper.min.js"></script>

<script src="https://cdn.staticfile.org/twitter-bootstrap/4.3.1/js/bootstrap.min.js"></script>





### 拆分按钮下拉菜单





### 垂直按钮组及下拉菜单





## 徽章（Badges）

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 **<span>** 元素上即可。 徽章可以根据父元素的大小的变化而变化。例如

<h1>测试标题 <span class="badge badge-secondary">New</span></h1>



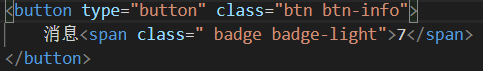
* 同样的，可以在添加.badge类后添加.badge-primary / .badge-secondary / .badge-success / .badge-danger / .badge-warning / .badge-info / .badge-light / .badge-light / .badge / dark设置指定意义的颜色：



* 添加.badge-pill类，设置徽章成药丸状；



* 将徽章嵌入到元素中，例如：button;

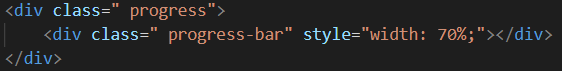




## [进度条](https://www.runoob.com/bootstrap4/bootstrap4-progressbars.html#content)

### 最基本的进度条

在div.progress中嵌套div.progress-bar，并用内联样式设置div.progress-bar的宽度；



下面所有结构都是基于此，先看看基本样式的效果：



### 设置进度的高度

在div.progress中用内联样式直接设置高度即可，默认高度16px;

### 在进度条中添加文本

直接在div.progress-bar中嵌套文本即可；效果如下：



### 设置进度条的颜色

直接在div.progress-bar中添加.bg-success/.bg-info/.bg-warning/.bg-danger等背景颜色类。

### 设置进度条有条纹样式

直接在div.progress-bar中添加.progress-bar-striped类，效果如下：



### 设置进度条有动画样式

直接在div.progress-bar中添加.progress-bar-animated类，div.progress-bar必须添加了.progress-bar-striped类才有效果。

### 设置混合色彩的进度条

通常div.progress中只嵌套了一个div.progress-bar，混合色彩其实就是嵌套多个div.progress-bar，每个div.progress-bar定义了不同色彩而已。



## 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。要创建一个基本的分页可以在 **<ul>** 元素上添加 **.pagination** 类。然后在 **<li>** 元素上添加 **.page-item** 类。然后在li中的a元素上添加.page-link类。最后生成的其实是一个分页导航栏（可在a元素设置href和target——目标元素）。效果如下（和百度搜索结果最下方的导航类似）：



### 设置当前页页码状态

在a元素上添加.active类：



### 不可点击的分页链接

在a元素上添加.disabled类：



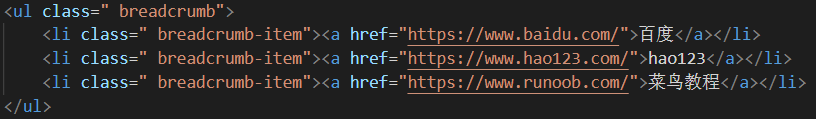
### 分页显示大小

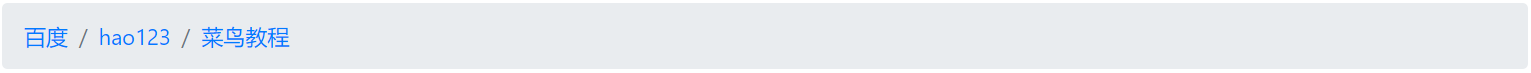
在div.pagination元素上添加.pagination-sm/.paginantion-lg类：



### 面包屑导航

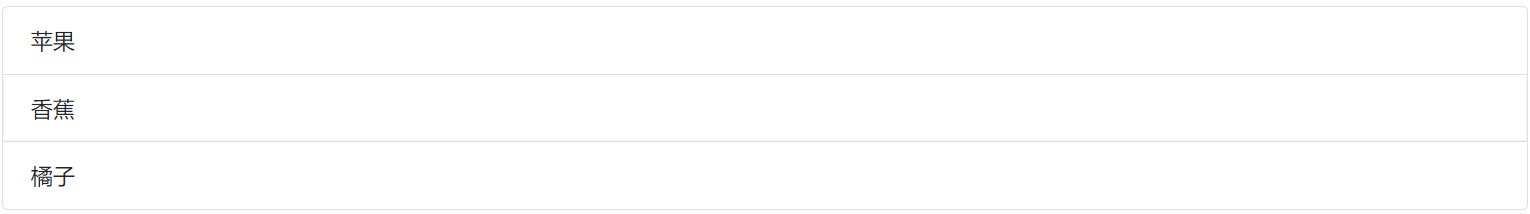
**.breadcrumb** 和 **.breadcrumb-item** 类用于设置面包屑导航：





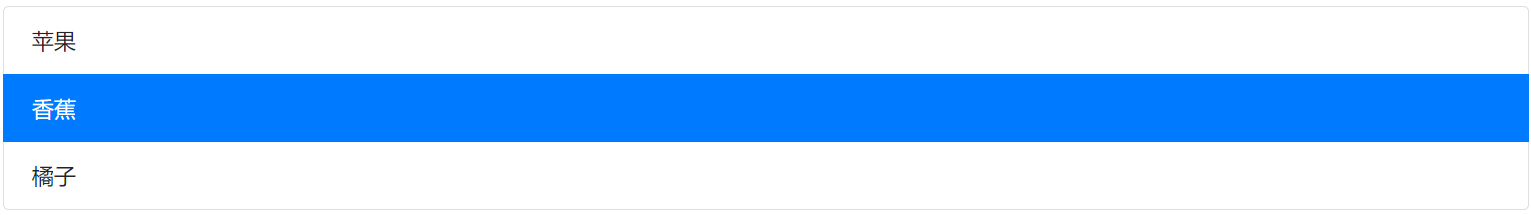
## [列表组](https://www.runoob.com/bootstrap4/bootstrap4-list-groups.html#content)

大部分基础列表组都是无序的, 要创建列表组，可以在 **<ul>** 元素上添加 **.list-group** 类, 在 **<li>** 元素上添加 **.list-group-item** 类:



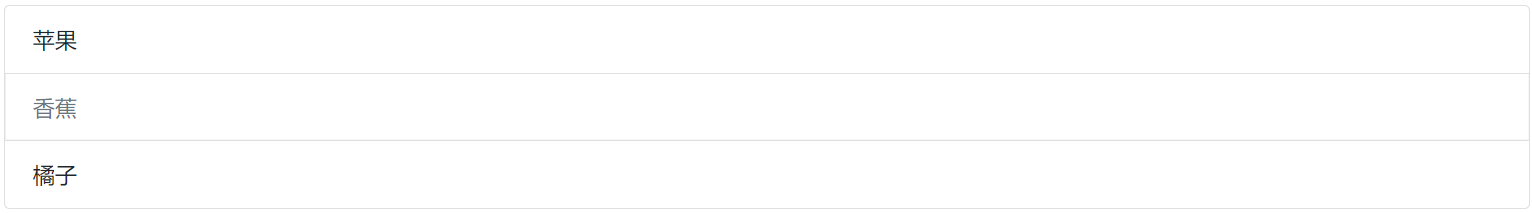
### 激活状态的列表项

通过添加 **.active** 类来设置激活状态的列表项：



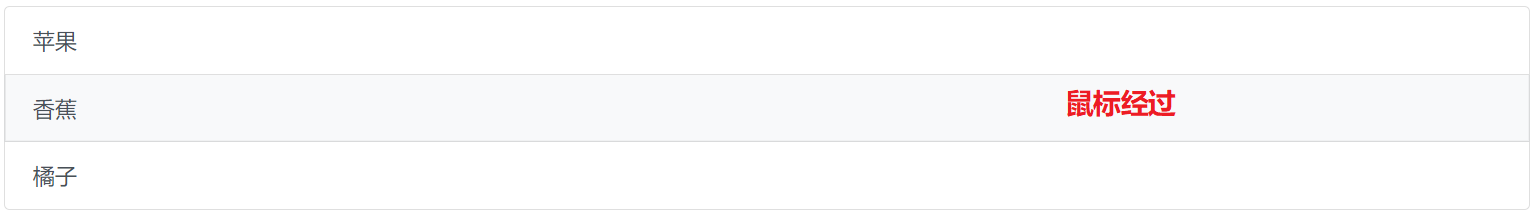
### 禁用的列表项

**.disabled** 类用于设置禁用的列表项:



### 链接列表项

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

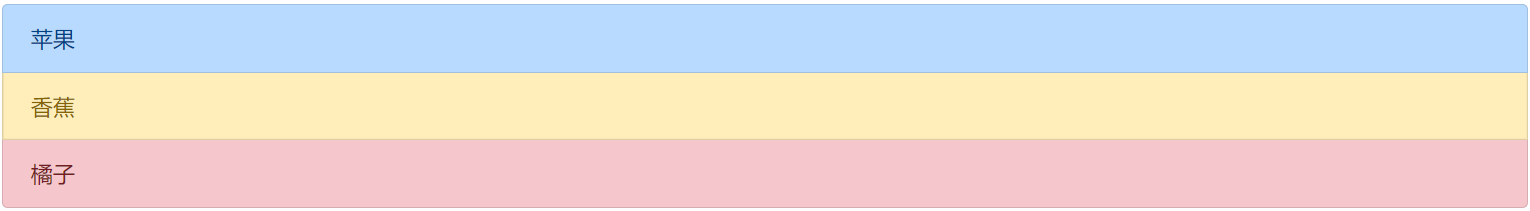


点击可跳转页面。

### 多种颜色列表项

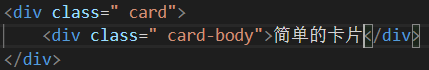
列表项目的颜色可以通过以下列来设置：**.list-group-item-success**, **list-group-item-secondary**, **list-group-item-info**, **list-group-item-warning**, **.list-group-item-danger**, **list-group-item-dark** 和 **list-group-item-light：**

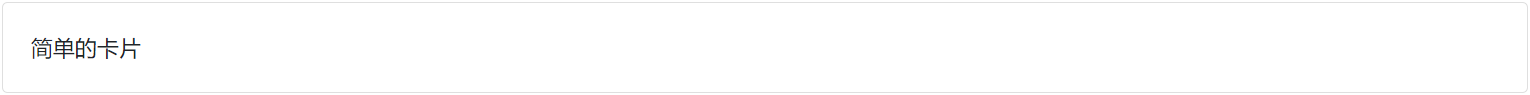
**对于普通列表性or链接列表项都适用。**



## [卡片](https://www.runoob.com/bootstrap4/bootstrap4-cards.html#content)

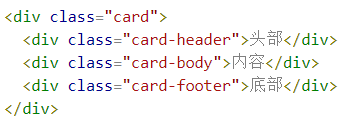
我们可以通过 Bootstrap4 的 **.card** 与 **.card-body** 类来创建一个简单的卡片:





### 头部和底部

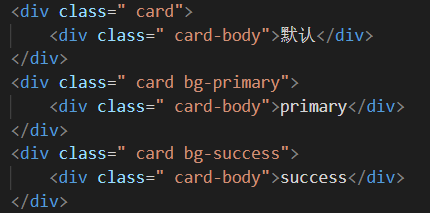
**.card-header**类用于创建卡片的头部样式， **.card-footer** 类用于创建卡片的底部样式：

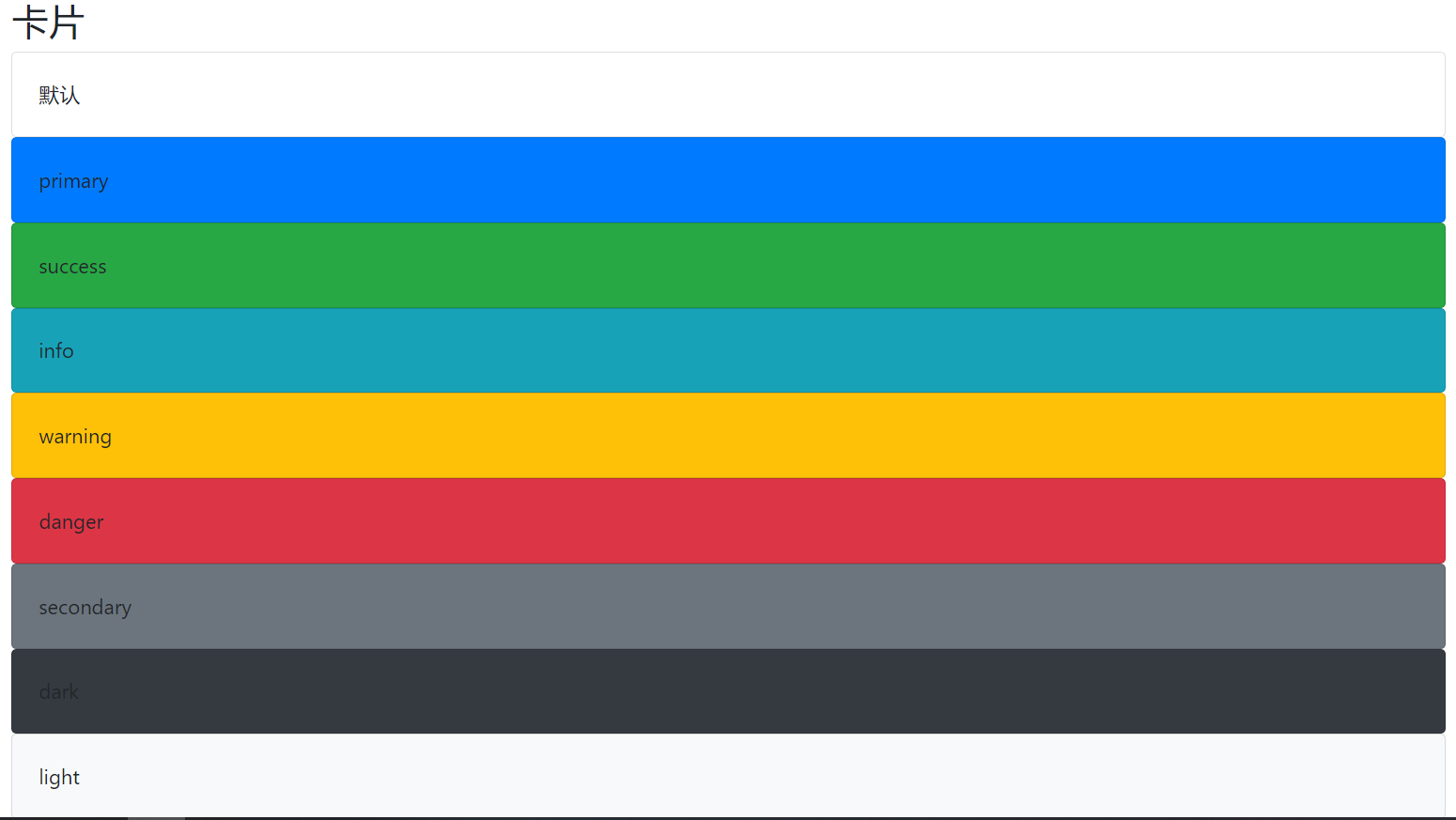




### 多种颜色卡片

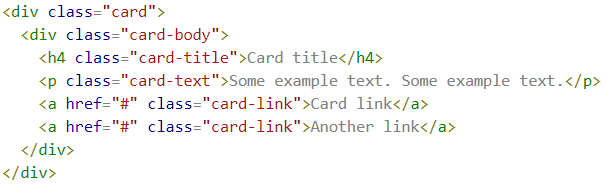
Bootstrap 4 提供了多种卡片的背景颜色类： **.bg-primary**, **.bg-success**, **.bg-info**, **.bg-warning**, **.bg-danger**, **.bg-secondary**, **.bg-dark** 和 **.bg-light**。

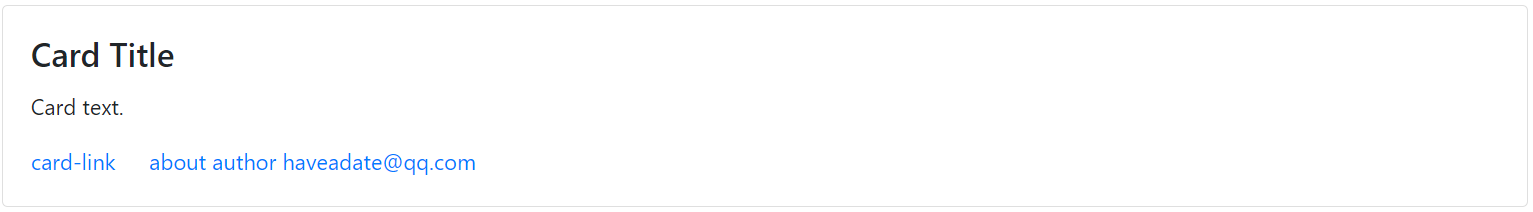




### 标题、文本和链接

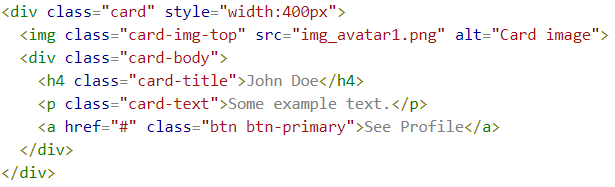
我们可以在头部元素上使用 **.card-title** 类来设置卡片的标题 。 **.card-text** 类用于设置卡片正文的内容。 **.card-link** 类用于给链接设置颜色。

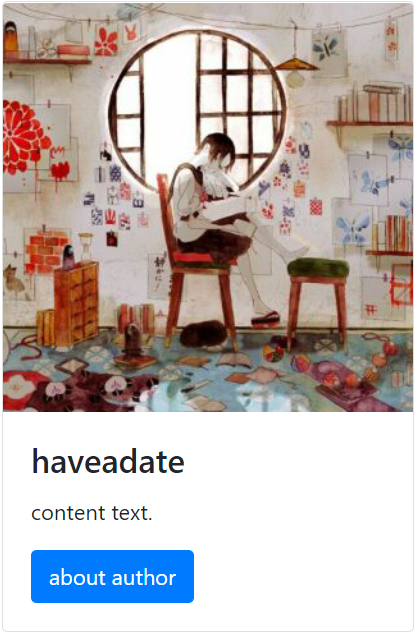




### 图片卡片

我们可以给 **<img>** 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：

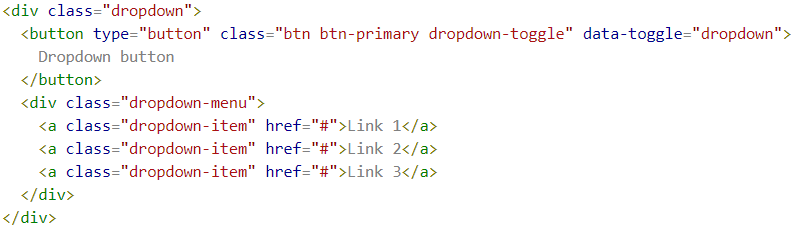




如果图片要设置为背景，可以使用 **.card-img-overlay** 类（在.card-body中添加）。

## [下拉菜单](https://www.runoob.com/bootstrap4/bootstrap4-dropdowns.html#content)

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。Bootstrap4 下拉菜单依赖于 popper.min.js。





**.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。我们也可以在 **<a>** 标签中使用。

### 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（穿插在.dropdown-item元素之间）。

### 下拉菜单中的标题

**.dropdown-header** 类用于在下拉菜单中添加标题（位置放到.dropdown-menu的第一个子元素）。

### 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。

### 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。

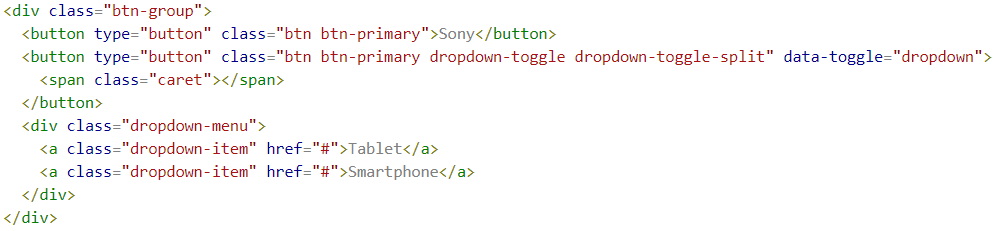
### 下拉菜单弹出方向设置

下拉菜单弹出方向默认为向下，当然我们也可以设置不同的方向。

* 如果你希望下拉菜单向右弹出，可以在 div 元素上添加 **"dropright"** 类;
* 如果你希望上拉菜单向上弹出，可以在 div 元素上添加 **"dropup"** 类;
* 如果你希望下拉菜单向上弹出，可以在 div 元素上添加 **"dropleft"** 类;

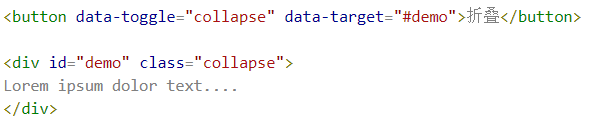
### 按钮中设置下拉菜单

我们可以在按钮中添加下拉菜单:



## [折叠](https://www.runoob.com/bootstrap4/bootstrap4-collapse.html#content)

Bootstrap4 折叠可以很容易的实现内容的显示与隐藏。



**.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:

