

---

## 1.1 Bootstrap 介绍

### 1.1.1 Bootstrap 简介

Bootstrap 是美国 Twitter 公司的设计师 Mark Otto 和 Jacob Thornton 合作基于 HTML、CSS、JavaScript 开发的简洁、直观、强悍的前端开发框架，使得 Web 开发更加快捷。Bootstrap 提供了优雅的 HTML 和 CSS 规范，它是由动态 CSS 语言 Less 写成。Bootstrap 一经推出后颇受欢迎，一直是 GitHub 上的热门开源项目，包括 NASA 的 MSNBC（微软全国广播公司）的 Breaking News 都使用了该项目。国内一些移动开发者较为熟悉的框架，如 WeX5 前端开源框架等，也是基于 Bootstrap 源码进行性能优化而来。

### 1.1.2 Bootstrap 的特性

Bootstrap 是开源的，它为开发人员创建接口提供了一个简洁统一的解决方案，包含了功能强大的内置组件，易于定制。Bootstrap 的特性如下：

- （1）移动设备优先：从 Bootstrap 3 开始，框架中包含了贯穿于整个库的移动设备优先的样式。
  - （2）浏览器支持广泛：所有的主流浏览器都支持 Bootstrap。包括 IE 浏览器、火狐浏览器、Opera 浏览器、Chrome 浏览器和 Safari 浏览器。
  - （3）易上手：只需具备 HTML 和 CSS 基础知识，就可以学习 Bootstrap。
  - （4）响应式设计：Bootstrap 的响应式 CSS 样式可以自适应于台式机、平板电脑和手机。
- Bootstrap 的响应式设计，如图 1.1 所示。



图 1.1 Bootstrap 的响应式设计

### 1.1.3 Bootstrap 的安装和配置

#### 1. 下载 Bootstrap

- （1）打开 Bootstrap 的官方主页：<http://getbootstrap.com/>，可以看到当前的最新版本是 V4.1。Bootstrap 官方主页，如图 1.2 所示。

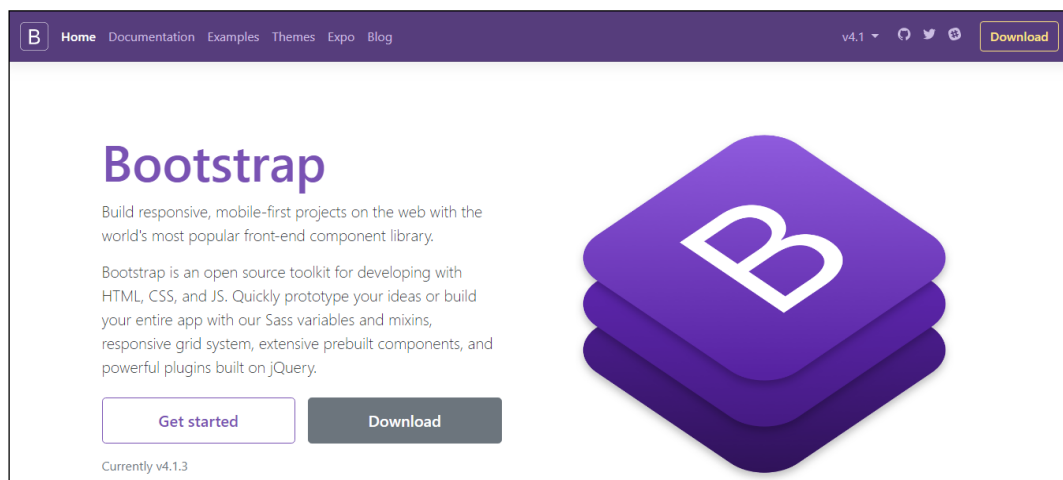


图 1.2 Bootstrap 官方主页

(2) 点击页面中的“Download”按钮进入到 Bootstrap 的下载页面。Bootstrap 下载页面，如图 1.3 所示。

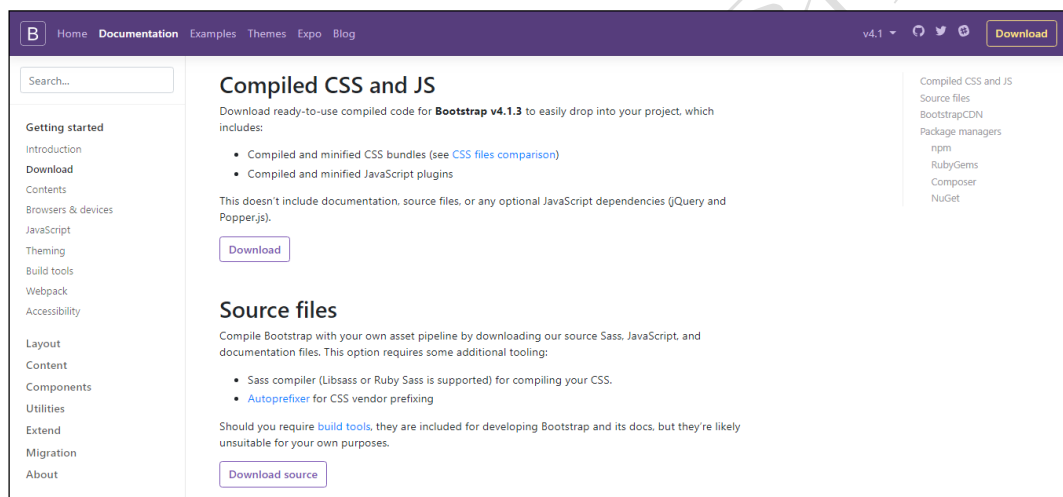


图 1.3 Bootstrap 下载页面

页面中有 2 个下载按钮，Download 和 Download source。

下载 Bootstrap 预编译文件：

点击 Download 按钮可下载经过编译的、可立即使用的安装包；下载安装包后，解压到任意目录，将看到如下所示的目录结构，预编译的 Bootstrap 目录结构，如图 1.4 所示。

```
bootstrap/
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   ├── bootstrap-grid.css
│   ├── bootstrap-grid.css.map
│   ├── bootstrap-grid.min.css
│   ├── bootstrap-grid.min.css.map
│   ├── bootstrap-reboot.css
│   ├── bootstrap-reboot.css.map
│   ├── bootstrap-reboot.min.css
│   └── bootstrap-reboot.min.css.map
└── js/
    ├── bootstrap.bundle.js
    ├── bootstrap.bundle.js.map
    ├── bootstrap.bundle.min.js
    ├── bootstrap.bundle.min.js.map
    ├── bootstrap.js
    ├── bootstrap.js.map
    ├── bootstrap.min.js
    └── bootstrap.min.js.map
```

图 1.4 预编译的 Bootstrap 目录结构

预编译的 Bootstrap 文件可直接引入到项目中，它提供了预先编译的 CSS 和 JS (bootstrap.\*) 文件，以及编译并压缩的 CSS 和 JS (bootstrap.min.\*) 文件。CSS source maps (bootstrap\*.map) 用于配合某些浏览器的开发者工具使用。打包后的 JS 文件 (bootstrap.bundle.js) 和压缩后的 JS 文件 (bootstrap.bundle.min.js) 已经包含了 Popper (一个扩展较好的 JS 插件，官方主页：<http://popper.js.org/>)，但是并没有包含 jQuery。

下载 Bootstrap 源代码文件：

点击 Download source 按钮可下载 Bootstrap 的源代码，下载 Bootstrap 源代码后，解压到任意目录，将看到如下所示的目录结构。Bootstrap 源代码的目录结构，如图 1.5 所示。

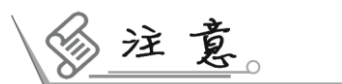
```
bootstrap/
├── scss/
├── js/
├── build/
├── dist/
│   ├── css/
│   └── js/
├── site/
│   └── docs/
│       ├── 4.1/
│       └── examples/
```

图 1.5 Bootstrap 源代码的目录结构

Bootstrap 的源代码包含了预先编译的 CSS 和 JavaScript 文件，并且还有 Sass、JavaScript 和文档的源码。scss/和 js/目录分别包含了 CSS 和 JS 的源码。dist/目录包含了预编译 Bootstrap 包内的所有文件。build/目标中包含了编译 Bootstrap 的构建工具。site/目录中提供了 Bootstrap 的网站介绍和文档的源码文件，examples/目录是 Bootstrap 官方提供的实例工程。除此之外，bootstrap 文件夹中还包含 Bootstrap 安装包的定义文件、许可证文件和编译脚本等。



为了更好的了解和更方便的使用，本书中将使用 Bootstrap 的预编译文件。



Bootstrap 4 与 Bootstrap 3 相比拥有了更多的具体类，把一些相关的部分变成了组件。同时 Bootstrap.min.css 的体积减少了 40% 以上。Bootstrap 4 放弃了对 IE8 以及 IOS 6 的支持，目前仅支持 IE9 以上以及 iOS 7 以上版本的浏览器。

## 1.2 创建第一个 Bootstrap 页面

### 1.2.1 开发工具介绍

(1) 打开 Visual studio code 的官方主页 (<https://code.visualstudio.com/>)。Visual studio code 的官方主页，如图 1.6 所示。

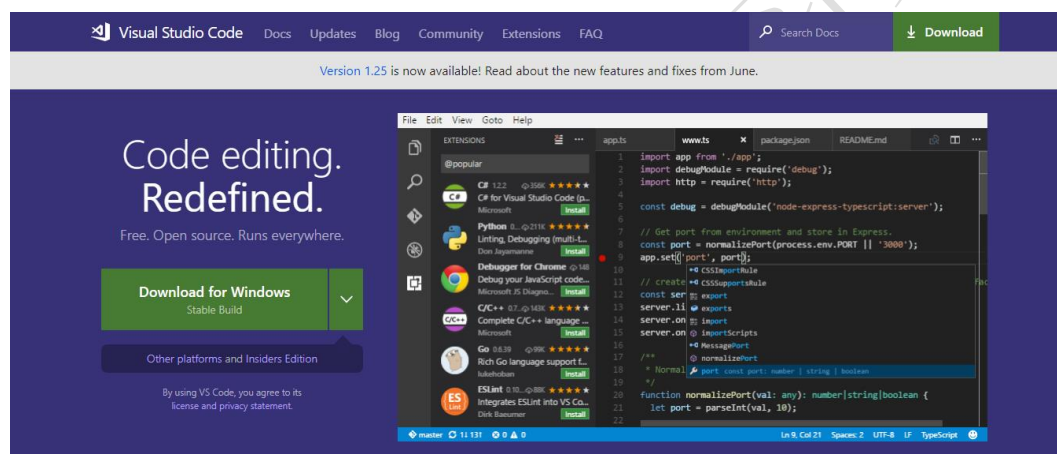


图 1.6 Visual studio code 的官方主页

点击页面中的“Download for Windows”按钮下载 Visual studio code 开发工具。

(2) 下载完成后双击安装文件进行安装。安装完成后打开 Visual studio code 开发工具。Visual studio code 开发工具，如图 1.7 所示。

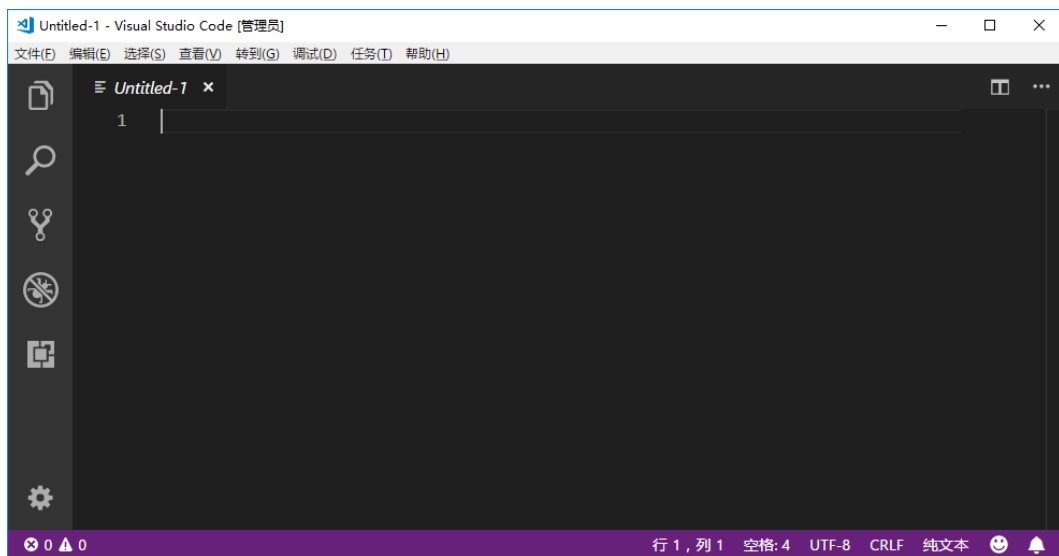


图 1.7 Visual studio code 开发工具

(3) 安装 Web Server 插件，点击 Visual studio code 开发工具左侧边栏中的扩展图标，在搜索框中输入“Preview on Web Server”，点击搜索到的结果，并在右侧的详情页面点击“安装”按钮。安装 Web Server 插件，如图 1.8 所示。

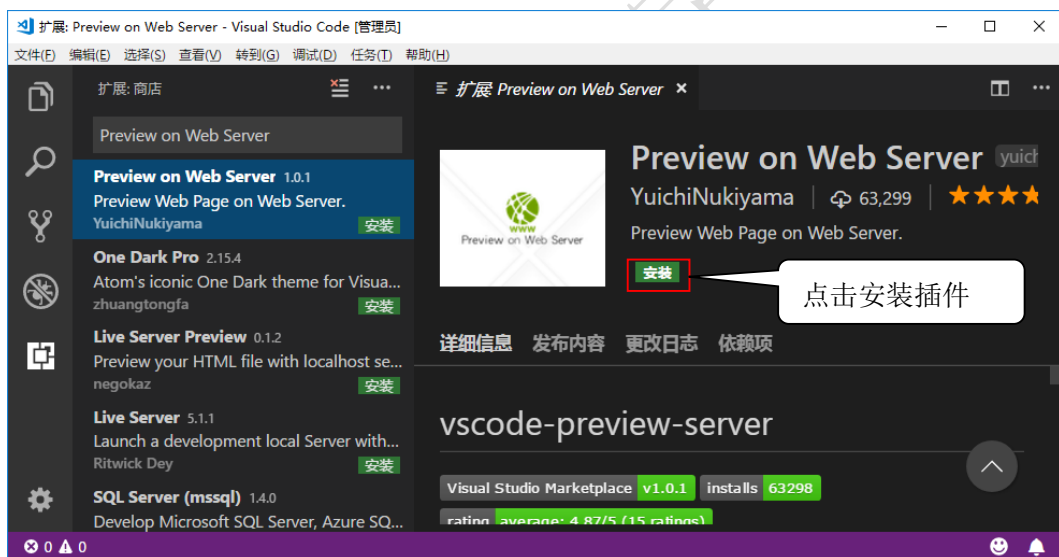


图 1.8 安装 Web Server 插件

安装完成后点击“重新加载”按钮即可。

### 1.2.2 创建项目

(1) 启动 Visual studio code 开发工具，在磁盘上创建“Bootstrap”文件夹作为项目的根目录，同时在“Bootstrap”文件夹下创建“js”文件夹用来放置 JS 文件，创建“css”文件夹用来放置 CSS 样式文件，创建“img”文件夹用来放置图片文件。在 Visual studio code 开发工具中点击菜单栏的“文件”，选择“打开文件夹”选项，选择“Bootstrap”文件夹并打开。打开 Bootstrap 项目文件夹，如图 1.9 所示。

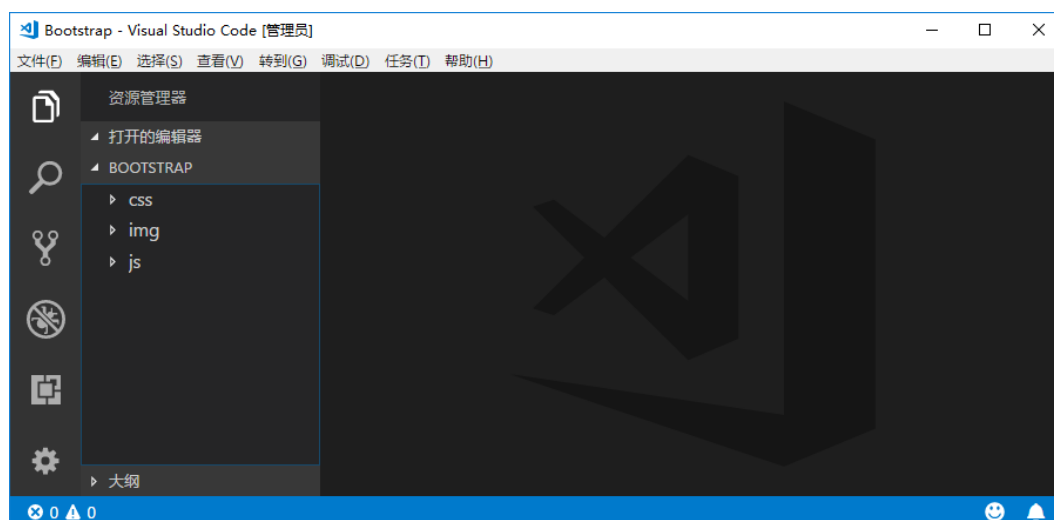


图 1.9 打开 Bootstrap 文件夹

(2) 打开下载的预编译 bootstrap 文件夹，将 css 文件夹下的“bootstrap.min.css”文件复制到项目根目录下的 css 文件夹中，将 js 文件夹下的“bootstrap.bundle.min.js”文件复制到项目根目录下的 js 文件夹中，将 jQuery 文件复制到项目根目录下的 js 文件夹中。复制 JQuery 和 Bootstrap 文件到项目中，如图 1.10 所示。

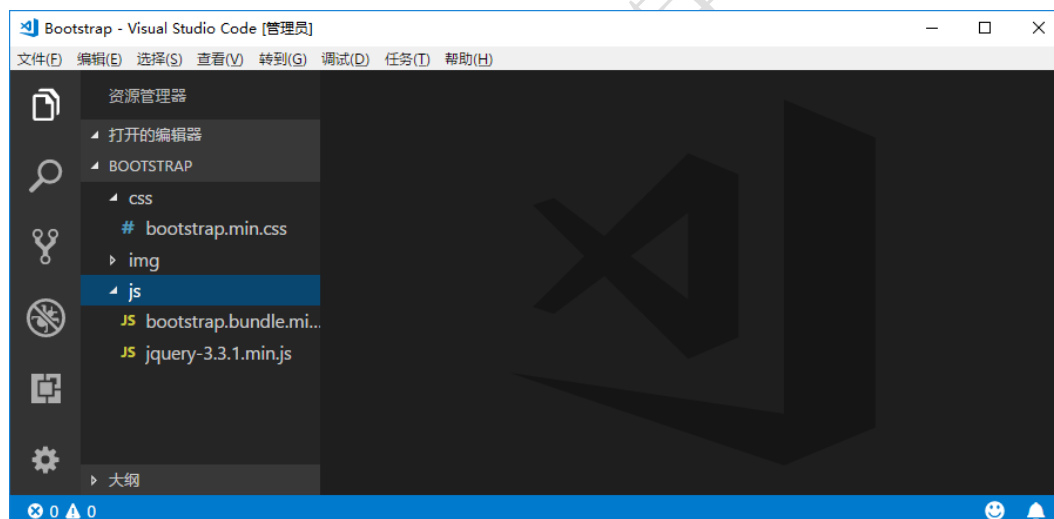


图 1.10 复制 JQuery 和 Bootstrap 文件到项目中

(3) 创建页面。在 Visual studio code 开发工具中点击菜单栏的“文件”，选择“新建文件”选项，并将文件命名为“hello-world.html”，保存到“Bootstrap”项目的根目录。创建“hello-world.html”页面，如图 1.11 所示。

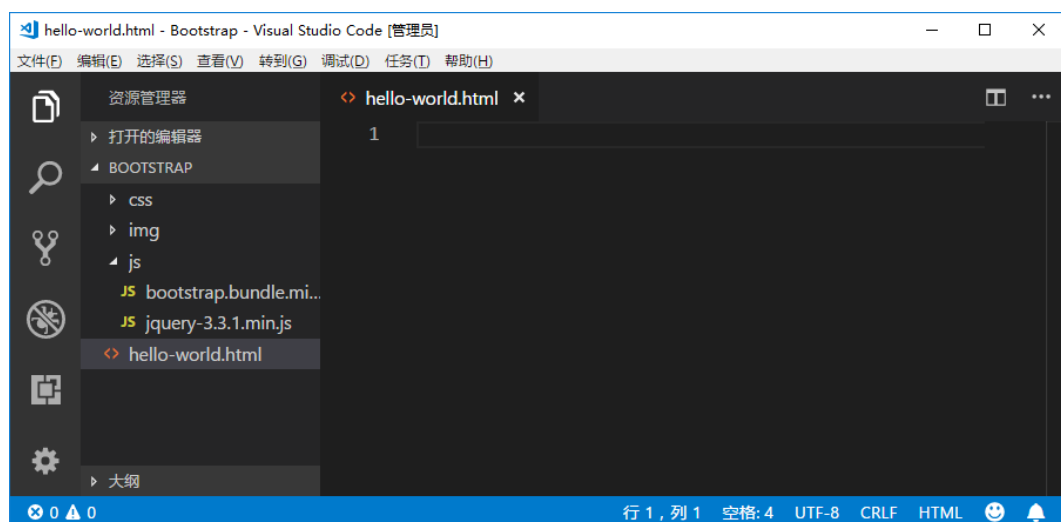
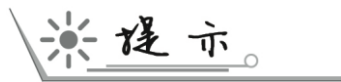


图 1.11 创建“hello-world.html”页面

### 1.2.3 引用框架

(1) Bootstrap 中使用了一些 HTML5 元素和 CSS 属性。为了让它们正常运行，需要在页面中使用 HTML5 文档类型 (Doctype)，在文档头部声明，并设置对应编码。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
</body>
</html>
```



在 Visual studio code 开发工具中，编辑 HTML 文件时输入“html:5”并按键盘的“Enter”键，可快速生成 HTML5 模板。

(2) 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 中添加 viewport meta 标签。

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

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。

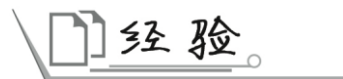
(3) 在页面中引入 JQuery 和 Bootstrap 文件，如下所示：

---

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <script src="js/jquery-3.3.1.min.js"></script>
  <script src="js/bootstrap.bundle.min.js"></script>
  <title>Document</title>
</head>
<body>
</body>
</html>
```

引入 Bootstrap 时不仅可以使⽤本地的 Bootstrap 文件，还可以使⽤ CDN 库，如下所示：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
  <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.bundle.min.js"></script>
  <title>Document</title>
</head>
<body>
</body>
</html>
```



CDN 服务国内推荐使用：<https://www.staticfile.org>，国际推荐使用：<https://cdnjs.com>。



Bootstrap 中的许多组件需要依赖 JavaScript 才能运行，所以必须将 JQuery 文件的引用放



置在其他 JS 文件的引用之前。

(3) 添加页面内容，如下所示：

```
<body>
  <h1>Hello World!</h1>
</body>
```

运行程序，查看运行结果。页面输出“Hello World!”如图 1.12 所示。

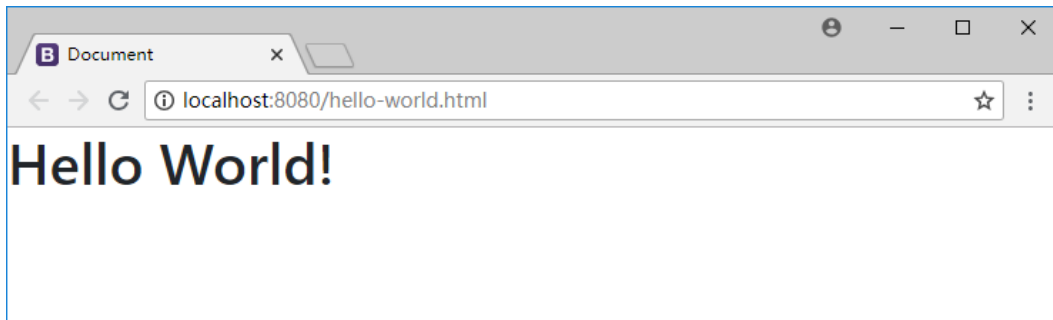


图 1.12 页面输出“Hello World!”



右键点击要预览的页面，在弹出的选项中选择“vscode-preview-sever:Launch on browser”选项，可在浏览器上预览页面效果，或者选择要预览的页面，按“Ctrl+Shift+L”键，也可在浏览器上预览页面效果。

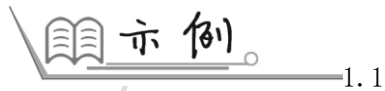
## 1.3 Bootstrap 网格系统

### 1.3.1 容器类

Bootstrap 4 中需要一个容器元素来包裹网站的内容。有以下两个容器类：

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

接下来分别使用一个案例来演示 `.container` 类和 `.container-fluid` 类的用法。



```
<body>
  <div class="container" style="background: gray; height: 300px;">
    .container
  </div>
</body>
```

运行程序，查看运行结果。`.container` 类的用法，如图 1.13 所示。

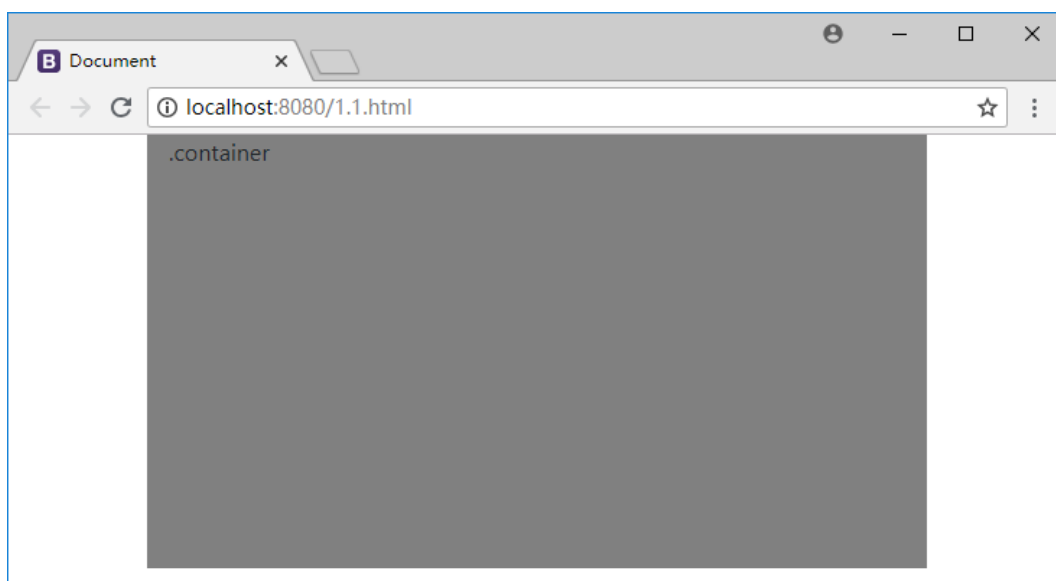
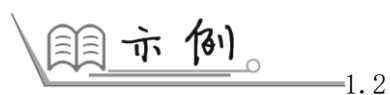


图 1.13 .container 类的用法



```
<body>  
  <div class="container-fluid" style="background: gray; height: 300px;">  
    .container  
  </div>  
</body>
```

运行程序，查看运行结果。.container-fluid 类的用法，如图 1.14 所示。

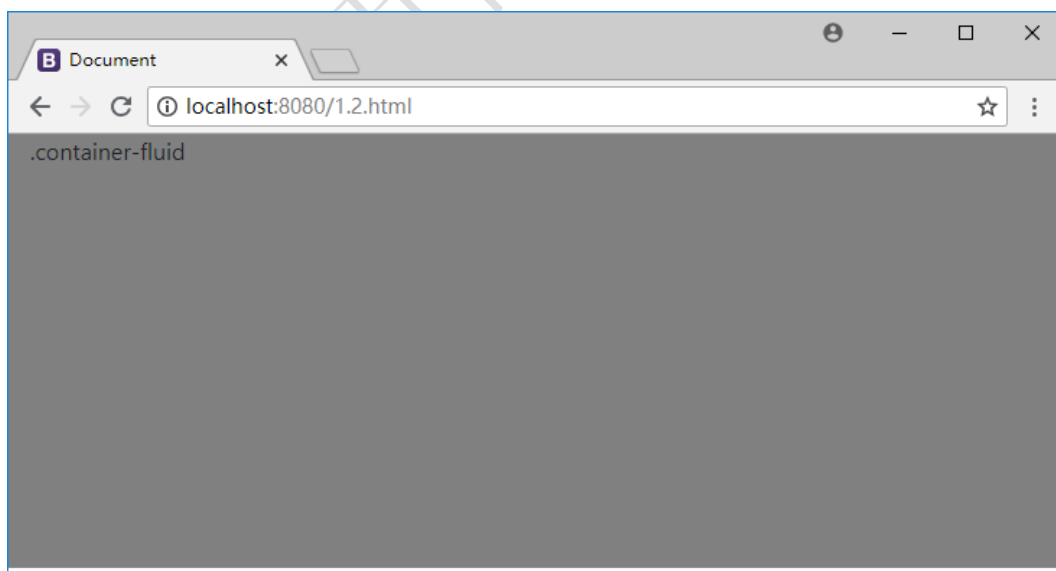


图 1.14 .container-fluid 类的用法

### 1.3.2 网格系统简介

Bootstrap 提供了一套响应式的、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。Bootstrap 网格布局，如图 1.15 所示。

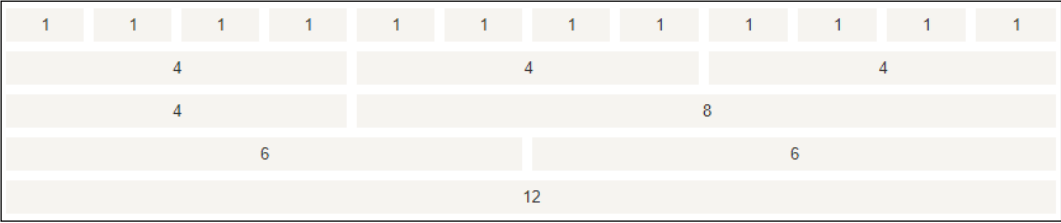


图 1.15 Bootstrap 网格系统



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。

1. 网格类

Bootstrap 4 网格系统提供了以下 5 个类：

- (1) `.col-(1-12)`：针对所有设备
  - (2) `.col-sm-(1-12)`：针对屏幕宽度等于或大于 576px 的设备，如平板
  - (3) `.col-md-(1-12)`：屏幕宽度等于或大于 768px，如桌面显示器
  - (4) `.col-lg-(1-12)`：屏幕宽度等于或大于 992px，如大桌面显示器
  - (5) `.col-xl-(1-12)`：屏幕宽度等于或大于 1200px，如超大桌面显示器
- 上述各个类可组合使用，从而创建更灵活的页面布局。

2. 网格系统规则

Bootstrap 4 中的网格系统规则：

- (1) 网格中的每一行都需要放置在 `.container` (固定宽度) 类或 `.container-fluid` (全屏宽度) 类的容器中，便于为页面自动设置一些外边距和内边距。
- (2) 使用行来创建水平的列组。
- (3) 页面的内容需要放置在列中，并且只有列可以是行的直接子节点。
- (4) 预定义的类（如 `.row` 和 `.col-sm-4`）可用于快速制作网格布局。
- (5) 列之间的间隙是通过列的边距来设置的，通过 `.row` 类上的负边距消除第一列和最后一列的边距。
- (6) 网格列是通过跨越指定的 12 个可用列来创建。例如，要在平板电脑上设置三个相等的列，则需要使用三个 `.col-sm-4` 类来设置。



Bootstrap 3 和 Bootstrap 4 最大的区别在于 Bootstrap 4 使用的是 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和等高列。Bootstrap 网格系统在不同设备上的显示情况，见表 1.1。

表 1.1 Bootstrap 网格系统在不同设备上的显示情况

超小设备	平板	桌面显示	大桌面显示	超大桌面显示
<576px	≥576px	器	器	器

	≥768px		≥992px	≥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				

### 1.3.3 网格系统结构

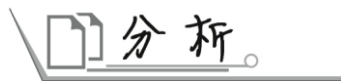
Bootstrap 4 网格的基本结构如下：

<!-- 示例 1：控制列的宽度以及在不同的设备上的显示情况 -->

```
<div class="row">
  <div class="col-*-*"></div>
</div>
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
```

<!-- 示例 2：Bootstrap 自动处理布局 -->

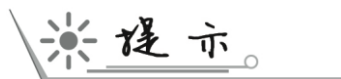
```
<div class="row">
  <div class="col"></div>
  <div class="col"></div>
  <div class="col"></div>
</div>
```



#### 分析

(1) 示例 1：创建一个行(<div class="row">)。然后添加需要的列(.col-\*\*)。第一个星号(\*)表示响应的设备，可以是：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。

(2) 示例 2：在每个 col 上不添加数字，让 bootstrap 自动处理布局，同一行的每个列宽度相等：如果是两列，则每个"col"为 50%的宽度；如果是三列，则每个"col"为 33.33%的宽度；如果是四列，则每个"col"为 25%的宽度，以此类推。



#### 提示

可以使用.col-sm|md|lg|x1 来设置列的响应规则。

接下来使用一个案例来演示在行中创建四个相等宽度的列。

### 示例 1.3

```
<body>
  <div class="container">
    <p>创建 4 个相等宽度的列</p>
    <div class="row">
      <div class="col bg-info">第一列</div>
      <div class="col bg-warning">第二列</div>
      <div class="col bg-danger">第三列</div>
      <div class="col bg-success">第四列</div>
    </div>
  </div>
</body>
```

运行程序，查看运行结果。Bootstrap 自动布局等宽列，如图 1.16 所示。

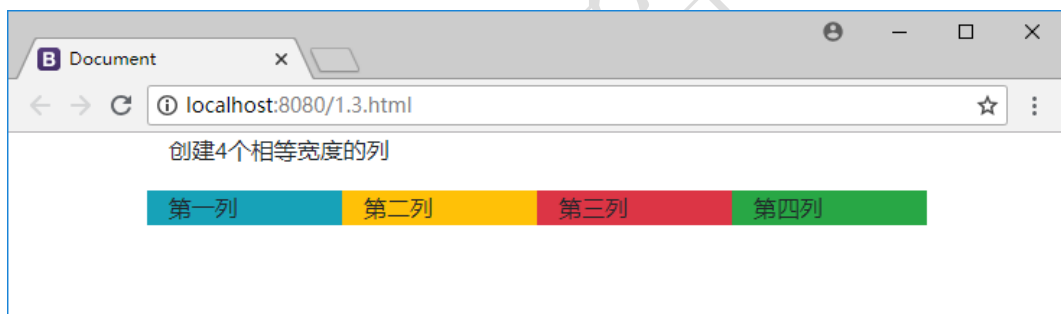


图 1.16 Bootstrap 自动布局等宽列

### 说明

- .bg-info 类：设置提示信息的文字背景颜色。
- .bg-warning：设置提示警告的文字背景颜色。
- .bg-danger：设置提示危险的文字背景颜色。
- .bg-success：设置操作成功的文字背景颜色。

接下来使用一个案例来演示在行中创建等宽式响应列。在平板及更大屏幕上四个列的宽度相等。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。

### 示例 1.4

```
<body>
  <div class="container">
```

```
<p>创建 4 个等宽式响应列</p>
<div class="row">
    <div class="col-sm-3 bg-info">第一列</div>
    <div class="col-sm-3 bg-warning">第二列</div>
    <div class="col-sm-3 bg-danger">第三列</div>
    <div class="col-sm-3 bg-success">第四列</div>
</div>
</body>
```

运行程序，查看运行结果。

调整浏览的宽度，当浏览器的宽度大于等于 576px 时，四列平分页面宽度。浏览器宽度大于等于 576px 时四列等宽，如图 1.17 所示。

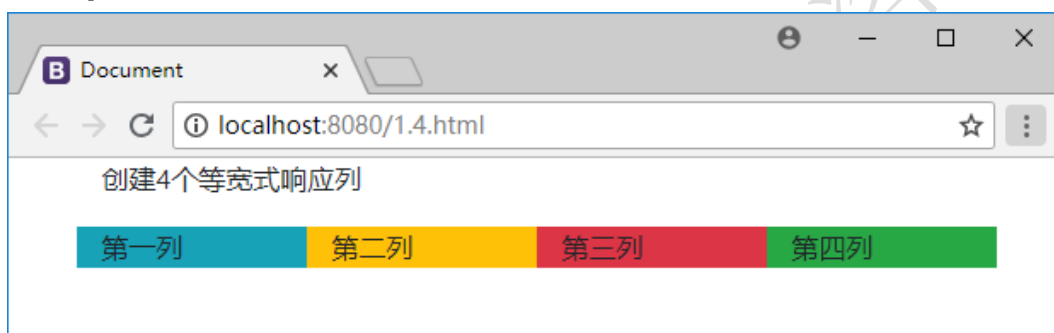


图 1.17 浏览器宽度大于等于 576px 时四列等宽

当浏览器的宽度小于 576px 时，四列上下堆叠排版。浏览器宽度小于 576px 时四列堆叠排版，如图 1.18 所示。

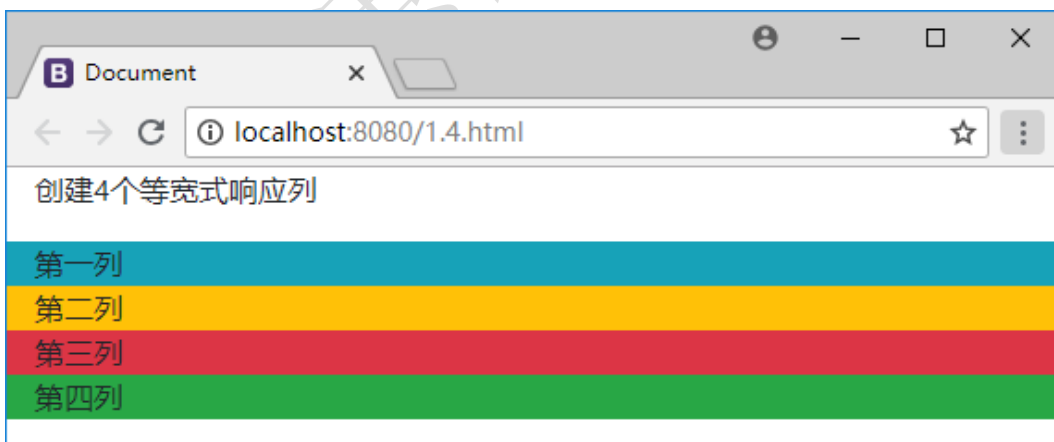


图 1.18 浏览器宽度小于 576px 时四列堆叠排版

接下来使用一个案例来演示在行中创建非等宽式响应列。在平板及更大屏幕上两列的宽度比为 1:2。在移动设备上，即屏幕宽度小于 576px 时，两列将会上下堆叠排版。

```
<body>
  <div class="container">
    <p>创建 2 个非等宽式响应列</p>
    <div class="row">
      <div class="col-sm-4 bg-info">第一列</div>
      <div class="col-sm-8 bg-warning">第二列</div>
    </div>
  </div>
</body>
```

运行程序，查看运行结果。

调整浏览的宽度，当浏览器的宽度大于等于 576px 时，列宽分别占页面宽度的 1/3 和 2/3。浏览器宽度大于等于 576px 时两列非等宽，如图 1.19 所示。

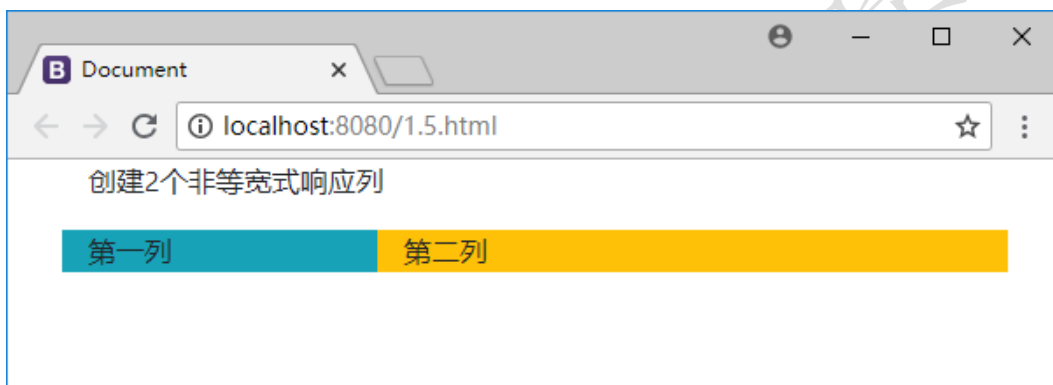


图 1.19 浏览器宽度大于等于 576px 时两列非等宽

当浏览器的宽度小于 576px 时，两列上下堆叠排版。浏览器宽度小于 576px 时两列堆叠排版，如图 1.20 所示。

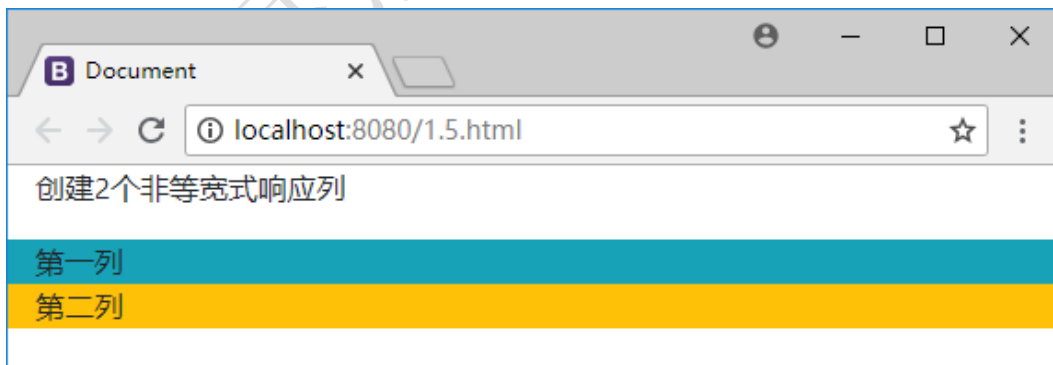


图 1.20 浏览器宽度小于 576px 时两列堆叠排版

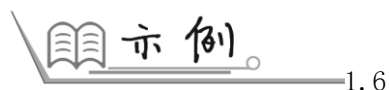
## 1.4 Bootstrap 响应式布局

### 1.4.1 不同设备的响应式布局

(1) 在平板设备和桌面设备上显示列的不同宽度

接下来使用一个案例来演示在平板设备上显示两列，列的宽度分别占页面的 1/4 和 3/4，在

桌面设备上这两列的宽度分别占页面宽度的一半。



```
<body>
  <div class="container">
    <p>在平板设备和桌面设备上显示列的不同宽度</p>
    <div class="row">
      <div class="col-sm-3 col-md-6 bg-info">第一列</div>
      <div class="col-sm-9 col-md-6 bg-warning">第二列</div>
    </div>
  </div>
</body>
```

运行程序，查看运行结果。

调整浏览的宽度，当浏览器的宽度在 576px 和 768px 之间时，列宽分别占页面宽度的 1/4 和 3/4。浏览器宽度在 576px 和 768px 之间时列宽比为 1:3，如图 1.21 所示。

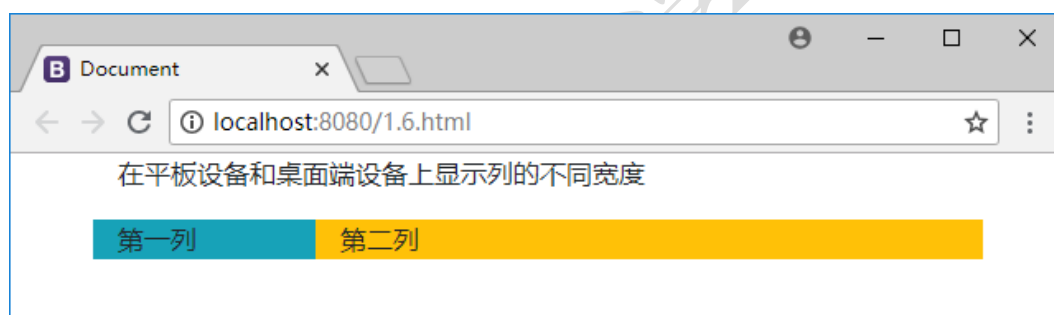


图 1.21 浏览器宽度在 576px 和 768px 之间时列宽比为 1:3

当浏览器的宽度大于 768px 时，列宽分别占页面宽度的一半。浏览器宽度大于 768px 时列宽比为 1:1，如图 1.22 所示。

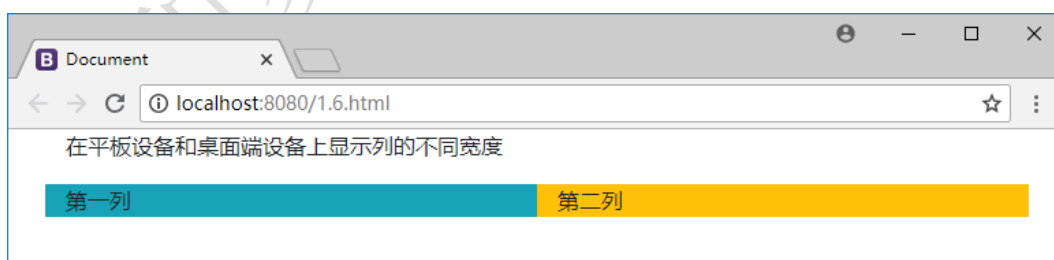


图 1.22 浏览器宽度大于 768px 时列宽比为 1:1

(2) 在平板设备、桌面设备、大桌面显示器和超大桌面显示器上显示列的不同宽度

接下来使用一个案例来演示在平板设备上显示两列，列的宽度分别占页面的 1/4 和 3/4；在桌面设备上这两列的宽度分别占页面宽度的一半；在大桌面显示器上这两列的宽度分别占页面宽度的 1/3 和 2/3；在超大桌面显示器上着两列的宽度分别占页面宽度的 1/6 和 5/6。



```
<body>
  <div class="container">
    <p>
      在平板设备、桌面设备、大桌面显示器和超大桌面显示器上显示列的不同宽度
    </p>
    <div class="row">
      <div class="col-sm-3 col-md-6 col-lg-4 col-xl-2 bg-info">
        第一列
      </div>
      <div class="col-sm-9 col-md-6 col-lg-8 col-xl-10 bg-warning">
        第二列
      </div>
    </div>
  </div>
</body>
```

运行程序，查看运行结果。

调整浏览的宽度，当浏览器的宽度在 576px 和 768px 之间时，列宽分别占页面宽度的 1/4 和 3/4，如图 1.22 所示；当浏览器的宽度在 768px 和 992px 之间时，列宽分别占页面宽度的一半，如图 1.23 所示；当浏览器的宽度在 992px 和 1200px 之间时，列宽分别占页面宽度的 1/3 和 2/3。浏览器宽度在 992px 和 1200px 之间时列宽比为 1:2，如图 1.23 所示。

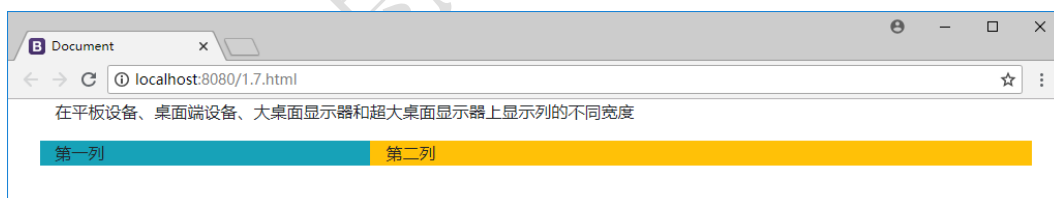


图 1.23 浏览器宽度在 992px 和 1200px 之间时列宽比为 1:2

当浏览器的宽度大于或等于 1200px 时，列宽分别占页面宽度的 1/6 和 5/6。浏览器宽度大于 1200px 时列宽比为 1:5，如图 1.24 所示。

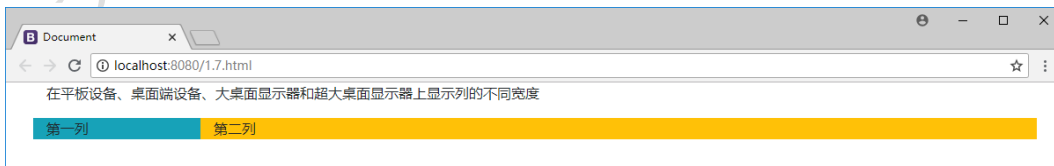


图 1.24 浏览器宽度大于 1200px 时列宽比为 1:5

## 1.4.2 列预定义类

### (1) 列偏移

偏移列可以通过 `offset-*` 类来设置。第一个星号(\*)可以是 `sm`、`md`、`lg` 或 `xl`，表示屏幕

的设备类型，第二个星号(\*)是 1 到 11 之间的数字。

例如需要在桌面显示器上使用偏移，则需要使用 `.offset-md-*` 类。这些类会把一个列的左外边距 (margin) 增加 \* 列，其中 \* 是 1 到 11 之间的数字。

例如：`.offset-md-4` 表示在桌面显示器上将列向右移动了四格。

接下来通过一个案例来演示列偏移的使用。

## 示例 1.8

```
<body>
  <div class="container">
    <div class="row">
      <div class="col-md-4 bg-info">
        第一行第一列，列宽 4 格
      </div>
      <div class="col-md-4 offset-md-4 bg-info">
        第一行第二列，列宽 4 格，向右偏移 4 格
      </div>
    <br>
    <div class="row">
      <div class="col-md-3 offset-md-3 bg-warning">
        第二行第一列，列宽 3 格，向右偏移 3 格
      </div>
      <div class="col-md-3 offset-md-3 bg-warning">
        第二行第二列，列宽 3 格，向右偏移 3 格
      </div>
    </div>
    <br>
    <div class="row">
      <div class="col-md-6 offset-md-3 bg-success">
        第三行第一列，列宽 6 格，向右偏移 3 格
      </div>
    </div>
  </div>
</body>
```

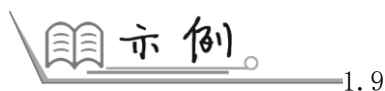
运行程序，查看运行结果。在桌面显示器上使用列的偏移，如图 1.25 所示。



图 1.25 在桌面显示器上使用列的偏移

## (2) 列包裹

如果一行中放置了超过 12 列，则超出的列将被包裹进一个新的行中。接下来通过一个案例来演示在平板设备上列包裹的使用。



```
<body>
  <div class="container">
    <div class="row">
      <div class="col-sm-9 bg-info">第 1 列</div>
      <div class="col-sm-4 bg-warning">第 2 列，
        由于行列宽之和大于 12 格 (9+4>12)，将该列放置在一个单独的行中
      </div>
      <div class="col-sm-6 bg-danger">第 3 列，
        由于行列宽之和小于 12 格 (4+6<12)，将该列放置在第二列的右边
      </div>
      <div class="col-sm-2 bg-success">第 4 列，
        由于行列宽等于 12 (4+6+2=12)，将该列放置在第三列的右边
      </div>
    </div>
  </div>
</body>
```

运行程序，查看运行结果。在平板设备上使用列包裹，如图 1.26 所示。



图 1.26 在平板设备上使用列包裹

### (3) 列排序

Bootstrap 网格系统的另一个完美的特性，就是可以很容易地以一种顺序编写列，然后以另一种顺序显示列。可以轻易地改变带有 `.order-md-*` 类的内置网格列的顺序，其中 `*` 的范围是 1 到 11。

接下来通过一个案例来演示 Bootstrap 中列排序的用法，案例中有两列布局，左列很窄，作为侧边栏，使用 `.order-md-*` 类互换两列的顺序。



1.10

```
<body>
  <div class="container">
    <p>排序前</p>
    <div class="row">
      <div class="col-md-4 bg-info">
        我在左边
      </div>
      <div class="col-md-8 bg-warning">
        我在右边
      </div>
    </div>
    <br>
    <p>排序后</p>
    <div class="row">
      <div class="col-md-4 order-md-8 bg-info">
        我在左边
      </div>
      <div class="col-md-8 order-md-4 bg-warning">
        我在右边
      </div>
    </div>
  </div>
```

```

        </div>
    </div>
</div>
</body>

```

运行程序，查看运行结果。使用 `.order-md-*` 类互换两列的顺序，如图 1.27 所示。

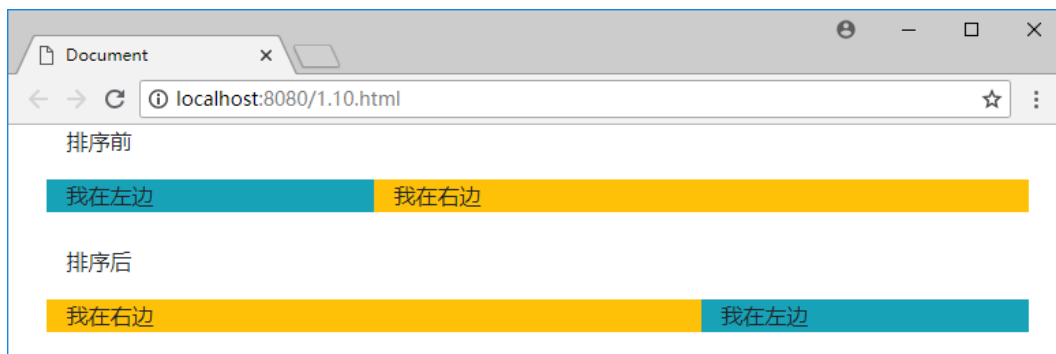


图 1.27 使用 `.order-md-*` 类互换两列的顺序

#### (4) 列嵌套

可以在 Bootstrap 的列中嵌套列，嵌套方法如下：

在已有的列中添加一个新的行（`.row`），在行中添加一组列（`.col-*-*`），这组列的个数之和不能大于 12。

接下来使用一个案例来演示 Bootstrap 中列嵌套的用法，案例中布局的第二列被分为两个盒子。



1.11

```

<body>
  <div class="container">
    <div class="row">
      <div class="col-sm-3 bg-warning" style="height:200px">
        第一行第一列
      </div>
      <div class="col-sm-9 bg-info" style="height:200px">
        第一行第二列
        <div class="row">
          <div class="col-sm-6 bg-danger" style="height:150px">
            嵌套行的第一列
          </div>
          <div class="col-sm-6 bg-success" style="height:150px">
            嵌套行的第二列
          </div>
        </div>
      </div>
    </div>
  </div>

```

```
        </div>
    </div>
</div>
</div>
</body>
```

运行程序，查看运行结果。布局的第二列被分为两个盒子，如图 1.28 所示。

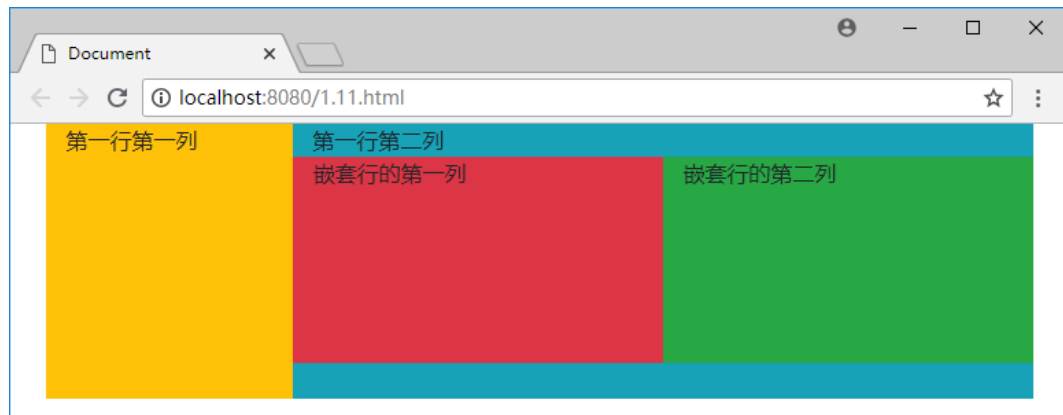


图 1.28 布局的第二列被分为两个盒子