

Bootstrap基础

2020年9月21日 8:33

前端框架Bootstrap

该框架已经帮你写好了很多页面样式，你如果需要使用，只需要下载它对应文件，之后直接复制即可

在使用Bootstrap的时候所有的页面样式都只需要你通过class来调节即可

版本选择建议使用v3版本: <https://v3.bootcss.com/>

注意

bootstrap的js代码是依赖于jQuery的，也就意味着你在使用Bootstrap动态效果的时候，一定要导入jQuery

布局容器

```
<div class="container">
    左右两侧有留白
</div>

<div class="container-fluid">
    左右两侧没有留白
</div>

# 后续在使用bootstrap做页面的时候 上来先写一个div class=container,之后在div内部
  书写页面
```

栅格系统

```
<div class="row"></div>
写一个row就是将所在的区域划分成12份

<div class="col-md-6"> 获取你所要的份数
# 在使用bootstrap的时候 脑子里面一定要做12的加减法 | 1
```

栅格参数

```
.col-xs- .col-sm- .col-md- .col-lg-  
# 针对不同的显示器 bootstrap会自动选择对应的参数  
# 如果你想要兼容所有的显示器 你就全部加上即可  
  
# 在一行如何移动位置  
<div class="col-md-8 c1 col-md-offset-2"></div>
```

表格

```
<table class="table table-hover table-striped table-bordered">  
  
  <tr class="success">  
    <td>1</td>  
    <td>jason</td>  
    <td>123</td>  
    <td>study</td>  
  </tr>  
  
  <tr class="active">...</tr>  
  <tr class="success">...</tr>  
  <tr class="warning">...</tr>  
  <tr class="danger">...</tr>  
  <tr class="info">...</tr>
```

表单

```
<div class="container">
  <div class="col-md-8 col-md-offset-2">
    <h2 class="text-center">登陆页面</h2>
    <form action="">
      <p>username:<input type="text" class="form-control"></p>
      <p>password:<input type="text" class="form-control"></p>
      <p>
        <select name="" id="" class="form-control">
          <option value="">111</option>
          <option value="">222</option>
          <option value="">333</option>
        </select>
      </p>
      <textarea name="" id="" cols="30" rows="10" class="form-control"></textarea>
      <input type="submit">
    </form>
  </div>
</div>
```

针对表单标签 加样式就用form-control

```
""
<input type="checkbox">222
<input type="radio">333
checkbox和radio我们一般不会给它加form-control，直接使用原生的即可
""
```

```
# 针对报错信息 可以加has-error (input的父标签加)
<p class="has-error">
  username:<input type="text" class="form-control">
</p>
```

按钮

```
<a href="https://www.mzitu.com/" class="btn btn-primary">点我</a>
<button class="btn btn-danger">按我</button>
<button class="btn btn-default">按我</button>
<button class="btn btn-success">按我</button>
<button class="btn btn-info">按我</button>
<button class="btn btn-warning">按我</button>

<button class="btn btn-warning btn-lg">按我</button>
<button class="btn btn-warning btn-sm">按我</button>
<button class="btn btn-warning btn-xs">按我</button>
<input type="submit" class="btn btn-primary btn-block"> | I
```

通过给按钮添加 `.btn-block` 类可以将其拉伸至父元素100%的宽度，而且按钮也变为了块级（block）元素。

图表

```
<h2 class="text-center">登陆页面 <span class="glyphicon glyphicon-user">
</span></h2>

<style>
  span {
    color: greenyellow;
  }
</style>

# 扩展
```

静态网页

页面上的数据是直接写死的 万年不变

动态网页

数据是实时获取的

eg:

1. 后端获取当前时间展示到html页面上
2. 数据是从数据库中获取的展示到html页面上

动态网页制作

```
import datetime
def get_time(env):
```

```
# 动态网页制作
import datetime
def get_time(env):
    current_time = datetime.datetime.now().strftime('%Y-%m-%d %X')
    # 如何将后端获取到的数据"传递"给html文件?
    with open(r'templates/03 mytime.html', 'r', encoding='utf-8') as f:
        data = f.read()
        # data就是一堆字符串
    data = data.replace('dwadasdsadsadasdas', current_time) # 在后端将
    # html页面处理好之后再返回给前端
    return data
```