

Python程序设计——第五次实验报告

2311095 宋卓伦 计算机科学与技术 2024.10.20

一、实验目的

利用序列、集合、字典和字符串的相关函数和方法编写程序，熟悉Python内置数据类型的使用方法和应用场景，掌握基于正则表达式进行文本关键信息提取的方法。

二、实验描述

实现一个简易的用户注册程序，具体要求如下：

- 使用字典存储用户信息，其中键为用户名，值为另一个字典，包含用户的电子邮件和密码；
- 使用正则表达式验证电子邮件格式是否正确；
- 使用正则表达式验证密码复杂度是否符合要求：密码必须至少包含8个字符，且至少包含一个大写字母、一个小写字母、一个数字和一个特殊字符 (@\$!%*?&)；
- 验证输入的用户名是否已经存在。如果已存在，则提示用户重新输入。

三、实验代码

首先导入re的库，实现正则运算：

```
1 import re
```

构建用户存储信息，并构建正则表达式：

```
1 # 用户信息存储字典
2 users = {}
3
4 # 电子邮件正则表达式
5 email_pattern = re.compile(r'^[a-zA-Z0-9_+]+@[a-zA-Z0-9]+\.[a-zA-Z0-9-]+\.$')
6
7 # 密码复杂度正则表达式
8 password_pattern = re.compile(r'^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$')
```

函数表达式：

```
1 def is_valid_email(email):
2     """验证电子邮件格式是否正确"""
3     return email_pattern.match(email) is not None
4
5
6 def is_valid_password(password):
```

```

7         """验证密码复杂度是否符合要求"""
8         return password_pattern.match(password) is not None
9
10
11 def register_user():
12     """用户注册函数"""
13     username = input("请输入用户名: ")
14     if username in users:
15         print("用户名已存在, 请重新输入。")
16         return
17
18     email = input("请输入电子邮件: ")
19     if not is_valid_email(email):
20         print("电子邮件格式不正确, 请重新输入。")
21         return
22
23     password = input("请输入密码: ")
24     if not is_valid_password(password):
25         print("密码复杂度不符合要求, 请确保密码至少8个字符, 且包含大写字母、小写字母、数字和特殊字符。")
26         return
27
28     # 存储用户信息
29     users[username] = {'email': email, 'password': password}
30     print("用户注册成功!")
31
32

```

主函数体:

```

1 # 主程序
2 if __name__ == "__main__":
3     while True:
4         register_user()
5         if input("是否继续注册新用户? (y/n): ") != 'y':
6             break
7

```

请输入用户名: *ads*

请输入电子邮件: *dasf21@asd.cd*

请输入密码: *adscSA21?*

用户注册成功!

{'ads': {'email': 'dasf21@asd.cd', 'password': 'adscSA21?'}}

是否继续注册新用户? (y/n): *y*

请输入用户名: *adax*

请输入电子邮件: *eqwv312@sad.cd*

请输入密码: *adsqwADS12?*

用户注册成功!

{'ads': {'email': 'dasf21@asd.cd', 'password': 'adscSA21?'}, 'adax': {'email': 'eqwv312@sad.cd', 'password': 'adsqwADS12?'}}

是否继续注册新用户? (y/n): *n*

进程已结束, 退出代码为 0

|

四、实验问题

1、正则表达式的描述

```
1 email_pattern = re.compile(r'^[a-zA-Z0-9_+-.]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+.$')
```

其中re的compile函数参数为 (pattern, flags)

2、在代码设计方面，报错信息不应该一致，比如正则表达式没有通过和用户名已经出现这两种情况不应该对应一个报错语句，否则出现问题的时候不容易发现是哪里出了问题。使用不同的报错语句能够方便调试。

3、电子邮件的形式较为多变，目前无法找到一个很好的正则表达式进行处理，（如果对于某几种电子邮件限制的话可以实现这个问题，例如126和163，QQ邮箱，nankai.edu.cn，gmail等等）

五、实验改进

1、在注册系统之外，设置一个登录系统：

```
1 def user_login():
2
3     """用户登录系统"""
4     un = input("请输入用户名: ")
5
6     if not un in user:
7         print("用户名不存在，请重新输入。")
8         return
9
10    pw = input("请输入密码: ")
11
12    if pw != user[un]["password"]:
13        print("密码不正确，请重新输入。")
14    else:
15        print("欢迎回来" + un)
16
17    return
18
```

2、设置密码找回（修改）系统：

```
1 """二者区别：找回是未知原始密码，修改是已知原始密码"""
2 """但是由于找回密码对安全性来说要求高，现有知识目前还无法解决"""
3 """需要通过邮箱传送验证码，口令正确方可修改"""
4
5 def userpw_change():
6
7     un = input("请输入用户名: ")
8
9     if un not in user:
10        print("用户名不存在，请重新输入。")
11        return
12
```

```

13     pw = input("请输入密码: ")
14
15     if pw != user[un]["password"]:
16         print("密码错误, 请重新输入。")
17         return
18
19     new_pw = input("请输入新密码: ")
20
21     if not password_pattern.match(new_pw):
22         print("密码复杂度不符合要求, 请确保密码至少8个字符, 且包含大写字母、小写字母、数字和特殊字符。")
23         return
24
25     new_pw0 = input("请再次输入新的密码: ")
26
27     if new_pw == new_pw0:
28         user[un]["password"] = new_pw
29     else:
30         print("新密码错误, 请重新输入。")
31
32     return
33

```

主程序修改如下:

```

1  if __name__ == "__main__":
2      while True:
3          opt = int(input("请输入选择: "))
4          if opt == 1:
5              user_register()
6          if opt == 2:
7              userpw_change()
8          if opt == 3:
9              user_login()
10         if input("是否继续? ") != 'y':
11             break

```

这样的话可以实现选择功能: 注册、改密和登录。同理还可以实现修改email的功能。

但是会存在一个问题: (英文是之前的程序, 但是问题是相同的)

```
Please enter your id: asddsfsf
{'adsa': 'asddsfsf'}
Wrong! Please try again!
Go on? n|
```

进程已结束，退出代码为 0

修改有误会使得密码和电子邮箱的信息全部丢失，后面操作也会报错。

3、需要注意的是，修改**用户名**设计修改字典中的键，**不可直接修改**。

但是可以通过建立一个**新的键并赋初值**进行处理，类似于temp的功能。

```
1 def usernm_change():
2     un = input("请输入用户名: ")
3     if un not in user:
4         print("用户名不存在，请重新输入!")
5         return
6     nm = input("请输入一个新的用户名: ")
7     user[nm] = {"email": user[un]['email'], "password": user[un]
8     ['password']}
9     del user[un]
10    print("用户名已被修改成功!")
11    return
```

4、上述密码修改功能如果其中一个步骤出错，需要从头重新输入，非常麻烦，我进行了如下改进。

```
1 """二者区别：找回是未知原始密码，修改是已知原始密码"""
2 """但是由于找回密码对安全性来说要求高，现有知识目前还无法解决"""
3 """需要通过邮箱传送验证码，口令正确方可修改"""
4
5 def userpw_change():
6     print("温馨提示：本模式只能提供已知密码的修改。")
7     while True:
8         un = input("请输入用户名: ")
9         if un not in user:
10            print("用户名不存在，请重新输入!")
11            continue
12        else:
13            break
14
15    while True:
16        pw = input("请输入密码: ")
17        if pw != user[un]["password"]:
18            print("密码错误，请重新输入!")
```

```

19         continue
20     else:
21         break
22
23     while True:
24         new_pw = input("请输入新的密码: ")
25         if not password_pattern.match(new_pw):
26             print("密码复杂度不符合要求, 请确保密码至少8个字符, 且包含大写字母、小写字母、数字和特殊字符。")
27             continue
28         else:
29             break
30
31     while True:
32         new_pw0 = input("请再次输入新密码: ")
33         if new_pw == new_pw0:
34             user[un]["password"] = new_pw
35             print("密码修改成功!")
36             break
37         else:
38             print("新密码错误! 请再次输入!")
39             continue
40
41     return
42

```

通过添加**while**循环和**continue**、**break**语句控制进程的走向, 防止“走回头路”。

5、接下来, 基于这种模式, 我们可以对每段代码进行如下修改, 结合上段代码的完整程序如下:

re包的导入和正则表达式的判断:

```

1  import re
2
3  user = {}
4
5  # username_pattern = re.compile(r'[A-Za-z0-9_]+$')
6  email_pattern = re.compile(r'^[a-zA-Z0-9_+-.]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+\.$')
7  password_pattern = re.compile(r'^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$')
8
9
10 def judge_email(email):
11     return email_pattern.match(email) is not None
12
13
14 def judge_password(password):
15     return password_pattern.match(password) is not None
16

```

注册用户名的函数:

```

1  def user_register():
2      print("注册界面: ")

```

```

3     while True:
4         username = input("请输入用户名: ")
5         if username in user:
6             print("用户名已存在, 请输入一个新的用户名! ")
7             continue
8         else:
9             break
10
11    while True:
12        email = input("请输入电子邮件: ")
13        if not judge_email(email):
14            print("电子邮件格式不正确, 请重新输入。")
15            continue
16        else:
17            break
18
19    while True:
20        password = input("请输入密码: ")
21        if not judge_password(password):
22            print("密码复杂度不符合要求, 请确保密码至少8个字符, 且包含大写字母、小写字母、数字和特殊字符。")
23            continue
24        else:
25            break
26
27    user[username] = {"email": email, "password": password}
28    print("用户注册成功! ")
29
30    return
31

```

登录用户名的函数:

```

1  def user_login():
2      print("登录界面: ")
3      while True:
4          un = input("请输入用户名: ")
5          if un not in user:
6              print("用户名不存在, 请重新输入! ")
7              continue
8          else:
9              break
10
11     while True:
12         pw = input("请输入密码: ")
13         if pw == user[un]["password"]:
14             print("欢迎回来! " + un)
15             break
16         else:
17             print("密码错误! 请重新输入。")
18             continue
19
20     return
21

```

修改用户名的函数：

```
1 def usernm_change():
2     while True:
3         un = input("请输入用户名: ")
4         if un not in user:
5             print("用户名不存在，请重新输入！")
6             continue
7         else:
8             break
9
10    nm = input("请输入新的用户名: ")
11    user[nm] = {"email": user[un]['email'], "password": user[un]
12               ['password']}
13    del user[un]
14
15    print('用户名修改成功! ')
16
17    return
```

修改电子邮箱的函数：

```
1 def userem_change():
2     while True:
3         un = input("请输入用户名: ")
4         if un not in user:
5             print("用户名不存在，请重新输入！")
6             continue
7         else:
8             break
9
10    while True:
11        em = input("请输入电子邮箱: ")
12        if em != user[un]["email"]:
13            print("电子邮箱错误，请重新输入！")
14            continue
15        else:
16            break
17
18    while True:
19        new_em = input("请输入新的电子邮箱: ")
20        if not email_pattern.match(new_em):
21            print("电子邮件格式不正确，请重新输入。")
22            continue
23        else:
24            break
25
26    while True:
27        new_em0 = input("请再次输入新电子邮箱: ")
28        if new_em == new_em0:
29            user[un]["email"] = new_em
30            print("电子邮箱修改成功！")
31            break
32        else:
```



```

33         print("电子邮箱错误！请再次输入！")
34         continue
35
36     return
37

```

主函数的程序：

```

1  if __name__ == "__main__":
2      while True:
3          opt = int(input("请输入你的选择（1：注册；2：改密；3：登录；4：改名）"))
4          if opt == 1:
5              user_register()
6          if opt == 2:
7              userpw_change()
8          if opt == 3:
9              user_login()
10         if opt == 4:
11             usernm_change()
12         if input("是否继续？") != 'y':
13             break
14

```

运行结果如图：

| | |
|---|---|
| <p>"C:\Program Files\Python312\python.exe" "C:\Users\Alan Soong\</p> <p>请输入你的选择（1：注册；2：登录；3：改密；4：改名；5：改邮箱）：1</p> <p>注册界面：</p> <p>请输入用户名：adsax</p> <p>请输入电子邮件：qwe</p> <p>电子邮件格式不正确，请重新输入。</p> <p>请输入电子邮件：asdqw321@ds.sa</p> <p>请输入密码：asdAD234?</p> <p>用户注册成功！</p> <p>是否继续？（y or n）：y</p> <p>请输入你的选择（1：注册；2：登录；3：改密；4：改名；5：改邮箱）：2</p> <p>登录界面：</p> <p>请输入用户名：as</p> <p>用户名不存在，请重新输入！</p> <p>请输入用户名：adsax</p> <p>请输入密码：asdAD234?</p> <p>欢迎回来！adsax</p> <p>是否继续？（y or n）：y</p> <p>请输入你的选择（1：注册；2：登录；3：改密；4：改名；5：改邮箱）：</p> | <p>请输入你的选择（1：注册；2：登录；3：改密；4：改名；5：改邮箱）：3</p> <p>温馨提示：本模式只能提供已知密码的修改。</p> <p>请输入用户名：a</p> <p>用户名不存在，请重新输入！</p> <p>请输入用户名：adsax</p> <p>请输入密码：asdAD234?</p> <p>请输入新的密码：asdAD234?!</p> <p>请再次输入新密码：asdAD234?!</p> <p>密码修改成功！</p> <p>是否继续？（y or n）：y</p> <p>请输入你的选择（1：注册；2：登录；3：改密；4：改名；5：改邮箱）：4</p> <p>请输入用户名：adsax</p> <p>请输入新的用户名：add</p> <p>用户名修改成功！</p> <p>是否继续？（y or n）：y</p> <p>请输入你的选择（1：注册；2：登录；3：改密；4：改名；5：改邮箱）：5</p> <p>请输入用户名：adsax</p> <p>用户名不存在，请重新输入！</p> <p>请输入用户名：add</p> <p>请输入电子邮箱：asdqw321@ds.sa</p> <p>请输入新的电子邮箱：asdqw321@ds.com</p> <p>请再次输入新电子邮箱：asdqw321@ds.com</p> |
|---|---|

请输入用户名: *adsax*

请输入新的用户名: *add*

用户名修改成功!

是否继续? (y or n): *y*

请输入你的选择 (1: 注册; 2: 登录; 3: 改密; 4: 改名; 5: 改邮箱): *5*

请输入用户名: *adsax*

用户名不存在, 请重新输入!

请输入用户名: *add*

请输入电子邮箱: *asdqw321@ds.sa*

请输入新的电子邮箱: *asdqw321@ds.com*

请再次输入新电子邮箱: *asdqw321@ds.com*

电子邮箱修改成功!

是否继续? (y or n): *y*

请输入你的选择 (1: 注册; 2: 登录; 3: 改密; 4: 改名; 5: 改邮箱): *2*

登录界面:

请输入用户名: *add*

请输入密码: *asdAD234?!*

欢迎回来! *add*

是否继续? (y or n): *n*

进程已结束, 退出代码为 *0*