目录

[一、 python概览 3](#_Toc37418096)

[1. 优点 3](#_Toc37418097)

[2. 缺点 3](#_Toc37418098)

[3. 小知识 3](#_Toc37418099)

[二、 Python运行 3](#_Toc37418100)

[1.运行脚本文件的格式 3](#_Toc37418101)

[2.代码格式（持续更新） 3](#_Toc37418102)

[（1）赋值 3](#_Toc37418103)

[（2）If，else 3](#_Toc37418104)

[（3）type函数用法 4](#_Toc37418105)

[（4）id函数的用法 4](#_Toc37418106)

[（5）==的用法 4](#_Toc37418107)

[（6）is函数的用法 4](#_Toc37418108)

[（7）载入随机数模块 4](#_Toc37418109)

[（8）字符串 4](#_Toc37418110)

[（9）列表 4](#_Toc37418111)

[（10）字典表 4](#_Toc37418112)

[（11）字典表中get函数的使用 5](#_Toc37418113)

[（12）tuple元组 5](#_Toc37418114)

[（13）占位符的使用 5](#_Toc37418115)

[（14）除号的使用：/和// 5](#_Toc37418116)

[(15)小数的取整 6](#_Toc37418117)

[（16）二进制，八进制，十六进制 6](#_Toc37418118)

[（17）转义符 6](#_Toc37418119)

[（17）字符串前**r**的使用（忽略所有转义符） 7](#_Toc37418120)

[（18）“““ ”””与注释 7](#_Toc37418121)

[（19）for循环的简单使用 7](#_Toc37418122)

[（20）字符串的简单使用 7](#_Toc37418123)

[（21）字符串中单个字符的替换 8](#_Toc37418124)

[（22）字符串和列表的相互转化 8](#_Toc37418125)

[（23）字符串以某个字符切割为列表 8](#_Toc37418126)

[（24）判断字符串是否为某个字母开头或结尾或包含某个字符 8](#_Toc37418127)

[（25）交换变量 8](#_Toc37418128)

[（26）将列表赋值给新列表 8](#_Toc37418129)

[（27）列表append函数与extend函数 9](#_Toc37418130)

[（28）列表的排序 9](#_Toc37418131)

[（29）列表间的赋值 9](#_Toc37418132)

[(30)字典表的两种声明方式 9](#_Toc37418133)

[（31）字典表的合并（添加） 10](#_Toc37418134)

[（32）字典表删除（弹出）某一项 10](#_Toc37418135)

[（33）查看字典表的键和值 10](#_Toc37418136)

[（34）元组的性质 10](#_Toc37418137)

[（35）namedtuple使用方法 10](#_Toc37418138)

[（36）文件的相关操作 10](#_Toc37418139)

[（37）pickle模块：存取python对象（以二进制方式） 11](#_Toc37418140)

[（38）浏览磁盘文件 11](#_Toc37418141)

[（39）字符串的赋值 12](#_Toc37418142)

[（40）print函数的使用 12](#_Toc37418143)

[（41）while的使用方法 12](#_Toc37418144)

[（42）接收用户输入的函数：***input( )*** 12](#_Toc37418145)

[（43）range函数和enumerate函数 12](#_Toc37418146)

[（44）二进制，版禁止，十进制，十六进制简称 13](#_Toc37418147)

1. python概览
2. 优点

软件质量，开发效率，可移植性，库支持丰富，组件集成

1. 缺点

运行效率低

1. 小知识

只有0~256才有固定的储存位置，其他的数字都是随机储存的

1. Python运行

1.运行脚本文件的格式

将结果输出至显示器：

C:\Users\duqnq>py C:\Users\duqnq\source\repos\MyPython\hello.py

将结果输出至某一文件：

C:\Users\duqnq>py C:\Users\duqnq\source\repos\MyPython\hello.py>C:\Users\duqnq\source\repos\MyPython\result.txt

2.代码格式（持续更新）

（1）赋值

Score=60（无需分号，无需提前声明变量类型）

Name=’Tom’或name=”Tom”

（2）If，else

>>> if 75<60***:***

print('good')

***elif*** 60>70: ***elif：else if***

print('verry good')

else:

print("not yet")

not yet

>>> score=60

>>> ***result='good' if score>=60 else 'bad'***

>>> print(result)

good

（Python不会自动缩进，需要自己对照位置）

（3）type函数用法

>>> score=60

>>> type(score)

<class 'int'>

（输出函数内变量的类型）

（4）id函数的用法

>>> score=60

>>> id(score)

140725058697328

（输出函数内变量的储存位置）

（5）==的用法

判断两个变量值是否相同

Score == age

结果为布尔型

（6）is函数的用法

判断储存位置是否相同

Score is age

结果为布尔型

（7）载入随机数模块

>>>import random

>>>random.random()

0.7028322386207699

（8）字符串

S=’abc’

（9）列表

>>> l=[1,2,3,4]

>>> l

[1, 2, 3, 4]

>>> len(l)

4

>>> l[0]

1

>>> l[2]='abc'

>>> l[3]=[2,3,4]

>>> l

[1, 2, 'abc', [2, 3, 4]]

（10）字典表

>>> d={'age':16,'name':'Tom'}

>>> d['age']

16

>>> d['name']

'Tom'

>>> ***def add(x):***

print(x)

>>> def defaut(x):

print("BUG")

>>> operation={'add':add,'delete':***lambda x:print(x\*\*2)***}

>>> operation.get('add')(10)

10

>>> operation.get('delete')(10)

100

>>> operation.get('delet',defaut)(10)

BUG

（列表初始化使用中括号，字典表初始化使用大括号）

（11）字典表中get函数的使用

>>> d.get('name')

'Tom'

若无查找项，可以赋默认值：

>>> d.get('job','teacher')

'teacher'

（12）tuple元组

初始化使用小括号

>>> d=(1,2,3,4)

>>> d

(1, 2, 3, 4)

与列表类似，但不可修改值

（13）占位符的使用

>>> '{0}'.format(2)

'2'

>>> "{0}".format(2)

'2'

>>> 'f={0}'.format(5)

'f=5'

({0}为占位符，即将被替换的位置)

>>> '{0:.2f}'.format(0.33333)

'0.33'

**.2作用：保留两位小数**

**f作用：浮点型**

>>> '{0:.2e}'.format(0.33333)

'3.33e-01'

**e作用：科学技术法显示**

>>> d={'age':16,'name':'Tom'}

>>> '姓名：{0},年龄：{1}'.format(d.get('name'),d.get('age'))

'姓名：Tom,年龄：16'

（14）除号的使用：/和//

>>> 10/4

2.5

>>> 10//4

2

>>> 10//4.0

2.0

**//可以舍去小数点后面的部分**

(15)小数的取整

>>> import math

>>> math.floor(3.7)

3

>>> math.trunc(3.1)

3

>>> round(3.7)

4

**导入math模块**

**floor：向下取整**

**trunc：截断小数部分**

**全局函数round：四舍五入法**

（16）二进制，八进制，十六进制

二进制：0b开头，转换函数：bin（）

八进制：0o开头，转换函数：oct（）

十六进制：0x开头，10为a，以此类推，转换函数：hex（）

>>> 0b10101

21

>>> 0o1027

535

>>> 0x23c

572

（17）转义符

|  |  |
| --- | --- |
| \a | 响铃(BEL) |
| \b | 退格(BS) ，将当前位置移到前一列 |
| \f | 换页(FF)，将当前位置移到下页开头 |
| \n | 换行(LF) ，将当前位置移到下一行开头 |
| \r | 回车(CR) ，将当前位置移到本行开头 |
| \t | 水平制表(HT) （跳到下一个TAB位置）向右移四个空格 |
| \v | 垂直制表(VT) |
| \\ | 代表一个反斜线字符''\' |
| \' | 代表一个单引号（撇号）字符 |
| \" | 代表一个双引号字符 |
| \? | 代表一个问号 |
| \0 | 空字符(NUL) |

（17）字符串前**r**的使用（忽略所有转义符）

>>> path='C:\abc\xyz.txt'

SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in position 6-7: truncated \xXX escape

>>> path=***r***'C:\abc\xyz.txt'

>>> path

'C:\\abc\\xyz.txt'

（18）“““ ”””与注释

“““ ”””注释，不会被编译器忽略

# 注释，会被编译器忽略

>>> """这不是一个注释"""

'这不是一个注释'

>>> #这是一个注释

（19）for循环的简单使用

>>> name='aeiou'

>>> for c in name:

print(c)

a

e

i

o

u

>>> for c in name:

print(c,',')

a ,

e ,

i ,

o ,

u ,

>>> for c in name:

print(c,end=',')

a,e,i,o,u,

（20）字符串的简单使用

>>> s='abcdefg'

>>> s[0]

'a'

>>> s[0:4]

'abcd'

>>> s[-1]

'g'

>>> s[len(s)-1]

'g'

>>> s[:]

'abcdefg'

>>> s[::2]

'aceg'

（21）字符串中单个字符的替换

>>> h='hello'

>>> h=h.replace('e','a')

>>> h

'hallo'

（22）字符串和列表的相互转化

>>> s='abcde'

>>> l=***list(s)***

>>> l

['a', 'b', 'c', 'd', 'e']

>>> s=***''.join(l)***

>>> s

>>> s='|'.***join(l)***

>>> s

'a|b|c|d|e'

'abcde'

（23）字符串以某个字符切割为列表

>>> s='abc,def,ghi'

>>> l=s.***split(',')***

>>> l

['abc', 'def', 'ghi']

（24）判断字符串是否为某个字母开头或结尾或包含某个字符

>>> s='abcde'

>>> s.***startswith***('a')

True

>>> s.***endswith***('de')

True

>>> s.***find***('c')

2（所在位置）

（25）交换变量

>>> ***a,b=b,a***

（26）将列表赋值给新列表

>>> l=[1,2,3,4,5]

>>> res=[]

>>> ***for i in l***:（循环遍历操作）

res.***append***(i***\*\****2)

列表追加函数

***\*\****2：平方，\*\*3:三次方

>>> res

[1, 4, 9, 16, 25]

精简版：

>>> l1=[i\*\*3 for i in l]

>>> l1

[1, 8, 27, 64, 125]

（27）列表append函数与extend函数

>>> l

[1, 2, 3, 4, 5]

>>> l.***append***(6) #追加一个字符

>>> l

[1, 2, 3, 4, 5, 6]

>>> l.***extend***([7,8,9]) #追加一组字符

>>> l

[1, 2, 3, 4, 5, 6, 7, 8, 9]

（28）列表的排序

>>> l=[1,6,8,4,3,7,9]

>>> l

[1, 6, 8, 4, 3, 7, 9]

>>> l.***sort***() ***正向排序***

>>> l

[1, 3, 4, 6, 7, 8, 9]

>>> l.***reverse***() ***反向排序***

>>> l

[9, 8, 7, 6, 4, 3, 1]

（29）列表间的赋值

>>> l=[1,2,3,4]

>>> l1=l ***指向同一对象***

>>> l1

[1, 2, 3, 4]

>>> l2=l[:] ***指向不同对象***

>>> l2

[1, 2, 3, 4]

>>> l3=l.copy() ***指向不同对象***

>>> l3

[1, 2, 3, 4]

(30)字典表的两种声明方式

>>> emp=***{'name':'Tom','age':16,'sex':'boy'}***

>>> emp

{'name': 'Tom', 'age': 16, 'sex': 'boy'}

>>> emmp=***dict(name='Tom',age=15,sex='boy')***

>>> emmp

{'name': 'Tom', 'age': 15, 'sex': 'boy'}

（31）字典表的合并（添加）

>>> emp

{'age': 16, 'name': 'Tom'}

>>> elp={'job':'teacher'}

>>> emp.***update***(elp)

>>> emp

{'age': 16, 'name': 'Tom', 'job': 'teacher'}

（32）字典表删除（弹出）某一项

>>> emp

{'age': 16, 'name': 'Tom', 'job': 'teacher'}

>>> emp.***pop***('age')

16

>>> emp

{'name': 'Tom', 'job': 'teacher'}

（33）查看字典表的键和值

>>> emp.***keys***()

dict\_keys(['name', 'job'])

>>> emp.***values***()

dict\_values(['Tom', 'teacher'])

>>> emp.***items***()

dict\_items([('name', 'Tom'), ('job', 'teacher')])

（34）元组的性质

任意对象的有序集合，不可修改，可以通过下标访问，可存储任意类型，可嵌套

***元组的合并***：

>>> 1,2

(1, 2)

>>> x =1,

>>> x

(1,)

>>> (1,2)+(3,4,5)

(1, 2, 3, 4, 5)

***寻找元素所在位置：index（）***

***统计出现几次：count（）***

（35）namedtuple使用方法

>>> ***from collections import namedtuple***

>>> employe=namedtuple('employeer',['name','age'])

>>> Jerry=employe('Tom',16)

>>> Jerry

employeer(name='Tom', age=16)

（36）文件的相关操作

***创建文件对象的函数：open(‘文件名’，‘打开方式’)***

***格式：***f=open（‘C:\Users\duqnq\source\hello.txt’,’***r’***）

逗号前输入地址和文件名，逗号后输入打开方式：r（read，读），w（write，写），a（append，添加），b（bin，二进制），+（添加或删除）

>>> f=open('D:\\Adobe\helloo.txt','w')

>>> f.***write***('你好\n') ***写入操作***

3

>>> f.close()

>>> f=open('D:\Adobe\helloo.txt','r')

>>> f.***read***() ***全部读取***

'你好\n'

>>> f.***readline***() ***逐行读取***

''

>>> l=open('D:\Adobe\helloo.txt','r').***readlines***() ***逐行读取并全部输入在一个列表中***

>>> l

['你好\n']

（37）pickle模块：存取python对象（以二进制方式）

>>> f=open('D:\Adobe\hello.pkl','***wb***') ***以二进制方式写入pkl格式的文件***

>>> l={'name':1,'age':2}

>>> ***import pickle 载入pickle模块***

>>> pickle.***dump***(l,f) ***将字典表l中的元素载入对象文件中***

>>> f.close()

>>> f=open('D:\Adobe\hello.pkl','***rb***') ***以二进制方式打开文件***

>>> f.read()

b'\x80\x03}q\x00(X\x04\x00\x00\x00nameq\x01K\x01X\x03\x00\x00\x00ageq\x02K\x02u.'

>>> f=open('D:\Adobe\hello.pkl','rb')

>>> data=pickle.***load***(f) ***将对象文件中的数据输出***

>>> data['name']

1

>>> data.get('age')

2

（38）浏览磁盘文件

C:\Users\duqnq>D:

D:\>

D:\>cd Adobe

D:\Adobe>

D:\Adobe>dir

驱动器 D 中的卷是 日常文件

卷的序列号是 1049-E498

D:\Adobe 的目录

2020/02/18 13:49 <DIR> .

2020/02/18 13:49 <DIR> ..

2019/10/28 13:12 <DIR> Adobe After Effects CC 2019

2020/02/09 15:49 <DIR> Adobe Illustrator CC 2019

2019/11/20 12:25 <DIR> Adobe Media Encoder CC 2019

2019/10/30 18:32 <DIR> Adobe Photoshop CC 2019

2019/10/28 13:14 <DIR> Adobe Premiere Pro CC 2019

2020/02/18 13:51 33 hello.pkl

1 个文件 33 字节

7 个目录 85,658,996,736 可用字节

（39）字符串的赋值

>>> s='duanqilong'

>>> a,b,c=s[0],s[1],s[2:-1]

>>> c

'anqilon'

>>> a,b,\*c=s

>>> a

'd'

>>> b

'u'

>>> c

['a', 'n', 'q', 'i', 'l', 'o', 'n', 'g']

（40）print函数的使用

>>> s,l='abc','xyz'

>>> print(s,l)

abc xyz

>>> print(s,l,***sep=','***)

abc***,***xyz

>>> print(s,l,***end=',,,'***)

abc xyz***,,,***

>>> print(s,l,***file=open('D:\\result.txt','w')***)

（41）while的使用方法

>>> x='abcd'

>>> ***while x:***

print(x,end=' ')

***x=x[1:]***

abcd bcd cd d

（42）接收用户输入的函数：***input( )***

>>> name=***input***('请输入年龄: ')

请输入年龄: 5

>>> name

'5'

（43）range函数和enumerate函数

***range（）：取固定值范围内所有数***

***enumerate（）：返回字符串内容及所在位置***

>>> for i in ***range(1,10,3):*** ***1：起始 10：终止且不包含 3：间隔***

print(i)

1

4

7

s='duanqilong'

>>> for num,press in ***enumerate(s):***

print('{}) {}'.format(num,press))

0) d

1) u

2) a

3) n

4) q

5) i

6) l

7) o

8) n

9) g

（44）二进制，八进制，十进制，十六进制简称

二进制：bin

八进制：oct

十进制：dec

十六进制：hex