# 一 Python简介

## 1 Hello World

在CMD中，输入：python -V 可以查看python版本；

输入 python 即可开启python的命令行

print(‘hello world’)

回车即可实现hello world程序。

当然我们也可以编写一个文件：test.py，将上述内容书写在这个文件内，

在命令行中使用 python test.py 一样可以实现hello world。

## 2 基本语法

### 2.1 注释

|  |
| --- |
| *# 我是单行注释* print(**'hello world'**)  **'''  我是多行注释  多行注释 '''** *#如果直接在程序中用到了中文，可能会报错，解决办法是加入以下注释 #coding=utf-8* print(**'你好'**) |

### 1.2.2 变量



在python中，只要定义了一个变量，而且它有数据，那么它的类型就已经确定了，不需要咱们开发者主动的去说明它的类型，系统会自动辨别

可以使用type(变量的名字)，来查看变量的类型。

|  |
| --- |
| num1 = 24 num2 = 32 print(num1 + num2) print(type(num2)) |

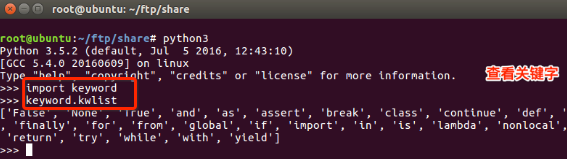
### 2.3 标识符

标识符由字母、下划线和数字组成，且数字不能开头，python中的标识符是区分大小写的。

标识符命名原则：驼峰命名法 myName

python一些具有特殊功能的标识符，这就是所谓的关键字。

可以使用命令查看python的关键字有哪些：



### 2.3 输出

|  |
| --- |
| *#格式化输出* age = 10 name = **'lisi'** print(**'我今年%d岁'**%age) print(**'我叫%s，我今年%d岁'**%(name,age)) *#换行输出* print(**'abcde\nfgh'**) |

常用格式对照表



### 2.4 输入

|  |
| --- |
| name = input(**'请输入你的名字：'**) print(**'你输入的名字是：'**,name) |

##### 2.4.1 python2中的输入

在python2中，输入的函数是 raw\_input()；

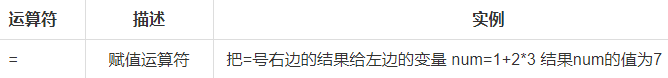
在python2中也有input()函数，用法也是输入，但是这个函数会把输入的一些式子进行解析并计算结果，并不是单纯的只记录内容。

### 2.5 运算符

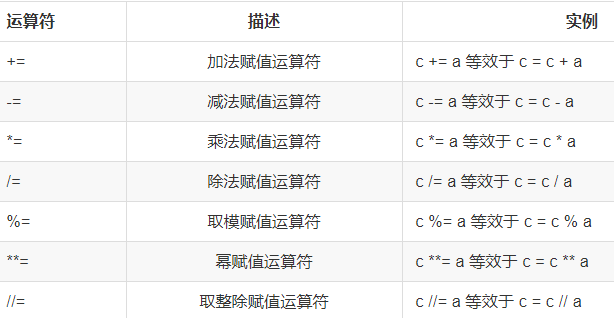
##### 2.5.1 算数运算符



##### 2.5.2 赋值运算符



##### 2.5.3 复合运算符



##### 2.5.4 关系运算符



##### 2.5.5 逻辑运算符



### 2.6 数据类型转换



### 2.7 判断语句

|  |
| --- |
| age = 19 **if** age > 18:  print(**'成年了'**) **else**:  print(**'未成年'**) |

##### 2.7.1 elif

|  |
| --- |
| score = 77 **if** score>=90 **and** score<=100:  print(**'本次考试，等级为A'**) **elif** score>=80 **and** score<90:  print(**'本次考试，等级为B'**) **elif** score>=70 **and** score<80:  print(**'本次考试，等级为C'**) **else**:  print(**'考试成绩不理想'**) |

注意：elif必须和if一起使用，否则出错。

##### 2.7.2 if嵌套



### 2.8 循环语句

##### 2.8.1 while循环

while循环嵌套的就乘法表

|  |
| --- |
| i = 1 **while** i <= 9:  j = 1  **while** j <= i:  print(**"%d\*%d=%-2d "** % (j, i, i \* j), end=**''**)  j += 1  print(**'\n'**)  i += 1 |

##### 2.8.2 for循环

|  |
| --- |
| name = **'' for** x **in** name:  print(x) **else**:  print(**"没有数据"**) |

##### 2.8.3 break与contune

break/continue只能用在循环中，除此以外不能单独使用

break/continue在嵌套循环中，只对最近的一层循环起作用

|  |
| --- |
| i = 0 **while** i < 10:  i = i + 1  print(**'----'**)  **if** i == 5:  **continue** print(i) |

|  |
| --- |
| i = 0 **while** i < 10:  i = i + 1  print(**'----'**)  **if** i == 5:  **break** print(i) |

### 2.9 字符串

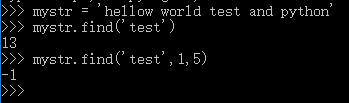
##### 2.9.1 下标与切片

|  |
| --- |
| *#[起始位置:结束位置:步长] 没有给出值默认从0开始* name = **'abcdefg'** print(name[0]) print(name[0:3]) *#切片：取下标0-2的字符* print(name[2:]) *#取下标2-最后的字符* print(name[1:-1]) *# 取下标为1开始 到 最后第2个之间的字符:bcdef* print(name[:3]) *#abc* print(name[:]) *#abcdefg* print(name[::]) *#abcdefg* print(name[::3]) *#adg---每次取值的时候跳跃3个长度 adg* print(name[5:1:2]) *#输出空白* print(name[::-2]) *#geca* print(name[5:1:-2]) *#fd #反转字符串* print(name[::-1]) |

##### 2.9.2 常见操作

find： 检测str是否在字符串中，在--返回开始的索引，不在--返回-1

mystr.find(str, start=0, end=len(mystr))



同理 rfind 从右边查找

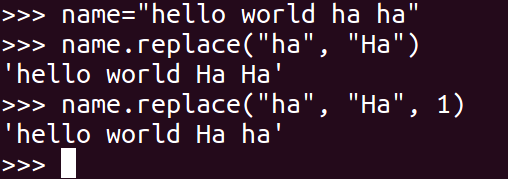
index: 与find一样，不过找不到会报异常；同理rindex

count： str在start和end之间，出现的次数

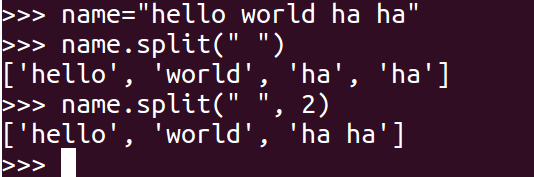
mystr.count(str, start=0, end=len(mystr))

replace： str1替换str2，如果count置顶，则不允许替换超过count次

mystr.replace(str1, str2, mystr.count(str1))



split： 以 str 为分隔符切片 mystr，如果 maxsplit有指定值，则仅分隔 maxsplit 个子字符串



capitalize：把字符串的第一个字符大写

mystr.capitalize()

title：把字符串的每个单词首字母大写

>>> a = "hello itcast"

>>> a.title()

'Hello Itcast'

startswith：检查字符串是否是以 obj 开头, 是则返回 True，否则返回 False

mystr.startswith(obj)

同理还有： endwith

lower：转换 mystr 中所有大写字符为小写

mystr.lower()

同理还有：upper

ljust：返回一个原字符串左对齐,并使用空格填充至长度 width 的新字符串

mystr.ljust(width)

同理还有右对齐 rjust

center：返回一个原字符串居中,并使用空格填充至长度 width 的新字符串

mystr.center(width)

lstrip：删除 mystr 左边的空白字符

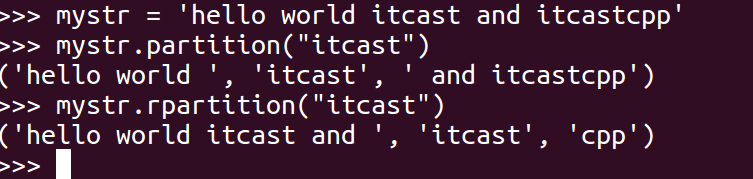
mystr.lstrip()

同理还有rstrip strip（删除两边空白字符）

partition：把mystr以str分割成三部分,str前，str和str后

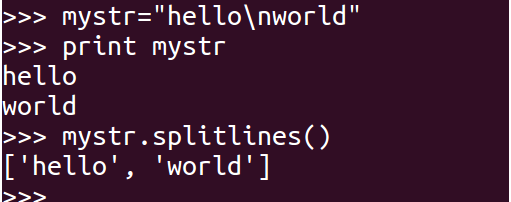
mystr.partition(str)

同理 rpartition，从右边开始



splitlines：按照行分隔，返回一个包含各行作为元素的列表

mystr.splitlines()



isalpha：如果 mystr 所有字符都是字母 则返回 True,否则返回 False

mystr.isalpha()

isdigit：如果 mystr 只包含数字则返回 True 否则返回 False

mystr.isdigit()

isalnum：如果 mystr 所有字符都是字母或数字则返回 True,否则返回 False

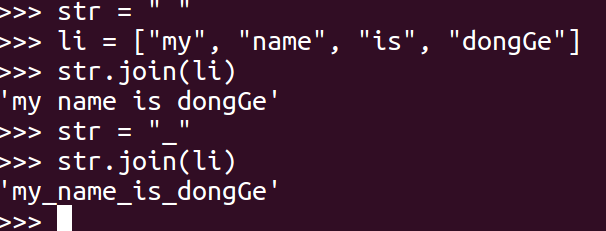
mystr.isalnum()

isspace：如果 mystr 中只包含空格，则返回 True，否则返回 False.

mystr.isspace()

join：mystr 中每个字符后面插入str,构造出一个新的字符串

mystr.join(str)



### 2.10 列表

##### 2.10.1 简介

|  |
| --- |
| *#和C语言不通的是列表可以存储不同类型元素* testList = [1, **'a'**] print(testList[0])  length = len(testList)  *#while遍历* i = 0 **while** i < length:  print(testList[i])  i += 1 |

##### 2.10.2 常见操作

|  |
| --- |
| testList = [**'a'**,**'b'**,**'c'**] *#添加元素-append* testList.append(**'ww'**) print(testList)  *#添加元素-extend 可以将另一个集合中的元素逐一添加* testList1 = [1,2,3] testList2 = [**'zs'**,**'ls'**] testList1.extend(testList2) print(testList1) *#使用append后则是整个列表加入  #插入元素-insert* testList.insert(1,5) print(testList)  *#修改元素-* testList[1] = **'z'** print(testList)  *#查找元素-in not in* **if 'z' in** testList:  print(**'true'**)  *#查找元素-index* listA = [**'a'**,**'b'**,**'c'**,**'d'**,**'e'**] listA.index(**'a'**,0,3) *#注意左闭右开区间 这里会报一个异常  #查找元素-count 返回这个元素的数目* listA.count(**'e'**)  *#删除元素-* **del** listA[1] *#del 根据下标进行删除* listA.pop() *#pop 删除最后一个元素* listA.remove(**'c'**) *#remove 根据元素的值进行删除  #排序-* listB = [1,2,5,3,4] listB.sort() *#sort方法是将list按特定顺序重新排列，默认为由小到大* print(listB) listB.sort(reverse=**True**) *# 参数reverse=True可改为倒序，由大到小。* print(listB) *#排序-reverse方法是将list逆置* |

##### 2.10.3 列表嵌套

schoolNames = [[**'北京大学'**,**'清华大学'**],  
 [**'南开大学'**,**'天津大学'**,**'天津师范大学'**],  
 [**'山东大学'**,**'中国海洋大学'**]]

### 2.11 元组

Python的元组与列表类似，不同之处在于元组的元素不能修改。元组使用小括号，列表使用方括号。

|  |
| --- |
| tuple = (**'hi'**,**'1'**,5,5) print(tuple[2]) *#不能修改、删除元素* tuple.index(**'1'**,1,4) tuple.count(5) |

### 2.12 字典

##### 2.12.1 简介

|  |
| --- |
| info = {**'name'**:**'班长'**, **'id'**:100, **'sex'**:**'f'**, **'address'**:**'地球亚洲中国北京'**} print(info[**'name'**]) *#如果键不存在，则报错 #不确定字典是否存在某个键，又想获取值，可以用get方法* age = info.get(**'age'**) *#age键不存在，所以返回None* age = info.get(**'age'**,18) *#如果info中没有age这个键，则返回默认值18* |

字典和列表一样，也能够存储多个数据

列表中找某个元素时，是根据下标进行的

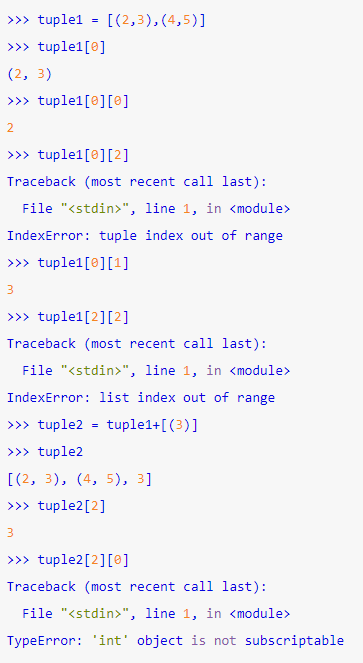
字典中找某个元素时，是根据'名字'（就是冒号:前面的那个值，例如上面代码中的'name'、'id'、'sex'）

字典的每个元素由2部分组成，键:值。例如 'name':'班长' ,'name'为键，'班长'为值

##### 2.12.2 常见操作

|  |
| --- |
| info = {**'name'**: **'班长'**, **'id'**: 100, **'sex'**: **'f'**, **'address'**: **'地球亚洲中国北京'**} *#修改元素* info[**'id'**] = 111 *#添加元素* info[**'age'**] = 30 *#如果age不存在，则是添加元素 #删除元素* **del** info[**'name'**] *# del info 删除整个字典 # info.clear() 清空整个字典* len(info) *#返回字典键值对个数* info.keys() *#返回一个包含字典所有key的列表* info.values() *#返回一个包含字典所有value的列表* info.items() *#返回一个包含所有键、值的元组* info.has\_key(**'id'**)*#返回True或False* |

### 2.13多为列表、元组



## 3 可变类型与不可变类型

可变类型，值可以改变：

列表 list

字典 dict

不可变类型，值不可以改变：

数值类型 int, long, bool, float

字符串 str

元组 tuple