

该文件夹目录树为：

```
mongo.py 主程序
tips.docx 该文件
.idea
  misc.xml
  modules.xml
  workspace.xml
  寻宝游戏flaskdemo.iml
inspectionProfiles
  profiles_settings.xml
templates
  buy.html
  equipinfo.html
  homepage.html
  index.html
  look_for_money.html
  look_for_treasure.html
  marinfo.html
  myinfo.html
  sell.html
  sellinfo.html
  sign.html
  signup.html
  stoinfo.html
  take_off.html
  take_on.html
```

idea 生成文件

网页文件

具体执行流程参考以下操作指南。

## Flask app 操作指南

### 1, 如何启动?

[1] 在 mongo.py 文件中有如下语句，可直接运行该 py 文件

```
if __name__ == '__main__':
    app.run(host='127.0.0.1', port=5005)
```

[2] 在命令行中运行：

可以使用 `flask` 命令或者 `python` 的 `-m` 开关来运行这个应用。在 运行应用之前，需要在终端里导出 `FLASK_APP` 环境变量：

```
$ export FLASK_APP=hello.py
```

```
$ flask run
```

```
* Running on http://127.0.0.1:5000/
```

如果是在 Windows 下，那么导出环境变量的语法取决于使用的是哪种命令行解释器。在 Command Prompt 下：

```
C:\path\to\app>set FLASK_APP=hello.py
```

```
C:\path\to\app>flask run
```

### 2, 前端如何向后端传输数据?

在 html 中使用 `form` 标签，`form` 表单的作用是收集标签中的内容，然后这些内容将会

被送到服务端。

在 form 表单中设置一个 action 表示目标网址，method 表示提交方式，可以分为 get 和 post。

**GET** - 从指定的资源请求数据。提交的参数会显示在 URL 上。

**POST** - 向指定的资源提交要被处理的数据。通过表单提交不会显示在 URL 上。

下图为 index.html 中的表单数据，<input>为输入框，type="submit" 表示提交按钮

```
<form action="/login" method="get">
  用户名: <input type="text" name="username"><br>
  密码: <input type="password" name="password"><br>
  <input type="submit" value="提交">
</form>
```

点击之后就会将 username 和 password 传递给后端的 login 函数，接下来就需要在后端代码进行相应的处理。

### 3, 后端如何接受前端数据?

在 login()函数中使用 flask 中的 request.args.get('name')可以获得前端向后传递的数据，然后就可以进行相应的处理。

```
@app.route('/login')
def login():
    if request.method == 'GET':
        name = request.args.get('username')
        pwd = request.args.get('password')
        re=[name]
        return render_template('homepage.html',re=re)
```

### 4, 后端如何路由到前端页面?

将 html 文件保存在 templates 文件夹中，如上图所示，在返回中写入 render\_template('homepage.html',re=re)，就可以返回相应的网页。

```
@app.route('/homepage')
def homepage():
    re = ['xxx']
    return render_template('homepage.html',re=re)
```

### 5, 后端如何向前端传输数据?

在返回的语句中加上该页面需要的数据，就可向前端传递。

```
@app.route('/myinfo')
def myinfo():
    name='111'
    wealth=60
    row1=[['xxx1','等级一'],['xxx2','等级二']]
    row2 = [['xxx1', '等级一'], ['xxx2', '等级二']]
    return render_template('myinfo.html',name=name,money=wealth,row1=row1,row2=row2)
```

### 6, 前端如何接受后端数据?

以 myinfo.html 为例，在网页中使用双重大括号显示需要从后端接受的数据，用%可以写入循环语句，如下图所示。

```

<p>name:{{name}}
<p>money:{{wealth}}

<p>cloths:
{% for i in row1 %}
<p>
<form action="/equip_info" method="get">
    装备id<input type="submit" name="equipments" value="{{i[1]}}">{{i[0]}}
</form>
{% endfor %}
<p>storage:
{% for i in row2 %}
<p></p>
<form action="/sto_info" method="get">
    装备id<input type="submit" name="storage" value="{{i[1]}}">{{i[0]}}
</form>
{% endfor %}

```

## 7, 前端如何调用后端方法?

如下图所示, 只需要在 `action` 中填入本次按钮相要调用的方法, 就进行可以相应的调用。

```

<form action="/look_for_treasure" method="get">
    寻宝<input type="submit" value="进入">
</form>

```

```

@app.route('/look_for_treasure')
def look_for_treasure():
    return render_template('look_for_treasure.html', equip_name='手套', tag=0)

```

## 8, 前端 html 如何使用变量, 选择结构, 循环结构?

变量: 将变量名写在两个大括号内

```

<p>name:{{name}}
<p>money:{{wealth}}

```

选择结构: 和正常语法相似, 不过需要将语法行都写作 `{% ... %}` 中。

```

{% if tag==0%}
<p>恭喜你, {{name}}, 注册成功
<p>送你十个金币, 一只帽子, 一个充电器作为见面礼, 愿你玩得开心
{% endif %}

```

循环结构: 和选择结构语法相似。

```

{% for i in row1 %}
<p>
<form action="/equip_info" method="get">
    装备id<input type="submit" name="equipments" value="{{i[1]}}">{{i[0]}}
</form>
{% endfor %}

```

如有功能不清楚, 请参考:

Flask: <https://flask.net.cn/installation.html>