**Python3自学**

# Python基础语法

## 1.1注释

#!/usr/bin/python3

# 第一个注释

# 第二个注释

'''

第三注释

第四注释

'''

"""

第五注释

第六注释

"""

print ("Hello, Python!")

## 1.2多行语句

Python 通常是一行写完一条语句，但如果语句很长，我们可以使用反斜杠(\)来实现多行语句，例如：

total = item\_one + \

item\_two + \

item\_three

在 [], {}, 或 () 中的多行语句，不需要使用反斜杠(\)，例如：

total = ['item\_one', 'item\_two', 'item\_three',

'item\_four', 'item\_five']

## 1.3数字(Number)类型

python中数字有四种类型：整数、布尔型、浮点数和复数。

int (整数), 如 1, 只有一种整数类型 int，表示为长整型，没有 python2 中的 Long。

bool (布尔), 如 True。

float (浮点数), 如 1.23、3E-2

complex (复数), 如 1 + 2j、 1.1 + 2.2j

## 1.4字符串

#!/usr/bin/python3

str='Runoob' # python中单引号和双引号使用完全相同。

print(str) # 输出字符串

print(str[0:-1]) # 输出第一个到倒数第二个的所有字符，Python 中的字符串有两种索引方式，从左往右以 0 开始，从右往左以 -1 开始。

print(str[0]) # 输出字符串第一个字符

print(str[2:5]) # 输出从第三个开始到第五个的字符

print(str[2:]) # 输出从第三个开始的后的所有字符

print(str \* 2) # 输出字符串两次

print(str + '你好') # 连接字符串

print('------------------------------')

print('hello\nrunoob') # 使用反斜杠(\)+n转义特殊字符

print(r'hello\nrunoob') # 在字符串前面添加一个 r，表示原始字符串，不会发生转义

## 1.5多个语句构成代码组

缩进相同的一组语句构成一个代码块，我们称之代码组。

像if、while、def和class这样的复合语句，首行以关键字开始，以冒号( : )结束，该行之后的一行或多行代码构成代码组。

我们将首行及后面的代码组称为一个子句(clause)。

如下实例：

if expression :

suite

elif expression :

suite

else :

suite

## 1.6Print 输出

print 默认输出是换行的，如果要实现不换行需要在变量末尾加上，

#!/usr/bin/python3

x="a"

y="b"

# 换行输出

print( x )

print( y )

print('---------')

# 不换行输出

print( x, end=" " )

print( y, end=" " )

print()

## 1.7import 与 from...import

在 python 用 import 或者 from...import 来导入相应的模块。

将整个模块(somemodule)导入，格式为： import somemodule

从某个模块中导入某个函数,格式为： from somemodule import somefunction

从某个模块中导入多个函数,格式为： from somemodule import firstfunc, secondfunc, thirdfunc

将某个模块中的全部函数导入，格式为： from somemodule import \*

导入 sys 模块

import sys

print('===============Python mport mode=========================');

print ('命令行参数为:')

for i in sys.argv:

print (i)

print ('\n python 路径为',sys.path)

导入 sys 模块的 argv,path 成员

from sys import argv,path # 导入特定的成员

print('================python from import===================================')

print('path:',path) # 因为已经导入path成员，所以此处引用时不需要加sys.path