1. Print与四则运算

#print

print('Hello You')

print('Hello'+' World')

print("okok")

print('I\'m ok')

print(222)

print(3,4,5,'hello')

print('hello'+str(6))

#===================

#四则运算

# + - \* / %(取余） //（取整）\*\*(求幂)

print(2+3)

print(2-3)

print(2\*3)

print(2/3)

print(9%4)

print(9//4)

print(3\*\*2)

print(3\*\*3)

1. 变量

print(1)

apple = 3

Apple = 8

print(apple)

print(Apple)

a,b,c = 1,2,3

print(a,b,c)

d = a+b+c

print(d)

# 数据类型转换

str1 = '6'

print(int(str1)\*\*2)

num1 = 7

print('a num is '+str(num1))

#格式

1. while循环

condition =1

while condition < 10:

print(condition)

condition += 1# condition = condition + 1

'''

while True:

do sth.

'''

while True:

print('666')

1. for循环

list = [1,2,3,4,5,6,1,2,3,4,59]

for item in list:

print(item)

print('内部')

print('good time')

'''

python结构

缩进4格

windows:ctrl+[ ]

'''

for i in range(1,10):# range左包含，右不包含

print(i)

for ai in range(1,10,2):# 第三个参数 step

print(ai)

1. if

#====if===========

a =1

b =1

c=3

if a >b:

print('a 大于b')

'''

if True:

do sth.

'''

if a<b<c:

print('a<b<c')

'''

< > <= >=

== !=

'''

if a!=b:

print('a!=b')

#=if=else=============

if a>b:

print('a>b')

else:# 注意缩进

print('a<=b')

#=if=elif=else==========

if a >1:

print('a>1')

elif a==1:

print('a==1')

elif a < 1:# change to 2 讲解elif

print('a<1')

else:

print('else')

1. def函数

#===基本def函数（无参数，无返回值）=======

a = 1

b = 2

c = a+b

print(c)

# 函数定义

def lets\_print():

print('我在打印')

'''

def func\_name(params):

do sth.

return sth.

'''

# 函数调用

lets\_print()

# =def 带参数=============

def lets\_minus(p1,p2):

result = p1-p2

print('lets\_minus: ',result)

lets\_minus(a,b)

lets\_minus(3,9)

lets\_minus(p1=3,p2=9)# 指定参数值,顺序无所谓

r1 = lets\_minus(p2=9,p1=3)

print(r1)

# =def 返回值

def lets\_minus(p1,p2):

result = p1-p2

print('lets\_minus: ',result)

return result

r = lets\_minus(99,3)

print('r:',r)

# ==def 默认值====

def drink(name,time='早上',drinks='水'):

print(name,'在',time,'喝了',drinks)

drink('小明','中午','牛奶')

drink('小明','中午')

drink('小明')

1. 全局、局部变量

# 变量的作用域

# 层级,内部可以访问外部，外部不能访问内部

A = 3

def func():

a = 666

global B# 强制在函数内部定义全局变量

B = 'bbb'

print(A)

func()

print(B)

1. 元组、列表

#=======tuple===list=====

# 元组和列表都是一连串有顺序的数字

#元组tuple

a\_tuple = (2,4,1,565,33)

b\_tuple = 5,7,8,1,1,11,3

a\_list = [3,56,67,1,1,6,6]

#取值 []索引

print(a\_tuple[0])

print(a\_list[1])

a\_list[0] = 4

##a\_tuple[0] = 8 # tuple 不可更改

# 循环输出

for item in a\_tuple:

print(item)

for item in a\_list:

print(item)

for i in range(len(a\_list)):

print(i,a\_list[i])

# 另一种循环输出enumerate

print('====================')

for i,item in enumerate(a\_list):

print(i,item)

#====list列表=================

print('====================')

#append 在末尾添加

print(a\_list)

a\_list.append(999)

print(a\_list)

#insert 在指定位置添加

a\_list.insert(1,666)# 在第1的位置添加666，后面依次顺延

print(a\_list)

#remove 移除

a\_list.remove(1)# 移除第一次出现值为1的数

print(a\_list)

#index索引

print(a\_list[0])

print(a\_list[-1])

print(a\_list[1:4])#[3, 666, 56, 67, 1, 6, 999]

#output? []

print(a\_list[4:])

print(a\_list[-2:])

# 打印索引

print(a\_list.index(666))# 第一次出现666的index是多少

print(a\_list.count(6))# 计算6出现的次数

# 排序

a\_list.sort()

print(a\_list)

a\_list.sort(reverse=True)

print(a\_list)

1. 多维列表

#=====多维列表=======

# numpy更适合多维矩阵的处理

# 1维

a = [1,2,3,4,5]

# 2维 每一个元素都是一个list

b = [[1,2,3],

[5,3,7],

[6,8,9]]

print(a,b)

# 索引

print(b[0][2])

1. 字典

#========字典dict======

# key : value

# 字典无序

d = {'xiaoming':'23','laowang':'45'}

d1 = {1:'a',2:'b','three':'c'}

print(d,d1)

print(d1[1])

print(d1['three'])

#删除

del d1[2]

print(d1)

#增加

d1['Monday']=99

print(d1)

def func():

print(888)

return 8887

#可以包含各种内容

d2 = {

'aa':[1,3,5,3],

'bb':{'qq':999,'ww':666},

'cc':func

}

r = d2['cc']()

print(r)

1. input

#======input===============

a\_input = input('input a number')# 输入会被解释为string

print('Square of your input is',int(a\_input)\*\*2)

1. 文件读写

#===========文件写入==============

text = "good morning.\ngood night.\n"

print(text)

# docs https://docs.python.org/3/

my\_file = open('this file.txt','w')

my\_file.write(text)

my\_file.close()

#===========文件追加写入===========

new\_text = "good tomorrow\n"

my\_file = open('this file.txt','a')

my\_file.write(new\_text)

my\_file.close()

#===========文件读取===============

my\_file = open('this file.txt','r')# 读入的是文件

content = my\_file.read() # 读入文件所有内容

print(type(my\_file))

content\_line = my\_file.readline()# 每调用一次读取一行

content\_lines = my\_file.readlines()# 读取所有内容，按照行放入list中

print(content\_line)

1. 类

#======类基础===========

class Person:# 首字母大写（约定成俗）

# 属性

name = "Xiao Ming"

age = 30

# 方法

def eat(self,food):# 类内方法的第一个参数都会指向当前的类

print(self.name,food)

self.food = food

print('name:',self.name)

# 实例化

a\_person = Person()

print(a\_person.name)

a\_person.eat('apple')

print(a\_person.food)

a\_person.eat('banana')

print(a\_person.food)

1. 类2

#======类基础+构造函数===========

class Person:# 首字母大写（约定成俗）

# 属性(可删)

name = "Xiao Ming"

age = 30

# 方法

def \_\_init\_\_(self,name='Xiao Ming',age):# 实例初始化的时候会运行一次

self.name = name

self.age = age

def eat(self,food):# 类内方法的第一个参数都会指向当前的类

print(self.name,food)

self.food = food

print('name:',self.name)

# 实例化

laowang = Person('LaoWang',20)

print(laowang.name,laowang.age)

1. module安装

# 模块

# 使用 pip install numpy 安装

# pip install -U numpy

1. import

#=====如何import=======

import time

print(time.localtime())

#===== # 自定义名称

import time as t # 自定义名称

print(t.time())

#=====导入指定功能

from time import time,localtime

print(localtime())

print(time())

#====导入所有功能

from time import \*

print(clock())

1. 作业

# 验证码生成器

# 生成200个不重复的6位验证码，由数字组成

# 存储至文件

# 要求+：不止数字，还有字符（ABC...）

==============

# 班级管理系统

'''

功能：

=. 录入。输入学生信息，保存到管理系统

=.\* 删除。删除某个指定学生

=.\* 学生信息修改

=. 查看所有学生

=. 本地保存和读取

=. 面向对象编程

=. 多情况处理

'''