**变量**

在内存中的位置可以存储不同的值，如整数，实数，布尔值，字符串，或者更复杂的数据结构，列表，字典，如果我们定义一个存储整形的变量和一个存储字符串的提示消息。为了把两个变量连接到一个字符串中，我们必须用str（）函数

port=23

banner="FTP Server"

print("[+]"+banner+"on port"+str(port)))

type（）函数可以识别变量类型

type(port)

**字符串**

python中字符串模块提供了一系列非常强大的字符串操作方法

常用函数：

upper() 小写字母转大写字母

lower() 大写字母转小写字母

replace(old,new)将old旧字符替换成new新字符

find()查找检测的字符有没有包含制定字符

banner = "FreeFloat FTP Server"

>>> print banner.upper()

FREEFLOAT FTP SERVER

>>> print banner.lower()

freefloat ftp server

>>> print banner.replace('FreeFloat','Ability')

Ability FTP Server

>>> print banner.find('FTP')

10

**列表**

python列表，提供了一种存储一组数据的方式。程序员可以构建任何数据的列表，另外有一些内置操作列表的方法，例如添加，删除，插入，弹出，获取索引，排序，计时，排序反转。

.append()函数添加元素建立列表。

.remove()删除莫元素

.sort()从从小排到大

protlist=[]

protilst.append(21)

protilst.append(80)

protilst.append(433)

protilst.append(25)

print(protilst)

[21,80,433,25]

portilst.sort() #从小到大排序

print(protilst)

[21,25,80,433]

portilst.remove(433)#删除其中某元素

print(protilst)

[21,25,80]

print(len(protilst)) #列出长度

3

**字典**

python数据结构字典，可以存储任何数量的python对象的哈系表，字典的元素由键和值组成。

当我们建立一个字典时，每个键和值用冒号隔开，同时我们用逗号分割元素。.keys()这个方法将返回字典中所有键的列表，.items()将返回字典的元素的一系列列表。

service={'ftp':21,'ssh':22,'http':80}

service.keys()

['ftp','ssh','http']

services.items()

[('ftp', 21), ('ssh', 22), ('http', 80)]

services.has\_key('ftp')

True

services['ftp']

21

**两种字符串截断匹配**

import binascii,re

s='466C6167317B63404E745F35655F6D457D'

flag=""

for i in range(len(s)):

flag= flag + str(binascii.a2b\_hex(s[4\*i:4\*i+4]))

print(flag)

s='466C6167317B63404E745F35655F6D457D'

result1 = re.findall('.{2}',s)

flag=""for i in result1:

flag= flag+ (chr(int(i,16)))

print(flag)

**zfill函数**

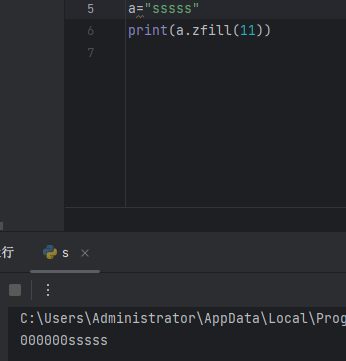
①语法说明：

str.zfill(width)

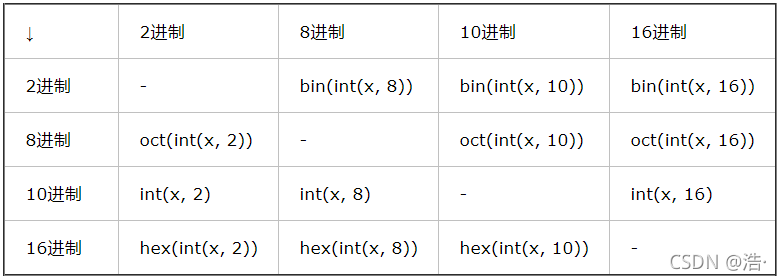
其中，str表示这里要对一个字符串类型string进行操作，如果是整型、浮点型这样的数字类型，要先通过str()函数转化为字符串，才可以进行相关的操作；

width表示进行补零之后的字符串的长度，如果补零后的长度小于或等于原字符串的长度，那么字符串将不会产生任何的变化（该长的短不了）

就是不够就在前面补0的一个函数



**进制字符转换：**



**chr函数：**

就是把10进制也可以是16进制的形式的数字，转换成整数

chr() 用一个范围在 range（256）内的（就是0～255）整数作参数，返回一个对应的字符。

以下是 chr() 方法的语法:

chr(i)

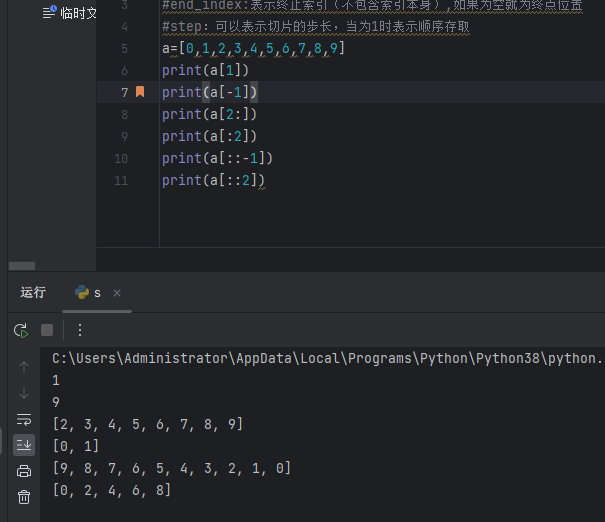
**切片：**

#object[start\_index:end\_index:step]

#start\_index：表示起始索引（包含索引本身）,如果为空就取端点位置

#end\_index:表示终止索引（不包含索引本身）,如果为空就为终点位置

#step：可以表示切片的步长，当为1时表示顺序存取



if \_\_name\_\_=='\_\_main\_\_':

    main()  #如果是直接执行就相当于

main() 代码的意思，如果是被import调用哪\_\_name\_\_就不等于\_\_main\_\_就不执行main()函数就达到保密的 效果

try:

    x = input("ssss:")

    y = input("kkkk:")

    print(x/y)

except :

    print("报错信息") #自定义报错信息

import os

os.system("ipconfig")#调用命令

with open('1.txt','w') as file:

 file.write('ssssssssssssssssssssss')

 file.close() #with写入字符到文件1.txt

------------------------------------

f = open('1.txt','r')

f.close #将文件内容读取并赋值到f

import time #线程睡眠指定时间

def a():

    for x in range(6):

        print("等六秒")

        time.sleep(1)

def b():

    for x in range(7):

        print("再等七秒")

        time.sleep(1)

def main():

    a()

    b()

if \_\_name\_\_ == '\_\_main\_\_':

    main()

import time

import threading

#单线程

def a():

    for x in range(6):

        print("等六秒")

        time.sleep(1)

def b():

    for x in range(7):

        print("再等七秒")

        time.sleep(1)

def main():

    a()

    b()

if \_\_name\_\_ == '\_\_main\_\_':

    main()

#多线程

def a():

    for x in range(6):

        print("等六秒")

        time.sleep(1)

def b():

    for x in range(7): #threading多线程使用方法

        print("再等七秒")

        time.sleep(1)

def main():

    t1 = threading.Thread(target=a)

    t2 = threading.Thread(target=b)

    t1.start()

    t2.start()

if \_\_name\_\_ == '\_\_main\_\_':

    main()

该函数执行一个命令并返回命令的结果，可用以替代os.system()。

pexpect.run('ls -la')#该函数执行一个命令并返回命令的结果，可用以替代os.system()。

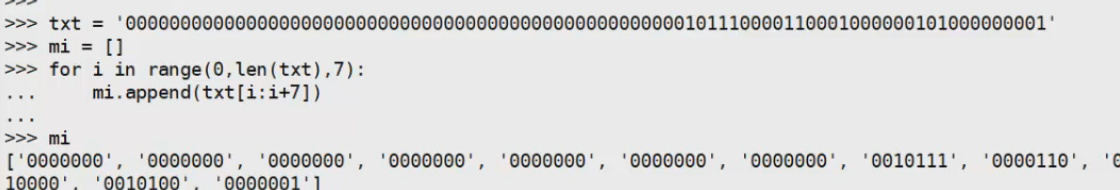
将字符串的值匹配长度8的值赋值给k字符串（可以实现二进制转换进制的操作）

import re

s="01010101010101010101100101011100101010101010101001111001110"

k=re.findall(r'.{8}',s) #匹配长度为8的值赋给k

按7个一组的分割字符串



**python模拟键盘鼠标基本操作**

import time

from pymouse import \*

from pykeyboard import \*

import pyperclip

from pykeyboard import PyKeyboard

m = PyMouse()

a = m.position() #获取当前坐标的位置(这个东西到时候可以新建个py 获取坐标)#

m.move(1510, 10) #鼠标移动到(x,y)位置#

a = m.position()m.click(1510, 10) #移动并且在(x,y)位置左击

time.sleep(1)

m.click(598, 51) #移动并且在(x,y)位置左击

m.click(598, 51) #移动并且在(x,y)位置左击

m.click(598, 51) #移动并且在(x,y)位置左击#

m.click(300, 300, 2) #(300,300)位置右击

pyperclip.copy('中文')

keyboard = PyKeyboard()

keyboard.press\_key(keyboard.control\_key) # 按下Ctrl键

keyboard.tap\_key('V') # 点击V键

keyboard.release\_key(keyboard.control\_key) # 松开Ctrl键

keyboard.press\_key(keyboard.enter\_key) #ent回车键

keyboard.press\_key(keyboard.function\_keys[12]) #按F12键

keyboard.press\_key(keyboard.alt\_key) # 按下Alt键

keyboard.press\_key(keyboard.function\_keys[4]) #按F4键

keyboard.release\_key(keyboard.alt\_key) #松开Alt键

keyboard.press\_key(keyboard.enter\_key) #ent回车键

print(dir(keyboard)) #查看按键所以属性

m.click(1510, 10) #移动并且在(x,y)位置左击

time.sleep(1)

m.click(598, 51) #移动并且在(x,y)位置左击

m.click(598, 51) #移动并且在(x,y)位置左击

m.click(598, 51) #移动并且在(x,y)位置左击#

m.click(300, 300, 2) #(300,300)位置右击

pyperclip.copy('中文')

keyboard = PyKeyboard()

keyboard.press\_key(keyboard.control\_key) # 按下Ctrl键

keyboard.tap\_key('V') # 点击V键

keyboard.release\_key(keyboard.control\_key) # 松开Ctrl键

keyboard.press\_key(keyboard.enter\_key) #ent回车键

keyboard.press\_key(keyboard.function\_keys[12]) #按F12键

keyboard.press\_key(keyboard.alt\_key) # 按下Alt键

keyboard.press\_key(keyboard.function\_keys[4]) #按F4键

keyboard.release\_key(keyboard.alt\_key) #松开Alt键

keyboard.press\_key(keyboard.enter\_key) #ent回车键

print(dir(keyboard)) #查看按键所以属性

**python图像识别**

#!/usr/bin/env python

# -\*- coding: utf-8 -\*-

#在同个目录里放zan.png就会找zan.png图标然后点击

import time

def zan():

time.sleep(0.5) # 等待 0.5 秒

left, top, width, height = pyautogui.locateOnScreen('zan.png') # 寻找 点赞图片；

center = pyautogui.center((left, top, width, height)) # 寻找 图片的中心

pyautogui.click(center) # 点击 print('点赞成功！')

while True:

if pyautogui.locateOnScreen('zan.png'):

zan() # 调用点赞函数

break #只执行一次

else:

pyautogui.scroll(-500) # 本页没有图片后，滚动鼠标；

print('没有找到目标，屏幕下滚~')

**python网站爬虫**

学习链接：[selenium-python中文文档](https://python-selenium-zh.readthedocs.io/zh_CN/latest/)

控制浏览器操作

控制浏览器窗口大小

driver.set\_window\_size(480, 800)

浏览器后退，前进

# 后退 driver.back()

# 前进 driver.forward()

刷新

driver.refresh() # 刷新

find\_element()

早期的selenium提供了针对id、name、xpath等多种方式的具体方法来定位到具体的元素，比如find\_element\_by\_id()、find\_element\_by\_name()等，在后续的升级中，这些方法被弃用了，现在统一使用find\_element(by=By.ID, value=None)方法，该方法包含了id、name、xpath等定位方式

find\_element(by=By.ID, value=None)是WebDriver对象用于定位元素的方法，返回对应元素对象（WebElement），我们的点击、拖拽等操作都是在元素对象的基础上进行的

参数说明

by：指定按照对应的方式来定位元素

By.ID，根据查找标签中的id属性来定位元素

By.NAME，根据查找标签中的name属性来定位元素

By.CLASS\_NAME，根据class属性指定的值来查找元素

By.CSS\_SELECTOR，根据css选择器的方式来查找元素

By.XPATH，根据XPath语法来查找元素

By.LINK\_TEXT，查找文本精确匹配的a标签元素

By.PARTIAL\_LINK\_TEXT，查找文本模糊匹配的a标签元素

By.TAG\_NAME，根据标签名称来查找元素，不太好用，不常用

value：元素位置，字符串类型



import time

from selenium import webdriver

from selenium.webdriver.common.by import By

---------------百度搜索-------------------

driver = webdriver.Chrome()

driver.get("https://www.baidu.com")

driver.find\_element(By.ID, "kw").send\_keys("单基强最帅") #打开百度的输入框输入time.sleep(5)

driver.find\_element(By.ID, "su").click() #点击搜索按钮time.sleep(5)

---------------哔哩哔哩搜索-------------------

driver = webdriver.Chrome()

driver.get("https://www.bilibili.com/")

time.sleep(5)

driver.find\_element(By.CLASS\_NAME,"nav-search-input").send\_keys("shan")

time.sleep(5)

driver.find\_element(By.CLASS\_NAME,"nav-search-btn").click()

time.sleep(5)

打开默认浏览器这样就有那些登录信息了

from selenium import webdriver

option = webdriver.ChromeOptions()

# 添加保持登录的数据路径：安装目录一般在C:\Users\\*\*\*\*\AppData\Local\Google\Chrome\User Data

option.add\_argument(r"user-data-dir=C:\Users\Administrator\AppData\Local\Google\Chrome\User Data")

如果找不到可以直接搜索Google一个一个试目录

driver = webdriver.Chrome(options=option)

driver.get("https://www.csdn.net/")

driver.maximize\_window()