Python语法速成

## 一、打印print

1.打印 print(‘xxxxx’) / print(“xxxxx”) 引号里的内容作为字符串直接打印；

2.字符串连接print(“hesda” + ”sad” + “sda”)当拼接的内容中包含变量、表达式或动态生成的字符串时必须使用“+”；

3.单双引号转义print(“he is ‘good’”)适用于引号内部还有引号的情况；print("he said \"hello\"")反斜杠+引号表示这个部分只是引号，为内容字符串的一部分；

4.换行print(“hello! \n hi”) \n表示换行符；

5.三引号换行符

Print("""dawdasdaw

Dawdawdasd

wadawdaw""")

括号里的内容可正常换行读取。

6.计算程序执行时间：

import time

start\_time = time.time()

xxxxxx

xxxxxx

end\_time = time.time()

print(**f**"Execution time: {end\_time - start\_time} seconds")

7.大写打印

S=”Hello”

Print(s.upper())

这样打印出的Hello则全是大写字母

## 二、变量

1.变量赋值：aaa=”111”或者直接aaa=111，打印print("wad”+aaa)；引号包裹的为字符串；

2.变量命名：大小写敏感，user\_age与user\_Age是两个变量；变量名不能占用python关键字；

## 三、数学运算

1.一般整数默认为Int类型，小数默认为Float类型；数学运算中\*为乘号，正斜杠/为除号，两个乘号\*\*为乘方；

2.更多的复杂的运算操作需要导入math库（import math）,math.sin(1)计算1弧度的正弦值；

## 四、注释

1.多行快速注释：除去直接使用#键外，ctrl+/可以同时添加好几行注释或者同时撤销好几行注释；或者使用三引号跨行注释”””xxx”””不进行赋值也不进行打印（即不进行任何操作）；

## 五、数据类型

1.字符串str: 通常要使用单/双引号包围，使用len()函数可得到字符串的长度，空格与符号都会占一个长度，转义符\n只占一个长度，”hello”[3]这个式子可以索引该字符串第三位的字符（但是是从0开始数的，所以这里的第三位其实是第四位）；

2.整数Int，浮点数Float，操作与其他编程语言的类似；

3.布尔类型Bool：真True与假False；

4.空值类型NoneType：None，这个类型不是0不是空字符串也不是False，它表示完全没有值，比如想知道一个变量，但不知道这个变量的值可以先定义为None；

5.Type函数：返回该变量的数据类型，Type(True)

## 六、与用户交互的Input函数

1.user\_age = input(“请输入您的年龄：”)，input()函数返回的全部是字符串str类型；

2.user\_age\_int = int(user\_age)，将字符串str类型更改为整数int类型，类似的也有float函数float()，str函数str()

## 七、条件运算

1.比较运算符：等于==（两个等号，一个等号为变量赋值）、不等号!=、大于>、大于等于>=、小于<、小于等于<=

2.if [布尔条件或使用比较运算符的布尔条件]：

[执行语句]

[执行语句]

所有缩进的执行语句都被看做条件为True的时候执行的内容

3.if [条件]：

[执行语句]

Else:

[执行语句]

4.嵌套条件语句

If [条件一]：

If[条件二]：

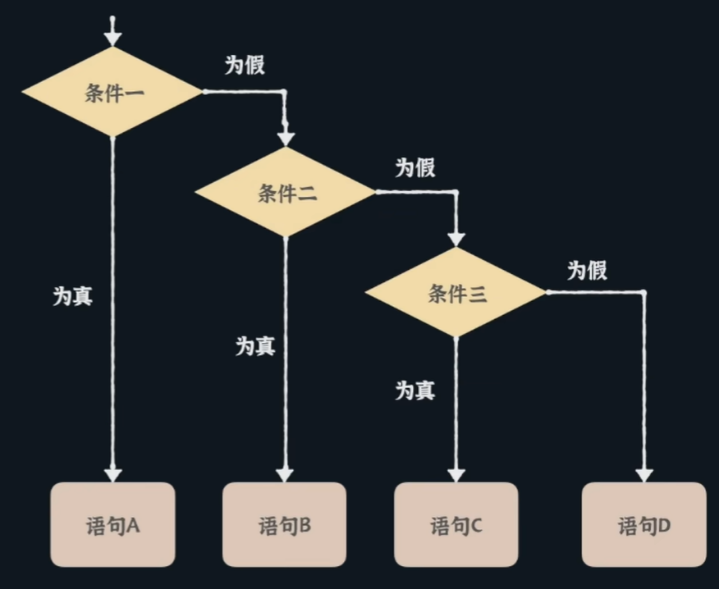
[语句A]

Else:

[语句B]

Else:

[语句C]

5.elif：

If[条件一]:

[语句A]

Elif[条件二]:

[语句B]

Elif[条件三]:

[语句C]

Else:

[语句D]

由于Python的编译是从上往下的，当条件二与条件三同时满足时也只会执行条件二的语句

## 八、逻辑运算

1.逻辑运算符号：and \ or \ not

2.x>5 and x<10 and x\*2==10，当所有条件都满足时返回值才为True

## 九、列表

1.定义列表Shopping\_list = [“键盘”, “键帽”]，空列表直接使用方括号定义如shopping\_list = []；往这个定义好的列表末尾里面加东西shopping\_list.append(“显示器”)，注意这里不应该对shopping\_list列表重新赋值了；要删除这个列表里面的某个元素可以使用shoppint\_list\_remove(“显示器”)，此处也是不应该对列表重新进行赋值；

2.索引：列表可以使用len()函数求长度，也可以进行索引提取出某一个元素，这里的索引也是从0开始，最后一个元素的位置为列表长度减去1；如果需要修改列表的某个元素可以利用索引赋值，直接修改该位置的元素shopping\_list[1]=”音响”；

3.列表的内置函数

Print(max(num\_list)) #打印列表里的最大值

Print(min(num\_list)) #打印列表里的最小值

Print(sorted(num\_list)) #打印从小到大排序好的列表

## 十、字典

1.用于储存键（key）值（value）对，键会有对应的值，键是用来查找值的，空的字典用花括号表示{}，即contacts={}；

2.字典的结构：

Contacts={“小明”：“123123”，

“小花”：“234123”}

查找时使用Contacts[“小明”]

3.键的类型必须是不可变的，即字符串、整数、浮点数或者元组；

4.多个不可变量确定一个键的时候需要用元组example\_tuple=(“键盘”,”键帽”)，与列表的区别是元组用小括号：

Contacts={(“张伟”,34):”1231245”,

(“张伟”,45):”2314512”,

(“张伟”,56):”2314123”}

找值时可以将整个元组作为键：zhangwei34\_number = contacts[(“张伟”,34)]

5.字典可以添加或删除键值对，添加时直接使用contacts[“美女A”]=”12314523”添加至字典的末尾（注意列表需要使用append函数），但如果这个键已经存在，则这个操作是将原本的值覆盖；删除键值对使用del contacts[“小明”]

6.查看键是否已经存在：”小明” in contacts，返回一个布尔值；

7.len()函数可以查看字典里面有多少键值对

8.字典的函数如tem\_dict.keys()返回所有键；tem\_dict.values()返回所有值；tem\_dict\_items()返回所有键值对

## 十一、for循环

1.可以对列表、字典和字符串作迭代；

2.具体的形式为

For 变量名 in 可迭代对象

#对每个变量做一些事情

#......若可迭代对象为列表，则这个变量名会被依次

#赋值为列表里的每一个元素

3.for staff\_id, tem in tem\_dict.items()

If tem>38

Print(staff\_id)

4.range(5,10)，其中5表示起始值，10表示结束值（结束值不在序列的范围内）；range(5,10,2)，末尾的2表示步长，即每次跨几个数字。求1依次加到100：

Total=0

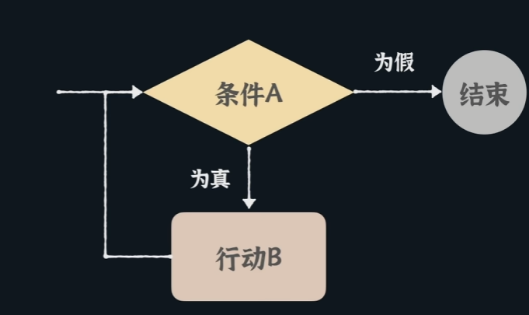
For I in range(1,101):

Total=total+i

Print(total)

## 十二、While循环

适用条件：执行多少次判断后条件为真或者为假，基本结构为：

While 条件A：

行动B

注：for循环在有明确循环对象或次数的时候更加方便，while循环次数未知

## 十三、格式化字符串

1.Format方法：

“””金{0}贺岁，给{1}及家人拜年了”””.format(year,name)，花括号表示会被替换的位置，里面的数字代表会被第几个参数替换；也可以直接给关键词来指定替换的对象：“””金{current\_year}贺岁，给{receive\_name}及家人拜年了”””.format(current\_year=year,recrive\_name=name)，这种情况下format括号里的变量顺序就无所谓了。

2.字符串前缀”f”：

在字符串前加前缀”f”： f“””金{year}贺岁，给{name}及家人拜年了”””，花括号里的内容会被直接求值并添加到字符串内；

3.浮点数格式：{1:.2f}表示打印出两位小数

## 十四、函数定义

Def calculate\_sector(central\_angle, radius):

Sector\_area = central\_angel+radius

调用：calculate\_sector(30,160)在括号里面直接进行赋值

在函数里定义的变量都只是局部变量，在函数外就无法进行取值了，要在函数完成调用后返回函数里的变量需要添加return语句：

Def func():

A=3

Print(A)

Return A

## 十五、引入模块

1.Import statistics

Statistics.median 表示使用引入的statistics模块里的median函数；

2.from statistics import median, mean

Median 使用from xxx import yyy, zzz 这样在使用yyy和zzz的时候就不再需要带上模块statistics的名字了；

3.from statistics import \*，这样将模块statistics里的函数全部引入，这样在使用的时候也不需要带上模块statistics的前缀了

4.pypi.org这个网站可以对第三方库进行搜索

## 十六、面相对象编程

1.面相过程编程：完成某个任务，函数，将要实现的事情拆分成一个个步骤依次完成；

2.面相对象编程：本质就是一个数据类型，不过和里面的参数和方法进行了绑定，在逻辑或信息复杂的时候，面相对象编程可以让逻辑和信息更清晰。

## 十七、文件操作

1.linux的目录为树状结构，终端那个叫做根目录，用斜杠/表示，一切的文件和目录均存放在根目录下面；而windows的每个磁盘分区都有自己的根目录，用磁盘分区加反斜杠表示如C:\；

2.绝对路径是从根目录出发的路径，以根目录为基础，对linux而言就是以斜杠/开头如/home，每个斜杠之间用斜杠/进行分割，最后以目标文件或目标目录结尾；对windows系统，绝对路径以如C:\的形式出现，每个路径之间也用反斜杠\进行分隔，同样最后以目标文件或路径结尾；

3.相对路径：即从参考位置来看所需要索引的文件或目录在哪个位置，用点.来表示参照文件当前所在的目录，用点点..表示更上一层的父目录，更上一层用../..进行分隔（需要根据操作系统进行具体处理）；相对路径的.\可以省略，若在同一目录下，可以直接使用文件名进行索引

4.读取文件：使用open()函数，如open(“/usr/demo/data.txt”, ”r”, encoding=”utf-8”)其中的”r”表示模式，r表示读取模式（只读），w表示写入模式（只写），encoding表示编码方式。Open()函数是python自带的，不需要进行import，需要注意的是代码适合读纯文本，读取的文本也被作为字符串进行返回，所以最好是读取.txt文件。

Print(f.read()) #会读取文件里面的全部内容并打印，该方式存储在内存里

Print(f.read()) #会读空字符串，并打印，因为程序会记录已经读到哪个位置了

在文件非常大的情况下不要用read，因为读取的文件传输在内存里会爆内存。如果不想读取所有内容可以使用Print(f.read(10))表示读从当前读取位置开始再读取十个字符的文件内容

Print(f.readline()) #会读一行文件内容，并打印

F = open(“./data.txt”,”r”,encoding=”utf-8”)

Line = f.readline() #读第一行

While line !=”” #判断当前行是否为空

Print(line) #不为空则打印当前行

Line = f.readline() #读取下一行

Readlines会读取全部文件内容，并把每行作为列表元素返回

5.关闭文件：close，释放资源

F = open(“./data.txt”)

Print(f.read()) #对文件的操作

f.close() #关闭文件，释放资源

6.with open操作：自动关闭文件释放资源

With open(“./data.txt”) as f:

Print(f.read()) #对文件操作

7.写文件：

With open(“./data.txt”,”w”,encoding=”utf-8”) as f #w模式下会清空原文件内容，a模式下可以附加内容，这两种模式下如果文件不存在都会直接创新新文件。

f.write(“hello!\n”) #其中的\n表示换行符

r+ 模式支持同时读写文件，a+同理

## 十八、报错类型

IndexError：索引错误

ZeroDivisionError：除零错误

FileNotFoundError：找不到文件错误

TypeError：类型错误

……

捕捉异常：

Try:

……

Except ValueError:

Print……

Except ZeroDivisionError:

Print……

Except: #这一行会捕捉所有的错误类型，但except语句从上往下运行，只有第一个符合条件的语句会运行。

Else: #当try里面的语句没有产生任何错误时执行

Finally: #无论错误发生与否，里面的语句都会被执行

## 十九、Debug

1.使用assert语句判断求出的值或者需要的目标是否与预想的一致，如asset len(“Hi”) == 2，如果条件判断为False程序就会出现AssertionError断言错误，程序会直接终止

2.从某个文件如my\_calculate.py里面引用某个函数需要from my\_calculate import my\_adder

## 二十、函数嵌套

1.形如calculate\_and\_print(3,calculate\_square)，这里直接把函数名作为参数传入；

2.匿名函数：calculate\_and\_print(7,lambda num: num\*5)，这里的lambda就是匿名函数，可以在该函数下直接调用