

## 本章学习目标

- 了解 jQuery
- 了解 Bootstrap

通过 HTML、CSS、JavaScript 原生代码编写 Web 客户端会出现代码量大、杂乱等缺点,因此 Web 客户端还可以通过 jQuery 和 Bootstrap 两个框架来简化开发,从而提高开发效率。本章主要讲解 jQuery 和 Bootstrap 的内容。

## 3.1 jQuery

在 Web 客户端开发中直接使用 JavaScript 进行开发,虽然工作都能完成,但效率明显不高。jQuery 将一些常用的 JavaScript 内容进行了封装,因此使用 jQuery 可以轻松、高效地对 Web 客户端进行开发。

### 3.1.1 jQuery 简介与使用

jQuery 提供一种简便的 JavaScript 设计模式来实现 HTML 文档操作、事件处理、动画设计和 Ajax 交互的优化。jQuery 的设计宗旨是“Write Less;Do More”,即“用最少的代码,完成更多的事情”。

jQuery 的使用步骤如下:

- (1) 下载 jQuery,并放在项目的 static 文件夹中。
- (2) 在文件中引入 jquery-3.2.1.min.js 文件。
- (3) 使用其中封装的函数,直接编写代码。

接下来演示 jQuery 的使用,如例 3-1 所示。

#### 【例 3-1】 jQuery 的使用

```
1 <!DOCTYPE html >
2 <html lang = "en">
3 <head >
4     <meta charset = "UTF-8">
5     <title> jquery 使用</title>
6     <script src = "static/js/jquery-3.2.1.min.js"></script >
7     <script type = "text/JavaScript">
8         $(document).ready(function () {
```

```
9         $("p").click(function () {
10             $(this).hide();
11         });
12     });
13 </script >
14 </head >
15 <body >
16 <div >
17     <span>单击千锋或 1000phone, 立即隐藏, 单击好程序员不隐藏</span >
18 </div >
19 <div >
20     <p>千锋</p >
21     <p>1000phone </p >
22 </div >
23 <div >
24     <span>好程序员</span >
25 </div >
26 </body >
27 </html >
```

例 3-1 运行结果如图 3.1 所示。



图 3.1 例 3-1 运行结果图

单击【千锋】, 运行结果如图 3.2 所示。



图 3.2 单击【千锋】运行结果

再单击【1000phone】, 运行结果如图 3.3 所示。

单击【好程序员】则没有变化。



图 3.3 单击【1000phone】运行结果

例 3-1 代码是 jQuery 的简单使用,使< p>标签内的内容单击以后隐藏,其他内容不作任何处理。

### 3.1.2 jQuery 选择器

jQuery 中的选择器与 CSS 中类似,只是 jQuery 中使用更丰富,具体如表 3.1 所示。

表 3.1 jQuery 选择器

选 择 器	举 例	作 用
*	\$ (" * ")	所有元素
# id	\$ (" # lastname")	id = "lastname" 的元素
. class	\$ (". intro")	所有 class = "intro" 的元素
element	\$ (" p")	所有< p>元素
. class, class	\$ (". intro. demo")	所有 class = "intro" 且 class = "demo" 的元素
:first	\$ (" p: first")	第一个< p>元素
:last	\$ (" p: last")	最后一个< p>元素
:even	\$ (" tr: even")	所有偶数< tr>元素
:odd	\$ (" tr: odd")	所有奇数< tr>元素
:eq(index)	\$ (" ul li: eq(3)")	列表中的第四个元素(index 从 0 开始)
:gt(no)	\$ (" ul li: gt(3)")	列出 index 大于 3 的元素
:lt(no)	\$ (" ul li: lt(3)")	列出 index 小于 3 的元素
:not(selector)	\$ (" input: not( : empty)")	所有不为空的 input 元素
:header	\$ (": header")	所有标题元素< h1> - < h6>
:contains(text)	\$ (": contains('W3School')")	包含指定字符串的所有元素
:empty	\$ (": empty")	无子(元素)节点的所有元素
:hidden	\$ (" p: hidden")	所有隐藏的< p>元素
:visible	\$ (" table: visible")	所有可见的表格
s1, s2, s3	\$ (" th, td, . intro")	所有带有匹配选择的元素
[attribute]	\$ (" [href]")	所有带有 href 属性的元素

### 3.1.3 jQuery action

jQuery action 实质上指的是 jQuery 的行为,即通过某些方法使整个网站实现动态效果,增加美感,最终达到开发要求。

#### 1. jQuery 效果

jQuery 封装了实现网页动态效果的一些方法,具体如表 3.2 所示。

表 3.2 jQuery 效果方法

方 法	描 述	举 例
hide(speed, callback)	隐藏 id 为 hide 的 <p> 元素内的内容	<code>\$("#hide").click(function(){ \$("#p").hide();});</code>
show(speed, callback)	显示 id 为 show 的 <p> 元素内的内容	<code>\$("#show").click(function(){ \$("#p").show();});</code>
toggle(speed, callback)	显示被隐藏的元素, 并隐藏已显示的元素	<code>\$("#button").click(function(){ \$("#p").toggle();});</code>
fadeIn(speed, callback)	淡入已隐藏的元素	<code>\$("#button").click(function(){ \$("#div1").fadeIn(); \$("#div2").fadeIn(1000);});</code>
fadeOut(speed, callback)	淡出可见元素	<code>\$("#button").click(function(){ \$("#div1").fadeOut(); \$("#div2").fadeOut(1000);});</code>
fadeToggle(speed, callback)	元素已淡出, 则 fadeToggle() 会向元素添加淡入效果; 元素已淡入, 则 fadeToggle() 会向元素添加淡出效果	<code>\$("#button").click(function(){ \$("#div1").fadeToggle(); \$("#div2").fadeToggle("slow"); \$("#div3").fadeToggle(3000);});</code>
fadeTo(speed, opacity, callback)	允许渐变为给定的不透明度(值介于 0 与 1 之间)	<code>\$("#button").click(function(){ \$("#div1").fadeTo("slow", 0.15);});</code>
slideDown(speed, callback)	向下滑动元素	<code>\$("#flip").click(function(){ \$("#panel").slideDown();});</code>
slideUp(speed, callback)	向上滑动元素	<code>\$("#flip").click(function(){ \$("#panel").slideUp();});</code>
slideToggle(speed, callback)	元素向下滑动, 则可向上滑动它们, 元素向上滑动, 则可向下滑动它们	<code>\$("#flip").click(function(){ \$("#panel").slideToggle();});</code>

jQuery 中还有很多实现动画的方法, 可以极大地丰富网页内容。

接下来以 hide() 与 show() 方法为例来演示 jQuery 的效果实现, 如例 3-2 所示。

### 【例 3-2】 jQuery 的效果实现

```

1  <!DOCTYPE html >
2  <html >
3  <head >
4  <meta charset = "utf - 8">
5  <script src = "static/js/jquery - 3.2.1.min.js"></script >
6  <script >
7  $(document).ready(function(){
8      $("#hide").click(function(){
9          $("#p").hide();
10     });
11     $("#show").click(function(){
12         $("#p").show();

```

```
13     });
14 });
15 </script>
16 </head>
17 <body>
18 <p>如果单击“隐藏”按钮,内容将会被隐藏。</p>
19 <button id="hide">隐藏</button>
20 <button id="show">显示</button>
21 </body>
22 </html>
```

例 3-2 运行结果如图 3.4 所示。



图 3.4 例 3-2 运行结果

单击【隐藏】按钮之后运行结果如图 3.5 所示。



图 3.5 单击【隐藏】按钮运行结果

再单击【显示】按钮显示结果如图 3.6 所示。



图 3.6 再单击【显示】按钮运行结果

例 3-2 是实现单击不同的按钮显示不同效果的功能。第 5 行是引入 jQuery。第 8~10 行是当 id 为 hide 的按钮单击以后将 <p> 标签内容进行隐藏。第 11~13 行是当 id 为 show 的按钮单击以后将 <p> 标签中隐藏的内容显示出来。

同理,大家可以自己动手尝试编写实现其他效果。

## 2. jQuery 操作 DOM

上一节 JavaScript 内容中讲解了对 DOM 对象的操作方法,而 jQuery 对这些操作方法都进行了封装,从而使得操作也更丰富,常用的操作方法如表 3.3 所示。

表 3.3 jQuery 操作 DOM 常用方法

方 法	描 述
.addClass(元素)	为每个匹配的元素添加指定的样式类名
.after(元素)	在匹配元素集合中的每个元素后面插入参数所指定的内容,作为其兄弟节点
.append(元素)	在每个匹配元素里面的末尾处插入参数内容
.attr(元素)	获取匹配的元素集合中的第一个元素的属性的值。设置每一个匹配元素的一个或多个属性
.appendTo(元素)	将匹配的元素插入到目标元素的最后面(内部插入)
.css(元素)	获取匹配元素集合中的第一个元素的样式属性的值。设置每个匹配元素的一个或多个 CSS 属性
.empty(元素)	从 DOM 中移除集合中匹配元素的所有子节点
.html()	获取集合中第一个匹配元素的 HTML 内容。设置每一个匹配元素的 html 内容
.position()	获取匹配元素中第一个元素的当前坐标,相对于 offset parent 的坐标
.remove(元素)	将匹配元素集合从 DOM 中删除
.text(元素)	得到匹配元素集合中每个元素的合并文本,包括他们的后代设置匹配元素集合中每个元素的文本内容为指定的文本内容

接下来以 append()方法为例来演示 jQuery 操作 DOM,如例 3-3 所示。

#### 【例 3-3】 jQuery 操作 DOM

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head>
4      <meta charset = "UTF - 8">
5      <script src = "static/js/jquery - 3.2.1.min.js"></script >
6      <script >
7          $(document).ready(function(){
8              $("p").append("<b>用良心做教育</b>");
9          });
10     </script >
11 </head >
12 <body >
13     <p>千锋教育: </p>
14 </body >
15 </html >

```

例 3-3 运行结果如图 3.7 所示。



图 3.7 例 3-3 运行结果

例 3-3 是实现在文档末尾添加内容的功能。第 5 行是引入 jQuery。第 7~9 行是在 <p> 标签的内容后添加字体加粗的“用心做教育”。第 13 行是 <body> 中的 <p> 标签,内容是“千锋教育:”。

同理,大家可以自己动手尝试编写实现其他操作方法。

jQuery 框架可以很好地丰富网页,原因是它封装了许多操作网页的方法,大家可以通过 jQuery 官网对其进行深入的研究学习。

### 3.1.4 jQuery 事件

事件是页面对不同访问者的响应。事件处理程序指的是当 HTML 中发生某些事件时所调用的方法。接下来讲解 jQuery 事件方法,具体如表 3.4 所示。

表 3.4 jQuery 事件方法

方 法	描 述
blur([[data],fn])	触发或将函数绑定到指定元素的 blur 事件
ready(fn)	文档就绪事件(当 HTML 文档就绪可用时)
on(eve,[selector],[data],fn)	在选择元素上绑定一个或多个事件的事件处理函数
off(eve,[selector],[fn])	在选择元素上移除一个或多个事件的事件处理函数
one(type,[data],fn)	向匹配元素添加事件处理器。每个元素只能触发一次该处理器
trigger(type,[data])	所有匹配元素的指定事件
triggerHandler(type,[data])	第一个被匹配元素的指定事件
hover([over,]out)	模仿悬停事件(鼠标移动到一个对象上面及移出这个对象)的方法。这是一个自定义的方法,它为频繁使用的任务提供了一种“保持在其中”的状态
load()	触发或将函数绑定到指定元素的 load 事件
toggle([speed],[easing],[fn])	绑定两个或多个事件处理器函数,当发生轮流的 click 事件时执行
change([[data],fn])	触发或将函数绑定到指定元素的 change 事件
click([[data],fn])	触发或将函数绑定到指定元素的 click 事件
dblclick([[data],fn])	触发或将函数绑定到指定元素的 double click 事件
error([[data],fn])	触发或将函数绑定到指定元素的 error 事件
focus([[data],fn])	触发或将函数绑定到指定元素的 focus 事件
focusin([data],fn)	当元素获得焦点时,触发 focusin 事件
focusout([data],fn)	当元素失去焦点时触发 focusout 事件
keydown([[data],fn])	触发或将函数绑定到指定元素的 key down 事件
keypress([[data],fn])	触发或将函数绑定到指定元素的 key press 事件
keyup([[data],fn])	触发或将函数绑定到指定元素的 key up 事件
mousedown([[data],fn])	触发或将函数绑定到指定元素的 mouse down 事件
mouseenter([[data],fn])	触发或将函数绑定到指定元素的 mouse enter 事件
mouseleave([[data],fn])	触发或将函数绑定到指定元素的 mouse leave 事件
mousemove([[data],fn])	触发或将函数绑定到指定元素的 mouse move 事件
mouseout([[data],fn])	触发或将函数绑定到指定元素的 mouse out 事件
mouseover([[data],fn])	触发或将函数绑定到指定元素的 mouse over 事件
mouseup([[data],fn])	触发或将函数绑定到指定元素的 mouse up 事件
resize([[data],fn])	触发或将函数绑定到指定元素的 resize 事件

方 法	描 述
scroll([[data],fn])	触发或将函数绑定到指定元素的 scroll 事件
select([[data],fn])	触发或将函数绑定到指定元素的 select 事件
submit([[data],fn])	触发或将函数绑定到指定元素的 submit 事件
unload([[data],fn])	触发或将函数绑定到指定元素的 unload 事件

接下来以 on() 方法为例来演示 jQuery 事件操作, 如例 3-4 所示。

#### 【例 3-4】 jQuery 事件操作

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head >
4      <meta charset = "UTF - 8">
5      <script src = "static/js/jquery - 3.2.1.min.js"></script >
6      <script >
7          $(document).ready(function() {
8              $("p").on("click", function () {
9                  alert( $(this).text());
10             });
11         });
12     </script >
13 </head >
14 <body >
15     <p>千锋教育</p >
16     <p>好程序员</p >
17     <p>扣丁学堂</p >
18 </body >
19 </html >

```

例 3-4 运行结果如图 3.8 所示。

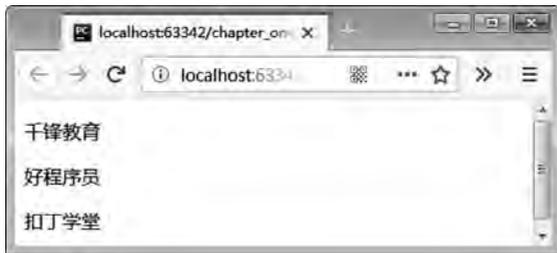


图 3.8 例 3-4 运行结果

单击【千锋教育】, 运行结果如图 3.9 所示。

单击“好程序员”, 运行结果如图 3.10 所示。

单击“扣丁学堂”, 运行结果如图 3.11 所示。

例 3-4 是使用 on() 函数实现事件绑定的功能。第 5 行是引入 jQuery。第 7~11 行是使用 on() 函数绑定 <p> 标签的 click 事件, 单击 <p> 标签, 则弹出标签 <p> 中的内容。



图 3.9 单击“千锋教育”运行结果



图 3.10 单击“好程序员”运行结果



图 3.11 单击“扣丁学堂”运行结果

第 15~17 行是< body>中的< p>标签。

同理,大家可以自己动手尝试编写实现其他事件绑定方法。

### 3.1.5 jQuery Ajax

异步的 JavaScript 和 XML(Asynchronous JavaScript and XML, Ajax)是用来实现异步传输数据的技术,即在网页不刷新的情况下进行数据传送。jQuery 中主要使用函数 \$.ajax() 来实现 Ajax 技术, \$.ajax() 函数包含很多参数。接下来讲解 \$.ajax() 函数中常用参数的作用及 jQuery 中 Ajax 技术的使用。

#### 1. 回调函数

Web 中, Ajax 技术主要是进行数据的传送,若要处理 Ajax 得到的数据,需要使用以下几种回调函数:

- **beforeSend**: 在发送请求之前调用,并且传入一个 XMLHttpRequest 作为参数。
- **error**: 在请求出错时调用。传入 XMLHttpRequest 对象,描述错误类型的字符串以及一个异常对象(如果存在异常情况)。
- **dataFilter**: 在请求成功之后调用。传入返回的数据以及 dataType 参数的值,并且必须返回新的数据(可能是处理后的)传递给 success 回调函数。
- **success**: 在请求之后调用。传入返回后的数据,以及包含成功代码的字符串。
- **complete**: 在请求完成之后调用,无论成功或失败。传入 XMLHttpRequest 对象,以及一个包含成功或错误代码的字符串。

## 2. 数据类型

参数 dataType 是指定服务器返回的数据类型,参数类型为 String。如果未指定类型, jQuery 将自动根据 http 包 mime 的信息返回 responseXML 或 responseText,并作为回调函数的参数进行传递。

其中可指定的类型有以下 6 种:

- **xml**: 返回 XML 文档,可用 jQuery 处理。
- **html**: 返回纯文本 HTML 信息;包含的 script 标签会在插入 DOM 时执行。
- **script**: 返回纯文本 JavaScript 代码。不会自动缓存结果。除非设置了 cache 参数。注意:在远程请求时(不在同一个域下),所有 post 请求都将转为 get 请求。
- **json**: 返回 JSON 数据。
- **jsonp**: JSONP 格式。使用 JSONP 形式调用函数时,例如 myurl? callback=?, jQuery 将自动替换后一个“?”为正确的函数名,以执行回调函数。
- **text**: 返回纯文本字符串。

## 3. 其他常用参数

- **url**: 发送请求的地址(默认为当前页地址),参数类型为 String。
- **type**: 请求方式(post 或 get)默认为 get,参数类型为 String。其他 http 请求方法,如 put 和 delete 也可使用,但仅部分浏览器支持。
- **data**: 发送到服务器的数据,参数类型为 Object 或 String。如果不是字符串,将自动转换为字符串。get 请求将 data 附加在 url 后。对象必须为 key/value 格式,如 {foo1:"bar1",foo2:"bar2"} 转换为 &foo1=bar1&foo2=bar2。如果是数组, jQuery 将自动将不同值对应同一个名称,如 {foo:["bar1","bar2"]} 转换为 &foo=bar1&foo=bar2。

## 4. 使用

\$.ajax() 函数的使用示例如下:

```
$.ajax({
  type: "POST",
  url: "test.py",
  data: {name: "1000phone",location: "BeiJing"},
  success: function(msg){
    alert("Data Saved: " + msg);
  }
});
```

上述示例是通过 \$.ajax() 函数实现使用 POST 方式请求 test.py 文件并将 data 中的数据保存到服务器中,成功之后显示保存的数据的功能。

## 3.2 Bootstrap

### 3.2.1 Bootstrap 简介

Bootstrap 是 Twitter 设计师 Mark Otto 和 Jacob Thornton 合作开发的前端框架,也是目前最受欢迎的前端框架。Bootstrap 基于 HTML、CSS、JavaScript,它简洁灵活,使 Web 开发更加快捷。

Bootstrap 之所以如此受欢迎,是因为它有以下优点:

- 适配移动端和 PC 端,并以移动设备优先。
- 所有主流浏览器都支持 Bootstrap。
- 容易上手。
- 响应式设计。
- 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。
- 提供了基于 Web 的定制。
- 开源。

Bootstrap 的安装及使用都非常简单,接下来分步骤讲解 Bootstrap 的安装与使用。

#### 1. 下载 Bootstrap

下载地址: <http://www.bootcss.com/>,即 Bootstrap 中文网首页,如图 3.12 所示。



图 3.12 Bootstrap 中文网首页

单击【Bootstrap3 中文文档(v3. 3. 7)】按钮,进入下载接入页如图 3. 13 所示。



图 3. 13 下载接入页

单击【下载 Bootstrap】进入下载页面,如图 3. 14 所示。



图 3. 14 Bootstrap 下载页

在图 3.14 中,有三种形式的 Bootstrap,本书选择第一种下载,即单击【下载 Bootstrap】即可下载。

## 2. 解压下载文件

解压后文件如图 3.15 所示。

```
bootstrap
├──css
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   ├── bootstrap-theme.css
│   ├── bootstrap-theme.css.map
│   ├── bootstrap-theme.min.css
│   └── bootstrap-theme.min.css.map
│
├──fonts
│   ├── glyphicons-halflings-regular.eot
│   ├── glyphicons-halflings-regular.svg
│   ├── glyphicons-halflings-regular.ttf
│   ├── glyphicons-halflings-regular.woff
│   └── glyphicons-halflings-regular.woff2
│
└──js
    ├── bootstrap.js
    └── bootstrap.min.js
```

图 3.15 解压后文件

其中 css 文件夹中常用的两个文件是 bootstrap.css 和 bootstrap.min.css; js 文件夹中常用的两个文件是 bootstrap.js 和 bootstrap.min.js。

## 3. 复制解压后的文件

将解压后的文件夹复制到项目中即可使用。

## 4. Bootstrap 使用

接下来以一个简单实例来讲解 Bootstrap 的使用,如例 3-5 所示。

### 【例 3-5】 Bootstrap 的使用

```
1 <!DOCTYPE html >
2 <html >
3 <head >
4   <meta charset = "utf - 8">
5   <title> Bootstrap 测试</title >
6   <link rel = "stylesheet" href = "bootstrap/css/bootstrap.min.css">
7   <script src = "static/js/jquery.min.js"></script >
8   <script src = "bootstrap/js/bootstrap.min.js"></script >
9 </head >
```

```

10 <body>
11     <h1>Hello, world!</h1>
12 </body>
13 </html>
    
```

Bootstrap 和 jQuery 的相关文件需要导入项目,如图 3.16 所示。

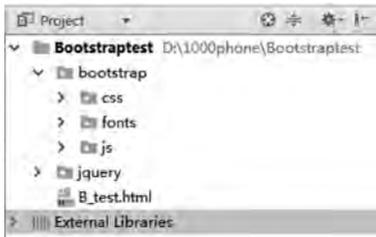


图 3.16 导入 Bootstrap 和 jQuery

例 3-5 运行结果如图 3.17 所示。



图 3.17 例 3-5 运行结果

图 3.17 是例 3-5 的运行结果图。第 5 行是链入 Bootstrap 中 css 文件,第 6 行是引入 jQuery,第 7 行是引入 Bootstrap 中 js 文件,最终使< h1 >标签中的内容更优美地显示在浏览器中。

Bootstrap 的功能强大,内容较多,因此本章将讲解 Bootstrap 中较为重要的两部分内容——全局 CSS 样式和组件。

### 3.2.2 Bootstrap 全局 CSS 样式

一个网站的搭建,首先要对整个网站进行布局,Bootstrap 提供了非常便利的响应式、移动设备优先的栅格系统来对网站进行布局,使网站不仅布局美观还可适配不同的屏幕大小。

栅格系统将整个网页分成最多 12 列,设计者可以根据内容来选择所占比例,其中栅格参数如表 3.5 所示。

表 3.5 栅格参数

	超小屏幕 ( $<768\text{px}$ )	小屏幕 ( $\geq 768\text{px}$ )	中等屏幕 ( $\geq 992\text{px}$ )	大屏幕 ( $\geq 1200\text{px}$ )
栅格系统行为	总是水平排列	开始堆叠在一起,当大于这些阈值时将变为水平排列		
.container 容器最大宽度	None(自动)	750px	970px	1170px
类前缀	.col-xs-	.col-sm-	.col-md-	.col-lg-

续表

	超小屏幕 ( $<768\text{px}$ )	小屏幕 ( $\geq 768\text{px}$ )	中等屏幕 ( $\geq 992\text{px}$ )	大屏幕 ( $\geq 1200\text{px}$ )
列(column)数	12			
最大列(column)宽	自动	~62px	~82px	~97px
槽(gutter)宽	30px(每列左右均有 15px)			
可嵌套	是			
偏移(Offsets)	是			
列排序	是			

接下来以一个简单的实例来讲解响应式栅格系统的使用,如例 3-6 所示。

### 【例 3-6】 响应式栅格系统的使用

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head >
4      <meta charset = "UTF - 8">
5      <title>栅格系统测试</title>
6      <link rel = "stylesheet" href = "bootstrap/css/bootstrap.min.css">
7      <script src = "bootstrap/js/bootstrap.min.js"></script >
8  </head >
9  <body >
10     <div class = "row">
11         <div class = "col - xs - 12 col - md - 8">.col - xs - 12 .col - md - 8</div >
12         <div class = "col - xs - 6 col - md - 4">.col - xs - 6 .col - md - 4</div >
13     </div >
14 </body >
15 </html >

```

例 3-6 运行结果如图 3.18 所示。

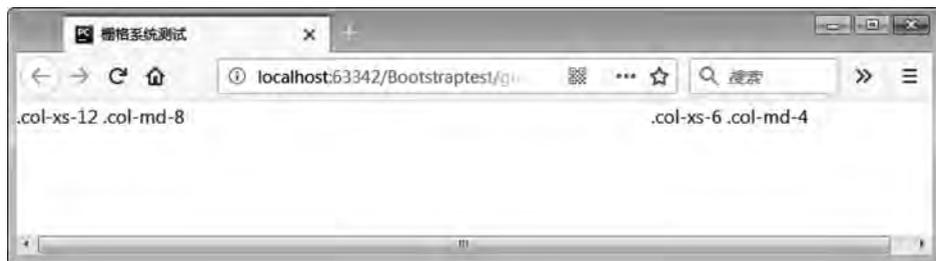


图 3.18 例 3-6 运行结果

当页面变为手机端时显示结果如图 3.19 所示。

例 3-6 是实现使用响应式的栅格系统来设计网页。第 6 行是链入 Bootstrap 中 css 文件;第 7 行是引入 Bootstrap 中 js 文件;第 11 行是当页面大于 768px 时<div>标签中内容占 8 列,当页面小于 768px 时<div>标签中内容占 12 列;第 12 行是当页面大于 768px 时<div>标签中内容占 4 列,当页面小于 768px 时<div>标签中内容占 6 列。

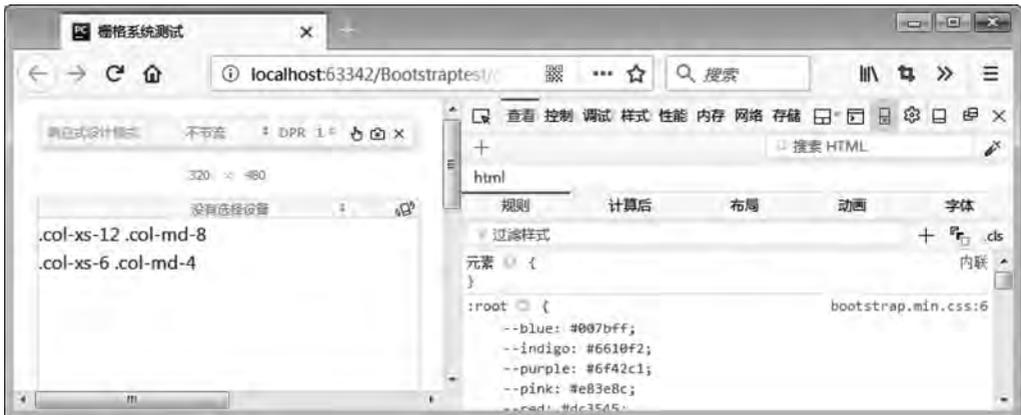


图 3.19 手机端显示

注意：Bootstrap 全局 CSS 样式中还有其他丰富的内容，可以借助官网进行学习。

### 3.2.3 Bootstrap 组件

#### 1. 导航栏

导航栏是每个网站必备的内容，接下来以一个简单的实例来讲解 Bootstrap 中导航栏的实现，如例 3-7 所示。

**【例 3-7】** Bootstrap 中导航栏的实现

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head >
4      <meta charset = "UTF - 8">
5      <title>导航栏</title>
6      <link rel = "stylesheet" href = "bootstrap/css/bootstrap.min.css">
7      <script src = "static/js/jquery - 3.2.1.min.js"></script >
8      <script src = "bootstrap/js/bootstrap.min.js"></script >
9  </head >
10 <body >
11 <nav class = "navbar navbar - expand - md bg - dark navbar - dark">
12     <! -- Brand -- >
13     <a class = "navbar - brand" href = "#">千锋教育</a >
14     <! -- Toggler/collapsible Button -- >
15     <button class = "navbar - toggler" type = "button"
16         data - toggle = "collapse" data - target = "#collapsibleNavbar">
17         <span class = "navbar - toggler - icon"></span >
18     </button >
19     <! -- Navbar links -- >
20     <div class = "collapse navbar - collapse" id = "collapsibleNavbar">
21         <ul class = "navbar - nav">
22             <li class = "nav - item">
23                 <a class = "nav - link" href = "#">好程序员</a >
24             </li >

```

```
25     <li class = "nav - item">
26         <a class = "nav - link" href = "#">扣丁学堂</a>
27     </li>
28     <li class = "nav - item">
29         <a class = "nav - link" href = "#">IT 培训</a>
30     </li>
31 </ul>
32 </div>
33 </nav>
34 </body>
35 </html>
```

例 3-7 运行结果如图 3.20 所示。



图 3.20 例 3-7 运行结果

当页面小于 768px 时显示界面如图 3.21 所示。

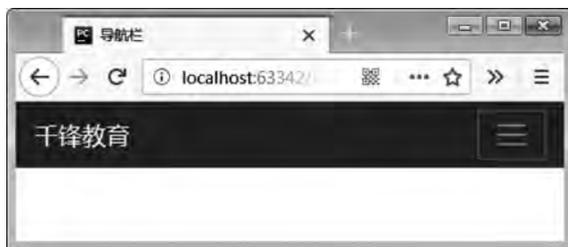


图 3.21 页面小于 768px 界面

单击右侧“三根横线”按钮显示结果如图 3.22 所示。



图 3.22 单击右侧按钮显示结果

例 3-7 实现了使用 Bootstrap 编写网站导航栏。第 11 行是包含整个导航栏的<nav>标签；第 13 行一般是网站所特有的商标，此处以文字“千锋教育”代替；第 15~18 行是当页面小于 768px 时页面显示的如图 3.21 所示右侧的按钮；第 20~32 行是整个导航栏的主体内容，包括“好程序员”“扣丁学堂”“IT 培训”。

## 2. 分页

当网站需要显示的数据量比较大时，需要进行分页展示，Bootstrap 提供了样式更优美的分页功能。接下来以一个简单的实例讲解分页的实现，如例 3-8 所示。

### 【例 3-8】 分页的实现

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head >
4      <meta charset = "UTF - 8">
5      <title>分页</title>
6      <link rel = "stylesheet" href = "bootstrap/css/bootstrap.min.css">
7      <script src = "bootstrap/js/bootstrap.min.js"></script >
8      <script src = "static/js/jquery - 3.2.1.min.js"></script >
9  </head >
10 <body >
11 <nav aria - label = "Page navigation">
12 <div class = "container">
13   <h2>分页</h2 >
14   <ul class = "pagination">
15     <li class = "page - item"><a class = "page - link" href = "#"> &laquo;</a ></li >
16     <li class = "page - item"><a class = "page - link" href = "#"> 1 </a ></li >
17     <li class = "page - item"><a class = "page - link" href = "#"> 2 </a ></li >
18     <li class = "page - item"><a class = "page - link" href = "#"> 3 </a ></li >
19     <li class = "page - item"><a class = "page - link" href = "#"> &raquo;</a ></li >
20   </ul >
21 </div >
22 </nav >
23 </body >
24 </html >

```

例 3-8 运行结果如图 3.23 所示。

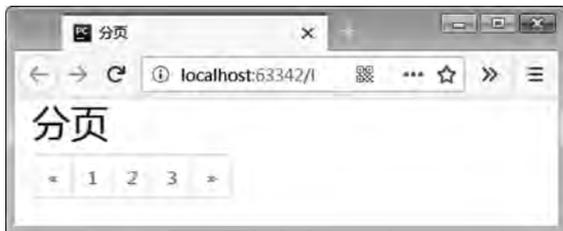


图 3.23 例 3-8 运行结果

例 3-8 是实现分页功能。第 14 行是整个分页的开始，标签<ul>中 pagination 类是标注分页的；第 15~19 行是分页的页数及链接内容。

**注意：**Bootstrap 中还有其他丰富的组件内容，可以借助官网进行学习测试，并最终运用到实际开发中。Bootstrap 可能在不同的浏览器中适配情况不太一致，本书选用的是火狐浏览器。

### 3.3 本章小结

本章主要是讲解 Web 客户端进阶的内容，包括 jQuery 和 Bootstrap 框架。在开发过程中合理运用以上两个框架可达到事半功倍的效果。jQuery 和 Bootstrap 框架还有很多丰富的内容，需要大家多掌握应用。

### 3.4 习 题

#### 1. 填空题

- (1) Web 客户端常用的两个框架是\_\_\_\_\_和\_\_\_\_\_。
- (2) jQuery action 实质上指的是\_\_\_\_\_。
- (3) JavaScript 中最常用的框架是\_\_\_\_\_。
- (4) \_\_\_\_\_是用来实现异步传输数据的技术，即在网页不刷新的情况下进行数据传送。
- (5) 栅格系统将整个网页分成最多\_\_\_\_\_列。

#### 2. 选择题

- (1) 下列选项中，( ) 不属于 jQuery 选择器。  
A. #id                      B. .class                      C. \*                      D. \*
- (2) 当屏幕小于 768px 时，栅格系统的( ) 类前缀起作用。  
A. col-md-                      B. col-sm-                      C. col-xs-                      D. col-lg-
- (3) 下列回调函数中，( ) 是在请求出错时调用。  
A. error                      B. success                      C. complete                      D. dataFilter
- (4) jQuery 的设计宗旨是( )。  
A. “Life is short, I use Python”                      B. “stay hungry stay foolish”  
C. “Write Less, Do More”                      D. “talk is cheap, show me your code”
- (5) 下列选项中，( ) 不属于 jQuery 事件。  
A. on()                      B. trigger()                      C. click()                      D. toggle()

#### 3. 思考题

- (1) 简述 jQuery 的概念，并简要讲解 jQuery 的使用步骤？
- (2) 简述 Ajax 的概念？

#### 4. 编程题

使用 jQuery 和 Bootstrap 编写一个动态网页，包括导航栏和 jQuery 的事件操作。

