

## 获取 POST 请求传递的数据

```
username = request.form.get('age')
```

调用 `request.form.get()` 获取 POST 方法传递的数据，方法的参数是前端 `<form>` 表单中 `<input>` 标签的 `name` 属性的值。

## 获取 GET 请求传递的数据

```
username = request.args.get('age')
```

调用 `request.args.get()` 即可获取 GET 方法传递的数据。

## 渲染 HTML 模板文件

```
from flask import Flask, render_template

@app.route('/hello')

def hello():

    return render_template('hi.html')
```

渲染模板文件使用 `render_template` 函数，需先从 flask 中导入。

## 重定向

```
redirect('/目标路由')
```

调用 `redirect` 函数即可完成重定向，`redirect` 函数的参数就是要重定向到的地址。

## find\_one 查询文档

```
db.collection.find_one( { 'username':'xxx' } )
```

调用 `find_one` 方法即可在数据库的集合中查询文档，`find_one` 的参数就是查询条件。上例中的代码会在集合 `collection` 中查询 `username` 是 `xxx` 的文档。

## insert\_one 插入文档

```
db.collection.insert_one({'username': 'xxx', 'gender': 'male' })
```

调用 `insert_one` 方法即可在数据库的集合中插入文档，参数中的字典即是要插入到数据库中的数据。

## update\_one 更新文档

```
db.collection.update_one({'username': 'xxx'}, {'$set': {'gender': 'female'}})

db.collection.update_one({'username': 'xxx'}, {'$inc': {'age': 1}})
```

调用 `update_one` 方法即可在数据库的集合中更新文档，参数中的第 1 个参数是条件，表示要更新哪条文档，第 2 个参数表示如何更新文档。

## session 数据存取

```
# 存数据

session['name'] = 'xxx'

# 取数据

name = session['name']
```

## 返回 JSON 格式数据

```
jsonify({'state': 'success', 'data': 10})
```

jsonify 函数的作用有两个，其一是使参数中的字典转成 JSON 格式的数据，其二是修改响应头告知浏览器响应数据内容是 JSON 数据。