1. **Hello World**

1、hello world

|  |
| --- |
| print("hello world!");  print"Hello world！"; |

1. 标识符

① 标识符中只能有字母、数字、下划线。但是不能以数字开头。区分大 小写。

② 以下划线开头是有特殊意义的：

> 以单下划线开头 (\_foo) 的代表不能直接访问的类属性，需通过类 提供的接口进行访问，不能用 “from xxx import” 而导入。

> 以双下划线开头的 (\_\_foo) 代表类的私有成员。

> 以双下划线开头和结尾的 (\_\_foo\_\_) 代表 python 里特殊方法专

用的标识，如：\_\_init\_\_() 代表类的构造函数。

1. 行和缩进

python 的代码块不适用大括号 ( {} ) 来控制类，函数以及其他逻辑判

断。使用缩进来写模块。

**Python 的所有代码块语句必须包含相同的缩进空白数量。**

|  |
| --- |
| if 9<5:  print "it is true!";  else:  Print "it is wrong!"; |

1. 多行语句

Python 使用斜杠（ \ ）将一行的语句分为多行显示。

|  |
| --- |
| print "abcdefg"+ \  "hijklmn!" |

Python 语句中包含 [] 或者 {} 或者 () 括号不需要使用多行连接符。

|  |
| --- |
| days = ['Monday', 'Tuesday',  'Wednesday','Thursday', 'Friday']  print days; |

1. Python 引导

Python 接受 单引号’’、双引号””、三引号””” 来表示字符串。

其中三引号可以由多行组成，编写多行文本的快捷语法。

|  |
| --- |
| print '这是单引号'  print "这是双引号"  print """第一行  第二行  第三行  这是三引号，可以包含多行""" |

1. 注释、

单行注释：#

多行注释：

三个单引号 ‘’’ 注释内容 ’’’ 或者 三个双引号 “”” 多行注释”””

|  |
| --- |
| '''  这是多行注释，使用单引号。  这是多行注释，使用单引号。  这是多行注释，使用单引号。  '''  """  这是多行注释，使用双引号。  这是多行注释，使用双引号。  这是多行注释，使用双引号。  """ |

1. Python 空行：空行不是语法的一部分，是程序代码的一部分。
2. 等待用户输入

|  |
| --- |
| raw\_input("\n\n\nplease enter key to exit!") |

当用户按下 enter 时，程序将退出！

1. 同一行显示多条语句

使用分号 ; 分割开。

1. **变量**

① Python 变量不需要类型声明。

② 每个变量在创建过程中都包括变量的标示、名称和数据这些信息。

③ 每个变量在使用前都必须赋值，变量赋值以后该变量才会被创建。

④ 使用等于 = 为变量赋值。

|  |  |
| --- | --- |
| #encoding=utf-8  age = 20 #赋值整型变量  height = 173.5;#浮点型  name = "wenka" #字符串  isBoy = True #布尔类型  print age;  print height;  print name;  print isBoy; | 20  173.5  wenka  True |

1. 标准数据类型

Number（数字）

String （字符串）

List （列表）

Tuple （元组）

Dictionary （字典）

1. **Python 数字：**是不可改变的数据类型。意味着改变数字数据类型会分配 一个新的对象。

> int：有符号整型

> long：长整型（也可以代表八进制和十六进制）

> float：浮点型

> complex：复数：可以用a + bj,或者complex(a,b)表示， 复数的实部a

和虚部b都是浮点型

del 语句删除一些对象的引用。

1. **Python 字符串：**是由数字、字母下划线组成的一串字符。

取值方式：

> 从左到右取值：索引从0开始，最大范围为字符串长度-1

> 从右到左取值：索引从-1开始，最大范围为字符串开头

截取字符串：

|  |
| --- |
| s = "i love python";  print s[-8] #取从右到左的第八个字符  print s[2:6]; #取从左到右 [2，6）之间的字符  print s[2:]; #取小标为2之后的所有字符，包括2  print s \* 3; # s \* N :代表重复输出N次  print s + " this is my python"; # + 拼接字符串 |
| **结果：**  e  love  love python  i love pythoni love pythoni love python  i love python this is my python |

1. **Python 列表**

支持字符，数字，字符串甚至包含列表（嵌套）

列表用 [] 标示。

+ 连接运算符 \* 重复操作

|  |
| --- |
| list = [ 'runoob', 786 , 2.23, 'john', 70.2 ]  tinylist = [123, 'john']  print list # 输出完整列表  print list[0] # 输出列表的第一个元素  print list[1:3] # 输出第二个至第三个的元素  print list[2:] # 输出从第三个开始至列表末尾的所有元素  print tinylist \* 2 # 输出列表两次  print list + tinylist # 打印组合的列表 |
| **结果:**  ['runoob', 786, 2.23, 'john', 70.2]  runoob  [786, 2.23]  [2.23, 'john', 70.2]  [123, 'john', 123, 'john']  ['runoob', 786, 2.23, 'john', 70.2, 123, 'john'] |

1. **元组**

类似于 List，用 () 标示。

|  |
| --- |
| tuple = ( 'runoob', 786 , 2.23, 'john', 70.2 )  tinytuple = (123, 'john')  print tuple # 输出完整元组  print tuple[0] # 输出元组的第一个元素  print tuple[1:3] # 输出第二个至第三个的元素  print tuple[2:] # 输出从第三个开始至列表末尾的所有元素  print tinytuple \* 2 # 输出元组两次  print tuple + tinytuple # 打印组合的元组 |
| **结果:**  ('runoob', 786, 2.23, 'john', 70.2)  runoob  (786, 2.23)  (2.23, 'john', 70.2)  (123, 'john', 123, 'john')  ('runoob', 786, 2.23, 'john', 70.2, 123, 'john') |

**注意：元组是不允许更新的，而列表是允许更新的**

1. **Python 元字典**

特点：无序、通过键取值、用 {} 标示。

|  |
| --- |
| dict = {}  dict['name'] = "wenka"  dict['age'] = 20  print dict; |
| {'age': 20, 'name': 'wenka'} |

输入所有key

|  |
| --- |
| print dict.keys() # 输出所有 key  print dict.values() # 输出所有 values |

1. Python 所有数据转换

|  |  |
| --- | --- |
| **函数** | **描述** |
| int(x [,base]) | 将x转换为一个整数 |
| long(x [,base] ) | 将x转换为一个长整数 |
| float(x) | 将x转换到一个浮点数 |
| complex(real [,imag]) | 创建一个复数 |
| str(x) | 将对象 x 转换为字符串 |
| repr(x) | 将对象 x 转换为表达式字符串 |
| eval(str) | 用来计算在字符串中的有效Python表达式,并返回一个对象 |
| tuple(s) | 将序列 s 转换为一个元组 |
| list(s) | 将序列 s 转换为一个列表 |
| set(s) | 转换为可变集合 |
| dict(d) | 创建一个字典。d 必须是一个序列 (key,value)元组。 |
| frozenset(s) | 转换为不可变集合 |
| chr(x) | 将一个整数转换为一个字符 |
| unichr(x) | 将一个整数转换为Unicode字符 |
| ord(x) | 将一个字符转换为它的整数值 |
| hex(x) | 将一个整数转换为一个十六进制字符串 |
| oct(x) | 将一个整数转换为一个八进制字符串 |

1. **运算符**

**1、算数运算符**

加（+）、减（-）、乘（\*）、除（/）、取余（%）、

求幂（\*\* [2\*\*3=8] ）、求商（//[9//2=4]）

|  |  |
| --- | --- |
| # 操作符  a = 2.0  b = 3.0;  print "2+3 =",a+b  print "2-3 =",a-b  print "2\*3 =",a\*b  print "2/3 =",a/b  print "2%3 =",a%b  print "2\*\*3 =",a\*\*b  print "2//3 =",a//b | 2+3 = 5.0  2-3 = -1.0  2\*3 = 6.0  2/3 = 0.666666666667  2%3 = 2.0  2\*\*3 = 8.0  2//3 = 0.0 |

**2、比较运算符**

相等（==）、不相等（!=）、是否不相等（<>）

大于（>）、小于（<）、大于等于（>=）、小于等于（<=）

所有的比较运算符返回 1 表示真，返回 0 表示假。

|  |  |
| --- | --- |
| # 比较运算符  print "a == b",a==b  print "a <> b",a<>b  if 0:  print True  else:  print False | a == b False  a <> b True  False |

**3、赋值运算符**

= 、 +=、-=、\*=、/=、%=、\*\*=、//=

**4、位运算符**

|  |  |  |
| --- | --- | --- |
| **符** | **描述** |  |
| & | 按位与运算符：参与运算的两个值,如果两个相应位都为1,则该位的结果为1,否则为0 |  |
| | | 按位或运算符：只要对应的二个二进位有一个为1时，结果位就为1。 |  |
| ^ | 按位异或运算符：当两对应的二进位相异时，结果为1 |  |
| ~ | 按位取反运算符：对数据的每个二进制位取反,即把1变为0,把0变为1 |  |
| << | 左移动运算符：运算数的各二进位全部左移若干位，由"<<"右边的数指定移动的位数，高位丢弃，低位补0。 |  |
| >> | 右移动运算符：把">>"左边的运算数的各二进位全部右移若干位，">>"右边的数指定移动的位数 |  |

**5、逻辑运算符**

and ：布尔“与”

or ：布尔“或”

not ：布尔“非”

|  |  |
| --- | --- |
| # 逻辑运算符  a = 2.0  b = 3.0  print a and b  print a or b  print not a  print not b | 3.0  2.0  False  False |

6、成员运算符

in ：如果在指定的序列中找到值返回 true，否则返回 false。

not in ：如果在指定的序列中没有找到值返回 true，否则返回false。

7、身份运算符

is ：判断两个标示符是不是引用同一个对象。

is not ：判断两个标示符是不是引用不对的对象。

1. **控制流程语句**

**1、条件语句**

|  |  |
| --- | --- |
| if 判断条件：  执行语句……  else：  执行语句…… | if 判断条件1:  执行语句1……  elif 判断条件2:  执行语句2……  elif 判断条件3:  执行语句3……  else:  执行语句4…… |

**2、循环语句**

① while 循环

|  |  |
| --- | --- |
| while 判断条件:  循环体… | count = 0;  while (count < 10):  print "循环：",count  count += 1  print "End..." |

② for 循环

|  |
| --- |
| for iterating\_var in sequence:  statements(s) |

例子：

|  |
| --- |
| for word in "hello world":  print word  print "End..."  print "-------------------------------------"  fruits = ["banana" , "apple" , "mango"]  for fruit in fruits:  print fruit  print "End..."  print "-------------------------------------"  for num in range(10,20): # 迭代 10 到 20 之间的数字  for i in range(2,num): # 根据因子迭代  if num%i == 0: # 确定第一个因子  j=num/i # 计算第二个因子  print '%d = %d \* %d' % (num,i,j)  break # 跳出当前循环  else: # 循环的 else 部分  print num, '是一个质数' |

③ 嵌套循环

|  |
| --- |
| #循环输出2~100之间的素数：  print range(5,6)  for num in range(2,100):  for i in range(2,num):  if num % i == 0:  break  else:  print num,"是素数！" |

④ break语句

1、break语句用来终止循环语句，即循环条件没有False条件或者序列还

没被完全递归完，也会停止执行循环语句。

2、break语句用在while和for循环中。

1. 如果您使用嵌套循环，**break语句将停止执行最深层的循环**，并开始执行下一行代码。若存在 else 语句，则停止执行最深层的循环后执行 else 语句。

⑤ continue语句

1、Python continue 语句跳出本次循环，而break跳出整个循环。

2、continue 语句用来告诉Python跳过当前循环的剩余语句，然后继续

进行下一轮循环。

⑥ pass 语句

1、Python pass是空语句，是为了保持程序结构的完整性。

2、pass 不做任何事情，一般用做占位语句。

1. **Python 日期和时间**

1、Python 提供了一个 time 和 calendar 模块可以用于格式化日期和时间。

2、时间间隔是以秒为单位的浮点小数。

3、每个时间戳都以自从1970年1月1日午夜（历元）经过了多长时间来表示。

4、import time; # 引入time模块

|  |  |
| --- | --- |
| import time; #引入 time 模块  print "当前时间为：".decode("utf-8"),time.time()  localTime = time.localtime(time.time())  print "本地时间为：",localTime | 当前时间为： 1482224204.4  本地时间为： time.struct\_time(tm\_year=2016, tm\_mon=12, tm\_mday=20, tm\_hour=16, tm\_min=56, tm\_sec=44, tm\_wday=1, tm\_yday=355, tm\_isdst=0) |

4、获取格式化时间asctime():

|  |  |
| --- | --- |
| t = time.asctime(localTime)  print "格式化后的时间为：",t | 格式化后的时间为： Tue Dec 20 17:00:03 2016 |

5、格式化日期 time.strftime(format[, t])

|  |
| --- |
| print time.strftime("%Y 年 %m 月 %d 日 %H:%M:%S",localTime) |
| 2016 年 12 月 20 日 17:06:45 |

python中时间日期格式化符号：

%y 两位数的年份表示（00-99）

%Y 四位数的年份表示（000-9999）

%m 月份（01-12）

%d 月内中的一天（0-31）

%H 24小时制小时数（0-23）

%I 12小时制小时数（01-12）

%M 分钟数（00=59）

%S 秒（00-59）

%a 本地简化星期名称

%A 本地完整星期名称

%b 本地简化的月份名称

%B 本地完整的月份名称

%c 本地相应的日期表示和时间表示

%j 年内的一天（001-366）

%p 本地A.M.或P.M.的等价符

%U 一年中的星期数（00-53）星期天为星期的开始

%w 星期（0-6），星期天为星期的开始

%W 一年中的星期数（00-53）星期一为星期的开始

%x 本地相应的日期表示

%X 本地相应的时间表示

%Z 当前时区的名称

%% %号本身

5、Calendar：import calendar

函数：

http://www.runoob.com/python/python-date-time.html

calendar.month(年份,月份)：获取某一月

|  |  |
| --- | --- |
| #encoding:utf-8  import calendar  print calendar.month(2016,12) | December 2016  Mo Tu We Th Fr Sa Su  1 2 3 4  5 6 7 8 9 10 11  12 13 14 15 16 17 18  19 20 21 22 23 24 25  26 27 28 29 30 31 |

**六、Python 函数**

函数的定义：

1、函数代码块以 **def** 关键字开头，后接函数的标示符名称和圆括号()。

2、任何传入参数和自变量必须放在圆括号中间。圆括号之间可以用来定义

参数。

3、函数的第一行语句可以选择性地使用文档字符 ---> 用于存放函数说明。

4、函数以冒号起始，并且缩进。

5、return [表达式] 结束函数，选择性返回一个值。不带表达式的 return

相当于返回None。

|  |
| --- |
| def functionname( parameters ):  "函数\_文档字符串"  function\_suite  return [expression] |

匿名函数：

1、lambda 只是一个表达式，函数体比 def 简单很多。

2、lambda 的主体是一个表达式，而不是一个代码块。

3、lambda 函数拥有自己的命名空间，且不能访问自有参数列表之外或全局

命名空间里的参数。

4、调用小函数时不占用栈内存从而增加运行效率。

|  |
| --- |
| lambda [arg1 [,arg2,.....argn]]:expression |

**七、**