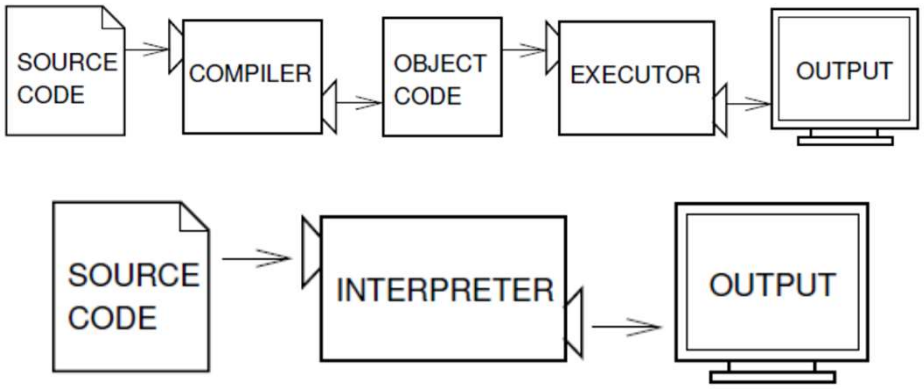
第1章　基础知识

1.0 Python是一种怎样的语言？(1/2)

- 跨平台、开源、免费的解释型高级动态编程语言



1.2 Python简单使用(5/5)

快捷键	功能说明
Alt+p	浏览历史命令（上一条）
Alt+n	浏览历史命令（下一条）
Ctrl+F6	重启Shell，之前的对象和模块全部失效
F1	打开Python帮助文档
Alt+/	自动补全前面曾经出现过的单词，如果之前有多个单词具有相同前缀，则在多个单词中循环选择
Ctrl+]	缩进代码块
Ctrl+[	取消代码块缩进
Alt+3	注释代码块
Alt+4	取消代码块注释。

1.4 Python基础知识

- 对象模型
- 变量
- 数字
- 字符串
- 运算符与表达式
- 内置函数
- 对象删除
- 输入输出
- 模块

1.4.1 Python的对象模型(1/2)

- 对象：python中处理的每样“东西”都是对象
- 内置对象：可直接使用

  如 数字、字符串、列表、del等；
- 非内置对象：需要导入模块才能使用

  如 sin(x)，random( )等。

1.4.1 Python的对象模型(2/2)

- 常用内置对象

对象类型	示例
数字	1234，3.14，3+4j
字符串	'swfu', "I'm student", '"Python "'
列表	[1, 2, 3]
字典	{1:'food',2:'taste',3:'import'}
元组	(2, -5, 6)
文件	f=open('data.dat', 'r')
集合	set('abc'), {'a', 'b', 'c'}
布尔型	True, False
空类型	None
编程单元类型	函数、模块、类

1.4.2 Python变量(2/12)

Python属于强类型编程语言

解释器会根据赋值或运算来自动推断变量类型

- 不同类型支持的运算也不完全一样
- 使用变量时需要程序员自己确定所进行的运算是否合适
- 同一个运算符对不同类型数据操作的含义和结果也不一样

1.4.2 Python变量(5/12)

- 创建变量或修改变量的值：

  变量出现在（复合）赋值运算符或赋值运算符的左边

- 引用该变量的值：其他情况。

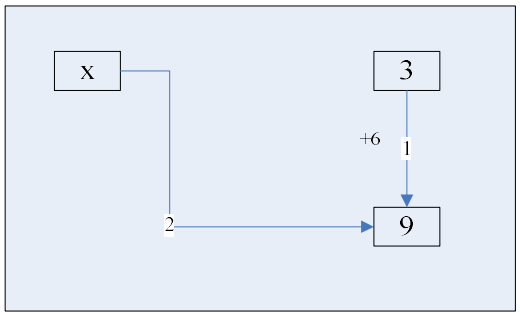
（适用于下标访问列表、字典等可变序列及自定义对象）

1.4.2 Python变量(9/12)

- Python采用基于值的内存管理方式

```
>>> x = 3
```


```
>>> x += 6
```



1.4.2 Python变量(12/12)

- 变量名必须以字母或下划线开头
- 变量名中不能有空格以及标点符号
- 不能使用关键字作变量名

```
>>> import keyword
>>> keyword.kwlist
```

- 不使用系统内置的模块名、类型名或函数名以及已导入的模块名及其成员名作变量名

```
>>> dir(__builtins__)
```

- 变量名对英文字母的大小写敏感

1.4.3 数字(1/4)

- 数字：python中最常用的不可变对象
- 可以表示任意大小的数字

```
>>> a=999999999999999999999999999999
>>> a*a
99999999999999999999999999999800000000000000
0000000000000001L
>>> a**3
99999999999999999999999999999700000000000000
00000000000000029999999999999999999999999999
99L
```

- Python的IDEL交互界面可以当做简便计算器来使用

1.4.3 数字(2/4)

- 十进制整数

  0、-1、9、123

- 十六进制整数，必须以0x开头

16个数字0、1、2、3、4、5、6、7、8、9、a、b、c、d、e、f来表示整数:　0x10、0xfa、0xabcdef

- 八进制整数，，必须以0o开头

需要8个数字0、1、2、3、4、5、6、7来表示整数

  0o35、0o11

- 二进制整数、必须以0b开头

需要2个数字0、1来表示整数：0b101、0b100

1.4.3 数字(3/4)

- 浮点数(小数)

  15.0、0.37、-11.2、1.2e2、314.15e-2

1.4.3 数字(4/4)

```
>>> a = 3+4j
>>> b = 5+6j
>>> c = a+b
>>> c
(8+10j)
>>> c.real #查看复数实部
8.0
>>> c.imag #查看复数虚部
10.0
>>> a.conjugate() #返回共轭复数
(3-4j)
>>> a*b #复数乘法
(-9+38j)
>>> a/b #复数除法
(0.6393442622950819+0.03278688524590165j)
```

1.4.4 字符串(1/3)

- 用单引号、双引号或三引号括起来的符号系列称为字符串
- 单引号、双引号、三单引号、三双引号可以互相嵌套，用来表示复杂字符串。

  'abc'、'123'、'中国'、"Python"
- 字符串属于不可变序列
- 空串表示为''或 ""
- 三引号'''或"""表示的字符串可以换行
- 三引号'''或"""可以在程序中表示较长的注释

1.4.4 字符串(3/3)

- 转义字符

\n：换行符
\t：制表符
\r：回车
\’：单引号
\”：双引号
\\：一个\
\ddd：3位八进制数对应的字符
\xhh：2位十六进制数对应的字符

* 字符串界定符前面加字母r表示原始字符串
* 字符串的最后一个字符不能是\

1.4.5 操作符和表达式(1/6)

运算符示例	功能说明
x+y	算术加法，列表、元组、字符串合并
x-y	算术减法，集合差集
x*y	乘法，序列重复
x/y	除法（在Python 3.x中叫做真除法）
x//y	求整商
-x	相反数
x%y	余数（对实数也可以进行余数运算），字符串格式化
x**y	幂运算
x<y; x<=y; x>y; x>=y	大小比较（可以连用），集合的包含关系比较
x==y; x!=y	相等（值）比较，不等（值）比较
x or y	逻辑或（只有x为假才会计算y）
x and y	逻辑与(只有x为真才会计算y)
not x	逻辑非
x in y; x not in y	成员测试运算符
x is y; x is not y	对象实体同一性测试（地址）
|、^、&、<<、>>、~	位运算符
&、|、^	集合交集、并集、对称差集

1.4.5 操作符和表达式(5/6)

- 合法表达式：

  单个任何类型的对象或常数

  使用运算符连接的变量和常量以及函数调用

```
>>> a = [1,2,3]
>>> b = [4,5,6]
>>> c = a+b
>>> c
[1, 2, 3, 4, 5, 6]
>>> d = map(str, c)
>>> d
['1','2','3','4','5','6']
```

```
>>> import math
>>> map(math.sin, c)
>>> 'Hello' + ' ' + 'world'
'Hello world'
>>> 'welcome ' * 3
'welcome welcome welcome '
>>>
('welcome,'*3).rstrip(',')+'!'
'welcome,welcome,welcome!'
```

1.4.6 常用内置函数(2 /5)

all(iterable)	如果对于可迭代对象中所有元素x都有bool(x)为True，则返回True。对于空的可迭代对象也返回True
any(iterable)	只要可迭代对象中存在元素x使得bool(x)为True，则返回True。对于空的可迭代对象，返回False
bin(x)	把数字x转换为二进制串
callable(object)	测试对象是否可调用。类和函数是可调用的，包含__call__()方法的类的对象也是可调用的
chr(x)	返回ASCII编码为x的字符
cmp(x, y)	比较大小，如果x<y则返回负数，如果x==y则返回0，如果x>y则返回正数。Python 3.x不再支持该函数
eval	计算字符串中表达式的值并返回
filter	返回序列中使得函数值为True的那些元素，如果函数为None则返回那些值等价于True的元素。如果序列为元组或字符串则返回相同类型结果，其他则返回列表

1.4.6 常用内置函数(3/5)

hex(x)	把数字x转换为十六进制串
id(obj)	返回对象obj的标识（地址）
input([提示内容字符串])	接收键盘输入的内容，返回字符串。
int(x[, d])	返回数字的整数部分，或把d进制的字符串x转换为十进制并返回，d默认为十进制
isinstance(object, class-or-type-or-tuple)	测试对象是否属于指定类型的实例
len(obj)	返回对象obj包含的元素个数，适用于列表、元组、集合、字典、字符串等类型的对象
list([x])、 set([x])、tuple([x])、dict([x])	把对象转换为列表、集合、元组或字典并返回，或生成空列表、空集合、空元组、空字典
map(函数,序列)	将单参数函数映射至序列中每个元素，返回结果列表

1.4.6 常用内置函数(4 /5)

open(name[, mode[, buffering]])	以指定模式打开文件并返回文件对象
ord(s)	返回1个字符s的编码
range([start,] end [, step])	返回一个等差数列（Python 3.x中返回一个range对象），不包括终值
reduce(函数,序列)	将接收2个参数的函数以累积的方式从左到右依次应用至序列中每个元素，最终返回单个值作为结果
reversed(列表或元组)	返回逆序后的迭代器对象
round(x [, 小数位数])	对x进行四舍五入，若不指定小数位数，则返回整数
str(obj)	把对象obj转换为字符串
sorted(列表[, cmp[, key[reverse]]]	返回排序后的列表。Python 3.x中的sorted()方法没有cmp参数
zip(seq1 [, seq2 [...]])	返回[(seq1[0], seq2[0] ...), (...)]形式的列表

1.4.7 对象的删除(1/3)

- Python自动内存管理功能：
  - 解释器跟踪所有值，没有变量指向的自动删除
  - 自动内存管理不保证及时释放内存
  - 显式释放申请的资源是程序员的好习惯和素养
- del命令
  - 显式删除对象
  - 解除与值之间的指向关系

1.4.9 模块的使用(1/4)

- Python默认安装仅包含部分基本或核心模块,用户可以用pip安装大量的扩展
- Python启动时仅加载了很少的一部分模块，需要时由程序员显式地加载（可能需要先安装）其他模块
- 减小运行的压力，仅加载真正需要的模块和功能，且具有很强的可扩展性。

1.4.9 模块的使用(4/4)

- Python首先在当前目录中查找需要导入的模块文件
- 否则从sys模块的path变量所指定的目录中查找
- 可以用sys模块的path变量查看导入模块时搜索路径
- 可以向sys.path中append自定义的目录以扩展搜索路径
- 导入模块时优先导入相应的pyc文件
- 如果相应的pyc文件与py文件时间不相符，则导入py文件并重新编译该模块

1.5 Python代码规范(1/3)

- 缩进
  - 类定义、函数定义、选择结构、循环结构，行尾的冒号
  - Python程序是依靠代码块的缩进来体现代码之间的逻辑关系的，缩进结束表示一个代码块结束
  - 同一个级别的代码块的缩进量必须相同
  - 一般以4个空格为基本缩进单位，可以通过下面的方法进行代码块的缩进和反缩进：

    Fortmat➔Indent Region/Dedent Region

1.5 Python代码规范(2/3)

- 注释
  - 30%以上注释
  - 以#开始，表示本行#之后的内容为注释
  - 包含在一对三引号'''...'''或"""..."""之间
  - IDLE 操作快速注释/解除注释大段内容：

    Format➔Comment Out Region/Uncomment Region

1.5 Python代码规范(3/3)

- 每个import只导入一个模块
- 语句太长，在行尾加上\来换行分成多行(建议使用括号来包含多行内容)
- 必要的空格与空行
  - 运算符两侧、函数参数间、逗号两侧建议使用空格分开
  - 代码块之间、函数定义之间建议增加一个空行
- 适当使用异常处理结构进行容错