

Bootstrap

一、Bootstrap 简介

- (一) 什么是 Bootstrap
- (二) 如何使用Bootstrap

二、Bootstrap 常用组件

- (一) 字体图标
- (二) 下拉菜单
- (三) 按钮组
- (四) 输入框组
- (五) 导航
- (六) 导航条
- (七) 分页导航
- (八) 标签
- (九) 徽章
- (十) 警告框
- (十一) 进度条
 - 1、带标签提示进度条
 - 2、根据情景变化效果
 - 3、条纹效果
 - 4、动态效果
 - 5、堆叠样式

三、Bootstrap 全局样式

- (一) 栅格系统
 - 1、从堆叠到水平排列
 - 2、手机、平板、桌面
 - 3、列偏移
 - 4、列嵌套
 - 5、列排序
- (二) 表单
 - 1、基本表单
 - 2、内联表单
 - 3、水平排列表单
- (三) 表格
 - 1、基础表格
 - 2、条纹表格
 - 3、边框表格
 - 4、鼠标悬停
 - 5、紧缩表格
 - 6、响应式表格
 - 7、状态类样式
- (四) 按钮

四、Bootstrap 插件

- (一) 插件概述
- (二) 模态框
- (三) 滚动监听

五、Bootstrap Table

- (一) Bootstrap Table简介
- (二) 入门
 - 1、导入文件
 - 2、基础用法
- (三) 表格操作
 - 1、常用属性
 - 2、API

- 2.1 表的各项(Table options)
- 2.2 列的各项(Column options)
- 2.3 事件
- 2.4 方法

Bootstrap

一、Bootstrap 简介

(一) 什么是 Bootstrap

Bootstrap 是最受欢迎的 HTML、CSS 和 JS 框架，用于开发**响应式布局**、移动设备优先的 WEB 项目。

为所有开发者、所有应用场景而设计。

Bootstrap 让前端开发更快速、简单。所有开发者都能快速上手、所有设备都可以适配、所有项目都适用。

中文官网：<https://v3.bootcss.com/>

为所有开发者、所有应用场景而设计

Bootstrap 让前端开发更快速、简单。所有开发者都能快速上手、所有设备都可以适配、所有项目都适用。



预处理工具

虽然可以直接使用 Bootstrap 提供的 CSS 样式表，但是不要忘记，Bootstrap 的源码是采用最流行的 CSS 预处理工具 Less 和 Sass 开发的。你可以直接采用预编译的 CSS 文件快速开发，也可以从 Bootstrap 源码自定义自己需要的样式。



一个框架、多种设备。

你的网站和应用能在 Bootstrap 的帮助下通过同一份源码快速、有效地适配手机、平板和 PC 设备，这一切都是 CSS 媒体查询 (Media Query) 的功劳。



功能完备

Bootstrap 提供了全面、美观的文档，你能在这里找到关于普通 HTML 元素、HTML 和 CSS 组件以及 jQuery 插件方面的所有详细文档。

(二) 如何使用Bootstrap

下载 Bootstrap 源码，导入相关资源文件 (CSS、JS) 。

```
1 <link href="css/bootstrap.css" type="text/css" rel="stylesheet"/>
2 <link href="css/bootstrap-theme.css" type="text/css" rel="stylesheet"/>
3
4 <script src="js/jquery-3.6.0.min.js"></script>
5 <script src="js/bootstrap.js"></script>
```

二、Bootstrap 常用组件

(一) 字体图标

具体样式参照: <https://v3.bootcss.com/components/>

```
1 <!--
2     class属性:
3         glyphicon: 字体图标基础样式
4         glyphicon-search: 具体字体图标形状
5 -->
6 <span class="glyphicon glyphicon-search" aria-hidden="true"></span>
```

(二) 下拉菜单

```
1 <!--
2     下拉菜单有按钮和列表组成, 所以需要使用div容器包裹
3     给容器添加 class="dropdown" 表示菜单展示在按钮下方; class="dropup" 表示菜单展示
    在按钮上方
4 -->
5 <div class="dropdown">
6     <!--
7         class="btn btn-default dropdown-toggle"
8         btn: 使用按钮基础样式
9         btn-default: 使用按钮默认样式
10        dropdown-toggle: 表示下拉切换样式
11        id="dropdownMenu1"
12        当前按钮的id标识
13        data-toggle="dropdown"
14        调用Bootstrap封装的dropdown事件
15    -->
16    <button class="btn btn-default dropdown-toggle" type="button"
17    id="dropdownMenu1"
18        data-toggle="dropdown" aria-haspopup="true" aria-
19    expanded="true">
20        Dropdown
21        <!-- 向下的三角 -->
22        <span class="caret"></span>
23    </button>
24    <!--
25        菜单列表
26        class="dropdown-menu"
27        使用下拉菜单样式
28        aria-labelledby="dropdownMenu1"
29        当前菜单和那个按钮关联
30    -->
31    <ul class="dropdown-menu" aria-labelledby="dropdownMenu1">
32        <li><a href="#">Action</a></li>
33        <li><a href="#">Another action</a></li>
34        <li><a href="#">Something else here</a></li>
35        <!-- 菜单项目之间的分割线 -->
36        <li role="separator" class="divider"></li>
37        <!-- 菜单项目标题 -->
38        <li class="dropdown-header">Dropdown header</li>
39        <li><a href="#">Separated link</a></li>
40        <!-- 禁用当前菜单项目, 需要搭配超链接标签使用 -->
```

```
39     <li class="disabled"><a href="#">Disabled link</a></li>
40   </ul>
41 </div>
```

(三) 按钮组

```
1  <!-- 按钮工具栏: class="btn-toolbar" -->
2  <div class="btn-toolbar" role="toolbar" aria-label="...">
3    <!-- 按钮组: class="btn-group" -->
4    <div class="btn-group" role="group" aria-label="...">
5      <button type="button" class="btn btn-default">Left</button>
6      <button type="button" class="btn btn-default">Middle</button>
7      <button type="button" class="btn btn-default">Right</button>
8    </div>
9    <!-- 按钮组 -->
10   <div class="btn-group" role="group" aria-label="...">
11     <button type="button" class="btn btn-default">Left</button>
12     <button type="button" class="btn btn-default">Middle</button>
13     <button type="button" class="btn btn-default">Right</button>
14     <!-- 组嵌套 -->
15     <div class="btn-group" role="group">
16       <button type="button" class="btn btn-default dropdown-toggle"
17         data-toggle="dropdown"
18         aria-haspopup="true" aria-expanded="false">
19         Dropdown
20         <span class="caret"></span>
21       </button>
22       <ul class="dropdown-menu">
23         <li><a href="#">Dropdown link</a></li>
24         <li><a href="#">Dropdown link</a></li>
25       </ul>
26     </div>
27   </div>
28 </div>
```

(四) 输入框组

```
1  <!-- 输入框组: class="input-group" -->
2  <div class="input-group">
3    <!-- 输入框组的标题样式: class="input-group-addon" -->
4    <span class="input-group-addon" id="basic-addon1">@</span>
5    <!-- 表单组件样式: class="form-control" -->
6    <input type="text" class="form-control" placeholder="Username" aria-
7      describedby="basic-addon1">
8  </div>
9
10 <div class="input-group">
11   <input type="text" class="form-control" placeholder="Recipient's
12     username" aria-describedby="basic-addon2">
13   <span class="input-group-addon" id="basic-addon2">@example.com</span>
14 </div>
15
16 <div class="input-group">
17   <span class="input-group-addon">$</span>
18 </div>
```

```

16     <input type="text" class="form-control" aria-label="Amount (to the
nearest dollar)">
17     <span class="input-group-addon">.00</span>
18 </div>
19
20 <label for="basic-url">Your vanity URL</label>
21 <div class="input-group">
22     <span class="input-group-addon" id="basic-
addon3">https://example.com/users/</span>
23     <input type="text" class="form-control" id="basic-url" aria-
describedby="basic-addon3">
24 </div>
25
26 <div class="input-group">
27     <input type="text" class="form-control" placeholder="Search for...">
28     <!-- 输入框组内的按钮 -->
29     <span class="input-group-btn">
30         <button class="btn btn-default" type="button">Go!</button>
31     </span>
32 </div>

```

(五) 导航

```

1  <!--
2      class="nav nav-tabs"
3      nav: 导航基础样式
4      nav-tabs: 标签页式导航
5  -->
6  <ul class="nav nav-tabs">
7      <!-- 当前导航激活: class="active" -->
8      <li role="presentation" class="active"><a href="#">Home</a></li>
9      <li role="presentation"><a href="#">Profile</a></li>
10     <li role="presentation"><a href="#">Messages</a></li>
11 </ul>
12
13 <!--
14     class="nav nav-pills"
15     nav: 导航基础样式
16     nav-pills: 胶囊式导航
17     nav-stacked: 纵向显示
18 -->
19 <ul class="nav nav-pills">
20     <li role="presentation" class="active"><a href="#">Home</a></li>
21     <li role="presentation"><a href="#">Profile</a></li>
22     <li role="presentation"><a href="#">Messages</a></li>
23     <li role="presentation" class="disabled"><a href="#">Disabled link</a>
24 </li>
25 </ul>

```

(六) 导航条

```
1  <!--
2      class="navbar navbar-default"
3      navbar: 导航条基础样式
4      navbar-default: 导航条默认样式
5      navbar-fixed-top: 导航条固定在顶部
6      navbar-fixed-bottom: 导航条固定在底部
7      navbar-inverse: 反色
8  -->
9  <nav class="navbar navbar-default">
10     <div class="container-fluid">
11         <!-- 手机端显示导航折叠后效果 -->
12         <div class="navbar-header">
13             <!--
14                 导航切换样式: class="navbar-toggle"
15                 导航折叠样式: class="collapsed"
16                 导航折叠展开的内容, 引入id: data-target="#bs-example-navbar-
collapse-1"
17             -->
18             <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse"
19                 data-target="#bs-example-navbar-collapse-1" aria-
expanded="false">
20                 <!-- 导航折叠后显示的内容 -->
21                 <span class="sr-only">Toggle navigation</span>
22                 <span class="icon-bar"></span>
23                 <span class="icon-bar"></span>
24                 <span class="icon-bar"></span>
25             </button>
26             <a class="navbar-brand" href="#">Brand</a>
27         </div>
28
29         <!--
30             包含连接、from表单、其他组件的导航条
31             class="collapse navbar-collapse"
32             折叠导航条
33             id="bs-example-navbar-collapse-1"
34             当前导航条的id标识
35         -->
36         <div class="collapse navbar-collapse" id="bs-example-navbar-
collapse-1">
37             <ul class="nav navbar-nav">
38                 <!-- 超链接 -->
39                 <li class="active"><a href="#">Link <span class="sr-only">
(current)</span></a></li>
40                 <li><a href="#">Link</a></li>
41                 <!-- 下拉列表 -->
42                 <li class="dropdown">
43                     <a href="#" class="dropdown-toggle" data-
toggle="dropdown" role="button"
44                         aria-haspopup="true" aria-expanded="false">Dropdown
45                     <span class="caret"></span>
46                 </a>
47                 <ul class="dropdown-menu">
```

```

48         <li><a href="#">Action</a></li>
49         <li><a href="#">Another action</a></li>
50         <li><a href="#">Something else here</a></li>
51         <li role="separator" class="divider"></li>
52         <li><a href="#">Separated link</a></li>
53         <li role="separator" class="divider"></li>
54         <li><a href="#">One more separated link</a></li>
55     </ul>
56 </li>
57 </ul>
58 <!--
59     表单
60     导航表单样式: class="navbar-form"
61     导航居左显示: class="navbar-left"
62 -->
63 <form class="navbar-form navbar-left">
64     <div class="form-group">
65         <input type="text" class="form-control"
placeholder="Search">
66     </div>
67     <button type="submit" class="btn btn-
default">Submit</button>
68 </form>
69 <!--
70     导航居左显示: class="navbar-right"
71 -->
72 <ul class="nav navbar-nav navbar-right">
73     <li><a href="#">Link</a></li>
74     <li class="dropdown">
75         <a href="#" class="dropdown-toggle" data-
toggle="dropdown" role="button"
76             aria-haspopup="true" aria-expanded="false">Dropdown
<span class="caret"></span></a>
77         <ul class="dropdown-menu">
78             <li><a href="#">Action</a></li>
79             <li><a href="#">Another action</a></li>
80             <li><a href="#">Something else here</a></li>
81             <li role="separator" class="divider"></li>
82             <li><a href="#">Separated link</a></li>
83         </ul>
84     </li>
85 </ul>
86 </div>
87 </div>
88 </nav>

```

(七) 分页导航

```

1 <nav aria-label="Page navigation">
2     <!-- 分页样式 class="pagination" -->
3     <ul class="pagination">
4         <!-- 禁用 -->
5         <li class="disabled">
6             <a href="#" aria-label="Previous">
7                 <span aria-hidden="true">&laquo;</span>

```

```

8         </a>
9     </li>
10    <!-- 激活 -->
11    <li class="active"><a href="#">1</a></li>
12    <li><a href="#">2</a></li>
13    <li><a href="#">3</a></li>
14    <li><a href="#">4</a></li>
15    <li><a href="#">5</a></li>
16    <li>
17        <a href="#" aria-label="Next">
18            <span aria-hidden="true">&raquo;</span>
19        </a>
20    </li>
21 </ul>
22 </nav>

```

(八) 标签

```

1 <span class="label label-default">Default</span>
2 <span class="label label-primary">Primary</span>
3 <span class="label label-success">Success</span>
4 <span class="label label-info">Info</span>
5 <span class="label label-warning">Warning</span>
6 <span class="label label-danger">Danger</span>

```

(九) 徽章

```

1 <!-- 徽章样式: class="badge" -->
2 <a href="#">Inbox <span class="badge">42</span></a>
3
4 <button class="btn btn-primary" type="button">
5     Messages <span class="badge">4</span>
6 </button>
7
8 <ul class="nav nav-pills" role="tablist">
9     <li role="presentation" class="active"><a href="#">Home <span
class="badge">42</span></a></li>
10    <li role="presentation"><a href="#">Profile</a></li>
11    <li role="presentation"><a href="#">Messages <span
class="badge">3</span></a></li>
12 </ul>

```

(十) 警告框

```

1 <div class="alert alert-success alert-dismissible" role="alert">
2     <button type="button" class="close" data-dismiss="alert" aria-
label="Close">
3         <span aria-hidden="true">&times;</span>
4     </button>
5     Well done! You successfully read this important alert message.
6 </div>
7 <div class="alert alert-info" role="alert">
8     Heads up! This alert needs your attention, but it's not super important.

```



```

9     </div>
10    <div class="alert alert-warning" role="alert">
11        warning! Better check yourself, you're not looking too good.
12    </div>
13    <div class="alert alert-danger" role="alert">
14        Oh snap! <a href="#" class="alert-link">Change a few things up and try
        submitting again.</a>
15    </div>

```

(十一) 进度条

1、带标签提示进度条

```

1  <!-- 带有提示标签的进度条 -->
2  <div class="progress">
3      <div class="progress-bar" role="progressbar" aria-valuenow="60" aria-
        valuelmin="0" aria-valuemax="100"
4          style="width: 60%;">
5          60%
6      </div>
7  </div>
8  <div class="progress">
9      <div class="progress-bar" role="progressbar" aria-valuenow="0" aria-
        valuelmin="0" aria-valuemax="100"
10         style="min-width: 2em;">
11         0%
12     </div>
13 </div>
14 <div class="progress">
15     <div class="progress-bar" role="progressbar" aria-valuenow="2" aria-
        valuelmin="0" aria-valuemax="100"
16         style="min-width: 2em; width: 2%;">
17         2%
18     </div>
19 </div>

```

2、根据情景变化效果

```

1  <!--
2      根据情境变化效果
3      class="progress-bar-success"
4      class="progress-bar-info"
5      class="progress-bar-warning"
6      class="progress-bar-danger"
7  -->
8  <div class="progress">
9      <div class="progress-bar progress-bar-success" role="progressbar" aria-
        valuenow="40" aria-valuemin="0"
10         aria-valuemax="100" style="width: 40%">
11         <span class="sr-only">40% Complete (success)</span>
12     </div>
13 </div>
14 <div class="progress">

```

```

15     <div class="progress-bar progress-bar-info" role="progressbar" aria-
    valuenow="20" aria-valuemin="0"
16         aria-valuemax="100" style="width: 20%">
17         <span class="sr-only">20% Complete</span>
18     </div>
19 </div>
20 <div class="progress">
21     <div class="progress-bar progress-bar-warning" role="progressbar" aria-
    valuenow="60" aria-valuemin="0"
22         aria-valuemax="100" style="width: 60%">
23         <span class="sr-only">60% Complete (warning)</span>
24     </div>
25 </div>
26 <div class="progress">
27     <div class="progress-bar progress-bar-danger" role="progressbar" aria-
    valuenow="80" aria-valuemin="0"
28         aria-valuemax="100" style="width: 80%">
29         <span class="sr-only">80% Complete (danger)</span>
30     </div>
31 </div>

```

3、条纹效果

```

1  <!-- 条纹效果: class="progress-bar-striped" -->
2  <div class="progress">
3      <div class="progress-bar progress-bar-success progress-bar-striped"
    role="progressbar" aria-valuenow="40"
4          aria-valuemin="0" aria-valuemax="100" style="width: 40%">
5          <span class="sr-only">40% Complete (success)</span>
6      </div>
7  </div>

```

4、动态效果

```

1  <!-- 动态效果: class="active" -->
2  <div class="progress">
3      <div class="progress-bar progress-bar-info progress-bar-striped active"
    role="progressbar"
4          aria-valuenow="20" aria-valuemin="0" aria-valuemax="100"
    style="width: 20%">
5          <span class="sr-only">20% Complete</span>
6      </div>
7  </div>

```

5、堆叠样式

```

1  <!-- 堆叠样式 -->
2  <div class="progress">
3      <div class="progress-bar progress-bar-success" style="width: 35%">
4          <span class="sr-only">35% Complete (success)</span>
5      </div>
6      <div class="progress-bar progress-bar-warning" style="width: 20%">
7          <span class="sr-only">20% Complete (warning)</span>
8      </div>
9      <div class="progress-bar progress-bar-danger" style="width: 10%">
10         <span class="sr-only">10% Complete (danger)</span>
11     </div>
12 </div>

```

三、Bootstrap 全局样式

(一) 栅格系统

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

- “行（row）”必须包含在 `.container`（固定宽度）或 `.container-fluid`（100% 宽度）中，以便为其赋予合适的排列（alignment）和内补（padding）。
- 通过“行（row）”在水平方向创建一组“列（column）”。
- 你的内容应当放置于“列（column）”内，并且，只有“列（column）”可以作为行（row）”的直接子元素。
- 类似 `.row` 和 `.col-xs-4` 这种预定义的 class，可以用来快速创建栅格布局。Bootstrap 源码中定义的 mixin 也可以用来创建语义化的布局。
- 如果一行（row）”中包含的“列（column）”大于 12，多余的“列（column）”所在的元素将被作为一个整体另起一行排列。
- 通过为“列（column）”设置 `padding` 属性，从而创建列与列之间的间隔（gutter）。通过为 `.row` 元素设置负值 `margin` 从而抵消掉为 `.container` 元素设置的 `padding`，也就间接为“行（row）”所包含的“列（column）”抵消掉了 `padding`。
- 负值的 `margin` 就是下面的示例为什么是向外突出的原因。在栅格列中的内容排成一行。
- 栅格系统中的列是通过指定 1 到 12 的值来表示其跨越的范围。例如，三个等宽的列可以使用三个 `.col-xs-4` 来创建。
- 栅格类适用于与屏幕宽度大于或等于分界点大小的设备，并且针对小屏幕设备覆盖栅格类。因此，在元素上应用任何 `.col-md-*` 栅格类适用于与屏幕宽度大于或等于分界点大小的设备，并且针对小屏幕设备覆盖栅格类。因此，在元素上应用任何 `.col-lg-*` 不存在，也影响大屏幕设备。

栅格参数

	超小屏幕 手机 (<768px)	小屏幕 平板 (≥768px)	中等屏幕 桌面显示器 (≥992px)	大屏幕 大桌面显示器 (≥1200px)
栅格系统行为	总是水平排列	开始是堆叠在一起的，当大于这些阈值时将变为水平排列		

	超小屏幕 手机 (<768px)	小屏幕 平板 (≥768px)	中等屏幕 桌面显示器 (≥992px)	大屏幕 大桌面显示器 (≥1200px)
<code>.container</code> 最大宽度	None (自动)	750px	970px	1170px
class 前缀	<code>.col-xs-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>
列 (column) 数	12			
最大列 (column) 宽	自动	~62px	~81px	~97px
槽 (gutter) 宽	30px (每列左右均有 15px)			
可嵌套	是			
偏移 (Offsets)	是			
列排序	是			

1、从堆叠到水平排列

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

```
1 <div class="row">
2   <div class="col-md-1">.col-md-1</div>
3   <div class="col-md-1">.col-md-1</div>
4   <div class="col-md-1">.col-md-1</div>
5   <div class="col-md-1">.col-md-1</div>
6   <div class="col-md-1">.col-md-1</div>
7   <div class="col-md-1">.col-md-1</div>
8   <div class="col-md-1">.col-md-1</div>
9   <div class="col-md-1">.col-md-1</div>
10  <div class="col-md-1">.col-md-1</div>
11  <div class="col-md-1">.col-md-1</div>
12  <div class="col-md-1">.col-md-1</div>
13  <div class="col-md-1">.col-md-1</div>
14 </div>
15 <div class="row">
```

```

16 <div class="col-md-8">.col-md-8</div>
17 <div class="col-md-4">.col-md-4</div>
18 </div>
19 <div class="row">
20 <div class="col-md-4">.col-md-4</div>
21 <div class="col-md-4">.col-md-4</div>
22 <div class="col-md-4">.col-md-4</div>
23 </div>
24 <div class="row">
25 <div class="col-md-6">.col-md-6</div>
26 <div class="col-md-6">.col-md-6</div>
27 </div>

```

2、手机、平板、桌面

.col-xs-12 .col-sm-6 .col-md-8

.col-xs-6 .col-md-4

.col-xs-6 .col-sm-4

.col-xs-6 .col-sm-4

.col-xs-6 .col-sm-4

```

1 <div class="row">
2 <div class="col-xs-12 col-sm-6 col-md-8">.col-xs-12 .col-sm-6 .col-md-
8</div>
3 <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
4 </div>
5 <div class="row">
6 <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
7 <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
8 <div class="clearfix visible-xs-block"></div>
9 <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
10 </div>

```

3、列偏移

.col-md-4

.col-md-4 .col-md-offset-4

.col-md-3 .col-md-offset-3

.col-md-3 .col-md-offset-3

.col-md-6 .col-md-offset-3

```

1 <div class="row">
2 <div class="col-md-4">.col-md-4</div>
3 <div class="col-md-4 col-md-offset-4">.col-md-4 .col-md-offset-4</div>
4 </div>
5 <div class="row">
6 <div class="col-md-3 col-md-offset-3">.col-md-3 .col-md-offset-3</div>
7 <div class="col-md-3 col-md-offset-3">.col-md-3 .col-md-offset-3</div>
8 </div>
9 <div class="row">
10 <div class="col-md-6 col-md-offset-3">.col-md-6 .col-md-offset-3</div>
11 </div>

```

4、列嵌套

Level 1: .col-sm-9

Level 2: .col-xs-8 .col-sm-6

Level 2: .col-xs-4 .col-sm-6

```
1 <div class="row">
2   <div class="col-sm-9">
3     Level 1: .col-sm-9
4     <div class="row">
5       <div class="col-xs-8 col-sm-6">
6         Level 2: .col-xs-8 .col-sm-6
7       </div>
8       <div class="col-xs-4 col-sm-6">
9         Level 2: .col-xs-4 .col-sm-6
10      </div>
11    </div>
12  </div>
13 </div>
```

5、列排序

.col-md-3 .col-md-pull-9

.col-md-9 .col-md-push-3

```
1 <div class="row">
2   <div class="col-md-9 col-md-push-3">.col-md-9 .col-md-push-3</div>
3   <div class="col-md-3 col-md-pull-9">.col-md-3 .col-md-pull-9</div>
4 </div>
```

(二) 表单

1、基本表单

单独的表单控件会被自动赋予一些全局样式。所有设置了 `.form-control` 类的 `<input>`、`<textarea>` 和 `<select>` 元素都将被默认设置宽度属性为 `width: 100%`。将 `label` 元素和前面提到的控件包裹在 `.form-group` 中可以获得最好的排列。

```
1 <form>
2   <!-- 表单组: class="form-group" -->
3   <div class="form-group">
4     <label for="exampleInputEmail">Email address</label>
5     <!-- 表单组件样式: class="form-control" -->
6     <input type="email" class="form-control" id="exampleInputEmail"
placeholder="Email">
7   </div>
8   <div class="form-group">
9     <label for="exampleInputPassword1">Password</label>
10    <input type="password" class="form-control"
id="exampleInputPassword1" placeholder="Password">
11  </div>
12 </div>
```

```

13     <label for="exampleInputFile">File input</label>
14     <input type="file" id="exampleInputFile">
15     <p class="help-block">Example block-level help text here.</p>
16 </div>
17 <div class="checkbox">
18     <label>
19         <input type="checkbox"> Check me out
20     </label>
21 </div>
22 <button type="submit" class="btn btn-default">Submit</button>
23 </form>

```

2、内联表单

为 `<form>` 元素添加 `.form-inline` 类可使其内容左对齐并且表现为 `inline-block` 级别的控件。只适用于视口 (viewport) 至少在 768px 宽度时 (视口宽度再小的话就会使表单折叠)。

```

1 <form class="form-inline">
2     <div class="form-group">
3         <label for="exampleInputName2">Name</label>
4         <input type="text" class="form-control" id="exampleInputName2"
placeholder="Jane Doe">
5     </div>
6     <div class="form-group">
7         <label for="exampleInputEmail2">Email</label>
8         <input type="email" class="form-control" id="exampleInputEmail2"
placeholder="jane.doe@example.com">
9     </div>
10    <button type="submit" class="btn btn-default">Send invitation</button>
11 </form>

```

3、水平排列表单

通过为表单添加 `.form-horizontal` 类，并联合使用 Bootstrap 预置的栅格类，可以将 `label` 标签和控件组水平并排布局。这样做将改变 `.form-group` 的行为，使其表现为栅格系统中的行 (row)，因此就无需再额外添加 `.row` 了。

```

1 <form class="form-horizontal">
2     <div class="form-group">
3         <label for="inputEmail3" class="col-sm-2 control-label">Email</label>
4         <div class="col-sm-10">
5             <input type="email" class="form-control" id="inputEmail3"
placeholder="Email">
6         </div>
7     </div>
8     <div class="form-group">
9         <label for="inputPassword3" class="col-sm-2 control-label">Password</label>
10        <div class="col-sm-10">
11            <input type="password" class="form-control" id="inputPassword3"
placeholder="Password">
12        </div>
13    </div>

```

```

14     <div class="form-group">
15         <div class="col-sm-offset-2 col-sm-10">
16             <div class="checkbox">
17                 <label>
18                     <input type="checkbox"> Remember me
19                 </label>
20             </div>
21         </div>
22     </div>
23     <div class="form-group">
24         <div class="col-sm-offset-2 col-sm-10">
25             <button type="submit" class="btn btn-default">Sign in</button>
26         </div>
27     </div>
28 </form>

```

(三) 表格

1、基础表格

```

1  <!-- class="table" -->
2  <table class="table">
3      <tr>
4          <th>1-1</th>
5          <th>1-2</th>
6          <th>1-3</th>
7          <th>1-4</th>
8          <th>1-5</th>
9      </tr>
10     <tr>
11         <td>2-1</td>
12         <td>2-2</td>
13         <td>2-3</td>
14         <td>2-4</td>
15         <td>2-5</td>
16     </tr>
17     <tr>
18         <td>3-1</td>
19         <td>3-2</td>
20         <td>3-3</td>
21         <td>3-4</td>
22         <td>3-5</td>
23     </tr>
24     <tr>
25         <td>4-1</td>
26         <td>4-2</td>
27         <td>4-3</td>
28         <td>4-4</td>
29         <td>4-5</td>
30     </tr>
31     <tr>
32         <td>4-1</td>
33         <td>4-2</td>
34         <td>4-3</td>
35         <td>4-4</td>

```



```

36         <td>4-5</td>
37     </tr>
38     <tr>
39         <td>4-1</td>
40         <td>4-2</td>
41         <td>4-3</td>
42         <td>4-4</td>
43         <td>4-5</td>
44     </tr>
45 </table>

```

2、条纹表格

```

1  <!-- class="table table-striped" -->
2  <table class="table table-striped">
3      <tr>
4          <th>1-1</th>
5          <th>1-2</th>
6          <th>1-3</th>
7          <th>1-4</th>
8          <th>1-5</th>
9      </tr>
10     <tr>
11         <td>2-1</td>
12         <td>2-2</td>
13         <td>2-3</td>
14         <td>2-4</td>
15         <td>2-5</td>
16     </tr>
17     <tr>
18         <td>3-1</td>
19         <td>3-2</td>
20         <td>3-3</td>
21         <td>3-4</td>
22         <td>3-5</td>
23     </tr>
24     <tr>
25         <td>4-1</td>
26         <td>4-2</td>
27         <td>4-3</td>
28         <td>4-4</td>
29         <td>4-5</td>
30     </tr>
31     <tr>
32         <td>4-1</td>
33         <td>4-2</td>
34         <td>4-3</td>
35         <td>4-4</td>
36         <td>4-5</td>
37     </tr>
38     <tr>
39         <td>4-1</td>
40         <td>4-2</td>
41         <td>4-3</td>
42         <td>4-4</td>

```

```
43         <td>4-5</td>
44     </tr>
45 </table>
```

3、边框表格

```
1  <!-- class="table table-bordered" -->
2  <table class="table table-bordered">
3      <tr>
4          <th>1-1</th>
5          <th>1-2</th>
6          <th>1-3</th>
7          <th>1-4</th>
8          <th>1-5</th>
9      </tr>
10     <tr>
11         <td>2-1</td>
12         <td>2-2</td>
13         <td>2-3</td>
14         <td>2-4</td>
15         <td>2-5</td>
16     </tr>
17     <tr>
18         <td>3-1</td>
19         <td>3-2</td>
20         <td>3-3</td>
21         <td>3-4</td>
22         <td>3-5</td>
23     </tr>
24     <tr>
25         <td>4-1</td>
26         <td>4-2</td>
27         <td>4-3</td>
28         <td>4-4</td>
29         <td>4-5</td>
30     </tr>
31     <tr>
32         <td>4-1</td>
33         <td>4-2</td>
34         <td>4-3</td>
35         <td>4-4</td>
36         <td>4-5</td>
37     </tr>
38     <tr>
39         <td>4-1</td>
40         <td>4-2</td>
41         <td>4-3</td>
42         <td>4-4</td>
43         <td>4-5</td>
44     </tr>
45 </table>
```

4、鼠标悬停

```
1  <!-- class="table table-hover" -->
2  <table class="table table-hover">
3      <tr>
4          <th>1-1</th>
5          <th>1-2</th>
6          <th>1-3</th>
7          <th>1-4</th>
8          <th>1-5</th>
9      </tr>
10     <tr class="active">
11         <td class="danger">2-1</td>
12         <td>2-2</td>
13         <td>2-3</td>
14         <td>2-4</td>
15         <td>2-5</td>
16     </tr>
17     <tr class="success">
18         <td>3-1</td>
19         <td>3-2</td>
20         <td>3-3</td>
21         <td>3-4</td>
22         <td>3-5</td>
23     </tr>
24     <tr class="info">
25         <td>4-1</td>
26         <td>4-2</td>
27         <td class="danger">4-3</td>
28         <td>4-4</td>
29         <td>4-5</td>
30     </tr>
31     <tr class="warning">
32         <td>4-1</td>
33         <td>4-2</td>
34         <td>4-3</td>
35         <td>4-4</td>
36         <td>4-5</td>
37     </tr>
38     <tr class="danger">
39         <td>4-1</td>
40         <td>4-2</td>
41         <td>4-3</td>
42         <td>4-4</td>
43         <td>4-5</td>
44     </tr>
45 </table>
```

5、紧缩表格

```
1  <!-- class="table table-condensed" -->
2  <table class="table table-condensed">
3      <tr>
4          <th>1-1</th>
```

```

5         <th>1-2</th>
6         <th>1-3</th>
7         <th>1-4</th>
8         <th>1-5</th>
9     </tr>
10    <tr>
11        <td>2-1</td>
12        <td>2-2</td>
13        <td>2-3</td>
14        <td>2-4</td>
15        <td>2-5</td>
16    </tr>
17    <tr>
18        <td>3-1</td>
19        <td>3-2</td>
20        <td>3-3</td>
21        <td>3-4</td>
22        <td>3-5</td>
23    </tr>
24    <tr>
25        <td>4-1</td>
26        <td>4-2</td>
27        <td>4-3</td>
28        <td>4-4</td>
29        <td>4-5</td>
30    </tr>
31    <tr>
32        <td>4-1</td>
33        <td>4-2</td>
34        <td>4-3</td>
35        <td>4-4</td>
36        <td>4-5</td>
37    </tr>
38    <tr>
39        <td>4-1</td>
40        <td>4-2</td>
41        <td>4-3</td>
42        <td>4-4</td>
43        <td>4-5</td>
44    </tr>
45 </table>

```

6、响应式表格

```

1  <!--
2      将任何 .table 元素包裹在 .table-responsive 元素内，即可创建响应式表格，
3      其会在小屏幕设备上（小于768px）水平滚动。当屏幕大于 768px 宽度时，水平滚动条消失。
4      class="table-responsive"
5  -->
6  <div class="table-responsive">
7      <table class="table">
8          ...
9      </table>
10 </div>

```

7、状态类样式

Class	描述
<code>.active</code>	鼠标悬停在行或单元格上所设置的颜色
<code>.success</code>	标识成功或积极的动作
<code>.info</code>	标识普通的提示信息或动作
<code>.warning</code>	标识警告或需要用户注意
<code>.danger</code>	标识危险或潜在的带来负面影响的动作

#	Column heading	Column heading	Column heading
1	Column content	Column content	Column content
2	Column content	Column content	Column content
3	Column content	Column content	Column content
4	Column content	Column content	Column content
5	Column content	Column content	Column content
6	Column content	Column content	Column content
7	Column content	Column content	Column content
8	Column content	Column content	Column content
9	Column content	Column content	Column content

```
1 <!-- 添加在行 -->
2 <tr class="active">...</tr>
3 <tr class="success">...</tr>
4 <tr class="warning">...</tr>
5 <tr class="danger">...</tr>
6 <tr class="info">...</tr>
7
8 <!-- 添加在列 (`td` or `th`) -->
9 <tr>
10   <td class="active">...</td>
11   <td class="success">...</td>
12   <td class="warning">...</td>
13   <td class="danger">...</td>
14   <td class="info">...</td>
15 </tr>
```

(四) 按钮

```
1 <!-- Standard button -->
2 <button type="button" class="btn btn-default">（默认样式）Default</button>
3
4 <!-- Provides extra visual weight and identifies the primary action in a set
of buttons -->
5 <button type="button" class="btn btn-primary">（首选项）Primary</button>
6
```

```

7 <!-- Indicates a successful or positive action -->
8 <button type="button" class="btn btn-success">(成功) Success</button>
9
10 <!-- Contextual button for informational alert messages -->
11 <button type="button" class="btn btn-info">(一般信息) Info</button>
12
13 <!-- Indicates caution should be taken with this action -->
14 <button type="button" class="btn btn-warning">(警告) Warning</button>
15
16 <!-- Indicates a dangerous or potentially negative action -->
17 <button type="button" class="btn btn-danger">(危险) Danger</button>
18
19 <!-- Deemphasize a button by making it look like a link while maintaining
    button behavior -->
20 <button type="button" class="btn btn-link">(链接) Link</button>

```

四、Bootstrap 插件

(一) 插件概述

jQuery 插件为 Bootstrap 的组件赋予了“生命”。可以简单地一次性引入所有插件，或者逐个引入到你的页面中。栅格系统用于通过一系列的行 (row) 与列 (column) 的组合来创建页面布局，你的内容就可以放入这些创建好的布局中。

(二) 模态框

- 静态模态框

```

1 <!-- 模态框容器: class="modal-dialog" -->
2 <div class="modal-dialog" role="document">
3     <!-- 模态框内容: class="modal-content" -->
4     <div class="modal-content">
5         <!-- 模态框标题: class="modal-header" -->
6         <div class="modal-header">
7             <button type="button" class="close" data-dismiss="modal" aria-
label="close">
8                 <span aria-hidden="true">&times;</span>
9             </button>
10            <h4 class="modal-title">Modal title</h4>
11        </div>
12        <!-- 模态框内容: class="modal-body" -->
13        <div class="modal-body">
14            <p>One fine body&hellip;</p>
15        </div>
16        <!-- 模态框内容: class="modal-footer" -->
17        <div class="modal-footer">
18            <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
19            <button type="button" class="btn btn-primary">Save
changes</button>
20        </div>
21    </div>
22 </div>

```

- 动态模态框

```
1 <!--
2     模态框切换: data-toggle="modal"
3     模态框内容, 指向id标识: data-target="#modal1"
4 -->
5 <button class="btn btn-default" data-toggle="modal" data-target="#modal1">
6     打开模态框
7 </button>
8
9 <!-- 模态框响应容器, 有动画效果: class="modal fade" -->
10 <div class="modal fade" id="modal1">
11     <!-- 模态框容器: class="modal-dialog" -->
12     <div class="modal-dialog" role="document">
13         <!-- 模态框内容: class="modal-content" -->
14         <div class="modal-content">
15             <!-- 模态框标题: class="modal-header" -->
16             <div class="modal-header">
17                 <button type="button" class="close" data-dismiss="modal"
aria-label="Close">
18                     <span aria-hidden="true">&times;</span>
19                 </button>
20                 <h4 class="modal-title">Modal title</h4>
21             </div>
22             <!-- 模态框内容: class="modal-body" -->
23             <div class="modal-body">
24                 <p>One fine body&hellip;</p>
25             </div>
26             <!-- 模态框内容: class="modal-footer" -->
27             <div class="modal-footer">
28                 <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
29                 <button type="button" class="btn btn-primary">Save
changes</button>
30             </div>
31         </div>
32     </div>
33 </div>
```

- 大小模态框

```
1 <!-- 大模态框 -->
2 <button type="button" class="btn btn-primary" data-toggle="modal" data-
target=".bs-example-modal-lg">Large
3     modal</button>
4
5 <div class="modal fade bs-example-modal-lg" tabindex="-1" role="dialog"
aria-labelledby="myLargeModalLabel">
6     <div class="modal-dialog modal-lg" role="document">
7         <div class="modal-content">
8             <!-- 模态框标题: class="modal-header" -->
9             <div class="modal-header">
10                 <button type="button" class="close" data-dismiss="modal"
aria-label="Close">
11                     <span aria-hidden="true">&times;</span>
```

```

12         </button>
13         <h4 class="modal-title">Modal title</h4>
14     </div>
15     <!-- 模态框内容: class="modal-body" -->
16     <div class="modal-body">
17         <p>One fine body&hellip;</p>
18     </div>
19     <!-- 模态框内容: class="modal-footer" -->
20     <div class="modal-footer">
21         <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
22         <button type="button" class="btn btn-primary">Save
changes</button>
23     </div>
24 </div>
25 </div>
26 </div>
27
28 <!-- 小模态框 -->
29 <button type="button" class="btn btn-primary" data-toggle="modal" data-
target=".bs-example-modal-sm">Small
modal</button>
30
31
32 <div class="modal fade bs-example-modal-sm" tabindex="-1" role="dialog"
aria-labelledby="mySmallModalLabel">
33     <div class="modal-dialog modal-sm" role="document">
34         <div class="modal-content">
35             <!-- 模态框标题: class="modal-header" -->
36             <div class="modal-header">
37                 <button type="button" class="close" data-dismiss="modal"
aria-label="Close">
38                     <span aria-hidden="true">&times;</span>
39                 </button>
40                 <h4 class="modal-title">Modal title</h4>
41             </div>
42             <!-- 模态框内容: class="modal-body" -->
43             <div class="modal-body">
44                 <p>One fine body&hellip;</p>
45             </div>
46             <!-- 模态框内容: class="modal-footer" -->
47             <div class="modal-footer">
48                 <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
49                 <button type="button" class="btn btn-primary">Save
changes</button>
50             </div>
51         </div>
52     </div>
53 </div>

```

- JS 操作模态框

```

1 $(``).modal('show'); // 打开模态框
2 $(``).modal('hide'); // 关闭模态框

```


名称	类型	默认值	描述
backdrop	boolean 或 字符串 'static'	true	操作模态框的背景。包括模态背景元素。或者，为不在单击时关闭模式的背景指定静态。
keyboard	boolean	true	键盘上的 esc 键被按下时关闭模态框。
show	boolean	true	模态框初始化之后就立即显示出来。
remote	path	false	<p>自v3.3.0以来，此选项已被弃用，并已在v4中删除。我们建议使用客户端模板或数据绑定框架，或者调用jQuery。给自己加载。</p> <p>如果提供的是 URL，将利用 jQuery 的 <code>load</code> 方法从此 URL 地址加载要展示的内容（只加载一次）并插入 <code>.modal-content</code> 内。</p> <p>如果使用的是 data 属性 API，还可以利用 <code>href</code> 属性指定内容来源地址。下面是一个实例：<code><a data-toggle="modal" href="remote.html" data-target="#modal">Click me</code></p>

```
1 <button type="button" onclick="openModal()">打开模态框</button>
2
3 <div id="modal1">
4     <!-- 省略内部 -->
5 </div>
6
7 <script>
8     function openModal(){
9         $("#modal1").modal('show');
10    }
11 </script>
```

(三) 滚动监听

```
1 <!-- 创建导航栏 -->
2 <div id="div1" class="navbar navbar-default navbar-fixed-top">
3     <ul class="nav navbar-nav">
4         <li><a href="#title1">标题1</a></li>
5         <li><a href="#title2">标题2</a></li>
6         <li><a href="#title3">标题3</a></li>
7         <li><a href="#title4">标题4</a></li>
8     </ul>
9 </div>
10
11 <!--
12     创建滚动内容
13     绑定滚动监听事件: data-spy="scroll"
14     当前滚动内容和哪个导航关联: data-target="#div1"
15     滚动内容对象必须设置:
16     position: relative; 相对定位;
```

```

17         overflow: scroll;隐藏多余内容显示滚动条
18     -->
19     <div data-spy="scroll" data-target="#div1" >
20         <h3 id="title1">标题1</h3>
21         ....
22         <h3 id="title2">标题2</h3>
23         ....
24         <h3 id="title3">标题3</h3>
25         ....
26         <h3 id="title4">标题4</h3>
27     </div>

```

五、Bootstrap Table

(一) Bootstrap Table简介

与一些最广泛使用的CSS框架集成的**扩展表**。（支持Bootstrap，语义UI，Bulma，Material Design，Foundation）

Bootstrap Table 旨在减少开发时间，并且不需要开发人员的特定知识。它既轻量级又功能丰富。

<https://www.bootstrap-table.com.cn/index.html>

(二) 入门

1、导入文件

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title></title>
6          <!-- 先导入Bootstrap 的基础样式 -->
7          <link href="css/bootstrap.css" rel="stylesheet" type="text/css"/>
8          <link href="css/bootstrap-theme.css" rel="stylesheet"
type="text/css"/>
9
10         <!-- 导入 BootstrapTable 的样式 -->
11         <link rel="stylesheet" href="css/bootstrap-table.css">
12     </head>
13     <body>
14
15         <!-- JS建议导入在body中，为了避免没有页面内容的时候就执行了js，会报错 -->
16         <!-- 导入 jQuery 的内容 -->
17         <script src="js/jquery-3.6.0.min.js" type="text/javascript">
</script>
18         <!-- 先导入 Bootstrap 的基础JS -->
19         <script src="js/bootstrap.js" type="text/javascript"></script>
20
21         <!-- 导入 BootstrapTable 的JS -->
22         <script src="js/bootstrap-table.js" type="text/javascript"></script>
23         <!-- 语言包 -->
24         <script src="js/bootstrap-table-zh-CN.js" type="text/javascript">
</script>
25     </body>

```

2、基础用法

- 使用属性创建表格

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title></title>
6          <!-- 先导入Bootstrap 的基础样式 -->
7          <link href="css/bootstrap.css" rel="stylesheet" type="text/css" />
8          <link href="css/bootstrap-theme.css" rel="stylesheet"
type="text/css" />
9
10         <!-- 导入 BootstrapTable 的样式 -->
11         <link rel="stylesheet" href="css/bootstrap-table.css">
12     </head>
13     <body>
14
15         <table data-toggle="table">
16             <thead>
17                 <tr>
18                     <th>Item ID</th>
19                     <th>Item Name</th>
20                     <th>Item Price</th>
21                 </tr>
22             </thead>
23             <tbody>
24                 <tr>
25                     <td>1</td>
26                     <td>Item 1</td>
27                     <td>$1</td>
28                 </tr>
29                 <tr>
30                     <td>2</td>
31                     <td>Item 2</td>
32                     <td>$2</td>
33                 </tr>
34             </tbody>
35         </table>
36
37         <!-- JS建议导入在body中, 为了避免没有页面内容的时候就执行了js, 会报错 -->
38         <!-- 导入 jQuery 的内容 -->
39         <script src="js/jquery-3.6.0.min.js" type="text/javascript">
</script>
40         <!-- 先导入 Bootstrap 的基础JS -->
41         <script src="js/bootstrap.js" type="text/javascript"></script>
42
43         <!-- 导入 BootstrapTable 的JS -->
44         <script src="js/bootstrap-table.js" type="text/javascript"></script>
45         <!-- 语言包 -->
46         <script src="js/bootstrap-table-zh-CN.js" type="text/javascript">
</script>

```

```
47     </body>
48 </html>
```

- 通过远程路径创建表格

远程路径对应的JSON文件

```
1  [
2    {
3      "no": "10001",
4      "name": "Tom",
5      "sex": "男",
6      "tel": "12345678901"
7    },
8    {
9      "no": "10002",
10     "name": "Jack",
11     "sex": "男",
12     "tel": "98765432101"
13   },
14   {
15     "no": "10003",
16     "name": "Lily",
17     "sex": "女",
18     "tel": "11166677789"
19   }
20 ]
```

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title></title>
6      <!-- 先导入Bootstrap 的基础样式 -->
7      <link href="css/bootstrap.css" rel="stylesheet" type="text/css" />
8      <link href="css/bootstrap-theme.css" rel="stylesheet"
9      type="text/css" />
10
11     <!-- 导入 BootstrapTable 的样式 -->
12     <link rel="stylesheet" href="css/bootstrap-table.css">
13   </head>
14   <body>
15
16     <!--
17     // 读取远程服务器的JSON文件
18     data-url="student.json">
19     -->
20     <table data-toggle="table" data-url="student.json">
21       <thead>
22         <tr>
23           <!--
24           获取JSON数据中的key，要写在表格的头信息上
25           data-field="no"
26           -->
27           <th data-field="no">学号</th>
```

```

27         <th data-field="name">姓名</th>
28         <th data-field="sex">性别</th>
29         <th data-field="tel">电话</th>
30     </tr>
31 </thead>
32 <tbody>
33 </tbody>
34 </table>
35
36 <!-- JS建议导入在body中，为了避免没有页面内容的时候就执行了js，会报错 -->
37 <!-- 导入 jQuery 的内容 -->
38 <script src="js/jquery-3.6.0.min.js" type="text/javascript">
</script>
39 <!-- 先导入 Bootstrap 的基础JS -->
40 <script src="js/bootstrap.js" type="text/javascript"></script>
41
42 <!-- 导入 BootstrapTable 的JS -->
43 <script src="js/bootstrap-table.js" type="text/javascript"></script>
44 <!-- 语言包 -->
45 <script src="js/bootstrap-table-zh-CN.js" type="text/javascript">
</script>
46 </body>
47 </html>

```

- 通过JS创建表格

```

1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta charset="utf-8">
5         <title></title>
6         <!-- 先导入Bootstrap 的基础样式 -->
7         <link href="css/bootstrap.css" rel="stylesheet" type="text/css" />
8         <link href="css/bootstrap-theme.css" rel="stylesheet"
type="text/css" />
9
10         <!-- 导入 BootstrapTable 的样式 -->
11         <link rel="stylesheet" href="css/bootstrap-table.css">
12     </head>
13     <body>
14         <!-- 创建表格容器即可 -->
15         <table id="mytable"></table>
16
17         <!-- JS建议导入在body中，为了避免没有页面内容的时候就执行了js，会报错 -->
18         <!-- 导入 jQuery 的内容 -->
19         <script src="js/jquery-3.6.0.min.js" type="text/javascript">
</script>
20         <!-- 先导入 Bootstrap 的基础JS -->
21         <script src="js/bootstrap.js" type="text/javascript"></script>
22
23         <!-- 导入 BootstrapTable 的JS -->
24         <script src="js/bootstrap-table.js" type="text/javascript"></script>
25         <!-- 语言包 -->
26         <script src="js/bootstrap-table-zh-CN.js" type="text/javascript">
</script>

```

```
27
28 <script>
29 // jQuery 页面加载完成之后的简介格式
30 $(function(){
31 // 通过ID操作指定表格
32 $("#mytable").bootstrapTable({
33 // 表格的相关数据
34 // url: 'student.json', // 请求后台的地址
35 url: '/student/studentJSON', // 请求后台的地址
36 columns: [{
37     checkbox: true, // 当前列的复选框
38 }, {
39     field: 'no', // 从后台获取JSON数据的key
40     title: '学号', // 页面表格当前列的题目, 表头
41     sortable: true, // 支持排序
42 }, {
43     field: 'name',
44     title: '姓名',
45     sortable: true, // 支持排序
46 }, {
47     field: 'sex',
48     title: '性别',
49     formatter: function(value, row, index){
50         if(value == 1){
51             return "男";
52         } else {
53             return "女";
54         }
55     }
56 }, {
57     field: 'tel',
58     title: '电话'
59 }, {
60     field: 'address',
61     title: '地址'
62 }, {
63     field: 'clazz',
64     title: '班级'
65 }, {
66     field: 'delFlag',
67     title: '状态'
68 }, {
69     title: '操作',
70     field: 'no',
71     align: 'center',
72     // formatter: 格式化当前列的内容
73     // value: 当前列遍历属性的值
74     // row: 当前行的所有数据
75     // index: 当前行的下标
76     formatter: function(value, row, index){
77         console.log(value);
78         console.log(row);
79         console.log(index);
80
81         var html =
```

```

82         '<button class="btn btn-danger">删除 (' +
value + ') </button> ' +
83         '<button class="btn btn-info">编辑 (' + value
+ ') </button>';
84
85         return html;
86     }
87     },],
88     });
89 });
90 </script>
91 </body>
92 </html>

```

后台Java代码：返回JSON数据

```

1  @Controller
2  @RequestMapping("/student")
3  public class StudentController {
4
5      @Resource
6      StudentService studentService;
7
8      @GetMapping("/studentJSON")
9      @ResponseBody
10     public List<Student> queryAll(Student student){
11
12         List<Student> list = studentService.queryAll(null);
13
14         return list;
15     }
16 }

```

(三) 表格操作

1、常用属性

常用表格属性：

```

1  pagination: true, // 分页组件
2  pageSize: 5, // 设置每页显示条数
3  pageList: [5,10,15,20], // 分页的每页显示条数的数值
4  search: true, // 搜索框，在页面进行搜索，不调用后台，直接在页面检索内容
5  showRefresh: true, // 显示刷新按钮
6  showToggle: true, // 显示切换视图按钮
7  showColumns: true, // 列的显示与否

```

常用列属性：

```

1  field: 'no', // 当前列的数据，获取的JSON数据中的key，通过key可以给表格填充数据
2  title: '学号', // 当前列的列名
3  sortable: true, // 当前列支持排序
4  checkbox: true, // 当前列设置为复选框

```

常用列方法：

```
1 // 格式化当前列内容的格式，通过return 字符串的形式，浏览器会自动解析为HTML
2 // value: 当前列的值
3 // row: 当前行的所有数据
4 // index: 当前行的下标，从0开始
5 formatter: function(value, row, index){
6
7 }
```

2、API

2.1 表的各项(Table options)

名称	属性	类型	默认值	作用描述
-	data-toggle	String	table	只要引入 jquery、bootstrap、bootstrap-table的包，不用在js里面定义就可以使用
classes	data-classes	String	table table-hover	表的class属性，如果没有自己定义，则默认有边框，并且当鼠标悬浮在那一行，背景会变为浅灰色。
sortClass	data-sort-class	String	undefined	声明表格td的class名，代表此列元素的class名将被排序
height	data-height	Number	undefined	表格的高度
undefinedText	data-undefined-text	String	-	当不写任何内容默认显示‘-’
striped	data-striped	Boolean	false	默认false，当设为true，则每行表格的背景会显示灰白相间
sortName	data-sort-name	String	undefined	定义哪一列的值会被排序，写data-field的自定义名，默认递增（asc）
sortOrder	data-sort-order	String	asc	同上面的结合使用，默认递增（asc），也可设为递减（desc）
sortStable	data-sort-stable	Boolean	false	（别看错了，是sortStable，sortable在下面）默认false，设为true，则和sort部分一样，区别是：在排序过程中，如果存在相等的元素，则原来的顺序不会改变。原文还有一句：（如果你把此属性设为了true）我们将为此行添加'_position'属性
iconsPrefix	data-icons-prefix	String	glyphicon	定义字体库（'Glyphicon' or 'fa' for FontAwesome),使用“fa”时需引用FontAwesome，并且配合 icons 属性实现效果 Glyphicon 集成于Bootstrap可免费使用
iconSize	data-icon-size	String	undefined	定义的图标大小： - undefined =>默认表示默认的按钮尺寸（btn） - xs =>超小按钮的尺寸（btn-xs） - sm =>小按钮的尺寸（btn-sm） - lg =>大按钮的尺寸（btn-lg）

名称	属性	类型	默认值	作用描述
buttonsClass	data-buttons-class	String	default	按钮的类，默认为default。- 可选的有：primary、danger、warning等等 - 写了之后会自动转换为btn-primary（蓝色）、btn-danger（红色）、btn-warning（黄色）等格式，所以前面不要再加“btn-”，默认为btn-default（白色） - 参考菜鸟教程： Bootstrap 按钮
icons	data-icons	Object	{ paginationSwitchDown: 'glyphicon-collapse-down icon-chevron-down', paginationSwitchUp: 'glyphicon-collapse-up icon-chevron-up', refresh: 'glyphicon-refresh icon-refresh', toggle: 'glyphicon-list-alt icon-list-alt', columns: 'glyphicon-th icon-th', detailOpen: 'glyphicon-plus icon-plus', detailClose: 'glyphicon-minus icon-minus' }	定义在工具栏、分页、详细视图中使用的图标 - 没办法解释，请参考菜鸟教程的图标： Bootstrap 字体图标
columns	-	Array	[]	默认空数组，在JS里面定义，field即data-field，title就是每列表头名等等。 - 请参考： 点击Bootstrap-table的右侧边栏Usage
data	-	Array	[]	被加载的数据。 - 也就是从服务器获取的数据，通过“.”运算符获取，例如“data.date/data.anything”，后面是服务器发来的字段名
dataField	data-data-field	String	rows	- 名称写自己定义的每列的字段名，也就是key，通过key才能给某行的某列赋value。 - 原文：获取每行数据json内的key - 例如：{"name": "zz", "age": 20}, name和age就是key，如果这是从服务端请求的json，那可能和每列定义的字段不同，但都是唯一的
ajax	data-ajax	Function	undefined	- ajax方法，和jQuery的ajax方法类似 - 参考 BootstrapTable实现定时刷新数据 ，只是参考，因为那是jQuery的ajax方法
method	data-method	String	get	向服务器请求远程数据的方式，默认为'get'，可选'post'

名称	属性	类型	默认值	作用描述
url	data-url	String	undefined	向服务器请求的url。 - 例如：server + “get_app_player”和server + ‘get_channel_list’，两者都是向server（server是自己定义的，例如“ http://kanshakan.nichousha.com/ ”）请求数据，只是接口不同，一个是请求该游戏玩家信息，一个是请求渠道信息 下面看看原文： - 向远程站点请求数据的URL - 请记住，如果服务端分页选项使用了自定义的，那么请求的服务器响应格式是不同的，参考下面两个数据格式： Without server-side pagination（没有启用服务端分页 - data1.json） With server-side pagination（启用服务端分页 - data2.json）
cache	data-cache	Boolean	true	默认缓存ajax请求，设为false则禁用缓存
contentType	data-content-type	String	application/json	请求数据的contentType（内容类型），用于定义文件的类型，决定接收方以什么形式读取这个文件。 - 默认application/json，用来告诉接收端从服务器发来的消息是序列化后的json字符串
dataType	data-data-type	String	json	期望从服务器获取的数据的类型，默认为json格式字符串
ajaxOptions	data-ajax-options	Object	{}	向服务器请求ajax时的附加项，默认无附加 - 参考 jQuery.ajax()
queryParams	data-query-params	Function	<pre>function(params){ return params;}</pre>	当请求数据时，你可以通过修改queryParams向服务器发送其余的参数。 - queryParamsType="limit"，它的参数包括：limit, offset, search, sort, order Else - params 包括：pageSize, pageNumber, searchText, sortName, sortOrder - 当return false，请求则终止
queryParamsType	data-query-params-type	String	limit	默认“limit”，设置该属性用来发送符合RESTful格式的参数
responseHandler	data-response-handler	Function	<pre>function(res){ return res;}</pre>	在加载服务器发送来的数据之前，处理数据的格式，参数res表示the response data（获取的数据）
pagination	data-pagination	Boolean	false	默认为false，表格的底部工具栏不会显示分页条（pagination toolbar），可以设为true来显示
paginationLoop	data-pagination-loop	Boolean	true	默认true，分页条无限循环
onlyInfoPagination	data-only-info-pagination	Boolean	false	前提：pagination设为true，启用了分页功能。 - 默认false，设为true显示总记录数

名称	属性	类型	默认值	作用描述
sidePagination	data-side-pagination	String	client	设置在哪进行分页，默认“client”，可选“server”，如果设置“server”，则必须设置url或者重写ajax方法 - 请记住，如果服务端分页选项使用了自定义的，那么请求的服务器响应格式是不同的，参考下面两个数据格式： Without server-side pagination (没有启用服务端分页) 、 With server-side pagination (启用服务端分页)
pageNumber	data-page-number	Number	1	前提：pagination设为true，启用了分页功能。- 默认第1页，用于设置初始的页数
pageSize	data-page-size	Number	10	前提：pagination设为true，启用了分页功能。- 默认每页显示10条记录，用于设置每页初始显示的条数
pageList	data-page-list	Array	[10, 25, 50, 100]	前提：pagination设为true，启用了分页功能。- 默认为[10, 25, 50, 100]，即可以选择“每页显示10/25/50/100条记录”，用于设置选择每页显示的条数
selectItemName	data-select-item-name	String	btSelectItem	radio（单选按钮）或checkbox（复选框）的字段名
smartDisplay	data-smart-display	Boolean	true	默认为true，会机智地（<_<）根据情况显示分页（pagination）或卡片视图（card view）
escape	data-escape	Boolean	false	跳过插入HTML中的字符串，替换掉特殊字符（后面符号没有逗号）： &, <, >, ", ` , '
search	data-search	Boolean	false	默认false不显示表格右上方搜索框，可设为true，在搜索框内只要输入内容即开始搜索
searchOnEnterKey	data-search-on-enter-key	Boolean	false	默认false不启用，设为true启用，比功能是与上面相比，在搜索框内输入内容并且按下回车键才开始搜索
strictSearch	data-strict-search	Boolean	false	设为true，开启精确搜索
searchText	data-search-text	String	""	前提：search设为true，启用了搜索功能。- 搜索框初始显示的内容，默认空字符串
searchTimeOut	data-search-time-out	Number	500	前提：search设为true，启用了搜索功能。- 设置搜索文件的超时时间 （原文：Set timeout for search fire，不知道是写错了还是我知识面太窄，“search fire”是什么意思，官方API错了？给我整蒙逼了）
trimOnSearch	data-trim-on-search	Boolean	true	默认true，自动忽略空格
showHeader	data-show-header	Boolean	true	默认为true显示表头，设为false隐藏
showFooter	data-show-footer	Boolean	false	默认为false隐藏表尾，设为true显示

名称	属性	类型	默认值	作用描述
showColumns	data-show-columns	Boolean	false	默认为false隐藏某列下拉菜单，设为true显示
showRefresh	data-show-refresh	Boolean	false	默认为false隐藏刷新按钮，设为true显示
showToggle	data-show-toggle	Boolean	false	默认为false隐藏视图切换按钮，设为true显示
showPaginationSwitch	data-show-pagination-switch	Boolean	false	默认为false隐藏每页数据条数选择，设为true显示

2.2 列的各项(Column options)

名称	属性	类型	默认值	作用描述
radio	data-radio	Boolean	false	默认false不显示radio（单选按钮），设为true则显示，radio宽度是固定的
checkbox	data-checkbox	Boolean	false	默认false不显示checkbox（复选框），设为true则显示，checkbox的每列宽度已固定
field	data-field	String	undefined	是每列的字段名，不是表头所显示的名字，通过这个字段名可以给其赋值，相当于key，表内唯一
title	data-title	String	undefined	这个是表头所显示的名字，不唯一，如果你喜欢，可以把所有表头都设为相同的名字
titleTooltip	data-title-tooltip	String	undefined	当悬浮在某控件上，出现提示 - 参考 Bootstrap 提示工具 (Tooltip) 插件
class	class/data-class	String	undefined	没什么好说的，就是class
rowspan	rowspan/data-rowspan	Number	undefined	每格所占的行数
colspan	colspan/data-colspan	Number	undefined	每格所占的列数
align	data-align	String	undefined	每格内数据的对齐方式，有：left（靠左）、right（靠右）、center（居中）

名称	属性	类型	默认值	作用描述
halign	data-halign	String	undefined	table header（表头）的对齐方式，有：left（靠左）、right（靠右）、center（居中）
falign	data-falign	String	undefined	table footer（表脚，就这样译，任性，其实随便译啦，知道就好）的对齐方式，有：left（靠左）、right（靠右）、center（居中）
valign	data-valign	String	undefined	每格数据的对齐方式，有：top（靠上）、middle（居中）、bottom（靠下）
width	data-width	Number (单位：px或%)	undefined	每列的宽度。- 如果没有自定义宽度，那么宽度会根据内容的宽度自适应。 - 如果表是左适应（left responsive）或者设置的宽度小于内容的宽度，那么，宽度还是会自适应（可以在class或其他的属性中使用 min-width 或 max-width）。- 你也可以使用“%”作为单位，这样的话，bootstrapTable将按百分比划分，如果你想使用“pixels（像素值）”，你可以只写数字（只要不加“%”，单位默认“px”，不嫌麻烦，或者想更清晰，你也可以加上“px”）
sortable	data-sortable	Boolean	false	默认false就默认显示，设为true则会被排序
order	data-order	String	asc	默认的排序方式为“asc（升序）”，也可以设为“desc（降序）”。- 与上面的结合使用，不然都不让排序了，你还考虑什么升降。

名称	属性	类型	默认值	作用描述
visible	data-visible	Boolean	true	默认为true显示该列，设为false则隐藏该列。 - 还是很有用的，例如隐藏自定义index列，按某属性排序后会变乱，你可以读取某行的index来进行赋值
cardVisible	data-card-visible	Boolean	true	默认为true显示该列，设为false则隐藏。
switchable	data-switchable	Boolean	true	默认为true显示该列，设为false则禁用列项目的选项卡。
clickToSelect	data-click-to-select	Boolean	true	默认true不响应，设为false则当点击此行的某处时，不会自动选中此行的checkbox（复选框）或radiobox（单选按钮）
formatter	data-formatter	Function	undefined	需要此列的对象。某格的数据转换函数，需要三个参数： -value: field (字段名) -row: 行的数据 -index: 行的 (索引) index
footerFormatter	data-footer-formatter	Function	undefined	需要此列的对象。某格的数据转换函数，需要一个参数： -data: 所有行数据的数组 函数需要返回 (return) footer某格内所要显示的字符串的格式，还要包括内容
events	data-events	Object	undefined	当某格使用formatter函数时，事件监听会响应，需要四个参数： -event: jQuery事件（参考Events）。 - value: 字段名 - row: 行数据 - index: 此行的index 举个例子： <th .. data-events="operateEvent"> var operateEvents = { 'click .like': function (e, value, row, index) {} };

名称	属性	类型	默认值	作用描述
sorter	data-sorter	Function	undefined	自定义的排序函数，实现本地排序，需要两个参数： - a: 第一个字段名 - b: 第二个字段名
sortName	data-sort-name	String	undefined	除了表头默认的sort-name或列的字段名，还可以使用自定义的sort-name 对特殊情况说明： - 一个列显示的内容或许是“html”代码，如： abc ，但是被排列的是html代码中的内容（content）：abc
cellStyle	data-cell-style	Function	undefined	对某格中显示样式（style）进行改变，需要三个函数： - value: 字段名 - row: 行数据 - index: 此行的index - field: 行的字段名 支持classes和css，用法如下： function cellStyle(value, row, index, field) { return { classes: 'text-nowrap another-class', css: {“color”: “blue”, “font-size”: “50px”} }; }
searchable	data-searchable	Boolean	true	默认true，表示此列数据可被查询
searchFormatter	data-search-formatter	Boolean	true	默认true，可使用格式化的数据查询
escape	data-escape	Boolean	false	跳过插入HTML中的字符串，替换掉特殊字符（后面符号没有逗号）： &, <, >, ", ` , '

2.3 事件

中的内容没显示前触发中的内容被加载完后或者在你所用的DOM中有定义则被触发中的内容被加载完后或者在你所用的DOM中有定义则被触发

事件名	定义在jQuery中的事件名	参数	作用描述
-----	----------------	----	------

事件名	定义在jQuery中的事件名	参数	作用描述
onAll	all.bs.table	name, args	当所有触发器被触发时都执行此函数，就是所有的事件都会触发该事件，参数包括： - name: 事件名 - args: 事件的数据
onClickRow	click-row.bs.table	row, \$element, field	当单击此行的任意一处，就会触发该事件，所需参数如下（只是按原文译过来的，在bootstrapable的js中不需要这些参数，因为单击该行就会获得一个row，row中包括此行所有数据，只要通过此行某列的字段名即可取出）： - row: 哪一行（从0开始） - \$element: 该行（tr） - field: 当点击该行，你想获取此行哪一格的数据，field就是此列字段名 - 参考： Bootstrap Table 双击或单击行获取该行内容
onDbClickRow	dbl-click-row.bs.table	row, \$element, field	和上面差不多，当双击（666）此行的任意一处，就会触发该事件，所需参数如下（只是按原文译过来的，在bootstrapable的js中不需要这些参数，因为双击该行就会获得一个row，row中包括此行所有数据，只要通过此行某列的字段名即可取出）： - row: 哪一行（从0开始） - \$element: 该行（tr） - field: 当点击该行，你想获取此行哪一格的数据，field就是此列字段名 - 参考： Bootstrap Table 双击或单击行获取该行内容
onClickCell	click-cell.bs.table	field, value, row, \$element	当单击某一格，就会触发该事件，所需参数如下： - field: 此格所在列的字段名 - value: 此格的数据 - \$element: 此行的此列，就是此格（td）
onDbClickCell	dbl-click-cell.bs.table	field, value, row, \$element	当双击某一格，就会触发该事件，所需参数如下： - field: 此格所在列的字段名 - value: 此格的数据 - \$element: 此行的此列，就是此格（td）
onSort	sort.bs.table	name, order	当对某列进行排序时触发该事件，所需参数如下： - name: 所排序的列的字段名 - order: 所排的顺序

事件名	定义在jQuery中的事件名	参数	作用描述
onCheck	check.bs.table	row, \$element	当选择 (check) 此行时触发, 所需的参数如下: - row: 所选择的行的字段名 - \$element: 此行的DOM元素
onUncheck	uncheck.bs.table	row, \$element	当取消选择 (uncheck) 此行时触发, 所需的参数如下: - row: 所取消选择的行的字段名 - \$element: 此行的DOM元素
onCheckAll	check-all.bs.table	rows	当全选行时触发, 所需的参数如下: - rows: 最近 (newly) 所选择的行的字段名的数组
onUncheckAll	uncheck-all.bs.table	rows	当取消全选行时触发, 所需的参数如下: - rows: 之前或上次 (previously) 所选择的行的字段名的数组
onCheckSome	check-some.bs.table	rows	当选择 (check) 某些行 (多行, rows) 时触发, 所需的参数如下: - rows: 之前或上次 (previously) 所选择的行的字段名的数组
onUncheckSome	uncheck-some.bs.table	rows	当取消选择 (uncheck) 某些行 (多行, rows) 时触发, 所需的参数如下: - rows: 之前或上次 (previously) 所选择的行的字段名的数组
onLoadSuccess	load-success.bs.table	data	当所有数据被加载时触发
onLoadError	load-error.bs.table	status, res	当加载某些数据出现错误时触发
onColumnSwitch	column-switch.bs.table	field, checked	当某列改变是否可见的状态时触发
onColumnSearch	column-search.bs.table	field, text	当某列被查询时触发
onPageChange	page-change.bs.table	number, size	当改变此页所显示的数据条数或改变页码时触发
onSearch	search.bs.table	text	当查询此表时触发
onToggle	toggle.bs.table	cardView	当改变表的视图时触发
onPreBody	pre-body.bs.table	data	当

事件名	定义在jQuery中的事件名	参数	作用描述
onPostBody	post-body.bs.table	data	当
onPostHeader	post-header.bs.table	none	当
onExpandRow	expand-row.bs.table	index, row, \$detail	当查看详细视图（点击查看detail的图标）时触发
onCollapseRow	collapse-row.bs.table	index, row	当关闭详细视图（再次点击查看detail的图标）时触发
onRefreshOptions	refresh-options.bs.table	options	当刷新各项后或在初始化表（也包括销毁了再初始化）之前触发
onResetView	reset-view.bs.table		当重置（reset）表的视图时触发
onRefresh	refresh.bs.table	params	当点击刷新按钮（refresh，不是浏览器的，而是表格右上角的刷新按钮）后触发

2.4 方法

方法名	参数	描述	举例
getOptions	none	返回各项的object	<code>\$table.bootstrapTable('getOptions');</code> - 请查看: getOptions
getSelections	none	返回被选择的行，当没有行被选择，则返回空数组	<code>\$table.bootstrapTable('getSelections');</code> - 请查看: getSelections
getAllSelections	none	返回所有被选的行（原文中：contain search or filter，其实就是你选择的筛选之后的数据），当没有行被选择，则返回空数组	<code>\$table.bootstrapTable('getAllSelections');</code> - 请查看: getAllSelections
showAllColumns	none	显示所有的列	<code>\$table.bootstrapTable('showAllColumns');</code> - 就是显示所有的列，并没什么可说的
hideAllColumns	none	隐藏所有的列	<code>\$table.bootstrapTable('hideAllColumns');</code> - 就是隐藏所有的列
getData	useCurrentPage	获取被加载的表的数据，如果你设置了使用当前页的数据（useCurrentPage），则返回当前页的数据	<code>\$table.bootstrapTable('getData');</code> - 请查看: getData
getRowByUniqueId	id	获取你想要的某行的数据（设置某行的id）	<code>\$table.bootstrapTable('getRowByUniqueId', 1);</code> 后面的1就是你要的那一行的ID - 请查看: getRowByUniqueId
load	data	向表中加载数据，原来的数据将被移除	<code>\$table.bootstrapTable('load', data);</code> - 请查看: load
append	data	就是追加，并不能在表的哪一行后添加，只能在最后一行后面	<code>\$table.bootstrapTable('append', data);</code> data可以是数组 - 请查看: append

方法名	参数	描述	举例
mergeCells	options	融合~（把多格合并为一格），所需的参数如下： - index：所要合并的格所在行的索引（index） - field：所在列的字段名 - rowspan：合并的行总数目 - colspan：合并的列总数目	<code>\$table.bootstrapTable('mergeCells', {index: 1, field: 'name', colspan: 2, rowspan: 3});</code> - 请查看： mergeCells
getHiddenColumns	-	获取所有隐藏的列	<code>\$table.bootstrapTable('getHiddenColumns');</code>