- 《Python程序设计与算法基础》第15章上机实践1-3
  - 第一题
    - 题目1
    - 答案1
  - 第二题
    - 题目2
    - 答案2
  - 第三题
    - 题目3
    - 答案3

1 《Python程序设计与算法基础》第15章上机实践1-3

15.8 上机实践

1. 完成本章中的例 15.1～例 15.16，熟悉 Python 语言中的字符串和文本处理程序设计。

2. 统计输入的字符串中英文字母、数字、空格和其他字符出现的次数，程序运行效果如图 15-1 所示。

```
请输入字符串：This is   a test. 123  45678~ End?
所有字母的总数为：   33
英文字母出现的次数：  13
数字出现的次数：      8
空格出现的次数：      9
其他字符出现的次数：  3
```

图 15-1 统计字符运行效果

提示：

本实践题使用表 15-8 中的字符串对象的方法和 str 类方法确定字符/字符串是否为字母、数字、空格等。

表 15-8 本实践题所使用的字符串对象的方法/str 类方法

方 法	功 能
isalpha()	判断字符/字符串是否全为字母
isdigit()	判断字符/字符串是否全为数字(0～9)
isspace()	判断字符/字符串是否只包含空白字符

3. 编写程序，分别输入 3 个字符串，依次验证其是否为有效的中华人民共和国电话号码、中华人民共和国邮政编码和网站网址格式，运行效果如图 15-2 所示。

```
请输入中国电话号码：021-12345678
021-12345678 是有效的电话号码格式吗？ True
请输入中国邮政编码：200062
200062 是有效的邮政编码格式吗？ True
请输入网站网址：http://www.ibm.com
http://www.ibm.com 是有效的网站网址格式吗？ True
```

(a) 有效格式

```
请输入中国电话号码：123456789
123456789 是有效的电话号码格式吗？ False
请输入中国邮政编码：123456789
123456789 是有效的邮政编码格式吗？ False
请输入网站网址：http@ecnu.edu.cn
http@ecnu.edu.cn 是有效的网站网址格式吗？ False
```

(b) 无效格式

图 15-2 验证有效格式运行效果

提示：

(1) 中华人民共和国电话号码(电话号码必须是 8 位号码，如果有区号，区号必须是 3 位)的正则表达式为^(\(\d{3}\)|\d{3}-)? \d{8}$。

(2) 中华人民共和国邮政编码(必须 6 位数字)的正则表达式为^\d{6}$。

(3) 网站网址(Internet URL)的正则表达式为^https?://\w+(?:\.[^\.]+)+(?:/.+)*$。

(4) 验证函数的参考代码如图 15-3 所示。

```
def check_phone(strPhone):  #中华人民共和国电话号码
    regex_phone = re.compile(r'^(\(\d{3}\)|\d{3}-)?\d{8}$')
    result = True if regex_phone.match(strPhone) else False
    return result
def check_ZIP(strZIP):      #中华人民共和国邮政编码
    regex_ZIP = re.compile(r'^\d{6}$')
    result = True if regex_ZIP.match(strZIP) else False
    return result
def check_URL(strURL):      #网站网址
    regex_URL = re.compile(r'^https?://\w+(?:\.[^\.]+)+(?:/.+)*$')
    result = True if regex_URL.match(strURL) else False
    return result
```

图 15-3 验证有效格式参考代码

1-1 第一题

I 题目1

完成本章中的例15.1~例15.16,熟悉 Python 语言中的字符串和文本处理程序设计。

II 答案1

```python
# 1

s1 = 'yellow ribbon'
s2 = 'Pascal Case'
s3 = '123'
s4 = 'iPhone7'

print(s1.islower())
print(s2.isupper())
print(s4.isalnum())
print(s3.isnumeric())
print(s1.isdigit())
print(s2.istitle())
```

```
mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main)> python3 15.1.py
True
False
True
True
False
True
```

```python
# 2

s1 = 'red car'
s2 = 'Pascal Case'
s3 = 'python3.7'
s4 = 'iPhoneX'

print(s1.capitalize())
print(s2.lower())
print(s3.upper())
print(s2.swapcase())
print(s1.title())
print(s4.casefold())
```

```
mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main)> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.1.2.py
Red car
pascal case
PYTHON3.7
pASCAL cASE
Red Car
iphonex
```

```python
# 3

s1 = '123'
s2 = ' 123'
print(len(s2))

print(s2.strip())
print(s2.lstrip())
print(s1.zfill(5))
print(s1.center(5,' '))
print(s1.ljust(5))
print(s1.rjust(5,'0'))
```

```
mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main)> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.1.3.py
```

```
4
123
123
00123
 123
123
00123

# 4

s1 = "abABabCD"
print(s1.startswith("AB"))
print(s1.startswith("AB",2))
print(s1.endswith("CD"))
print(s1.count("ab"))
print(s1.index("AB"))
print(s1.find("cd"))
print(s1.find("CD"))
print(s1.replace("ab","xyz"))

ub/3.codes/Python/Python程序设计作业/第八次作业/15.1.4.py
False
True
True
2
2
-1
6
xyzABxyzCD

# 5

s1 = 'one,two,three'
print(s1.split(','))
print(s1.rsplit(',',1))
print(s1.partition(','))
print(s1.rpartition(','))
s2 = 'abc\n123\nxyz'
print(s2.splitlines())
print(s2.splitlines(True))
s3 = ('a','b','c')
s4 = ':'
print(s4.join(s3))
print(s4.join('123'))

mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main)> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.1.5.py
['one', 'two', 'three']
['one,two', 'three']
('one', ',', 'two,three')
('one,two', ',', 'three')
['abc', '123', 'xyz']
['abc\n', '123\n', 'xyz']
a:b:c
1:2:3

# 6

table1 = str.maketrans('1234567','一二三四五六日')
s1 = '1 3 4 9'
```

```
print(s1.translate(table1))
weeks = {'1': 'M一','2': 'T二','3': 'W三','4': 'T四','5': 'F五','6': 'S六','7': 'S日'}
table2 = str.maketrans(weeks)
print(s1.translate(table2))

mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main)> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.1.6.py
一 三 四 9
M一 W三 T四 9

# 7

s1 = input('请输入字符串: ') # 'The quick brown fox jumps over the lazy dog
s2 = s1.upper() #转换为大写
countall = len(s1) #字符串长度
counta = s2.count('A');counte = s2.count('E');counti = s2.count('I')
counto = s2.count('O');countu = s2.count('U')
print('所有字母的总数为: ',countall)
print('元音字母出现的次数和频率分别为: ')
print('A:{0}\t{1:2.2f}%'.format(counta, counta/countall * 100))
print('E:{0}\t{1:2.2f}%'.format(counte, counte/countall * 100))
print('I:{0}\t{1:2.2f}%'.format(counti, counti/countall * 100))
print('O:{0}\t{1:2.2f}%'.format(counto, counto/countall * 100))
print('U:{0}\t{1:2.2f}%'.format(countu, countu/countall * 100))

mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main) [1]> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.1.7.py
请输入字符串: The quick brown fox jumps over the lazy dog
所有字母的总数为:  43
元音字母出现的次数和频率分别为:
A:1     2.33%
E:3     6.98%
I:1     2.33%
O:4     9.30%
U:2     4.65%

# 8

# 15.1.8.py

file_name = "text_count.txt"
line_counts = 0
word_counts = 0
character_counts = 0
with open(file_name, 'r', encoding = 'utf8') as f:
    for line in f:
        words = line.split()
        line_counts += 1
        word_counts += len(words)
        character_counts += len(line)
print("行数:", line_counts)
print("单词个数:", word_counts)
print("字符个数:", character_counts)

# text_count.txt

this is a word.
and this is a testing of text count.

# shell
```

```
mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main) [1]> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.1.8.py
行数: 2
单词个数: 12
字符个数: 52

# 9

import re
print(re.findall('d','godness'))

mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main)> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.1.9.py
['d']

# 10

import os, re
def check_email(strEmail):
    regex_email = re.compile(r'^[\w\.\-]+@([\w\-]+\.)+[\w\-]+$')
    result = True if regex_email.match(strEmail) else False
    return result

# test
if __name__ == '__main__':
    str1 = "hjiang@yahoo.com"
    str2 = "hjiang.yahoo.com"
    print(str1, 'is correct?', check_email(str1))
    print(str2, 'is correct?', check_email(str2))

mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main)> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.1.10.py
hjiang@yahoo.com is correct? True
hjiang.yahoo.com is correct? False

# 11

import os,re
def tasklist():
    regex_tast = re.compile(r'([\w.]+(?:[\w.]+)*)\s\s+(\d+)\w+\s\s+\d+\s\
s+([\d,]+K)')
    with os.popen('tasklist /nh','r') as f:
        for line in f:
            print(regex_tast.findall(line.strip()))

if __name__ == '__main__':
    tasklist()

mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main)> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.1.11.py
/bin/sh: tasklist: command not found

# 12

import re
def html_txt(htmlwithtag):
    regex_href = re.compile(r'<.+?>')
    return regex_href.sub('',htmlwithtag)
```

```
# test
if __name__ == '__main__':
    htmltext = r'<a href=\"index.html\"Welcome to Python world!</a>'
    print(html_txt(htmltext))

mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main)> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.1.12.py

# 13

import re,urllib.request
def url_extract(homepage):
    regex_href = re.compile(r'href="(.+?)"')
    f = urllib.request.urlopen(homepage)
    webcontents = f.read().decode()
    matches = regex_href.finditer(webcontents)
    for m in matches:
        print(m.group(1))
# test
if __name__ == '__main__':
    www = r'http://www.baidu.com'
    url_extract(www)

mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main)> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.1.13.py
/favicon.ico
/content-search.xml
//www.baidu.com/img/baidu_85beaf5496f291521eb75ba38eacbd87.svg
//dss0.bdstatic.com
//dss1.bdstatic.com
//ss1.bdstatic.com
//sp0.baidu.com
//sp1.baidu.com
//sp2.baidu.com
https://psstatic.cdn.bcebos.com/video/wiseindex/aa6eef91f8b5b1a33b454c401_1660835115
000.png
//
/
javascript:;
https://passport.baidu.com/v2/?login&tpl=mn&u=http%3A%2F%2Fwww.baidu.com%2F&sms=5
http://news.baidu.com
https://www.hao123.com?src=from_pc
http://map.baidu.com
http://tieba.baidu.com/
https://haokan.baidu.com/?sfrom=baidu-top
http://image.baidu.com/
https://pan.baidu.com?from=1026962h
http://www.baidu.com/more/
https://passport.baidu.com/v2/?login&tpl=mn&u=http%3A%2F%2Fwww.baidu.com%2F&sms=5
javascript:;
javascript:;
//www.baidu.com/s?wd=%E7%99%BE%E5%BA%A6%E7%83%AD%E6%90%9C&amp;sa=ire_dl_gh_logo_texi
ng&amp;rsv_dl=igh_logo_pcs
/
javascript:;
javascript:;
javascript:;
https://www.baidu.com/s?wd=2022%E5%8D%A1%E5%A1%94%E5%B0%94%E4%B8%96%E7%95%8C%E6%9D%A
F&rsv_dl=Worldcup_PC_2022_index_tips
https://top.baidu.com/board?platform=pc&sa=pcindex_entry
```

```
https://www.baidu.com/s?wd=%E5%AF%B9%E4%BA%8E%E5%B0%81%E6%8E%A7%E4%BA%BA%E5%91%98+%E5%A6%82%E4%BD%95%E4%BF%9D%E9%9A%9C%E6%AD%A3%E5%B8%B8%E5%B0%B1%E5%8C%BB%EF%BC%9F&amp;sa=fyb_n_homepage&amp;rsv_dl=fyb_n_homepage&amp;from=super&amp;cl=3&amp;tn=baidutop10&amp;fr=top1000&amp;rsv_idx=2&amp;hisfilter=1
https://www.baidu.com/s?wd=%E6%8B%AF%E6%95%91%E4%BA%86%E7%94%9F%E5%91%BD%E7%9A%84%E4%B8%AD%E5%9B%BD%EF%BC%8C%E4%B8%8D%E5%BA%94%E8%A2%AB%E8%AF%AF%E8%A7%A3&amp;sa=fyb_n_homepage&amp;rsv_dl=fyb_n_homepage&amp;from=super&amp;cl=3&amp;tn=baidutop10&amp;fr=top1000&amp;rsv_idx=2&amp;hisfilter=1
https://www.baidu.com/s?wd=%E5%AA%92%E4%BD%93%E6%8E%A2%E8%AE%BF%E6%A0%B8%E5%AD%90%E5%9F%BA%E5%9B%A0%3A%E5%88%9B%E5%A7%8B%E4%BA%BA%E7%88%B6%E4%BA%B2%E4%B8%BA%E6%95%99%E5%B8%88&amp;sa=fyb_n_homepage&amp;rsv_dl=fyb_n_homepage&amp;from=super&amp;cl=3&amp;tn=baidutop10&amp;fr=top1000&amp;rsv_idx=2&amp;hisfilter=1
https://www.baidu.com/s?wd=%E5%B9%BF%E5%B7%9E%E6%B5%B7%E7%8F%A0%E8%AF%95%E8%A1%8C%E2%80%9C%E9%97%AD%E7%8E%AF%E6%B3%A1%E6%B3%A1%E2%80%9D%E6%8E%A8%E8%BF%9B%E5%A4%8D%E5%B7%A5&amp;sa=fyb_n_homepage&amp;rsv_dl=fyb_n_homepage&amp;from=super&amp;cl=3&amp;tn=baidutop10&amp;fr=top1000&amp;rsv_idx=2&amp;hisfilter=1
https://www.baidu.com/s?wd=%E9%9F%A9%E5%9B%BDvs%E5%8A%A0%E7%BA%B3&amp;sa=fyb_n_homepage&amp;rsv_dl=fyb_n_homepage&amp;from=super&amp;cl=3&amp;tn=baidutop10&amp;fr=top1000&amp;rsv_idx=2&amp;hisfilter=1
https://www.baidu.com/s?wd=%E5%AA%92%E4%BD%93%3A%E6%96%B0%E7%96%86%E6%9A%B4%E9%9B%AA+%E6%9C%89%E7%89%A7%E6%B0%91%E5%A4%B1%E8%81%94%E7%89%9B%E7%BE%8A%E5%86%BB%E6%AD%BB&amp;sa=fyb_n_homepage&amp;rsv_dl=fyb_n_homepage&amp;from=super&amp;cl=3&amp;tn=baidutop10&amp;fr=top1000&amp;rsv_idx=2&amp;hisfilter=1
//home.baidu.com
http://ir.baidu.com
//www.baidu.com/duty
//help.baidu.com
https://e.baidu.com/?refer=1271
http://www.beian.gov.cn/portal/registerSystemInfo?recordcode=11000002000001
https://beian.miit.gov.cn
//www.baidu.com/licence/
https://www.baidu.com/s?rtt=1&amp;bsst=1&amp;cl=2&amp;tn=news
http://v.baidu.com/v?ct=301989888&amp;rn=20&amp;pn=0&amp;db=0&amp;s=25&amp;ie=utf-8
http://image.baidu.com/i?tn=baiduimage&amp;ps=1&amp;ct=201326592&amp;lm=-1&amp;cl=2&amp;nc=1&amp;ie=utf-8
http://zhidao.baidu.com/q?ct=17&amp;pn=0&amp;tn=ikaslist&amp;rn=10&amp;fr=wwwt
http://wenku.baidu.com/search?lm=0&amp;od=0&amp;ie=utf-8
http://tieba.baidu.com/f?fr=wwwt
https://map.baidu.com/?newmap=1&amp;ie=utf-8&amp;s=s
https://b2b.baidu.com/s?fr=wwwt
http://www.baidu.com/more/
//www.baidu.com/s?rtt=1&bsst=1&cl=2&tn=news&word=
http://ss.bdimg.com/static/superman/css/ubase_sync-d600f57804.css?v=md5

# 14

# 15.1.14.py

import re
import collections
def analyze_text(text):
    paragraphs = re.split("\n\n",text)
    paragraph_count=len(paragraphs)
    print("段落数:{0}".format(paragraph_count))
    lines=re.split("\n",text)
    line_count=len(lines)
    print("行数:{0}".format(line_count))
    sentences=re.split("[.?!]",text)
    sentence_count=len(sentences)
    print("句数:{0}".format(sentence_count))
```

```
    words=re.split(r"\W+",text)
    word_count=len(words)
    print("单词数:{0}".format(word_count))
    freqs=collections.Counter(words)
    print("频率最高的10个单词:")
    for (w,n) in freqs.most_common(10):
        print("{0:10}:{1:10}".format(w,n))
if __name__ == "__main__":
    filename = "tomsawyer.txt"
    with open(filename,"r") as f:
        text=f.read()
    analyze_text(text.strip())

# tomsawyer.txt

tomsawyer.txt
hello
hi
hello
what

# shell

mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main)> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.1.14.py
段落数:1
行数:5
句数:2
单词数:6
频率最高的10个单词:
hello     :         2
tomsawyer :         1
txt       :         1
hi        :         1
what      :         1

# 15

# 15.1.15.py

def isPotentialGene(dna):
    # 基因长度为3的倍数，否则返回False
    if len(line) % 3 != 0:
        return False
    # 基因以ATG开始，否则返回False
    if not dna.startswith('ATG'):
        return False
    # 基因以TAG、TAA或者TGA结束，否则返回False
    if dna[-3:] not in ('TAG', 'TAA', 'TGA'):
        return False
    # 基因中间部分不包括密码子TAG、TAA或者TGA，否则返回False
    for i in range(3, len(dna)-3,3):
        if dna[i:i+3] in ('TAG', 'TAA', 'TGA'):
            return False
        return True

if __name__ == "__main__":
    filename = "gene.txt"
    for lineno, line in enumerate(open(filename,"r")):
        if isPotentialGene(line.strip()):
```

```
            print("{0}:{1}".format(lineno+1,line.strip()))

# gene.txt

ATGCGCCTGCGTCTGTATAG
ATGCGCCTGCGTCTGTATAA
ATGCGCCTGCGTCTGTATGA

# shell

mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main)> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.1.15.py
1:ATGCGCCTGCGTCTGTATAG
2:ATGCGCCTGCGTCTGTATAA

# 16

from itertools import cycle
def crypt(text, key):
    result = []
    keys = cycle(key)
    for ch in text:
        result.append(chr(ord(ch)^ord(next(keys))))
    return ''.join(result)
# test
if __name__ == "__main__":
    plain = 'The quick brown fox jumps over the lazy dog'
    key = 'Python_1'
    print('Before crypt:{}'.format(plain))
    encrypted = crypt(plain,key)
    print('After crypt:{}'.format(encrypted))
    decrypted = crypt(encrypted,key)
    print('Decrypted:{}'.format(decrypted))

mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main)> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.1.16.py
Before crypt:The quick brown fox jumps over the lazy dog
After crypt:HR;Y16
                 HA#Y

8T&4
Decrypted:The quick brown fox jumps over the lazy dog
```


1-2 第二题

I 题目2

统计输入的字符串中英文字母、数字、空格和其他字符出现的次数，程序运行效果如图15-1所示。

```
请输入字符串：This is   a test. 123  45678~ End?
所有字母的总数为：   33
英文字母出现的次数：  13
数字出现的次数：      8
空格出现的次数：      9
其他字符出现的次数：  3
```

图 15-1　统计字符运行效果

提示：

本实践题使用表 15-8 中的字符串对象的方法和 str 类方法确定字符/字符串是否为字母、数字、空格等。

表 15-8　本实践题所使用的字符串对象的方法/str 类方法

方　法	功　能
isalpha()	判断字符/字符串是否全为字母
isdigit()	判断字符/字符串是否全为数字(0～9)
isspace()	判断字符/字符串是否只包含空白字符

II 答案2

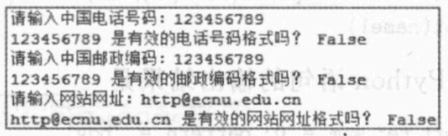
```
# 15.2.py

import re
import collections
def analyze_text(text):
    # calculate all alpha
    alpha = re.findall(r"[a-zA-Z0-9]",text)
    alpha_count = len(alpha)
    print(alpha)
    print("Alphabet: {0}".format(alpha_count))

    # calculate English alpha
    eng_alpha = re.findall(r"[a-zA-Z]",text)
    eng_alpha_count = len(eng_alpha)
    print(eng_alpha)
    print("English Alphabet: {0}".format(eng_alpha_count))

    # calculate Number
    number = re.findall(r"\d",text)
    number_count = len(number)
    print(number)
    print("Number: {0}".format(number_count))

    # calculate Space
    space = re.findall(r"\s",text)
    space_count = len(space)
    print(space)
    print("Space: {0}".format(space_count))

    # calculate Other
    other = re.findall(r"[^a-zA-Z0-9 ]",text)
    other_count = len(other)
    print(other)
    print("Other: {0}".format(other_count))


if __name__ == "__main__":
    filename = "calc.txt"
    with open(filename,"r") as f:
        text = f.read()
    analyze_text(text.strip())

# calc.txt

hello 123>>>???
```


1-3 第三题

I 题目3

编写程序,分别输入3个字符串,依次验证其是否为有效的中华人民共和国电话号码、中华人民共和国邮政编码和网站网址格式,运行效果如图 15-2 所示。

(a) 有效格式



(b) 无效格式

图 15-2　验证有效格式运行效果

提示：

(1) 中华人民共和国电话号码(电话号码必须是 8 位号码，如果有区号，区号必须是 3 位)的正则表达式为^(\(\d{3}\)|\d{3}-)? \d{8}$。

(2) 中华人民共和国邮政编码(必须 6 位数字)的正则表达式为^\d{6}$。

(3) 网站网址(Internet URL)的正则表达式为^https?://\w+(?: \.[^\.]+)+(?: /.+)* $。

(4) 验证函数的参考代码如图 15-3 所示。

```
def check_phone(strPhone):  #中华人民共和国电话号码
    regex_phone = re.compile(r'^(\(\d{3}\)|\d{3}-)?\d{8}$')
    result = True if regex_phone.match(strPhone) else False
    return result
def check_ZIP(strZIP):      #中华人民共和国邮政编码
    regex_ZIP = re.compile(r'^\d{6}$')
    result = True if regex_ZIP.match(strZIP) else False
    return result
def check_URL(strURL):      #网站网址
    regex_URL = re.compile(r'^https?://\w+(?:\.[^\.]+)+(?:/.+)*$')
    result = True if regex_URL.match(strURL) else False
    return result
```

图 15-3　验证有效格式参考代码

II 答案3

```
import re

def judgeNumber(number):
    zhNumber = re.compile(r'^(\(\d{3}\)|\d{3}-)?\d{8}$')
    result = True if zhNumber.match(number) else False
    return result

def judgeMailCode(code):
    zhMailCode = re.compile(r'^\d{6}$')
    result = True if zhMailCode.match(code) else False
    return result

def judgeWebSite(web):
    WebSite =
re.compile(r'^(?:https?:\/\/)?[a-zA-Z0-9][-a-zA-Z0-9]{0,62}(?:\.[a-zA-Z0-9][-a-zA-Z0
-9]{0,62})+$')
    result = True if WebSite.match(web) else False
    return result

if __name__ == "__main__":
    strNumber = str(input())
    print(strNumber,' is correct? ', judgeNumber(strNumber))
    strMailCode = str(input())
    print(strMailCode,' is correct? ', judgeMailCode(strMailCode))
    strWeb = str(input())
    print(strWeb,' is correct? ', judgeWebSite(strWeb))

mikeshinoda@Mikes-Air ~/G/3/P/P/第八次作业 (main)> /opt/homebrew/bin/python3
/Users/mikeshinoda/Github/3.codes/Python/Python程序设计作业/第八次作业/15.3.py
021-12345678
021-12345678  is correct?  True
123456
123456  is correct?  True
https://www.r2coding.com
https://www.r2coding.com  is correct?  **True**
```