

计算机与信息工程学院

教 案

课程名称: Python 全栈开发

任课教师: 陈永松

教研室: 计算机应用技术

任课班级: 2019、2020 计算机应用技术

总学时: 64

学期: 2020~2021 年 2 学期

课程信息	
基本说明	授课类型：理论、实践各半 授课人数：153 课型：新授课 课程总学时/周学时：64/4
教材信息	教材名称： Python Flask Web 开发入门与项目实战 作者：钱游 出版社：机械工业出版社 教材特色：企业教材
教学目标	知识与技能目标： 掌握 Flask 开发环境的搭建,创建和运行 Flask 工程的方法,Jinja2 模版的使用,Flask 视图高级技术 ,Flask 数据交互 ,Flask 访问数据库, Memcached 缓存系统的使用。具备开发 Flask 应用程序的能力，具备开发前后端分离程序能力，具备 Flask 访问数据库能力，具备开发 Flask 可视化网站的应用能力，具备调试应用程序的能力： 课程思政目标：了解新一代软件技术所处的社会环境和政治环境，能够正确理解和评价新一代软件技术对环境、安全、文化以及社会可持续发展的影响。 劳动教育目标：在完成实训任务的同时完成实训场地的清洁卫生、实训工具的规范摆放；强调全身心参与 Python 全栈项目开发，手脑并用，亲历实际的劳动过程；培养务实求精的工匠精神和爱岗敬业的劳动态度
教学重点及难点	教学重点： Jinja2、Flask 数据交互、访问数据库 教学难点：访问数据库
教学策略	教学方法：强调学生技能，淡化理论，注重动手与企业实际需求相结合 学习方法：课前预习，认真听课，多写代码，注重操作

第 1 教学单元		共 3 讲
基本说明	教学单元名称: Flask 基础知识 授课类型: 理论、实践各半 教学单元总学时/周学时: 12/4	
教学目标	技能目标: 掌握 Flask 开发环境的搭建,创建和运行 Flask 工程的方法,Jinja2 模版的使用 劳动教育目标: 强调全身心参与 Flask 开发环境的搭建,手脑并用,亲历实际的劳动过程	
教学重点及难点	教学重点: Flask 传参、Jinja2 模版 教学难点: Jinja2 模版	
单元教学安排	1.Flask 入门 4 课时 2.Jinja2 模版初级 4 课时 3. Jinja2 模版高级 4 课时	

章节 (内容): 第 1、2 章 Flask 入门		第 1 讲:
基本说明	教学时间: 180 分钟 时间分配: 理论 90 分钟, 实践 90 分钟 课时: 4	
教学过程	第一部分 理论授课 (90 分钟) 第一阶段: 课程导入 (5 分钟) 课程介绍及要求 第二阶段: 讲授新课 Python Flask 前景及就业需求等 软件安装: 1. Anaconda (安装后会自动安装 python3, 同时自动安装一些常用的库) 2. Pycharm 新建 my01.py 文件, 代码如下: <pre> from flask import Flask app=Flask('main') @app.route('/') def Flask01(): return 'Hello World' app.run() </pre> 修改 my01.py 文件: <pre> from flask import Flask app=Flask('main') @app.route('/') def Flask01(): return 'Hello World' app.run(port=8880,debug=True,host='0.0.0.0') </pre> 修改 my01.py 文件: <pre> from flask import Flask app=Flask(__name__) @app.route('/') def Flask01(): return 'Hello World' if __name__ == '__main__': app.run(port=8881) else: app.run(port=8882) </pre> 添加 my02.py 文件: <pre> import my01 my01() </pre> 调试 my02 修改 my01.py <pre> from flask import Flask app=Flask(__name__) @app.route('/') def Flask01(): return 'Hello World' @app.route('/user/<name>') </pre>	

```

def Flask02(name):
    return f'接收参数为{name}'
app.run(debug=True)
修改 my01.py
from flask import Flask
app=Flask(__name__)
@app.route('/')
def Flask01():
    return 'Hello World'
@app.route('/user/<name>')
def Flask02(name):
    return f'接收参数为{name}'
@app.route('/sqrt/<int:num>')
def Flask03(num):
    return f'{num}的平方根是{num**0.5}'
app.run(debug=True)
修改 my01.py
from flask import Flask
app=Flask(__name__)
@app.route('/')
def Flask01():
    return 'Hello World'
@app.route('/user/<name>')
def Flask02(name):
    return f'接收参数为{name}'
@app.route('/sqrt/<int:num>')
def Flask03(num):
    return '%d 的平方根是%d'%(num,num**0.5)
app.run(debug=True)
修改 my01.py
from flask import Flask,url_for
app=Flask(__name__)
@app.route('/')
def Flask01():
    url1=url_for('Flask02',name='chen')
    return f'反转内容为: {url1}'
@app.route('/user/<name>')
def Flask02(name):
    return f'接收参数为{name}'
@app.route('/sqrt/<int:num>')
def Flask03(num):
    return '%d 的平方根是%d'%(num,num**0.5)
app.run(debug=True)
修改 my01.py
from flask import Flask,url_for,redirect
app=Flask(__name__)
@app.route('/')
def Flask01():
    url1=url_for('Flask02',name='chen')

```

```

return redirect(url1)
@app.route('/user/<name>')
def Flask02(name):
    return f'接收参数为{name}'
@app.route('/sqrt/<int:num>')
def Flask03(num):
    return '%d 的平方根是%d'%(num,num**0.5)
app.run(debug=True)

```

小节课程内容，注意事项等

二 实验内容

Flask 入门

三 实验准备

预习与本次实验内容相关知识

四 实验步骤

(一)、练习利用 Anaconda 编写 python 程序方法。

1.通过开始菜单打开 Anaconda。

说明：如果下图不出现，可将谷歌浏览器设置为默认浏览器，再按上图操作

2.按下图编写代码，然后运行。

3.打开浏览器，在地址栏中按下图输入，查看结果

(二)、练习利用 Pycharm 编写 python 程序方法。

1.通过开始菜单打开 Pycharm

2.启动 Pycharm 后，创建一个新项目（按下图输入 d:\Flask01 后点击右侧...）

按下图设置（如果 C:\ProgramData 找不到可复制）

3.滚动条往下拉，双击 python.exe

4.右击 Flask01，再按下图选择，并输入文件名，然后回车

5.在代码窗口中输入第（一）大题代码

6.选择菜单“run”“run”，再按下图选择

7.按第一题方法，在浏览器地址栏中输入地址，查看结果。

8.修改代码如下

9.再次运行程序，然后在浏览器地址栏中输入下图地址，查看结果

10.修改代码如下

11.再次运行程序，然后在浏览器地址栏中输入下图地址（思考为什么是 8881），查看结果

12.在项目中添加 my02.py 文件（方法如第 3 题），代码如下

13.运行 my02.py，然后在浏览器地址栏中输入下图地址（思考为什么是 8882），查看结果

14.修改 my01.py

15.运行 my01.py，然后在浏览器地址栏中输入下图地址（思考为什么分别显示 zhang 和 1.1），查看结果

16.修改 my01.py

17.运行 my01.py，然后在浏览器地址栏中输入下图地址，查看结果

18.修改 my01.py

19.运行 my01.py，然后在浏览器地址栏中输入下图地址，查看结果

20.修改 my01.py

21.运行 my01.py，然后在浏览器地址栏中输入下图地址，查看结果

基本说明	<p>教学时间：180 分钟 时间分配：理论 90 分钟，实践 90 分钟 课时：4</p>
教学过程	<p>第一部分 理论授课（90 分钟） 第一阶段：课程导入（5 分钟） 复习上一讲教学内容。 第二阶段：讲授新课 复习引入 模版引擎是一个将页面模板和要显示的数据结合起来生成 HTML 页面的工具。 新建 Flask02 项目，venv 文件夹下新建 templates 文件夹， templates 文件夹下新建 index.html</p> <pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>首页</title> </head> <body> <h1>首页文字</h1> </body> </html></pre> <p>templates 文件夹下新建 user.html</p> <pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>用户</title> </head> <body> <h1>用户中心</h1> </body> </html></pre> <p>venv 文件夹下新建 app01.py</p> <pre>from flask import Flask,render_template app=Flask(__name__) @app.route('/') def Flask01(): return render_template('index.html') @app.route('/user/<username>') def Flask02(username): return render_template('user.html')</pre> <p>app.run(debug=True) 调试 app01.py 修改 user.html</p> <pre><!DOCTYPE html> <html lang="en"> <head></pre>

```
<meta charset="UTF-8">
<title>用户</title>
</head>
<body>
  <h1>欢迎您: {{name}}</h1>
</body>
</html>
修改 app01.py
from flask import Flask,render_template
app=Flask(__name__)
@app.route('/')
def Flask01():
    return render_template('index.html')
@app.route('/user/<username>')
def Flask02(username):
    return render_template('user.html',name=username)
app.run(debug=True)
传递多个参数, 修改 app01.py
from flask import Flask,render_template
app=Flask(__name__)
@app.route('/')
def Flask01():
    title='学习模版'
    author='zhang'
    return render_template('index.html',title=title,author=author)
@app.route('/user/<username>')
def Flask02(username):
    return render_template('user.html',name=username)
app.run(debug=True)
传递多个参数, 修改 index.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>首页</title>
</head>
<body>
  <h1>首页文字</h1>
  <h2>标题: {{title}}</h2>
  <h2>作者: {{author}}</h2>
</body>
</html>
调试 app01.py
修改 app01.py, 将 title=title,author=author 改为**locals()
调试 app01.py, 结果相同
修改 app01.py
from flask import Flask,render_template
app=Flask(__name__)
@app.route('/<author>')
```

```
def Flask01(author):
    title='学习模版'
    return render_template('index.html',**locals())
@app.route('/user/<username>')
def Flask02(username):
    return render_template('user.html',name=username)
app.run(debug=True)
调试 app01.py
修改 app01.py
from flask import Flask,render_template
app=Flask(__name__)
@app.route('/<author>')
def Flask01(author):
    title='学习模版'
    return render_template('index.html',**locals())
@app.route('/user/<username>')
def Flask02(username):
    return render_template('user.html',name=username)
@app.route('/')
def Flask03():
    title='学习模版'
    return render_template('index.html',**locals())
app.run(debug=True)
修改 index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>首页</title>
</head>
<body>
    <h1>首页文字</h1>
    <h2>标题: {{title}}</h2>
    <h2>作者:
        {%if author%}
            {{author}}
        {%else%}
            null
        {%endif%}
    </h2>
</body>
</html>
修改 app01.py, 增加代码
import random
@app.route('/score')
def Flask04():
    score=random.randint(0,100)
    return render_template('rand.html',**locals())
添加 rand.html
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>成绩</title>
</head>
<body>
<h1>{{score}}分,
  {%if score>=90 %}
    优秀
  {%elif score>=60 %}
    合格
  {%else%}
    不合格
  {%endif%}
</h1>
</body>
</html>
添加 shop.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>商品</title>
</head>
<body>
  <table>
    <tr><th>商口名称</th><th>商品价格</th></tr>
    {%for g in goods%}
    <tr><td>{{g.name}}</td><td>{{g.price}}</td></tr>
    {%endfor%}
  </table>
</body>
</html>
修改 app01.py, 增加代码
@app.route('/goods')
def Flask05():
    goods=[{'name':'西装','price':138.00},{'name':'夹克衫','price':95.00}]
    return render_template('shop.html',**locals())
调试 app01.py
修改 index.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>首页</title>
</head>
<body>
  <h1>首页文字</h1>
```

```
<h2>标题: {{title}}</h2>
<h2>作者:
    {{author|default('null')}}
</h2>
</body>
</html>
调试 app1.py
修改 app01.py, 添加黄色代码
@app.route('/<author>')
def Flask01(author):
    title='学习模版'
    discount=author.endswith('vip') and 0.8
    return render_template('index.html',**locals())
修改 index.html, 添加代码
<h3>折扣: {{discount|default('无')}}</h3>
调试 app01.py
修改 index.html
<h3>折扣:
{{discount|default('无',True)}}</h3>
第二部分 上机实践 (90 分钟)
一 实验目的
熟悉 Jinja2 模板
二 实验内容
熟悉 Jinja2 模板
编写 Jinja2 模板基本程序
三 实验准备
预习与本次实验内容相关知识
四 实验步骤
(一)、练习利用 Pycharm 编写 python 程序方法。
1.通过开始菜单打开 Pycharm
2.启动 Pycharm 后, 创建一个新项目 (按下图输入 d:\Flask01 后点击右侧...)
按下图设置 (如果 C:\ProgramData 找不到可复制)
3.滚动条往下拉, 双击 python.exe
4.右击 Flask01, 再按下图选择, 并输入文件名, 然后回车
5.在代码窗口中输入代码
6.新建 templates 文件夹 (右击"Flask01"选择"New""Directory")
7.在 templates 文件夹内新建 Html file, 文件名为 index.html, 代码如下:
8..在 templates 文件夹内新建 Html file, 文件名为 user.html, 代码如下:
9.调试 app01.py, 浏览器结果如下
10.浏览地址栏添加/user/zhang, 结果如下
11.修改 user.html
12.修改 app01.py
13.调试 app01.py, 浏览地址栏添加/user/zhang, 结果如下
14.按下图修改 app01.py, 再次调试 app01.py,并且地址栏添加/user/wang,
查看结果, 理解**locals()的作用
15.修改 app01.py
16.修改 user.html
17.调试 app01.py, 结果如下
18.修改 app01.py
```

	<p>19.调试 app01.py</p> <p>20.修改 user.html</p> <p>21.调试 app01.py</p> <p>22.修改 user.html</p> <p>23.调试 app01.py</p> <p>24.修改 app01.py</p> <p>25.在 templates 文件夹内新建 Html file，文件名为 score.html，代码如下：</p> <p>26.调试 app01.py，可多次刷新（按 F5 键）查看结果变化</p> <p>27.修改 app01.py</p> <p>28.在 templates 文件夹内新建 Html file，文件名为 shop.html，代码如下：</p> <p>29.调试 app01.py</p>
<p>章节（内容）：第 3 章 Jinja2 模版（二）</p>	
<p>第 3 讲：</p>	
<p>基本说明</p>	<p>教学时间：180 分钟</p> <p>时间分配：理论 90 分钟，实践 90 分钟</p> <p>课时：4</p>
<p>教学过程</p>	<p>实验三 Jinja2 模版高级</p> <p>一 实验目的</p> <p>熟悉 Jinja2 模版上机环境</p> <p>Jinja2 模版高级</p> <p>二 实验内容</p> <p>熟悉 Jinja2 模版上机环境</p> <p>编写 Jinja2 模版高级程序</p> <p>三 实验准备</p> <p>预习与本次实验内容相关知识</p> <p>四 实验步骤</p> <p>（一）、练习利用 Pycharm 编写 Jinja2 模版程序方法。</p> <p>1.通过开始菜单打开 Pycharm</p> <p>2.启动 Pycharm 后，创建一个新项目（按下图设置后点击 create，说明：anaconda3\python.exe 不一定是下图文件夹，请根据实际安装的 anaconda3 位置选择）</p> <p>按下图设置（如果 C:\ProgramData 找不到可复制）</p> <p>滚动条往下拉，双击 python.exe</p> <p>3.右击 venv，再按下图选择，并输入文件名，然后回车</p> <p>4.编写 app02.py 代码</p> <p>5.在 venv 文件夹内新建文件夹(Directory)，名称为 templates。</p> <p>6.在 templates 内添加 Html 文件，名称为 shop.html，代码如下</p> <p>7.调试 app02.py，结果如下</p> <p>8.在 templates 内添加 Html 文件，名称为 form.html，代码如下</p> <p>9.修改 app02.py 代码</p> <p>10.调试 app02.py，结果如下</p> <p>11.在 templates 内添加 Html 文件，名称为 form2.html，代码如下</p> <p>12.修改 app02.py 代码</p> <p>13.调试 app02.py，结果如下</p> <p>14.在 templates 内添加 Html 文件，名称为 header.html，代码如下</p> <p>15.在 templates 内添加 Html 文件，名称为 footer.html，代码如下</p>

	<p>16.在 templates 内添加 Html 文件，名称为 pyindex.html，代码如下</p> <p>17.修改 app02.py 代码</p> <p>18.调试 app02.py，结果如下</p> <p>19.在 venv 文件夹内新建文件夹(Directory)，名称为 static，在 static 文件夹内新建文件夹 css，将 bootstrap.min.css 复制到 css 文件夹内。</p> <p>20.修改 pyindex.html</p> <p>21.调试 app02.py，结果如下</p> <p>22.修改 pyindex.html</p> <p>23.调试 app02.py，结果如下</p> <p>24.修改 header.html</p> <p>25.修改 footer.html</p> <p>26.调试 app02.py，结果如下</p> <p>27.修改 pyindex.html</p> <p>28.调试 app02.py，结果如下</p> <p>29.在 templates 内添加 Html 文件，名称为 model.html，代码如下</p> <p>30.在 templates 内添加 Html 文件，名称为 flask.html，代码如下(自动生成的代码全删除)</p> <p>31.在 templates 内添加 Html 文件，名称为 other.html，代码如下(自动生成的代码全删除)</p> <p>32.修改 app02.py 代码</p> <p>33.调试 app02.py，结果如下</p>	
<p>第 2 教学单元</p>		<p>共 3 讲</p>
<p>基本说明</p>	<p>教学单元名称：Flask 数据交互</p> <p>授课类型：理论、实践各半</p> <p>教学单元总学时/周学时：12/4</p>	
<p>教学目标</p>	<p>技能目标：掌握处理表单、上传操作、Cookie、Session 的使用</p> <p>劳动教育目标：强调全身心参与处理、上传操作，手脑并用，亲历实际的劳动过程</p>	
<p>教学重点及难点</p>	<p>教学重点：上传操作、Session 的使用</p> <p>教学难点：Cookie</p>	

单元教学 安排	<ol style="list-style-type: none">1. 处理表单、上传操作 4 课时2.浏览文件夹的文件、Cookie 4 课时3.Session 的使用 4 课时
------------	---

章节（内容）： 第 5 章 Flask 数据处理		第 4 讲：
基本说明	教学时间：180 分钟 时间分配：理论 90 分钟，实践 90 分钟 课时：4	
教学过程	第一部分 理论授课（90 分钟） 第一阶段：课程导入（5 分钟） 复习上节课内容 第二阶段：讲授新课 添加 app02.py <pre> from flask import Flask,render_template,request app=Flask(__name__) @app.route('/') def Flask01(): return render_template('index.html') @app.route('/login',methods=['GET','POST']) def Flask02(): if request.method=='GET': return 'GET 请求' else: return 'POST 请求' app.run() </pre> 调试程序 新建 static 和 templates 文件夹，将 css 文件夹复制到 static 文件夹 templates 文件夹添加 index.html <pre> <link rel="stylesheet" href="../static/css/bootstrap.min.css"> <div class="container border mt-1" style="width: 380px"> <form action="/login" class="text-center"> <h2>登录</h2> <input placeholder="请输入用户名" class="form-control my-2"> <input type="password" placeholder="请输入密码" class="form-control my-2"> <button class="btn btn-primary mb-2 form-control">登录 </button> </form> </div> </pre> 调试程序 点击登录按钮 修改 index.html <pre> <form action="/login" class="text-center" method="post"> </pre> 调试程序，访问 http://127.0.0.1，通过表单点击登录按钮 在 static 文件夹内新建文件夹 upload 修改 app02.py	

```

from werkzeug.utils import secure_filename
@app.route('/upload',methods=['POST','GET'])
def Flask03():
    if request.method=='GET':
        return render_template('upload.html')
    else:
        f=request.files['file']
        filename=secure_filename(f.filename)
        f.save(f'static/upload/{filename}')
        return '上传文件成功'

```

添加 upload.html, body 内代码如下:

```

<form action="/upload" method="post" enctype="multipart/form-data">
    <input type="file" name="file">
    <button>上传</button>
</form>

```

调试程序, 地址栏加/upload 查看结果
问题?

使用 secure_filename 如果全中文名, 则上传的文件名变成 .扩展名。

可利用 pypinyin 库, 将中文改为拼音

按 alt+F12, 输入命令(说明: 第 1 条语句表示将 pip 下载地址改为清华)

```
pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
```

```
pip install pypinyin
```

然后修改 app02.py

```

from pypinyin import lazy_pinyin
filename=""'.join(lazy_pinyin(f.filename))

```

调试程序, 地址栏加/upload 查看结果

问题? 上传文件时, 如果不选文件直接单击上传按钮出错

修改 app02.py

```

def Flask03():
    if request.method=='GET':
        return render_template('upload.html')
    else:
        f = request.files['file']
        if f:
            filename=""'.join(lazy_pinyin(f.filename))
            f.save(f'static/upload/{filename}')
            return '上传文件成功'
        else:
            return '请上传文件'

```

调试程序, 地址栏加/upload 查看结果

问题? 如果两次上传同文件名文件, 则第一次传的被覆盖

修改 app03.py

```
import time
```

```
if f:
```

```
filename=""'.join(lazy_pinyin(f.filename))
arr=filename.rsplit('.',1)
filename=f'{arr[0]}{int(time.time())}-{arr[1]}'
f.save(f'static/upload/{filename}')
return '上传文件成功'
```

```
else:
    return '请上传文件'
```

调试程序，地址栏加/upload 查看结果

问题：限制只能上传图片文件

修改 app03.py

```
f = request.files['file']
if f:
    if f.content_type.startswith("image"):
        filename=""'.join(lazy_pinyin(f.filename))
        arr=filename.rsplit('.',1)
        filename=f'{arr[0]}{int(time.time())}-{arr[1]}'
        f.save(f'static/upload/{filename}')
        return '上传成功'
    else:
        return '只能上传图片文件'
```

```
else:
    return '请上传文件'
```

调试程序，地址栏加/upload 查看结果

问题：判断上传文件夹是否存在，如果不存在先创建文件夹再上传图片

修改 app03.py

```
import os
uppath='static/upload'
if not os.path.exists(uppath):
    os.makedirs(uppath)
f.save(f'{uppath}/{filename}')
```

调试程序，地址栏加/upload 查看结果

第二部分 上机实践（90 分钟）

实验四 表单与文件上传

一 实验目的

熟悉 Jinja2 模版上机环境

表单与上传文件

二 实验内容

熟悉 Jinja2 模版上机环境

编写表单与上传文件程序

三 实验准备

预习与本次实验内容相关知识

四 实验步骤

(一)、练习利用 Pycharm 编写 Jinja2 模版程序方法。

1.通过开始菜单打开 Pycharm

2.启动 Pycharm 后，创建一个新项目（按下图设置后点击 create，说明：anaconda3\python.exe 不一定是下图文件夹，请根据实际安装的 anaconda3 位置选择）

按下图设置（如果 C:\ProgramData 找不到可复制）

滚动条往下拉，双击 python.exe

3.右击 venv，再按下图选择，并输入文件名，然后回车

4.编写 app02.py 代码

5.在调试 app02.py，结果如下

6.在 venv 文件夹内新建文件夹(Directory)，名称为 static，在 static 文件夹内新建文件夹 css，将 bootstrap.min.css 复制到 css 文件夹内。

7.在 venv 文件夹内新建文件夹(Directory)，名称为 templates。

8.在 templates 内添加 Html 文件，名称为 index.html，代码如下

9.在调试 app02.py，结果如下

10.在上图中输入用户名和密码，点击登录，结果如下：

11.修改 index.html 代码，将 method="post"改为 method="get"，重做第 9、10 题，结果如下

12.在 templates 内添加 Html 文件，名称为 upload.html，代码如下

13.在 pythram 中，按 ALT 键不放再按 F12，然后复制下面两条命令并按回车。

```
pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
```

```
pip install pypinyin
```

14.修改 app02.py 代码

15.调试 app02.py

16.在上图中选择 1 个文件上传，结果如下(原因是 static 文件夹下没有 upload 文件夹)：

17.修改 app02.py

18.调试 app02.py，重做第 15、16 题，结果如下：

19.上传一个中文文件名的文件，然后查看上传之后的文件名

20.调试 app02.py，重做第 15 题，不选择任何文件，直接点上传，结果如下：

21.修改 app02.py

22.重做第 15 题，不选择任何文件，直接点上传，结果如下：

23.修改 app02.py

24.修改 app02.py，重做第 15、16 题，然后查看上传之后的文件名变化。

25.修改 app02.py

26.调试 app02.py，分别上传图片，和非图片文件，查看结果

章节（内容）： 第 5 章 Flask 处理表单		第 5 讲：
基本说明	教学时间：180 分钟 时间分配：理论 90 分钟，实践 90 分钟 课时：4	
教学过程	第一部分 理论授课（90 分钟） 第一阶段：课程导入（5 分钟） 复习 Flask 基本操作 第二阶段：讲授新课 修改 app03.py <pre> from flask import Flask,render_template,request,send_from_directory @app.route('/image/<filename>') def Flask04(filename): dirpath='static/upload' return send_from_directory(dirpath,filename,as_attachment=True) </pre> 调试 app03.py 修改 app03.py, 删除,as_attachment=True, 调试 app03 修改 app03.py <pre> return '上传成功' </pre> 改为 <pre> return redirect(url_for("Flask04",filename=filename)) </pre> 调试 app03 查看结果 修改 upload.html <pre> <style> .div1{height: 180px;width: 380px;border:1px solid #8A8989;margin:0 auto} .input{display: block;width: 250px;height: 30px;margin: 10px auto} .button{background: #2066c5;color:white;height: 30px;border-radius:4px} </style> </head> <body> <div class="div1"> <form action="/upload" method="post" enctype="multipart/form-data"> <input type="file" name="file" class="input"> <button class="input button">上传</button> </form> </div> </pre> 浏览文件夹 修改 app03.py <pre> def Flask05(): uppath = 'static/upload' files = os.listdir(uppath) </pre>	

```
return render_template('images.html',**locals())
```

添加 images.html

```
<link rel="stylesheet" href="/static/css/style.css">
<script>
    function sel(e) {
        e.classList.toggle("selected");
    }
</script>
<ul id="list">
    {%for f in files%}
    <li onclick="sel(this)">
        
    </li>
    {%endfor%}
</ul>
```

调试 app03.py

第二部分：上机实践（90 分钟）

实验五 浏览文件夹

一 实验目的

掌握浏览文件夹操作

二 实验内容

掌握浏览文件夹操作

三 实验准备

复习与本次实验内容相关知识

四 实验步骤

一、MongoDB 操作

1.启动 NoSQL Manager for MongoDB，按下图操作：

2.将"nosql.txt"的代码复制到代码窗口中，按 F5 执行。

3.右击"Databases"选"Refresh"(刷新)，可以看到 edut 和 project 数据库已建立。

二、在 project 数据库中进行以下操作。

(一) 建立"project.shell"(如下图)

(二) 在"project.shell"按下面各题要求输入代码，完成相应操作。

- 1.在 employee 集合查询基本工资(basepay)小于等于津贴(bonus)的文档
- 2.在 employee 集合查询津贴(bonus)等于保险(insurance)的文档
- 3.在 Employee 集合中查询文档个数。
- 4.在 Employee 集合中查询津贴(bonus)等于保险(insurance)的文档个数。
- 5.在 Employee 集合中查询前 3 个文档。
- 6.在 employee 集合中查询第 5 个文档。
- 7.在 employee 集合中查询在 employee 集合查询第 5 页数据，每页 2 个文档。
- 8.在 Employee 集合中按 bonus 从小到大顺序显示文档。
- 9.在 Employee 集合中查询所有文档的 eid,ename,bonus,insurance 元素值。
- 10.在 Employee 集合中查询所有文档的 eid,ename,bonus,insurance 元素值，

	<p>并且不显示_id</p> <p>11.在 Employee 集合中查询第一个基本工资(basepay)小于 3200 的文档</p> <p>12.在 employee 集合中将 ename 为刘志文的 basepay 改为 2018, insurance 改为 2019。</p> <p>13.在 employee 集合中将所有文档的 basepay 在原来基础上增加 20。</p> <p>14.在 employee 集合中将 ename 包含“英”的 basepay 全部改为 2018。</p> <p>15.在 employee 集合中将 ename 为刘志文的文档删除。</p> <p>三、在 educ 数据库中进行以下操作。</p> <p>(一) 建立“educ.shell”</p> <p>(二) 在“educ.shell”按下面各题要求输入代码，完成相应操作。</p> <p>1.在 student 集合同时插入以下 2 个文档（要求 birth 为日期型数据。）。</p> <pre>sid sname sex birth 101103 林中杰 男 1998-06-18 101104 冯慧妍 女 1999-04-05</pre> <p>2.在 student 集合中查询 1999-1-1 日之后出生(birth 大于 1999-1-1)的文档</p> <p>四、Mongojs 操作 MongoDB</p> <p>1.在 D 盘创建文件夹 dbOp,启动 vscode，打开创建的文件夹。</p> <p>2.选择菜单“查看 输出”，然后单击“终端”，在终端中输入 npm config set registry https://registry.npm.taobao.org，回车，输入 npm i mongojs，回车。</p> <p>3.在 vscode 的 dbOp 文件夹中创建一个 db.js 文件，代码如下：</p> <p>4.db.js 添加代码</p> <p>5.保存 db.js，调试程序查看结果。</p> <p>6. 在 NoSQL Manager for MongoDB 的 “project.shell” 中通过 db.employee.find()查看第 4 题运行结果是否正确。</p> <p>7.注释第 4 题代码，添加代码，功能是在 employee 集合中插入以下 2 个文档（以下各题代码类似第 7 章课件 16 至 17 页），然后保存调试程序查看结果。</p> <pre>eid ename sex basepay 1012 张传海 男 1950 1013 李爱英 女 2000</pre> <p>8.注释第 7 题代码，添加代码，功能是查询 employee 集合所有文档。然后保存调试程序查看结果</p> <p>9.注释第 8 题代码，添加代码，功能是查询 employee 集合 ename 包含英的所有文档。然后保存调试程序查看结果</p> <p>10.注释第 9 题代码，添加代码，功能是定义变量 var name='^杨'，然后查询 employee 集合中 ename 与变量 name 对应正则表达式相符合的所有文档。然后保存调试程序查看结果</p>
<p>章节（内容）： 第 5 章 Flask 数据处理</p>	<p>第 6 讲：</p>
<p>基本说明</p>	<p>教学时间：180 分钟</p> <p>时间分配：理论 90 分钟，实践 90 分钟</p> <p>课时：4</p>

<p>教学过程</p>	<p>第一部分 理论授课 (90 分钟)</p> <p>第一阶段: 课程导入 (5 分钟)</p> <p>复习 MongoDB 基本操作</p> <p>第二阶段: 讲授新课</p> <p>7.10 使用 monk 操作 MongoDB</p> <p>1. npm i monk</p> <p>2.var monk = require('monk');</p> <p>var db = monk('localhost:27017/educ')</p> <p>3.插入数据</p> <p>4.查询 5.更新 6.删除</p> <p>第二部分: 上机实践 (90 分钟)</p> <p>实验五 monk、mongojs 操作 MongoDB</p> <p>一 实验目的</p> <p>掌握 monk、mongojs 操作 MongoDB 的方法</p> <p>二 实验内容</p> <p>MongoDB 基本操作</p> <p>monk、mongojs 操作 MongoDB</p> <p>三 实验准备</p> <p>复习与本次实验内容相关知识</p> <p>四 实验步骤</p> <p>一、MongoDB 操作</p> <p>1.启动 NoSQL Manager for MongoDB</p> <p>2.将"nosql.txt"的代码复制到代码窗口中, 按 F5 执行。</p> <p>3.右击"Databases"选"Refresh"(刷新), 可以看到 edut 和 project 数据库已建立。</p> <p>二、monk 操作 MongoDB</p> <p>1.在 D 盘创建文件夹 monkLx,启动 vscode, 打开创建的文件夹。</p> <p>2.选择菜单“查看 输出”, 然后单击“终端”, 在终端中输入 npm config set registry https://registry.npm.taobao.org, 回车, 在终端中输入 npm i monk, 回车。</p> <p>3.在 monkLx 文件夹中创建一个 db.js 文件, 代码如下:</p> <p>4.db.js 添加代码</p> <p>5.保存 db.js, 调试程序查看结果。</p> <p>6.在 NoSQL Manager for MongoDB 的“educ.shell”中通过 db.student.find()查看第 4 题运行结果是否正确。</p> <p>7.注释第 4 题代码, 添加代码, 功能是查询 student 集合所有文档。然后保存调试程序查看结果。</p> <p>8.注释第 7 题代码, 添加代码, 功能是查询 student 集合指定_id 的一个文档 (id 号在 mongoDB 中通过 db.student.find()查看)。然后保存调试程序查看结果。</p> <p>9.注释第 8 题代码, 添加代码, 功能是查询 student 集合 sname 包含志的所有文档。然后保存调试程序查看结果。</p> <p>10.注释第 9 题代码, 添加代码, 功能是查询 student 集合 birth 小于等于 1998-6-30 的所有文档。然后保存调试程序查看结果。(参考课件 23 页)</p>
-------------	---

11.注释第 10 题代码, 添加代码, 功能是查询 student 集合查询第 5 页的所有文档 (每页 2 个文档)。然后保存调试程序查看结果。(参考课件 23 页)

12.注释第 11 题代码, 添加代码, 功能是将 sname 为唐海贤的文档的出生日期改为 1998-4-15。

13.注释第 12 题代码, 添加代码, 功能是将 sname 中以“王”开头的出生日期全都改为 1999-12-10。

14.注释第 13 题代码, 添加代码, 功能是将 sname 中以“李”开头的文档删除。

15.注释第 14 题代码, 添加代码, 功能是将 sname 中第一个以“王”开头的文档删除。(调试后在 mongoDB 中通过 db.student.find()查看第一个姓王的文档是否删除)

16.选择菜单“查看 | 输出”, 然后单击“终端”, 在终端中输入 npm i express, 回车。

17.在 vscode 的 monkLx 文件夹中创建一个 browse.js 文件, 运行时在浏览器显示 student 集合指定_id 的一个文档。(注意课件是用 mongojs 完成操作, 请改用 monk 完成, 数据库为 educ, 集合名称为 student)。

18.调试 browse.js, 在浏览器中输入 http://127.0.0.1:3000/id 号 (id 号在 mongoDB 中通过 db.student.find()查看), 查看结果。

三、mongojs 操作 MongoDB

1.在 D 盘创建文件夹 mongojsLx,启动 vscode, 打开创建的文件夹。

2.选择菜单“查看 | 输出”, 然后单击“终端”, 在终端中输入 npm i mongojs, 回车。

3.在 mongojsLx 文件夹中创建一个 db.js 文件, 代码如下:

4.db.js 添加代码, 功能是查询 employee 集合查询第 5 页的所有文档 (每页 2 个文档)。然后保存调试程序查看结果。(参考课件 18 页)

5.注释第 4 题代码, 添加代码, 功能是在 employee 集合查询指定_id 的 (id 号在 mongoDB 中通过 db.employee.find()查看) 一个文档。(参考课件 18 页)

6.注释第 5 题代码, 添加代码, 功能是在 employee 集合中将 ename 为李小珊的文档的 basepay 改 2345、insurance 改为 942。(参考课件 19 页)

7.注释第 6 题代码, 添加代码, 功能是在 employee 集合中将所有文档的 insurance 在原来基础上加 10。(参考课件 19 页)

8.注释第 7 题代码, 添加代码, 功能是将 employee 集合中 ename 以“芬”结束的文档删除。(参考课件 20 页)

9.注释第 8 题代码, 添加代码, 功能是将 employee 集合中第一个 ename 以“李”开头的文档删除。(参考课件 20 页)

10.选择菜单“查看 | 输出”, 然后单击“终端”, 在终端中输入 npm i express, 回车。

11.在 vscode 的 mongojsLx 文件夹中创建一个 page.js 文件, 运行时在浏览器显示 employee 集合第 n 页数据 (每页 m 个文档)。(参考课件 26 页, 注意课件是用 monk 完成操作, 请改用 mongojs 完成, 数据库为 project, 集合名称为 employee)。

12.调试 page.js, 在浏览器中输入 http://127.0.0.1:3000?m=2&n=3, 查看结果。