

第 1 章

Bootstrap 入门

本章技能目标

- 掌握 Bootstrap 的概念
- 掌握 Bootstrap 的文件结构
- 掌握 Bootstrap 入门知识

本章简介

随着互联网的不断成熟以及人们越来越多地使用各种移动终端设备访问互联网，Web 设计师和 Web 开发者的工作也变得越来越复杂。

几年前，一切都还简单得多。那个时候，大部分用户都是台式计算机。网页设计人员的开发工作主要就是跟十几个桌面浏览器打交道，并通过添加几个浏览器的 hack，来兼容旧版本的 IE 浏览器。时至今日，随着手持电子设备的突飞猛进，一切都变了样。各种大小的智能手机和平板电脑层出不穷，电子阅读器以及电视设备上的浏览器等也不断涌现，设备的多样性日益丰富。

同时，浏览设备的屏幕大小是不一样的。因此不能使用固定网页宽度，在不知道浏览设备屏幕的大小时，最主要的策略就是使用响应式网页设计。它是一种根据设备浏览窗口的尺寸大小来输出相应页面布局的方法。

响应式网页设计实现起来并不困难，但是要让它在所有的目标设备上都能正常运作会有一点小棘手。框架可以让这一工作变得简单。利用框架，用户可以花最少的力气创建响应式且符合标准的网站，一切都很简单并且具有一致性。



1 Bootstrap 简介

1.1 Bootstrap 简介

响应式网页设计利用同样的 HTML 文档来适配所有的终端设备，响应式网页设计会根据设备屏幕的大小加载不同的样式，从而在不同的终端设备上呈现最优的网页布局。举个例子，当用户在大屏幕桌面浏览器中查看一个网页时，网页的内容可能是分为很多列的，并且有常见的导航条。如果用户在小屏幕的智能手机上查看同样的页面，用户会发现页面的内容呈现在同一列中，并且导航按钮足够大，点击起来很方便。

在实际的开发中，使用框架的情况很多，使用框架有很多好处，比如说简单快速，以及在不同的设备之间具有一致性等。框架最大的优势就是简单易用，即使只掌握少量的 Web 知识，也可以毫无障碍地使用它们。

Bootstrap 绝对是目前最流行、用得最广泛的一款框架。它是一套优美、直观并且功能强大的 Web 设计开源工具包，可以用来开发跨浏览器兼容、功能丰富并且美观的页面。它提供了很多流行的样式简洁的 UI 组件，栅格系统以及一些常用的 JavaScript 插件。

Bootstrap 是基于 HTML5 和 CSS3 开发的，它在 jQuery 的基础上进行了更为个性化和人性化的完善，形成一套自己独有的网站风格，并兼容大部分 jQuery 插件。Bootstrap 框架提供了很多常见的各种 CSS 和 JavaScript 集合，在布局、版式、控件、特效方面都非常让人满意，预置了丰富的效果，开发人员可以拿来直接使用，极大地方便了用户开发。Bootstrap 目前的稳定版本是 v3.3.5。

Bootstrap 有如下特性：

- 代码开源。
- 有一套完整的基础 CSS 插件。
- 丰富的预定义样式表。
- 一组基于 jQuery 的 JS 插件。
- 一个非常流行的栅格系统，崇尚移动先行的思想。

Bootstrap 的目标是在最新的桌面和移动浏览器上有最佳的表现，也就是说，在较老旧的浏览器上可能会导致某些组件表现出的样式有些不同，但功能是完整的。表 1.1 列出了不同的浏览器对 Bootstrap 的支持情况。

表 1.1 不同系统的浏览器对 Bootstrap 的支持情况

	Chrome	Firefox	IE	Opera	Safari
Android	支持	支持	N/A	不支持	N/A
iOS	支持	N/A	N/A	不支持	支持
Mac OS X	支持	支持	N/A	支持	支持
Windows	支持	支持	支持	支持	不支持

其中 N/A 表示在该系统中不能安装该浏览器。通过表 1.1 可以看出大部分系统的浏览器对 Bootstrap 还是支持的，开发人员也可以放心大胆地去使用了。

1.2 Bootstrap 文件结构和标准模板

示例 1 是 Bootstrap 官网提供的最简单的一个模板：

➡ 示例 1

```
<!DOCTYPE html>
<html lang="zh-CN">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- 上述 3 个 meta 标签*必须*放在最前面，任何其他内容都*必须*跟随其后！ -->
    <title>Bootstrap 101 Template</title>
    <!-- 引入 Bootstrap 的 CSS 样式表 -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- 以下两个插件是用于使 IE8 支持 HTML5 和媒体查询 -->
    <!-- 注意：Respond.js 不支持 file:// 方式访问 -->
    <!--[if lt IE 9]>
      <script src="//cdn.bootcss.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="//cdn.bootcss.com/respond.js/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>你好，世界！</h1>
    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="//cdn.bootcss.com/jquery/1.11.3/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

这段代码演示了如何在 HTML5 中使用 Bootstrap。

Bootstrap 必须使用在 HTML5 中，因此，在模板的最上面使用 HTML5 代码：

```
<!DOCTYPE html>
```

这里面还要注意下面这行代码：

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

通常使用 Bootstrap 时，这段代码必须添加到 head 中，主要是因为在 CSS 中一般使用 px 作为单位，在桌面浏览器中 CSS 的 1 个像素往往都是对应着电脑屏幕的 1 个物理像素，这可能会造成一个错觉，那就是 CSS 中的像素就是设备的物理像素。但实际情况却并非如此，CSS 中的像素只是一个抽象的单位，在不同的设备或不同的环境中，CSS 中的 1px 所代表的设备物理像素是不同的。

在为桌面浏览器设计的网页中，不需要考虑这个问题，但在移动设备上，必须弄明白这

点。在早先的移动设备中，屏幕像素密度都比较低，如 iPhone 3，它的分辨率为 320*480，在 iPhone 3 上，一个 CSS 像素确实是等于一个屏幕物理像素的。后来随着技术的发展，移动设备的屏幕像素密度越来越高，从 iPhone 4 开始，分辨率提高了一倍，变成 640*960，但屏幕尺寸却没变化，这就意味着同样大小的屏幕上，像素却多了一倍。而如果不做处理，同样大小的字体，在 iPhone 4 上，会比在 iPhone 3 上缩小一半。其他的移动设备也有同样的问题。因此，在开发移动设备的网站时，最常见的一个动作就是把下面的代码添加到 head 标签中：

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0">
```

该 meta 标签的作用是让当前 viewport 的宽度等于设备的宽度，同时不允许用户手动缩放。通常最常用的是使 viewport 的宽度等于设备的宽度，而是否允许用户缩放可根据不同的需求设定，一般常用的属性写法就是示例 1 中所示的写法：

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

最简单的用法就是将这行代码直接复制到 head 中即可。

viewport 有 6 个属性如表 1.2 所示。

表 1.2 viewport 的属性

属性	描述
width	设置 viewport 的宽度，正整数，或"width-device"
initial-scale	设置页面的初始缩放值，为一个数字
minimum-scale	允许用户的最小缩放值，为一个数字
maximum-scale	允许用户的最大缩放值，为一个数字
height	设置 viewport 的高度，很少使用
user-scalable	是否允许用户进行缩放，值为"no"或"yes"，no 代表不允许，yes 代表允许

除了创建 HTML5 页面之外，要想在网站中使用 Bootstrap 首先要下载 Bootstrap 源文件，读者可以在网站上直接下载 Bootstrap 框架文件，Bootstrap 的官网地址是：<http://www.bootcss.com/>，在官网上有对 Bootstrap 各个版本的介绍，本书使用的是 v3.3.5 版，读者可在官网上跳转到 v3.3.5 版的页面进行下载，下载的页面地址是：<http://v3.bootcss.com/getting-started/#download>。在这个页面上可以下载 Bootstrap 的 v3.3.5 版本，如图 1.1 所示。



图 1.1 Bootstrap 下载页面

由图 1.1 可以看出,Bootstrap 下载页面有 3 个下载版本,分别是用于生产环境的 Bootstrap、Bootstrap 源码和 Sass 版本。如果只作为开发人员使用,则用于生产环境的 Bootstrap 就可以,本书也是使用的这一版本。

点击图 1.1 所示的“下载 Bootstrap”链接,会下载一个 bootstrap-3.3.5-dist.zip 的文件,将文件解压缩会得到 3 个文件夹,如图 1.2 所示。

其中 CSS 文件夹中存放的是 Bootstrap 的样式文件,如图 1.3 所示。js 文件夹中存放的是 Bootstrap 的 JavaScript 类库文件,如图 1.4 所示。

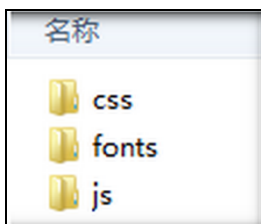


图 1.2 Bootstrap 文件夹

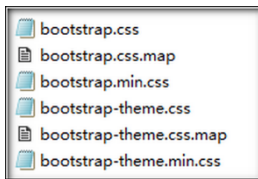


图 1.3 CSS 文件夹

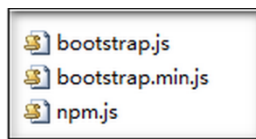


图 1.4 js 文件夹

图 1.3 中的 bootstrap.css 是 Bootstrap 设计好的样式文件,bootstrap.min.css 是经过压缩的样式文件。和 CSS 一样,JavaScript 也有一个压缩版的文件 bootstrap.min.js。使用 Bootstrap 时建议引入 bootstrap.min.css 和 bootstrap.min.js 两个文件。除了下载文件之外,还有另外一种方法引入 Bootstrap 所需要的文件,也就是使用 Bootstrap 中文网提供的免费 CDN 加速服务。使用时将如下代码引入到 head 之间即可。

```
<!--Bootstrap 核心 CSS 文件-->
<link rel="stylesheet" href="//cdn.bootcss.com/bootstrap/3.3.5/css/bootstrap.min.css">
<!--jQuery 文件。务必在 bootstrap.min.js 之前引入-->
<script src="//cdn.bootcss.com/jquery/1.11.3/jquery.min.js"></script>
<!--最新的 Bootstrap 核心 JavaScript 文件-->
<script src="//cdn.bootcss.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>
```

通过引入下载的 Bootstrap.min.css 以及引用 Bootstrap 中文网提供的免费 CDN 加速服务的地址就可以在网站中使用 Bootstrap 了。

操作案例：在页面中使用 Bootstrap

需求描述

在页面中使用 Bootstrap。

- 登录 Bootstrap 中文网, 下载用于生产环境的 Bootstrap, 在页面中添加 Bootstrap 文件引用。
- 使用 Bootstrap 中文网提供的免费 CDN 加速服务构建 Bootstrap 文件。

技能要点

- 会下载并创建 Bootstrap 页面。
- 会使用 CDN 加速服务构建 Bootstrap 文件。
- 理解 Bootstrap 的搭建和使用。

关键代码

```
<!--新 Bootstrap 核心 CSS 文件-->
<link rel="stylesheet" href="//cdn.bootcss.com/bootstrap/3.3.5/css/bootstrap.min.css">
<!--jQuery 文件。务必在 bootstrap.min.js 之前引入-->
<script src="//cdn.bootcss.com/jquery/1.11.3/jquery.min.js"></script>
<!--最新的 Bootstrap 核心 JavaScript 文件-->
<script src="//cdn.bootcss.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>
<!--设置移动视口-->
<meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0,
minimum-scale=1.0"/>
<link rel="stylesheet" href="../bootstrap.min.css"/>
```

实现步骤

- (1) 登录 <http://v3.bootcss.com/getting-started/#download>, 下载用于生产环境的 Bootstrap。
- (2) 解压 bootstrap-3.3.5-dist.zip 文件, 在 CSS 文件夹中提取 bootstrap.min.css 文件, 在 js 文件夹中提取 bootstrap.min.js 文件, 下载 jQuery 文件。
- (3) 在 HTML5 页面中编写代码:

```
<meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0,
minimum-scale=1.0"/>
```

按顺序引入 bootstrap.min.css、jQuery、bootstrap.min.js 文件。

2 Bootstrap 功能介绍

程序员虽然擅长编写复杂的应用, 但常对设计望而却步, 一个优秀的程序员设计出的页面难登大雅之堂也是经常发生的事情。比如程序员从不介意使用 Comic Sans 字体, 但设计行业却极为排斥它, 设计师或那些拥有美学情结的人往往对之不屑一顾。程序员想要制作一个精美的网站, 让自己的网站看起来更加吸引人, 必须有设计人员的配合。

可现实是, 写代码就是程序员工作的全部, 程序员对设计并不熟悉, 甚至有些畏惧。对于外行来说, 设计是由很多只可意会不可言传的规则以及所谓的设计灵感混合作用而成的, 知识壁垒很高。

然而, Bootstrap 诞生之后, 程序员经过简单的操作就可以让网站看上去更加专业, 虽然比不上设计师设计出来的效果, 但对于没有设计基础的人来说已经足够了。

2.1 Bootstrap 构成模块

Bootstrap 的构成模块从大的方面可以分为布局框架、页面排版、基本组件、jQuery 插件以及变量编译的 LESS 等部分。同时还有很多有趣的模块, 例如, 布局框架中的响应式布局, 页面排版中的 icon, 基本组件中的进度条, 菜单响应式展示等, 完全可以满足项目常用的交互效果。下面简单介绍一下 Bootstrap 中各模块的功能。

1. 页面布局

布局对于每个项目都必不可少。Bootstrap 创建出一套优秀的栅格布局, 而在响应式布局

中有更强大的功能，用法也相当简单，只需要按照 HTML 模板应用，让栅格布局成 1~12 列，能适应各种设备。使用这种栅格布局可轻松构建所需的布局效果。此外，改变模板中的类名就能实现不同的布局风格。例如，要实现常见的布局，只需在 HTML 中添加 `container` 类名，然后添加 `div` 元素，并给这个 `div` 元素添加 `class='row'`，然后在该 `div` 内部添加 `<div class="col-*-*"></div>` 即可实现所需要的栅格系统。

2. 页面排版

页面排版的好坏直接影响产品风格，也就是说页面设计是不是好看。在 Bootstrap 中，页面的排版都是从全局的概念上出发，定制了主体文本、段落文本、标题、导航风格、按钮、表单、表格等格式。

3. 基本组件

基本组件是 Bootstrap 的精华之一，其中都是开发者平时需要用到的交互组件。例如，网站导航、标签页、工具条、超大屏幕、徽章、分页栏、提示标签、多媒体对象、提示信息块和进度条等。这些组件都配有 jQuery 插件，运用它们可以大幅度提高用户的交互体验，使产品不再呆板、增强吸引力。

4. jQuery 插件

Bootstrap 组件仅仅是个开始。Bootstrap 自带 12 种 jQuery 插件，扩展了功能，可以给站点添加更多的互动。即使开发人员不是一名高级的 JavaScript 开发人员，也可以着手学习 Bootstrap 的 JavaScript 插件。利用 Bootstrap 数据 API (Bootstrap Data API)，大部分的插件可以在不编写任何代码的情况被触发。引用 Bootstrap 插件的方式有两种：

- 单独引用：使用 Bootstrap 的个别的*.js 文件。一些插件和 CSS 组件依赖于其他插件。如果单独引用插件，应先弄清这些插件之间的依赖关系。
- 编译（同时）引用：使用 `bootstrap.js` 或压缩版的 `bootstrap.min.js`。

不要尝试同时引用这两个文件，因为 `bootstrap.js` 和 `bootstrap.min.js` 都包含了所有的插件。

Bootstrap 中的 jQuery 插件主要用来帮助开发者实现与用户交互的功能。Bootstrap 提供了多种常见插件，所有的插件都依赖于 jQuery，所以必须在插进文件之前引用 jQuery。

（1）下拉框

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，可以通过与下拉菜单 (Dropdown) JavaScript 插件的互动来实现。如需使用下拉菜单，只需要在 `class.dropdown` 内加上下拉菜单即可。

（2）模态框 (Modal) 插件

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进行一些互动。子窗体可提供信息、交互等。

（3）滚动监听 (Scrollspy)

实现滚动条位置的效果，如在导航中有多个标签，用户单击其中一个标签，滚动条会自动定位到导航中标签对应的文本位置。

（4）标签页 (Tab)

这个插件能够快速实现本地内容的切换，动态切换标签页对应的本地内容。

（5）工具提示 (Tooltip)

一款优秀的 jQuery 插件，无需加载任何图片，采用 CSS3 新技术，动态显示 `data-attributes`

存储的标题信息。

5. 动态样式语言 LESS

LESS 是动态 CSS 语言,它基于 JavaScript 引擎或者服务器端对传统的 CSS 进行动态扩展,具有更强大的功能和更好的灵活性。基于 LESS,编辑 CSS 就可以像使用编程语言一样,定义变量、嵌入声明、混合模式、运算等。

Bootstrap 中有一套编辑好的 LESS 框架,开发者可以将其应用到自己的项目中,也可以通过 Less.js、Less.app 或 Node.js 等方法来编辑 LESS 文件。LESS 文件一旦编译,Bootstrap 框架就仅包含 CSS 样式,这意味着没有多余的图片、Flash 之类的元素。本书不对 LESS 进行讲解。

2.2 Bootstrap 的特色和功能介绍

Bootstrap 之所以能广泛流行,其易用性是很重要的一个原因。Twitter 在 GitHub 上提供了方便的自定义 Bootstrap 的工具,用户可以自由地选择想要的元素和组件,并且自定义每个颜色的具体数值,然后再让它自动生成一份属于自己的定制版的 Bootstrap。所有不需要的代码已经被去掉,以节省网络下载时间。虽然整个 Bootstrap 并不大,但是对于大流量网站来说,为每个用户节省 1KB 流量,总量也是很可观的。下面简单介绍一下 Bootstrap 的特色和功能,以便更详细地了解它。

2.2.1 Bootstrap 的特色

(1) 适应各种技术水平

Bootstrap 适应不同技术水平的从业者,无论是设计师,还是程序开发人员,无论是经验丰富的开发者,还是刚入门的“菜鸟”。使用 Bootstrap 既能开发简单的小东西,也能构造十分复杂的应用。

(2) 跨设备、跨浏览器

最初设想的 Bootstrap 只支持现代浏览器,不过新版本已经能支持所有主流浏览器,甚至包括 IE7。从 Bootstrap 2 开始,提供对平板电脑和智能手机的支持,Bootstrap 3 已经提出移动设备优先的理念。

(3) 提供 12 列栅格布局

栅格系统不是万能的,不过在应用的核心层有一个稳定和灵活的栅格系统确实可以让开发变得更简单。可以选用内置的栅格,或是自己手写。

(4) 支持响应式设计

从 Bootstrap 2 开始,提供完整的响应式特性。所有的组件都能根据分辨率和设备灵活缩放,从而提供一致性的用户体验。

(5) 样式化的文档

与其他前端开发工具包不同,Bootstrap 优先设计了一个样式化的使用指南,不仅用来介绍特性,更用以展示最佳实践、应用以及代码示例。

(6) 支持 HTML5

Bootstrap 支持 HTML5 标签和语法,要求建立在 HTML5 文档类型基础上进行设计和开发。

(7) 支持 CSS3

Bootstrap 支持 CSS3 所有属性和标准，逐步改进组件以达到最终效果。

Bootstrap 还有很多其他的特色，这里不再列出，读者在使用的时候可以自己体会。

2.2.2 媒体查询

Bootstrap 之所以能够在不同的设备以及不同的视口大小下显示不同的页面效果，是使用了一项叫做媒体查询（也叫做媒介查询）的技术。

媒体查询在 CSS 中是一项十分实用的技术，具体来说，就是可以根据客户端的介质和屏幕大小，提供不同的样式表或者只展示样式表中的一部分。通过响应式布局，可以达到只使用单一文件提供多平台的兼容性，省去了诸如浏览器判断之类的代码。

当然这种设计也存在着缺点，很多使用响应式布局的设计在适配移动端时大量使用 `display:none` 隐藏富媒体元素，这样势必会导致大量不必要的流量。因此，如果有较为重要的移动端需求，那么还是开发专门的移动版页面为好。不过，对于诸如内容较少的页面或者单页式网站来说，响应式布局依然不失为一种好方法。

如今屏幕分辨率的范围已经从 320px（iPhone）扩展到 2560px（大显示器），甚至更高。用户不单单在桌面电脑上浏览网站，还会使用移动电话、小型笔记本、平板设备（比如 iPad 或者 Playbook）来访问互联网，所以传统的固定宽度设计不再适用了。Web 设计需要有自适应能力。页面布局要可以自动地去适应所有的分辨率和设备。CSS3 媒体查询的目的就是适应跨浏览器的响应式设计。

所谓响应式设计就是同样的 html 结构和 CSS 样式在不同的视口下显示不同的界面效果，从而适应不同的设备。

在页面中使用媒体查询有两种方式：

一种是在 `<head>` 链接 CSS 文件时提供判断语句，选择性加载不同的 CSS 文件：

```
<link rel="stylesheet" href="middle.css" media="screen and (min-width:400px)">
```

该行代码的作用是在满足 media 的判断语句 `screen and(min-width:400px)`，即在屏幕最小宽度不小于 400px 的介质上面使用 `middle.css`。

第二种方式是在 CSS 文件中分段书写不同设备的代码：

```
/* CSS Code 代码段 1*/
```

```
@media screen and (min-width:600px) { /* CSS Code 代码段 2*/ }
```

```
@media screen and (max-width:599px) { /* CSS Code 代码段 3*/ }
```

写在 `@media` 语句段外的是共用代码（Code1），任何设备都可使用。第一个 `@media` 语句段是屏幕以及最小宽度 600px，使用代码段 2。第二个 `@media` 语句段是屏幕以及最大宽度 599px，使用代码段 3。

除了能够指定视口的宽度之外还可以依据设备进行限制，如下代码：

```
@media only screen and (max-width:640px) {
```

```
...
```

```
}
```

该代码段表示这段 CSS 代码只适用于彩色设备并且最大宽度为 640px。上段代码中使用

了 `only` 操作符, `only` 操作符用来限定某种特别的媒体类型, 对于支持 Media Queries 的移动设备来说, 如果存在 `only` 关键字, 移动设备的 Web 浏览器会忽略 `only` 关键字并直接根据后面的表达式应用样式文件。对于不支持 Media Queries 的设备但能够读取 Media Type 类型的 Web 浏览器, 遇到 `only` 关键字时会忽略这个样式文件。

`not` 操作符用来排除某些特定的设备, 比如 `@media not print` (非打印设备)。`all` 操作符表示不进行限制, 使用所有设备。

CSS 的媒体类型如表 1.3 所示。

表 1.3 媒体类型

媒体类型	描述
all	用于所有设备
print	用于打印机和打印预览
screen	用于电脑屏幕、平板电脑、智能手机等
speech	应用于屏幕阅读器等发声设备

`@media` 是非常有用的, 使用 `@media` 定义查询, 可以针对不同的媒体类型定义不同的样式, 从而可以针对不同的屏幕尺寸设置不同的样式。同时在用户重置浏览器大小的过程中, 页面也会根据浏览器的宽度和高度重新渲染页面。如果不同的代码段有冲突或者重叠, 会按照 CSS 原本的代码优先级排序, 即后方代码替代前方代码等。

示例 2 演示了 CSS 媒体查询的使用方法。

➡ 示例 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>媒体查询</title>
  <style>
    @media screen and (max-width:980px) {
      /*样式一*/
      #pagewrap {
        width:95%;
      }
    }
    /*样式二*/
    @media screen and (max-width:650px) {
      #pagewrap {
        width:100%;
      }
    }
    /*样式三*/
```



```
@media screen and (max-width:480px) {  
    #pagewrap {  
        width:auto;  
    }  
}  
  
</style>  
</head>  
<body>  
<div id="pagewrap"></div>  
</body>  
</html>
```

示例 2 演示了媒体查询的用法，当视口宽度在 980px 以下时使用样式一，当视口最大为 650px 时使用样式二，当视口小于 480px 时使用样式三。例如视口宽度为 575px，满足样式一和样式二的样式，但不满足样式三的样式，所以会使用样式一和样式二的样式，由于样式一和样式二有冲突，因此使用样式二的样式。这样就为不同的设备显示不同的样式了。

2.2.3 Bootstrap 主要功能

Bootstrap 的目的是提供一个便捷工具，方便快速开发项目，样式部分使用 LESS 编写，也提供了一些 jQuery 插件形式的扩展，在样式方面，Bootstrap 提供了如下解决方案：

(1) 栅格系统

栅格系统使用起来十分简单，仅仅可以通过配置几个参数，自定义栅格。同时还可以依据 class 参数适用于不同的设备。

(2) 布局

布局主要包括一个固定宽度的居中布局、一个可变宽度的浮动布局。

(3) 字体样式

字体风格比较明显，<title>、、、<label>等语义标签都配了一些默认样式。然后通过样式属性实现不同的字体样式效果，如示例 3 所示。

➡ 示例 3

```
<!DOCTYPE html>  
<html lang="zh-CN">  
<head>  
    <meta charset="UTF-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
    <!-- 上述 3 个 meta 标签*必须*放在最前面，任何其他内容都*必须*跟随其后! -->  
    <title>Bootstrap 基本模板</title>  
    <!--Bootstrap-->  
    <link href="../../bootstrap.min.css" rel="stylesheet">  
    <style>  
        .container {
```

```

        margin-top: 20px;
    }
    .label {
        font-size: 20px;
    }
</style>
<!--以下两个插件是用于在 IE8 中支持 HTML5 元素和媒体查询-->
<!--注意: Respond.js 不支持 file://方式访问-->
<!--[if lt IE 9]>
<script src="//cdn.bootcss.com/html5shiv/3.7.2/html5shiv.min.js"></script>
<script src="//cdn.bootcss.com/respond.js/1.4.2/respond.min.js"></script>
<![endif]-->
</head>
<body>
<div class="container">
    <span class="label label-default">默认样式</span>
    <span class="label label-success">成功样式</span>
    <span class="label label-warning">警告样式</span>
    <span class="label label-danger">危险样式</span>
    <span class="label label-primary">重要样式</span>
    <span class="label label-info">信息样式</span>
</div>
<!--如果要使用 Bootstrap 的 js 插件就必须引入 jQuery 插件-->
<script src="../../jquery-1.11.1.min.js"></script>
<script src="../../bootstrap.min.js"></script>
</body>
</html>

```

运行效果如图 1.5 所示。



图 1.5 字体样式

(4) 多媒体展现

多媒体列表比较简单, 定义了 3 种缩略图尺寸: 330px*230px、210px*150px 和 90px*90px。

(5) 表单

Bootstrap 对表单进行了比较充分的定制, 风格比较鲜明, 如圆角输入框, 正确、错误的状态, 表单的字号, 前缀字符, 输入、复选框等。要想实现更个性的解决方案, 需要与 JavaScript 配合设计。表单的按钮设计如表 1.4 所示。

表 1.4 按钮样式

按钮样式类	描述
.btn	为按钮添加基本样式
.btn-default	默认/标准按钮
.btn-primary	原始按钮样式（未被操作）
.btn-success	表示成功的动作
.btn-info	该样式可用于要弹出信息的按钮
.btn-warning	表示需要谨慎操作的按钮
.btn-danger	表示一个危险动作的按钮操作
.btn-link	让按钮看起来像个链接（仍然保留按钮行为）

使用按钮类的方式很简单，如下代码所示：

```

<!--标准的按钮-->
<button type="button" class="btn btn-default">默认按钮</button>
<!--提供额外的视觉效果，标识一组按钮中的原始动作-->
<button type="button" class="btn btn-primary">原始按钮</button>
<!--表示一个成功的或积极的动作-->
<button type="button" class="btn btn-success">成功按钮</button>
<!--信息警告消息的上下文按钮-->
<button type="button" class="btn btn-info">信息按钮</button>
<!--表示应谨慎采取的动作-->
<button type="button" class="btn btn-warning">警告按钮</button>
<!--表示一个危险的或潜在的负面动作-->
<button type="button" class="btn btn-danger">危险按钮</button>
<!--并不强调是一个按钮，看起来像一个链接，但同时保持按钮的行为-->
<button type="button" class="btn btn-link">链接按钮</button>
    
```

效果如图 1.6 所示。



图 1.6 按钮样式

（6）导航

网站的全局导航栏保持一致，使用样式可实现背景色渐变，将导航栏固定在头部和尾部，并且不需要考虑浏览器兼容问题。此外，还实现了提示、警告、弹出对话框等设计风格的统一。甚至同一个导航在不同的视口大小显示不同的风格，如示例 4 所示。

➡ 示例 4

```

<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>导航</title>
  <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0,
minimum-scale=1.0"/>
  <link rel="stylesheet" href="../bootstrap.min.css"/>
</head>
<body>
<div class="navbar navbar-inverse navbar-fixed-top">
  <div class="navbar-header">
    <!-- 自适应隐藏导航展开按钮 -->
    <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target=
      "#bs-example-navbar-collapse-1">
      <span class="sr-only">Toggle navigation</span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
    </button>
    <!-- 导航条 LOGO -->
    <a class="navbar-brand" href="#">Boot</a>
  </div>
  <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">首页</a></li>
      <li><a href="#">博文</a></li>
      <li><a href="#">留言</a></li>
      <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown">博客信息
          <span class="caret"></span></a>
        <ul class="dropdown-menu" role="menu">
          <li><a href="#">我的博客</a></li>
          <li><a href="#">我的好友</a></li>
          <li><a href="#">我的信箱</a></li>
        </ul>
      </li>
    </ul>
    <form class="navbar-form navbar-left" role="search">
      <div class="form-group">
        <input type="text" class="form-control" placeholder="Search">
      </div>
    </form>
  </div>
</div>

```

```
</div>
<button type="submit" class="btn btn-default">Submit</button>
</form>
<ul class="nav navbar-nav navbar-right">
  <li><a href="#">登入</a></li>
  <li><a href="#">退出</a></li>
</ul>
</div>
</div>
<script src="../jquery-1.10.2.min.js"></script>
<script src="../bootstrap.min.js"></script>
</body>
</html>
```

示例 4 的页面效果如图 1.7 和图 1.8 所示，注意图 1.8 的效果是当用户点击右上角的菜单展开后的效果。

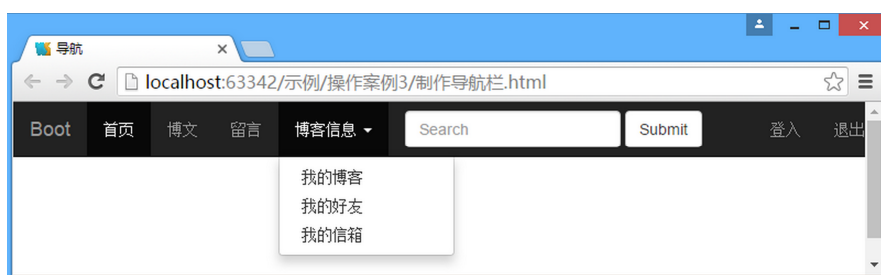


图 1.7 大屏幕设备的导航效果

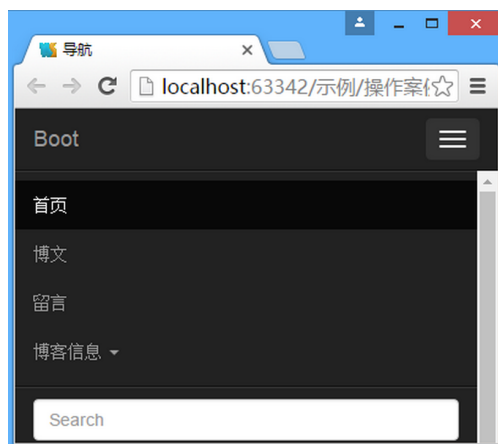


图 1.8 小屏幕设备导航展开效果

同样的代码在不同大小的屏幕上显示的效果完全不同，如果要手写样式，需要写比较复杂的代码，而使用 Bootstrap 仅需要添加几个必需的类即可实现。

以上样式的部分是 Bootstrap 框架的核心。在代码上，基本把样式重置与定制都做了，除了一些比较明显的组件（如面包屑、翻页等），基本都直接用标签作选择器。有许多风格是利

用 CSS3 样式属性设计出来的,如背景色渐变与圆角。因此,若为了节省时间且不介意早期浏览器的效果,Bootstrap 是个好选择,因为它省时省力,而且美观大方。在样式之外,Bootstrap 还提供了十几个常用的 JavaScript 插件,如模态对话框、下拉菜单、滚动监听、标签页、工具提示等 jQuery 插件。

2.3 Bootstrap 优秀插件

由于 Bootstrap 的易用性、兼容性以及优秀的响应式设计、栅格系统以及自定义的 jQuery 插件等优势,因此出现了很多以 Bootstrap 为基础的扩展技术插件,下面列举几个比较著名的插件:

1. Sco.js

Sco.js 不仅能够增强 Bootstrap 中现有的 js 组件,而且自身也带有部分插件,Sco.js 中的插件可以和 Bootstrap 一起使用,也可以单独使用。Sco.js 中还包含了 Bootstrap 中没有的插件。所有插件都进行了单元测试,并且有生产环境的使用案例。每个插件都可以通过 `data-attributes data-trigger="pluginName"` 或者 js 代码 `var $modal = $.scojs_modal({...})` 的方式使用。

2. Chart.js

Chart.js 是一个为设计者和开发者准备的简单的面向对象的图表绘制工具库。使用时需要在页面中引入 Chart.js 文件。此工具库在全局命名空间中定义了 Chart 变量。然后创建 Chart 对象。在页面中添加 canvas,即可在 canvas 中绘制各种图表。如曲线图、柱状图、雷达图、饼图等。

3. jQuery UI Bootstrap

jQuery UI Bootstrap 是一个基于 Bootstrap 样式的 jQuery UI 控件,允许在使用 jQuery UI 控件时充分利用 Bootstrap 的样式,而且不会出现样式不统一的现象,使 Bootstrap 和 jQuery UI 可以完美融合。

4. Flat UI

Flat UI 是一套精美的扁平风格 UI 工具包,基于 Twitter Bootstrap 实现。这套界面工具包含许多基本的和复杂的 UI 部件,例如按钮、输入框、组合按钮、复选框、单选按钮、标签、菜单、进度条和滑块、导航元素等。

5. Metro UI CSS

Metro UI CSS 是一套用来创建类似于 Windows 8 Metro UI 风格网站的样式。这组风格被开发成一个独立的解决方案。Metro UI CSS 包含两种类型的许可证:MIT 和 Commercial。

6. HTML5 Boilerplate

HTML5 Boilerplate 是一个由 Paul Irish (Google Chrome 开发人员、jQuery 项目成员、Modernizr 作者、yayQuery 播客主持人) 主导的前端开发模版。HTML5 Boilerplate 是一套具有非常多先进特性的框架。

2.4 Bootstrap 版本变化

了解 Bootstrap 版本变化的过程,能够更直观地了解 Bootstrap 在 Web 开发中的地位和价

值。通过其版本演变和功能变化，能够把握未来 Web 前端开发技术的发展方向，对于学习 Bootstrap 也有很大的帮助。

1. Bootstrap1

2011 年 8 月，推特（Twitter）推出了用于快速搭建 Web 应用程序的轻量级前端开发工具 Bootstrap，Bootstrap 是一套用于开发 Web 应用程序，符合 HTML 和 CSS 各项标准和规范的库。Bootstrap 由动态 CSS 语言 LESS 写成，经过编译后，Bootstrap 就是众多 CSS 的合集，而 Bootstrap 本身就是一套 CSS 库。

Bootstrap 便于开发团队快速部署 Web 应用程序，对于不懂前端或前端基础弱的个人或团队来讲，在某种程度上 Bootstrap 可以让他们在没有设计师的情况下完成一个 UI 较为理想的作品。

2. Bootstrap2

Bootstrap2 在 Bootstrap1 的基础上修改了一些网页元素的默认样式，除去了 Bootstrap1 中的若干个 bug，同时完善了说明文档。Bootstrap2 在原有特性的基础上着重改进了用户的体验和交互性，比如新增加的媒体展示功能，适用于智能手机上多种屏幕规格的响应式布局，另外还新增了 12 款 jQuery 插件，可以满足 Web 页面常用的用户体验和交互功能。Bootstrap2 的主要特点如下：

- 添加了响应式设计特性。
- 在文档中增加了视网膜屏幕相关的资源。
- 增加了 HTML5 Boilerplate 打印样式。
- 将 placeholder.it（用于创建占位图片）更换为 Holder.js。
- 修复了按钮组中字体大小 bug。
- 重构了 popover 箭头，修复了 IE8 中显示不正常的 bug。
- 更新了 popover 插件，在 .popover-content 中移除了 <p> 标记。popover 文本和 HTML 可直接插入到 .popover-content。
- 空标签自动折叠。

3. Bootstrap3

Bootstrap3 相对于 Bootstrap2 的改动比较大，在 Bootstrap3 中，重写了整个框架，使其一开始就是对移动设备优先的。这次不是简单的增加一些可选的针对移动设备的样式，而是直接融合进了框架的内核中。也就是说，Bootstrap3 是移动设备优先的。针对移动设备的样式融合进了框架的每个角落，而不是一个单一的文件。当然为了确保适当的绘制和触屏缩放，需要在 <head> 之中添加 viewport 元数据标签。

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

4. Bootstrap4

2015 年 8 月 19 日发布了 Bootstrap4 的内测版本，Bootstrap4 是一次重大更新，几乎涉及每行代码。Bootstrap4 中有很多重大的更新，下面是一些颇受关注的更新亮点：

- 从 LESS 迁移到 Sass：现在，Bootstrap 已加入 Sass 的大家庭中。得益于 LibSass，Bootstrap 的编译速度比以前更快。
- 改进网格系统：新增一个网格层适配移动设备，并整顿语义混合。
- 支持选择弹性盒模型（flexbox）：这是一项划时代的功能——只要修改一个 Boolean 变量，就可以利用 flexbox 的优势快速布局。

- 废弃了 Wells、Thumbnails 和 Panels，使用 Cards 代替：Cards 是一个全新的概念，但使用起来与 Wells、Thumbnails 及 Panels 很像，且更方便。
- 将所有 HTML 重置样式表整合到 Reboot 中：在用不了 Normalize.css 的地方可以使用 Reboot，它提供了更多选项。例如 box-sizing: border-box、margin tweaks 等都存放在一个单独的 Sass 文件中。
- 新的自定义选项：不再像上个版本一样，将渐变、淡入淡出、阴影等效果分放在单独的样式表中，而是将所有选项都移到一个 Sass 变量中。想要给全局或考虑不到的角落定义一个默认效果，只要更新变量值，然后重新编译就可以了。
- 不再支持 IE8，使用 rem 和 em 单位：放弃对 IE8 的支持意味着开发者可以放心地利用 CSS 的优点，不必研究 CSS hack 技巧或回退机制了。使用 rem 和 em 代替 px 单位，更适合做响应式布局，控制组件大小。如果要支持 IE8，只能继续用 Bootstrap3。
- 重写所有 JavaScript 插件：为了利用 JavaScript 的新特性，Bootstrap4 用 ES6 重写了所有插件。现在提供 UMD 支持、泛型拆解方法、选项类型检查等特性。
- 改进工具提示和 popovers 自动定位：这部分要感谢 Tether 工具的帮助。
- 改进文档：所有文档以 Markdown 格式重写，添加了一些方便的插件组织示例和代码片段，文档使用起来会更方便，搜索的优化工作也在进行中。
- 更多变化：支持自定义窗体控件、空白和填充类，此外还包括新的实用程序类等。

除了发布 Bootstrap4 alpha 外，官方还发布了 Bootstrap 主题。这些主题自己就有很多工具集，和 Bootstrap 本身一样。刚开始发布的主题有：dashboard、application 和 marketing，使用 multiple-use license 协议。可登录 <http://themes.getbootstrap.com/> 查看主题，如图 1.9 所示。



图 1.9 Bootstrap4 主题

3 Bootstrap 优秀网站示例

当前有很多优秀的网站使用的是 Bootstrap 框架。如 Readly (<https://cn.readly.com/>)，该网站采用 Bootstrap 框架，页面精美，可适应不同的使用设备，图 1.10 和图 1.11 分别演示了在 PC 端和移动端的效果。

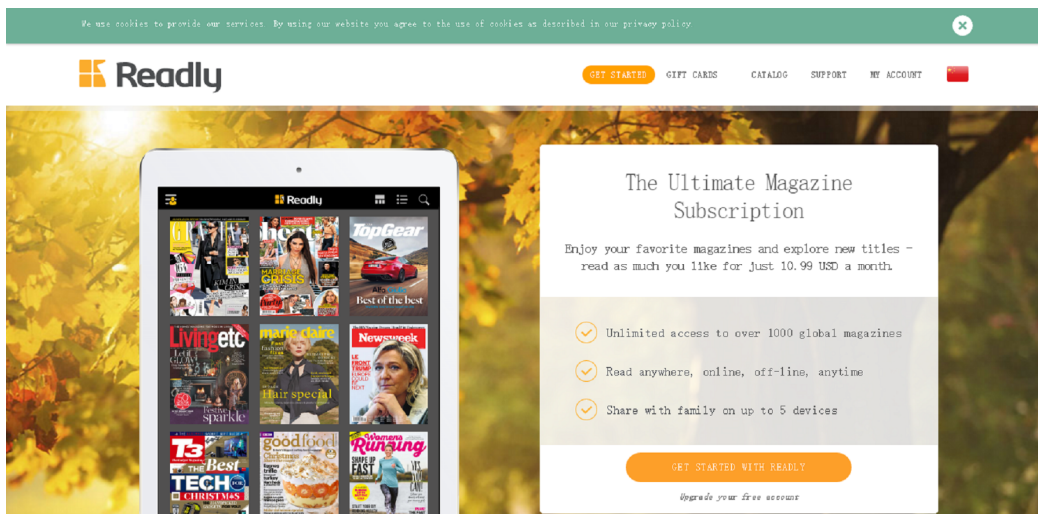


图 1.10 在 PC 端的首页效果

同样的代码，使用 Bootstrap 可以显示完全不同的效果，下图显示的是移动端的效果。

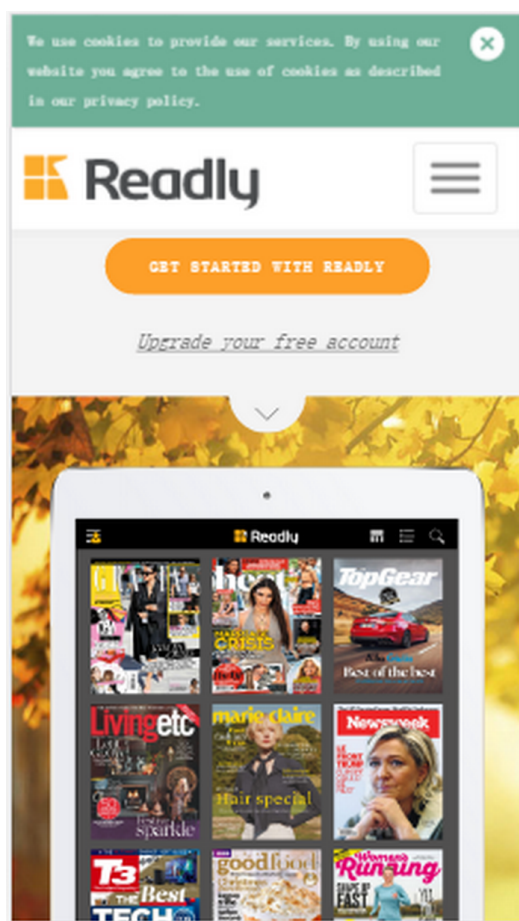


图 1.11 在移动端显示效果

本章总结

- Bootstrap 是目前最流行、用得最广泛的一款框架。它是基于 HTML5 和 CSS3 开发的，在 jQuery 的基础上进行了更为个性化和人性化的完善，开发人员可以拿来直接使用，极大方便了用户开发。
- 使用 Bootstrap 可以将从官网下载的 bootstrap.min.css 样式文件和 bootstrap.min.js 两个文件引入网页。还有另外一种方法引入 Bootstrap 所需要的文件，也就是使用 Bootstrap 中文网提供的免费 CDN 加速服务。
- Bootstrap 主要通过 CSS 组件和 JavaScript 组件实现响应式设计，注意 Bootstrap 并不仅仅有 CSS 组件和 JS 插件。
- CSS 即层叠样式，是一种用来美化 HTML 的一种语言，在网页设计中有很多的应用，
- CSS 目前最新版本为 CSS3。
- 使用 @media 定义查询，可以针对不同的媒体类型定义不同的样式，从而可以针对不同的屏幕尺寸设置不同的样式。

本章作业

1. 请描述如何在页面中使用 Bootstrap?
2. 请说明 CSS 媒介查询的作用以及优点?
3. 请举例说一下你所了解的 Bootstrap 网站。
4. 请说出 Bootstrap 新版本的特点。
5. 请登录课工场，按要求完成预习作业。