

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)
```

2. 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
```

3. while 循环

```
condition =1
while condition < 10:
    print(condition)
    condition      +=      1#
condition = condition + 1
```

```
'''
```

```
while True:
    do sth.
```

```
'''
```

```
while True:
    print('666')
```

4. 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)
```

5. 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')
```

6. 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')
```

7. 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)
```

8. 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)

```

9. 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])
```

10. 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)

```

11. 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)

```

12. 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)
13. 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)
14. 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 = (ABC...)
Person('LaoWang',20)
print(laowang.name,laowang.age)
# 班级管理系统

15. module 安装
# 模块
# 使用 pip install numpy 安装
# pip install -U numpy

16. 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())

17. 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

```