Flask

安装

```
$ pip install flask
```

编程快速入门 http://docs.jinkan.org/docs/flask/quickstart.html#quickstart

在项目根目录下构建：

1. webapp包目录，存放flask代码，包内有__init__.py文件
2. templates目录，存放模板文件
3. static目录，存放js、css等静态文件。其下建立js目录，放入jquery的js文件
4. app.py，入口文件

基本组成

```python
# webapp/__init__.py
from flask import Flask, jsonify

# 创建应用
app = Flask('web')

# 路由和视图函数
@app.route('/')
def index():
    return 'hello flask'

@app.route('/json', methods=['GET']) # 列表中指定多个方法
def getjson():
    d = {'a':1, 'b':[1, 2, 3]}
    return jsonify(d) # Mime是application/json

@app.route('/info') # 重要信息
def getinfo():
    import io
    sio = io.StringIO()
    print(*app.__dict__.items(), sep='<br>', file=sio, end='<hr>')
    print(app.name, app.template_folder, app.root_path,
          app.static_folder,   # 静态绝对路径
          app.static_url_path, # 静态路径url
          sep='<br>', file=sio)
    print(app.view_functions)
    print(app.url_map)
    return sio.getvalue()
```

应用：创建出来提供WEB服务的实例，也是wsgi的入口

视图函数：执行内部代码输出响应的内容

路由：通过route装饰器创建path到视图函数的映射关系

`jsonify(*args, **kwargs)`封装数据返回Json响应报文，但是args和kwargs传参只能选择一种，否则抛异常。

```
# app.py
from webapp import app

if __name__ == '__main__':
    app.run('0.0.0.0', 80, True)
```

启动应用程序app.py，访问试试。默认127.0.0.1:5000

调试时，把debug=True，会自动重新加载改变。

模板

Flask使用了Jinja2模板。

对于应用app来说其模板是，根目录下的templates，其下新建index.html

```
('template_folder', 'templates')
('root_path', 'E:\\ClassProjects\\test')
('_static_folder', 'static')
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>首页</title>
</head>
<body>
<h2>马哥教育欢迎你</h2>
<hr>
{% for x in userlist %}
{{x}}
{% endfor %}
</body>
</html>
```

修改index视图函数，使用模板渲染函数。

```
from flask import Flask, jsonify, render_template

# 创建应用
app = Flask('web')

# 路由和视图函数
@app.route('/')
def index():
    userlist = list(range(100, 110))
    return render_template('index.html', userlist=userlist)
```

在app.jinja_loader属性中，有下面的语句

```
if self.template_folder is not None:
    return FileSystemLoader(os.path.join(self.root_path,
                                         self.template_folder))
```

说明，不管是app，都是使用自己的root_path和模板路径拼接成模板路径。

可以通过app查看`app.jinja_loader.searchpath`模板搜索路径。

可以在jinja2.loaders.FileSystemLoader.get_source函数中下断点看搜索路径。

蓝图

到目前为止，最大的问题是，当路由的路径映射达到一定数量，这个代码堆积在一起非常不好。能否把代码进一步分出模块吗？这就是Flask的模块化，称为蓝图Blueprint。

新建一个模块webapp/books.py

```
from flask import Blueprint, jsonify, render_template

books = Blueprint('book', __name__, template_folder='book_templates',
url_prefix='/abcd') # /abcd 斜杠一定要有

@books.route('/')
def list():
    return jsonify({
        'count':3,
        'results':[
            (1, 'python'),
            (2, 'java'),
            (3, 'javascript')
        ]
    })
    # return render_template('a.html')

print('+' * 30)
print(books.root_path)
print(books.template_folder)
print(books.static_folder)
print(books.static_url_path)
print(books.jinja_loader.searchpath)
print('+' * 30)
```

蓝图创建好了，一定要记得使用前要注册

Blueprint构造参数

- name，蓝图名字，注册在app的蓝图字典中用的key
- import_name，用来计算蓝图模块所在的路径，一般写`__name__`
- root_path，指定蓝图模块所在的路径，如果为None，使用import_name计算得到
- template_folder
    - 模块化后，蓝图也可以有自己的模板路径，就不用和app的模板混在一起
    - 蓝图的模板路径为**books.root_path** 下的 **template_folder**

    - 模板是一批搜索路径，会依次搜索这些模板路径，找到第一个返回。首先搜索app的模板路径，然后依次搜索蓝图的模板路径。如果在找到本蓝图模板之前，有一个同名的模板，那么本蓝图模板文件就不能被找到了
- url_prefix，指定本蓝图模块的路径前缀。app.register_blueprint注册蓝图时，也可以对当前蓝图指定url_prefix，它将覆盖蓝图中的定义。

特别注意，输出的root_path路径，说明蓝图有自己一套路径。

不管是app还是蓝图，都是自己的root_path路径拼上模板路径，就是模板的绝对路径。

```
from flask import Flask, jsonify, render_template
from .books import books
app = Flask('web')

#app.register_blueprint(books) # 默认蓝图的所有url都挂到站点根路径上，所以最好提供url_prefix
app.register_blueprint(books, url_prefix='/abc')

@app.route('/')
def index():
    return render_template('index.html', userlist=[1, 2, 3, 'a', 'b'])

@app.route('/json', methods=['get'])
def get_json():
    return jsonify({'a':100, 'b':[1,2,3]})

@app.route('/info')
def get_info():
    import io
    sio = io.StringIO()
    print(app.__dict__.keys())
    print(app.name, app.root_path, app.static_folder, app.static_url_path,
          app.template_folder,
          file=sio, sep='<hr>')
    return sio.getvalue()

print('~' * 30)
print(app.blueprints)
for x in app.blueprints.values():
    print(x.name, x.root_path, x.template_folder)

print('=' * 30)
print(app.url_map)
print(app.root_path) # 物理的
print(app.template_folder)
print(app.static_folder)
print(app.static_url_path)
print(app.jinja_loader.searchpath)
print('=' * 30)
```

最后app.register_blueprint(books, url_prefix='/bbb')，url_prefix一定要以/开始，否则报错。最后路径以注册的url_prefix为准。

马哥教育
IT人的高薪职业学院