实验一 Python 基础

1. Python 简介

Python 的创始人为吉多·范罗苏姆（Guido van Rossum）。1989 年的圣诞节期间，吉多·范罗苏姆为了在阿姆斯特丹打发时间，决心开发一个新的脚本解释程序，作为 ABC 语言的一种继承。之所以选中 Python 作为程序的名字，是因为他是 BBC 电视剧——蒙提·派森的飞行马戏团（Monty Python's Flying Circus）的爱好者。

ABC 是由吉多参加设计的一种教学语言。就吉多本人看来，ABC 这种语言非常优美和强大，是专门为非专业程序员设计的。但是 ABC 语言并没有成功，究其原因，吉多认为是非开放造成的。吉多决心在 Python 中避免这一错误，并获取了非常好的效果，完美结合了 C 和其他一些语言。就这样，Python 在吉多手中诞生了。实际上，第一个实现是在 Mac 机上。可以说，Python 是从 ABC 发展起来，主要受到了 Modula-3（另一种相当优美且强大的语言，为小型团体所设计的）的影响。并且结合了 Unix shell 和 C 的习惯。

Python 2.0 于 2000 年 10 月 16 日发布，增加了实现完整的垃圾回收，并且支持 Unicode。同时，整个开发过程更加透明，社区对开发进度的影响逐渐扩大。Python 3.0 于 2008 年 12 月 3 日发布，此版不完全兼容之前的 Python 源代码。不过，很多新特性后来也被移植到旧的 Python 2.6/2.7 版本。

Python 可用于 WEB 开发、网络编程、爬虫、云计算、人工智能、自动化运维、金融分析、科学运算、游戏开发等，许多公司使用 python 进行软件开发，如谷歌、CIA、NASA、YouTube、Dropbox、Instagram、Facebook、豆瓣、知乎等。

2. 基础语法

2.1 中文编码

Python 中默认的编码格式是 ASCII 格式，在没修改编码格式时无法正确打印汉字，在读取中文时会报错。解决方法为只要在文件开头加入 # -*-

coding: UTF-8 -*- 或者 #coding=utf-8。当然也可以为源码文件指定其他不同的编码。

代码：输出‘南京大学’

```
# coding = utf-8
print(‘南京大学’)
```

2.2 标识符

标识符的第一个字符必须是字母表中字母或下划线，其他部分由字母、数字和下划线组成。

标识符对大小写敏感。

代码：标识符

```
# 定义变量
class_01 = 7
Doc_students = []
```

2.3 保留字

保留字即关键字，我们不能把它们用作任何标识符名称。Python 的标准库提供了一个 keyword 模块，可以输出当前版本的所有关键字。

代码：保留字

```
# 查看 python 保留字
import keyword

print(keyword.kwlist)
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue',
'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if',
'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise',
'return', 'try', 'while', 'with', 'yield']
```

2.4 注释

python 中单行注释采用#开头，多行注释可以用多个 # 号，还有 ''' 和 """。

代码：用三种方式写注释

```
# 输出’hello world’

‘’’
```

```
输出’hello world’
‘hello Nanjing’
‘’’

"""
第 n 行注释
第 n+1 行注释
"""

print(‘hello world’)
```

2.5 行与缩进

python 最具特色的就是使用缩进来表示代码块，不需要使用大括号{}。

缩进的空格数是可变的，但是同一个代码块的语句必须包含相同的缩进空格数，语句缩进空格数的不一致会导致运行错误。

代码：缩进不一致导致运行错误

```
# 输出’ hello world’

if True:
    print ("Answer")
    print ("True")
else:
    print ("Answer")
  print ("False")    # 缩进不一致，会导致运行错误
```

输出：

```
File "test.py", line 6
    print ("False")    # 缩进不一致，会导致运行错误
IndentationError: unindent does not match any outer indentation level
```

2.6 基本数据类型

Python 中的变量不需要声明。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。用等号“=”来给变量赋值。Python 允许同时为多个变量赋值，也可以为多个对象指定多个变量。

代码：变量赋值

```
a = b = c = 1
a, b, c = 1, 2, "daisy"
```

Python 3 中有六个标准的数据类型：Numbers（数字），String（字符串），List（列表），Tuple（元组），Set（集合），Dictionary（字典）。内置的 type() 函数可以用来查询变量所指的对象类型。

代码：查看数据类型

```
a = 8
print(type(a))
```

输出：

```
<type 'int'>
```

- Numbers

  Python 支持 int、float、bool、complex（复数），在 Python 3 里，只有一种整数类型 int，表示为长整型，没有 python2 中的 Long。

- String

  String（字符串或串）是由数字、字母、下划线组成的一串字符，用单引号 ' 或双引号 " 括起来。反斜杠 \ 用来转义特殊字符，如果不想让反斜杠发生转义，可以在字符串前面添加一个 r，表示原始字符串。

  字符串截取的语法格式：变量[头下标:尾下标]。python 的字串列表有 2 种取值顺序：从左到右索引默认 0 开始的，最大范围是字符串长度少 1；从右到左索引默认-1 开始的，最大范围是字符串开头。下标为空，表示从头取，或者取到尾。

代码：字符串操作

```
print('s\noopy')
print(r's\noopy ')

string = 'I love python'
print(string[4])                # 输出索引为 4 的字符
print(string[-2])               # 输出索引为-2 的字符，即倒数第 2 个字符
print(string[2: 6])             # 输出索引为 2 和索引为 6(不包含)之间的字符
print(string[ : -7])            # 输出开头到索引为-7(不包含)之间的字符
```

输出：

```
S
oopy
s\noopy
v
o
love
I love
```

- List

  List（列表）是 Python 中使用最频繁的数据类型。列表可以完成大多数集合类的数据结构实现。它支持字符，数字，字符串甚至可以包含列表（嵌套）。列表用[ ]标识，[ ] 中的元素用‘,’分隔开。列表的索引方法与字符串相似。列表经常用于 for 循环。

代码：List 声明与索引

```
lt = [1, 2, 'abc', [3,4,5]]       # 新建一个由数字构成的列表
print(lt[0])         # 输出列表第一个元素
print(lt[1:3])       # 从第二个开始输出到第三个元素
print(lt[2:])        # 输出从第三个元素开始的所有元素
```

输出：

```
1
[2, 'abc']
['abc',[3,4,5]]
```

代码：List 在 for 循环中的应用

```
lt = [1, 2, 3, 4, 5]       # 新建一个由数字构成的列表

for i in lt:               # 遍历列表内的元素
    print(i * 100)
```

输出：

```
100
200
300
400
500
```

List 相关的函数和方法：

序号	函数	描述
1	len(list)	返回列表元素个数
2	max(list)	返回列表元素最大值
3	min(list)	返回列表元素最小值
4	list(seq)	将元组转换为列表

序号	方法	描述
1	list.append(obj)	在列表末尾添加新的对象
2	list.count(obj)	统计某个元素在列表中出现的次数
3	list.extend(seq)	在列表末尾一次性追加另一个序列中的多个值（用新列表扩展原来的列表）
4	list.index(obj)	从列表中找出某个值第一个匹配项的索引位置
5	list.insert(index, obj)	将对象插入列表
6	list.pop([index=-1])	移除列表中的一个元素（默认最后一个元素），并且返回该元素的值
7	list.remove(obj)	移除列表中某个值的第一个匹配项
8	list.reverse()	反向列表中元素
9	list.sort(key=None, reverse=False)	对原列表进行排序
10	list.clear()	清空列表
11	list.copy()	复制列表

- Tuple

  Tuple（元组）与列表类似，不同之处在于元组的元素不能修改。元组写在小括号 () 里，元素之间用逗号隔开。元组中的元素类型也可以不相同。

  元组与字符串类似，可以被索引且下标索引从 0 开始，-1 为从末尾开始的位置，也可以进行截取。可以把字符串看作一种特殊的元组。

代码：tuple 实例

```
tuple = ('abcd', 786 , 2.23, 'runoob', 70.2  )
```

- Set

  Set（集合）是由一个或数个形态各异的大小整体组成的，构成集合的事物或对象称作元素或是成员。基本功能是进行成员关系测试和删除重复元素。

  可以使用大括号 { } 或者 set() 函数创建集合，注意：创建一个空集合必须用 set() 而不是 { }，因为 { } 是用来创建一个空字典。

代码：set 实例

```
sites = {'Google', 'Taobao', 'Runoob', 'Facebook', 'Zhihu', 'Baidu'}


print(sites)   # 输出集合，重复的元素被自动去掉


# 成员测试
if 'Runoob' in sites :
    print('Runoob 在集合中')
else :
    print('Runoob 不在集合中')


# set 可以进行集合运算
a = set('abracadabra')
b = set('alacazam')
print(a)
print(a - b)     # a 和 b 的差集
print(a | b)     # a 和 b 的并集
print(a & b)     # a 和 b 的交集
print(a ^ b)     # a 和 b 中不同时存在的元素
```

输出：

```
{'Zhihu', 'Baidu', 'Taobao', 'Runoob', 'Google', 'Facebook'}
Runoob 在集合中
{'b', 'c', 'a', 'r', 'd'}
{'r', 'b', 'd'}
{'b', 'c', 'a', 'z', 'm', 'r', 'l', 'd'}
{'c', 'a'}
{'z', 'b', 'm', 'r', 'l', 'd'}
```

- Dictionary

  列表是有序的对象集合，Dictionary（字典）是无序的对象集合。两者之间的区别在于：字典当中的元素是通过键来存取的，而不是通过偏移存取。

字典是一种映射类型，字典用 { } 标识，它是一个无序的 键(key) : 值(value) 的集合。

键(key)必须使用不可变数据类型（当变量的值发生了改变，对应的内存地址也会发生改变，如整型、字符串、元组）。在同一个字典中，键(key)必须是唯一的。

代码：Dictionary 实例

```
tinydict = {'name': 'bbs','code':1, 'site': 'www.bbs.com'}
print(tinydict ['name'])      # 输出键为 'name' 的值
```

输出：

```
bbs
```

2.7 条件语句

Python 中，可以使用 if、elif、else 来表达条件语句。

代码：条件语句实现

```
List = ['Peking Univ.', 'Nanjing Univ.', 'Wuhan Univ.']
for i in List:
    if i == 'Peking Univ.':
        print('PKU ', i)
    elif i == 'Nanjing Univ.':
        print('NJU ', i)
    else:
        print('WHU ', i)
```

输出：

```
PKU  Peking Univ.
NJU  Nanjing Univ.
WHU  Wuhan Univ.
```

2.8 循环语句

Python 中的循环语句可以用 for 和 while 来实现。循环语句可以用 break 和 continue 来实现，用法与 C 语言相似。

代码：循环语句实现

```
for i in [1, 2, 3, 4]:
    print(i),


i = 0
while i < 5:
    print i,
```

```
    i += 1
```

输出：

```
1 2 3 4
0 1 2 3 4
```

2.9 函数

Python 允许自定义函数。函数代码块以 def 关键词开头，后接函数标识符名称和圆括号()。任何传入参数和自变量必须放在圆括号中间。圆括号之间可以用于定义参数。

函数内容以冒号起始，函数内主体内容缩进。return [表达式] 用来结束函数，并返回一个值给调用方。不带 return 表达式的相当于返回 None。

代码：加法运算，并返回结果

```
def add(a1, a2):
    ans = a1 + a2
    return ans


a1 = 4
a2 = 5
print(str(a1) + ' + ' + str(a2) + ' = ', add(a1, a2))       # '+'表示连接字符串
```

输出：

```
4 + 5 = 9
```

2.10 错误和异常

Python 有两种错误很容易辨认：语法错误和异常。

Python 的语法错误或者称之为解析错，即便 Python 程序的语法是正确的，在运行它的时候，也有可能发生错误。运行期检测到的错误被称为异常。大多数的异常都不会被程序处理，都以错误信息的形式展现出来。

Python assert（断言）用于判断一个表达式，在表达式条件为 false 的时候触发异常。

代码：异常实例

```
>>> 10 * (1/0)            # 0 不能作为除数，触发异常
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
ZeroDivisionError: division by zero


>>> 4 + spam*3            # spam 未定义，触发异常
```

```
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
NameError: name 'spam' is not defined


>>> '2' + 2             # int 不能与 str 相加，触发异常
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str (not "int") to str
```

异常处理：try/except 语句、try/except...else 语句、try-finally 语句。

代码：try/except 语句

```
# 让用户输入一个合法的整数
while True:
    try:
        x = int(input("请输入一个数字: "))
        break
    except ValueError:
        print("您输入的不是数字，请再次尝试输入！")
```

代码：try-finally 语句

```
try:
    print('执行代码')
except:
    print('发生异常时执行的代码')
else:
    print('没有异常时执行的代码')
finally:
    print('不管有没有异常都会执行的代码')
```

2.11 常用标准库

序号	标准库	描述
1	os	操作系统接口
2	shutil	日常的文件和目录管理任务
3	glob	文件通配符；模块提供了一个函数用于从目录通配符搜索中生成文件列表
4	sys	命令行参数

5	math	数学
6	urllib.request	用于访问互联网以及处理网络通信协议
7	smtplib	发送电子邮件
8	datetime	日期和时间处理
9	zlib	支持通用的数据打包和压缩格式：zlib，gzip，bz2，zipfile，以及 tarfile

2.12 常用扩展库

序号	扩展库	描述
1	Numpy	提供数组支持，以及相应的高效处理函数
2	Scipy	提供矩阵支持，以及矩阵相关的数值计算模块
3	Matplotlib	强大的数据可视化工具、作图库
4	Pandas	强大、灵活的数据分析和探索工具
5	StatsModels	统计建模和计量经济学，包括描述统计、统计模型估计和推断
6	Scikit-Learn	支持回归、分类、聚类等的强大机器学习库
7	Keras	深度学习库，用于建立神经网络以及深度学习模型
8	Tensorflow	深度学习库，用于建立神经网络以及深度学习模型
9	Gensim	文本主题的库，可用于文本挖掘

参考网址：
https://www.tutorialspoint.com/python/python_overview.htm
https://www.runoob.com/python3/python3-data-type.html

python 扩展库下载：
https://www.lfd.uci.edu/~gohlke/pythonlibs/#twisted