

关键字

逻辑关系

- and: 与操作
- not: 非操作
- or: 或操作

引用库函数

- from
- import

条件判断和循环

- if else elif: 条件判断
- while: 条件成立时, 执行循环
- for: 遍历
- in: 经常用在for语句中, 执行循环。in也可以用来判断数字是否在数组中。如: 1 in range(10), 返回值为true。
- break: 中断全部循环
- continue: 跳出本次循环

错误调试

- assert: 断言, assert之后的布尔表达式的值必须为真, 否则报错
- try except finally: try后执行程序, 报错执行except

In [1]:

```
try:
    a = 1/2
    print(a)
    b = 1/0
    print(b)
    print(c)
except ZeroDivisionError:
    print("math error")
except NameError:
    print("wrong name")
finally:
    print("successful")
```

0.5

math error

successful

- with as:可以用来替换try except finally。参考：<https://blog.csdn.net/qigicos/article/details/79200089> (<https://blog.csdn.net/qigicos/article/details/79200089>) 和 <https://www.ibm.com/developerworks/cn/opensource/os-cn-pythonwith/> (<https://www.ibm.com/developerworks/cn/opensource/os-cn-pythonwith/>)
- raise: 引发自定的异常（错误），之后的代码不再执行。参考：<http://www.runoob.com/python/python-exceptions.html> (<http://www.runoob.com/python/python-exceptions.html>)
- as: 重命名，import as 和 with as

函数定义

- def: 定义函数
- return: 返回某个值

其他关键字

- is: 比较的是两个实例对象是不是完全相同，它们是不是同一个对象，占用的内存地址是否相同。
- del: 删除变量，解除引用。参考：<https://blog.csdn.net/windsccloud/article/details/79732014> (<https://blog.csdn.net/windsccloud/article/details/79732014>)
- print: 打印
- class: 定义类
- global: 定义全局变量。函数中的变量一般是局部变量，但是使用global可以在函数中定义全局变量。

- pass: 占位符，在程序不完整时，用占位符替代，程序可以顺利执行，不会报错。
- yield: 生成器，用于迭代功能，但是与一般数组不同，它只能迭代一次，下次迭代将寻找下一个 yield，所以它占的内存比较小。参考资料：1.<https://www.cnblogs.com/maoxiaolv/p/6425875.html>
(<https://www.cnblogs.com/maoxiaolv/p/6425875.html>)
2.<https://www.ibm.com/developerworks/cn/opensource/os-cn-python-yield/>
3.<https://blog.csdn.net/u013205877/article/details/70332612>

In [2]:

```
def addlist(alist):  
    for i in alist:  
        yield i + 1  
  
alist = [1, 2, 3, 4]  
for x in addlist(alist):  
    print(x)
```

2
3
4
5

- exec: 执行储存在字符串或文件中的 Python 语句，，也就是把字符串当做代码执行，相比于 eval，exec可以执行更复杂的 Python 代码。
- lambda: 匿名函数。用法：“lambda 参数: 函数表达式”

数据类型

针对每一种数据类型，都举出一些例子来，例如，针对string，你可以举出一些字符串。

True: 真值。1==1

False: 假。1 == 2

None: 和任何数据类型比较永远返回False。

string: 字符串。任何输入的数据都是字符串，如“哈哈哈哈哈”。

number: 数字。

floats:浮点型。

lists: 列表，用 [] 表示。

字符串转义序列

转义序列	说明
\\	反斜杠
\'	单引号
\"	双引号
\a	响铃(alarm)
\b	退格 (backspace)
\f	换页
\n	换行
\r	回车
\t	水平制表符
\v	垂直制表符

字符串格式化	说明
%d	十进制整型数字
%i	十进制整型数字（两者基本没有区别）
%o	无符号整数（八进制）
%u	无符号整数（十进制）
%x	无符号整数（十六进制）
%X	无符号整数（十六进制大写字符）
%e	浮点数字（科学计数法）
%E	浮点数字（科学计数法，用E替代e）
%f	浮点数
%g	动调整将整数、浮点数转换成 浮点型或科学计数法表示（超过6位数用科学计数法e）
%G	动调整将整数、浮点数转换成 浮点型或科学计数法表示（超过6位数用科学计数法E）
%c	字符及ASCII码
%r	原始数据
%s	字符串
%%	%号

操作符

符号	操作
+	加号
-	减号
*	乘号
**	指数
/	除号
//	取整除
%	取余数
>	大于
<	小于
>=	大于或等于
<=	小于或等于
==	等于
!=	不等于
<>	不等于
@	装饰器，将下面一个函数的返回值作为变量传递给@之后的函数。
.	操作符
+=	$a+=b$ 等效于 $a=a+b$
-=	$a-=b$ 等效于 $a=a-b$
* =	$a=b$ 等效于 $a=ab$
/=	$a/=b$ 等效于 $a=a/b$

- @的用法详解：https://blog.csdn.net/tyhj_sf/article/details/77417455 (https://blog.csdn.net/tyhj_sf/article/details/77417455) 和 <https://blog.csdn.net/xiangxianghehe/article/details/77170585> (<https://blog.csdn.net/xiangxianghehe/article/details/77170585>)

In [3]:

```
a=1
b=2
a+=b
print(a,b)
```

3 2

python 数据类型

<http://www.runoob.com/python3/python3-data-type.html> (<http://www.runoob.com/python3/python3-data-type.html>)

数据类型	创建方法	特点
数字	直接创建	各类数字
字符串	直接创建	元素不可改变；两种索引方式，从左往右以0开始，从右往左以-1开始
列表	a=[1, 2, 3]或list(空或迭代对象)	元素可变、可修改
元组	a=(1,2,3)	与列表类似，但是元素不可改变
集合	a={1,2,3}或a=set(1,2,3) ,空集必须用set () 创建	重复的集合元素会自动去除，集合之间可进行运算
字典	a={key1:value1,key2:value2}或者 a=dict([(key,value),(key,value)])	一个键对应一个键值，键必须是唯一的