

# Flask项目中使用bootstrap

原创 藕丝空间 藕丝空间 2018-03-03



flask的插件中有一个flask-bootstrap项目，但是用起来不怎么方便，如果大家知道的话，还是直接在项目中约会bootstrap。

## 烧瓶项目的目录结构

```
项目名称/  
  app/  
    static/ # 静态文件夹  
      css/ # 项目中引入的 css 样式表  
      js/ # 项目中引入的 js  
      fonts/ # 项目中需要的字体，一般是 bootstrap 所依赖的字体  
      images/ # 项目中的图片文件夹  
    templates/ # html 模板文件夹  
      common/ # 通用打模板文件  
      ...  
    views.py  
    __init__.py  
  env/ # python 虚拟环境  
  run.py # 启动文件
```

## bootstrap

### 下载bootstrap

将bootstrap的dist文件夹下的css, js, fonts三个文件夹复制到项目的静态文件夹下。在以后的学习过程中，你会发现，其实不需要完全复制的，你可以精简一些无用的文

件。

当然，由于bootstrap是基于jquery的，所以从网上下载jquery.js并复制至js文件夹下。具体下载的步骤，不再赘述。

## 制作基础模板

在template / common文件夹下建立一个base.html文件。内容如下：

```
<!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">
    <!-- The above 3 meta tags *must* come first in the head; any other he
    <meta name="description" content="藕丝空间">
    <meta name="author" content="东方鸮">
    <link rel="icon" href="{% url_for('static', filename='images/favicon.i

    <title>{% block title %}我的第一个前端页面{% endblock %}</title>
    <!-- 设置网页的标题，通过这样的设置，你可以利用 jinja2 的一些特性，将每个网页的标

    {% block css %} <!--css模块区域-->
    <!-- Bootstrap core CSS -->
    <link rel="stylesheet" href="{% url_for('static', filename='css/bootst
    <!-- 可选的Bootstrap主题文件（一般不用引入） -->
    <link rel="stylesheet" href="{% url_for('static', filename='css/bootst
    <!-- Custom styles for this template -->
    <link href="{% url_for('static', filename='css/offcanvas.css') %}" rel:
    {% endblock %}    </head>

  <body>
    {% include "common/header.html" %}<!--引入网页的头部模板-->
    <div class="container">
      <div class="row row-offcanvas row-offcanvas-right">
        {% block content %}<!--自定义模板区域-->

        {% endblock %}
        {% include "common/sidebar.html" %}<!--引入网页的侧边模板-->
      </div><!--/row-->
      {% include "common/footer.html" %}<!--引入网页的底部模板-->
    </div>

    {% block js %}<!--js模块区域-->
    <!-- Bootstrap core JavaScript
    ===== -->
    <!-- Placed at the end of the document so the pages load faster -->
    <script src="{% url_for('static', filename='js/jquery.js') %}"></scrip
    <script src="{% url_for('static', filename='js/bootstrap.min.js') %}">
```

```
<script src="{{ url_for('static', filename='js/offcanvas.js') }}"></script>
{% endblock %}    </body></html>
```

此模板是基于bootstrap的项目实例 - offcanvas制作的，你可以在bootstrap的源码中找到该实例。

此时，你的基础模板上是有错误存在的。因为还没有制作头部，底部和侧边模板。

## 制作头部模板

头部模板被放置在templates / common文件夹下，命名为header.html。

```
<nav class="navbar navbar-fixed-top navbar-inverse">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">Project name</a>
    </div>
    <div id="navbar" class="collapse navbar-collapse">
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#about">About</a></li>
        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button">Dropdown</a>
          <ul class="dropdown-menu">
            <li><a href="#">Action</a></li>
            <li><a href="#">Another action</a></li>
            <li><a href="#">Something else here</a></li>
            <li role="separator" class="divider"></li>
            <li><a href="#">Separated link</a></li>
            <li role="separator" class="divider"></li>
            <li><a href="#">One more separated link</a></li>
          </ul>
        </li>
      </ul>
    </div><!-- /.nav-collapse -->
  </div><!-- /.container --></nav><!-- /.navbar -->
```

## 制作底部模板

底部模板被放置在templates / common文件夹下，命名为footer.html。

```
<hr><footer>
  <p>&copy; 2016 Company, Inc.</p></footer>
```

## 制作侧边模板

侧边模板被放置在templates / common文件夹下，命名为sidebar.html。

```
<div class="col-xs-6 col-sm-3 sidebar-offcanvas" id="sidebar">
  <div class="list-group">
    <a href="#" class="list-group-item active">
      <button type="button" class="btn btn-block btn-danger" aria-label="Left Alig
        <span class="glyphicon glyphicon-align-left" aria-hidden="true"
      </button>
    </a>
    <a href="#" class="list-group-item">
      <button type="button" class="btn btn-block btn-info" aria-label="Left Alig
        <span class="glyphicon glyphicon-align-center" aria-hidden="true"
      </button>
    </a>
    <a href="#" class="list-group-item">
      <button type="button" class="btn btn-primary" aria-label="Left Alig
        <span class="glyphicon glyphicon-align-right" aria-hidden="true"
      </button>
    </a>
    <a href="#" class="list-group-item">
      <button type="button" class="btn btn-success" aria-label="Left Alig
        <span class="glyphicon glyphicon-align-justify" aria-hidden="true"
      </button>
    </a>
    <a href="#" class="list-group-item">
      <button type="button" class="btn btn-block btn-warning" aria-label="Left Alig
        <span class="glyphicon glyphicon-align-left" aria-hidden="true"
      </button>
    </a>
    <a href="#" class="list-group-item">Link</a>
    <a href="#" class="list-group-item">Link</a>
    <a href="#" class="list-group-item">Link</a>
    <a href="#" class="list-group-item">Link</a>
    <a href="#" class="list-group-item">Link</a>
  </div></div><!--/.sidebar-offcanvas-->
```

## 制作项目的主页

现在需要在项目的主页中约会基础模板，然后自定义自己的部分模板即可。**index.html**在**templat**s文件夹下，和头部，底部和侧边模板不在一个目录下，而在上级目录里。

```
{% extends "common/base.html" %}<!--继承基础模板-->{% block content %}<!--自定义模板-->
<p class="pull-right visible-xs">
  <button type="button" class="btn btn-primary btn-xs" data-toggle="offcanvas">
</p>
<div class="jumbotron">
  <h1>Hello, world!</h1>
  <p>This is an example to show the potential of an offcanvas layout pattern</p>
</div>
<div class="row">
  <div class="col-xs-6 col-lg-4">
    <h2>Heading</h2>
    <p>Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac consectetur
    <p><a class="btn btn-default" href="#" role="button">View details &raquo;
  </div><!--/.col-xs-6.col-lg-4-->
  <div class="col-xs-6 col-lg-4">
    <h2>Heading</h2>
    <p>Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac consectetur
    <p><a class="btn btn-default" href="#" role="button">View details &raquo;
  </div><!--/.col-xs-6.col-lg-4-->
  <div class="col-xs-6 col-lg-4">
    <h2>Heading</h2>
    <p>Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac consectetur
    <p><a class="btn btn-default" href="#" role="button">View details &raquo;
  </div><!--/.col-xs-6.col-lg-4-->
  <div class="col-xs-6 col-lg-4">
    <h2>Heading</h2>
    <p>Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac consectetur
    <p><a class="btn btn-default" href="#" role="button">View details &raquo;
  </div><!--/.col-xs-6.col-lg-4-->
  <div class="col-xs-6 col-lg-4">
    <h2>Heading</h2>
    <p>Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac consectetur
    <p><a class="btn btn-default" href="#" role="button">View details &raquo;
  </div><!--/.col-xs-6.col-lg-4-->
  <div class="col-xs-6 col-lg-4">
    <h2>Heading</h2>
    <p>Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac consectetur
    <p><a class="btn btn-default" href="#" role="button">View details &raquo;
  </div><!--/.col-xs-6.col-lg-4-->
</div><!--/row--></div><!--/.col-xs-12.col-sm-9-->{% endblock %}
```

到此，模板部分已经设置完毕。为了能够看到和测试我们的劳动成果，我们需要把flask项目写完整。

## 完善项目并展示

---

### \_\_init\_\_.py代码

```
from flask import Flask

app = Flask(__name__)from app import views
```

### views.py代码

```
from app import appfrom flask import render_template@app.route('/', methods=['GET'])
def index():
    return render_template('index.html')
```

### 运行

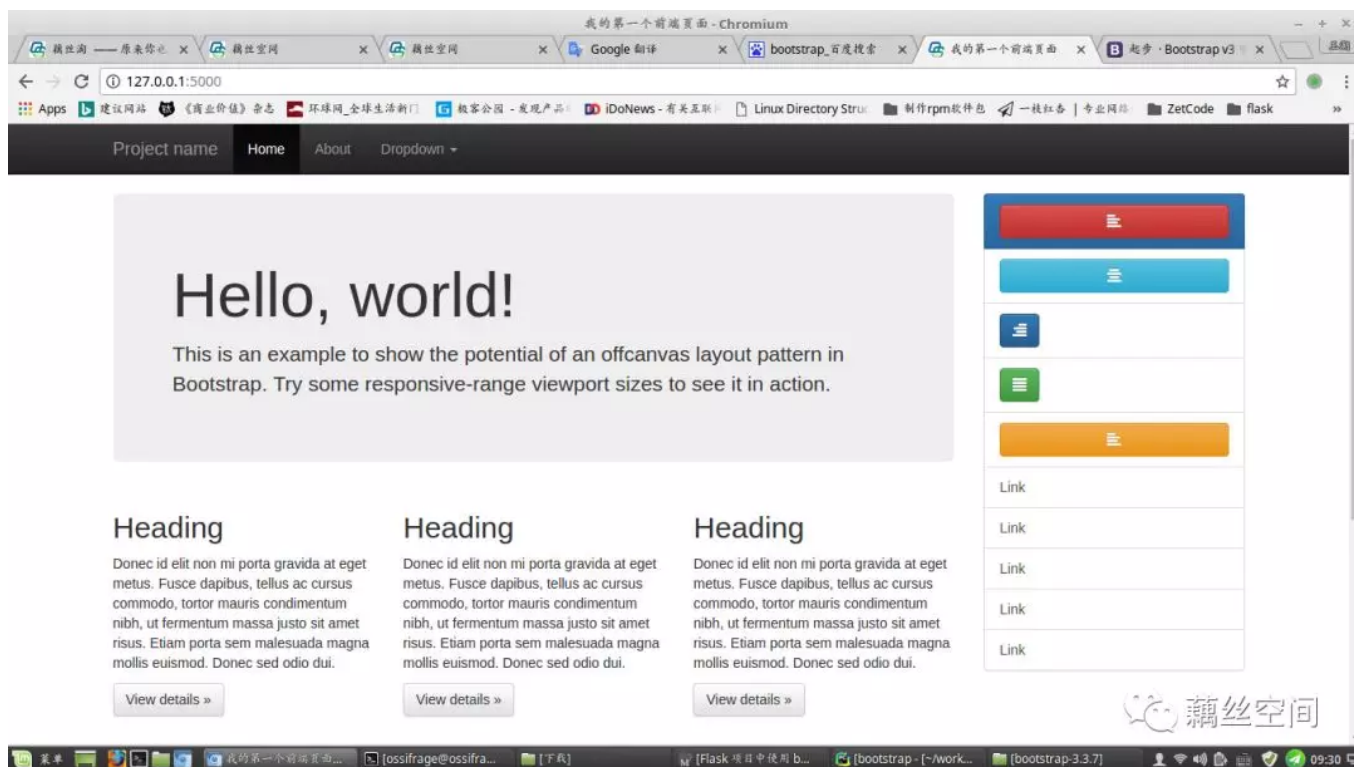
```
from app import app

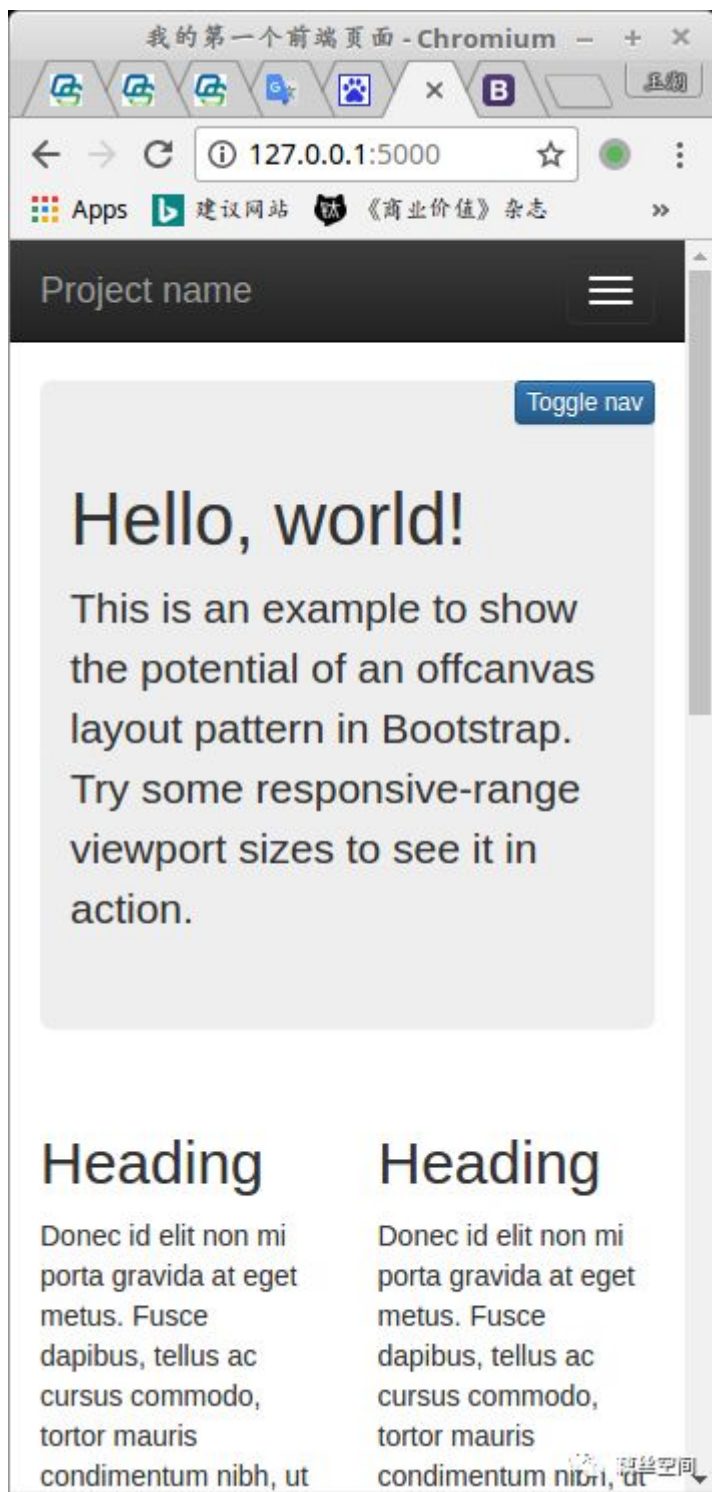
app.run(debug=True)
```

至此，项目已经能够展示了。

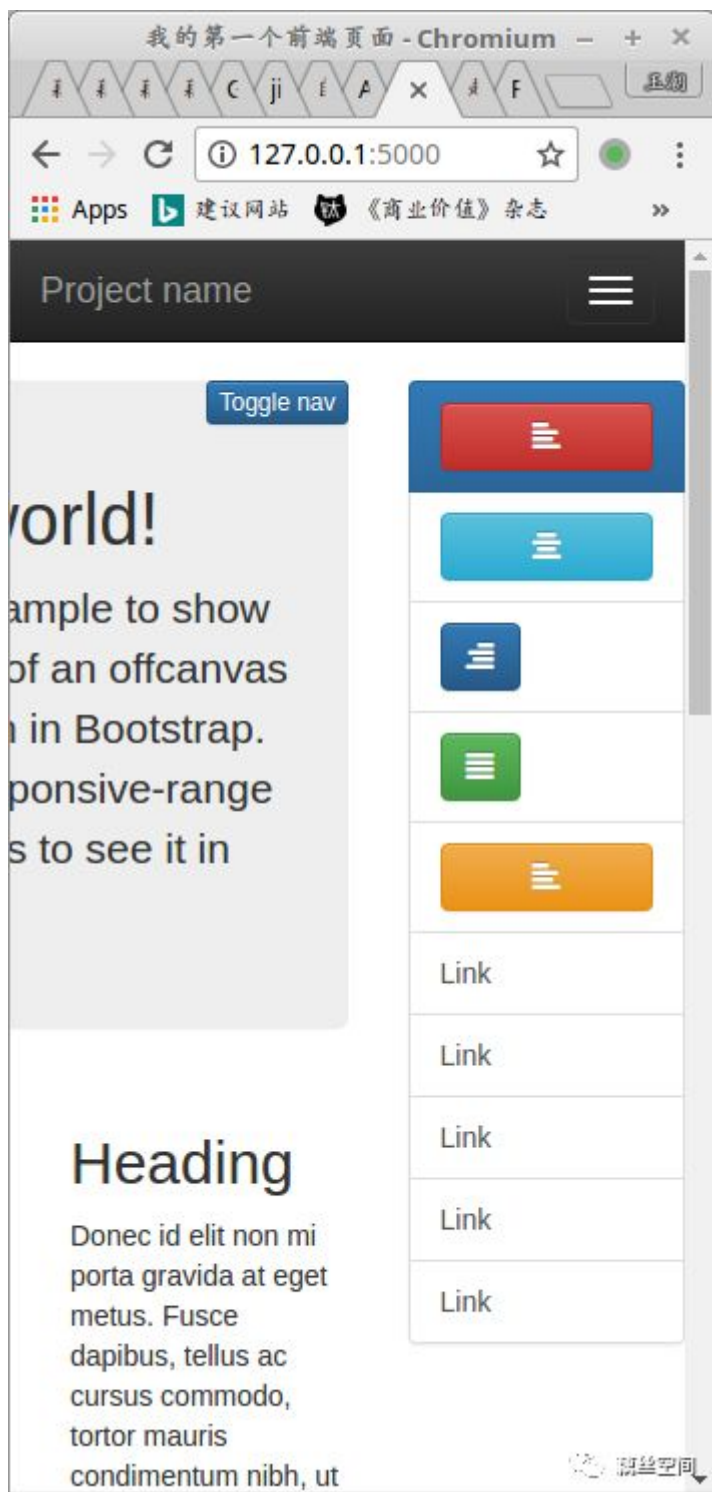
### 展示

---









## 进阶

### 在模板中使用jinja2自定义的过滤器

首先自定义一个函数

```
def jfloat(s):  
    return float(s)
```

本函数是为了在模板中将字符串，整型等其他类型转换为浮点型。

为了能够在jinja2模板中使用该函数。需要在项目的根\_\_init\_\_.py文件里的工厂函数create\_app () 中写入如下代码：

```
...def create_app(config_name):  
    ...  
    app.jinja_env.filters['jfloat'] = jfloat
```

这样，你就可以在模板中使用该过滤器啦。

## 参考

<http://docs.jinkan.org/docs/jinja2/api.html#writing-filters>

<http://www.tuicool.com/articles/7v6Ffiq>

阅读报纸