# Python知识点总结

## 基础语法

* 1. **编码**

默认情况下，Python3 的源码文件以UTF-8编码，所有字符串都是unicode字符串。

* 1. **标识符**

**第一个字符必须是字母表中的字母或下划线\_；**

**标识符的其他部分有字母、数字、下划线组成**

**标识符对大小写敏感**

**1.3行与缩进**

**Python的特点是用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，但是同一代码块的语句必须包含相同的缩进格数。**

**如果代码语句缩进的格数不一致则会导致“**IndentationError: unindent does not match any outer indentation level**”**

**1.4空行**

**函数之间或类之间的方法用空行分隔，表示一段新的代码的开始。类核函数入口之间也用一行空行分隔，一突出函数入口的开始。空行也是代码的一部分**

**1.5等待用户输入**

**Input (“\n\n按下enter键后退出”)**

**执行上述语句之后，按下回车键就会等待用户输入，其中\n\n在结果输出前会输出两个新的空行，一旦用户按下enter键，程序就会退出。**

**1.6 Print输出**

**Print默认输出是换行的，如果要实现不换行，需要在变量末尾加上end=””**

**1.7 import与 form...import**

**在 python 用 import 或者 from...import 来导入相应的模块。**

**将整个模块(somemodule)导入，格式为： import somemodule**

**从某个模块中导入某个函数,格式为： from somemodule import somefunction**

**从某个模块中导入多个函数,格式为： from somemodule import firstfunc, secondfunc, thirdfunc**

**将某个模块中的全部函数导入，格式为： from somemodule import \***

## Python基本数据类型

Python中六个标准的数据类型：

**不可变数据：**Number（数字）、String（字符串）、Tuple（元祖）；

**可变数据：**List(列表）、Dictionary（字典）、Set（集合）

### 2.1Numbei（数字）

Number支持int、float、bool、complex(复数）。Python中只有一种整数类型int。

其中type(var1)和isinstance(var1)可用来查询变量所指对象类型，他们的区别在于type（）不会认为自雷是父类类型，isinstance（）会认为子类是父类类型。

del()语句可以用来删除一些数字对象的引用，常见语法del(var1 ,var2 ...)

Python可以同时为多个变量赋值，如a, b = 1, 2;

一个变量可以通过赋值指向不同类型的对象;

数值的除法包含两个运算符：/ 返回一个浮点数，// 返回一个整数;

在混合计算时，Python会把整型转换成为浮点数。

Python支持的三种不同数值类型：

Int(整型）：通常被称为整型或者整数，是正或负整数，不带小数点。Python3中整型是没有大小限制的，可以当做Long类型来使用，布尔（bool）是整型的子类型。

Float（浮点型）：浮点型是由整数部分和小数部分组成，浮点型也可以使用科学计数法（2.5e2=2.5\*102）

Complex（复数）：复数由实数部分和虚数部分构成

>>> 5 + 4  # 加法  
9  
>>> 4.3 - 2 # 减法  
2.3  
>>> 3 \* 7  # 乘法  
21  
>>> 2 / 4  # 除法，得到一个浮点数  
0.5  
>>> 2 // 4 # 除法，得到一个整数  
0  
>>> 17 % 3 # 取余  
2  
>>> 2 \*\* 5 # 乘方  
32

### Python数字类型转换

有时候，我们需要对数据内置的类型进行转换，数据类型的转换，只需要将数据类型作为函数名即可。

Int(x)：将x转换到一个整数

float（x)：将x转换到一个浮点数

Complex(x)：将x转换到一个复数，实数部分是x，虚数部分是0

Complex(x,y)：将x和y 转换到一个复数，实数部分为x，虚数部分为y。x和y都是数字表达式。

### 2.1.2数字运算

注意：在整数除法中，除法/总是返回一个浮点数，如果只想得到整数的结果，丢弃可能的分数部分，可以使用运算符//。

>>> 17 / 3 # 整数除法返回浮点型

5.666666666666667

>>>>>> 17 // 3 # 整数除法返回向下取整后的结果

5

1. >> 17 % 3 # ％操作符返回除法的余数

2

>>> 5 \* 3 + 2

17

注意：//得到的并不一定是整数类型的数，他与分母分子的数据类型有关系。等号 = 用于给变量赋值。赋值之后，除了下一个提示符，解释器不会显示任何结果。Python 可以使用 \*\* 操作来进行幂运算。变量在使用前必须先"定义"（即赋予变量一个值），否则会出现错误。

### 2.1.3数学函数

|  |  |
| --- | --- |
| 函数 | 返回值 |
| Abs(x) | 返回数字的绝对值 |
| Ceil(x) | 返回数组的上入整数，入math.ceil(4.1)=5 |
| exp(x) | 返回e的x次幂 |
| Fabs(x) | 返回数字的绝对值 |
| Floor(x) | 返回数字的下舍整数，如math,floor(4.9)=4 |
| Log(x)  Log10(x) | 如math.log(math.e)返回1.0,math.log(100,10)返回2.0  返回以10为基数的x的对数，如math.log10(100)返回 2.0 |
| Max(x1,x2,....) | 返回给定参数的最大值，参数可以是序列 |
| Mod(x) | |  | | --- | | 返回x的整数部分与小数部分，两部分的数值符号与x相同，整数部分以浮点型表示 | |
| Pow(x,y) | X\*\*y运算后的值 |
| Round(x[,n]) | 返回浮点数 x 的四舍五入值，如给出 n 值，则代表舍入到小数点后的位数。  其实准确的说是保留值将保留到离上一位更近的一端。 |
| Sqrt(x) | 返回数字x的平方根。 |

### 2.1.4随机数函数

|  |  |
| --- | --- |
| 函数 | 描述 |
| Choice(seq) | 从系列的元素中随机挑选一个元素，比如random.choice(range(10))，从0到9中随机挑选一个整数 |
| Randrange([start],stop[,step]) | 从指点范围内，按指定技术递增的集合中货去一个随机数，基数默认值为1 |
| Random() | 随机生成下一个实数，在【0，1）范围内 |
| Seed([x]) | 改变随机数生成器的种子seed， |
| Shuffle（lst) | 将序列的所有元素随机排序 |
| Uniform(x,y) | 随机生成下一个实数，它在[x,y]范围内 |
| **random.randint(x,y)** | 随机生一个整数int类型，可以指定这个整数的范围 |
| **random.sample(sequence,length)** | 可以从指定的序列中，随机的截取指定长度的片断，不修改原序列。 |

## 

### Python3字符串

字符串是Python中最常用的数据类型，我们可以用单引号或者双引号来创建字符串。

Var1 = 'Hello World!'

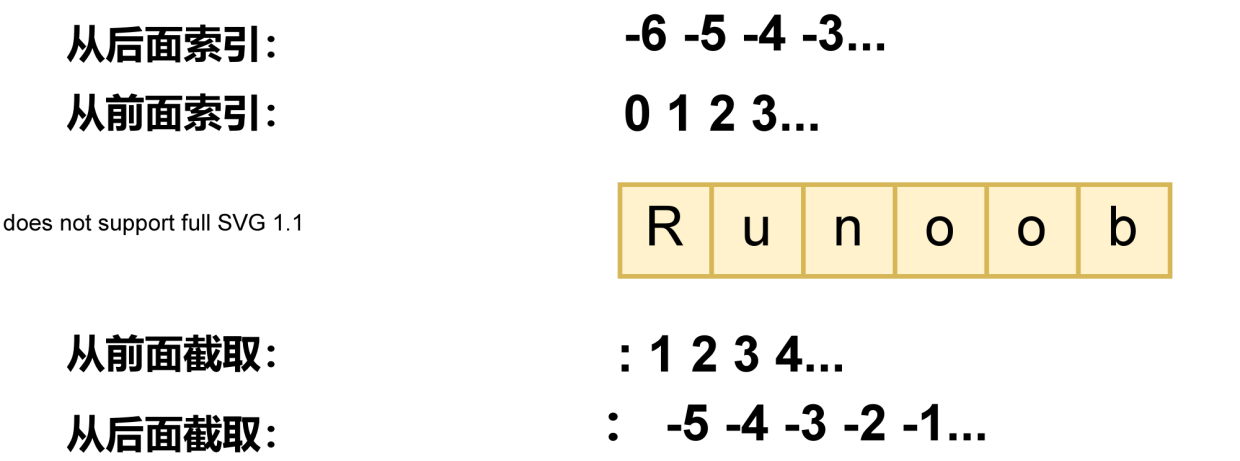
var2 = "Runoob"

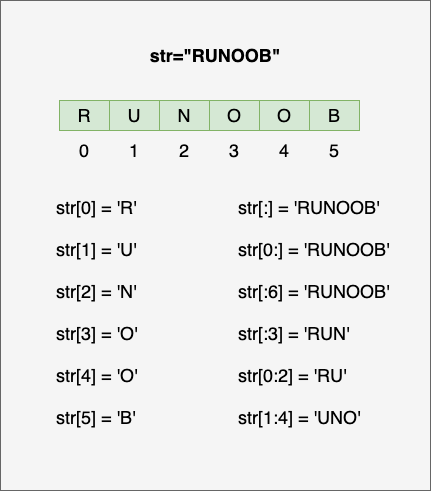
### 访问字符串中的值

他访问子字符串，可以使用方括号[]来截取字符串，字符串截取的格式为：

变量[投下标：尾下标]

索引值以0为开始值，-1为末尾的开始位置





### 转义字符

在需要在字符中使用特殊字符时，Python用反斜杠\转义字符。

\（在行尾时）表示续行符；\\反斜杠符；\’单引号；

\” 双引号; \a 响铃（执行后电脑有响声）；\b 退格；

\000 空； \n 换行；

\v 纵向制表符>>>

print("Hello \v World!")

Hello

World!

|  |  |  |
| --- | --- | --- |
| \t | 横向制表符 | >>> print("Hello \t World!")  Hello      World!>>> |

\r 回车，将\r后面的内容移到字符串开头，并逐一替换开头部分字符，直至将\r后面的内容完全替换完成。

print("Hello\rWorld!")

》》》World!

print('google runoob taobao\r123456')

》》》123456 runoob taobao

**\f 换页**

\yyy 八进制数，y代表0~7的字符，例如：\012代表换行

print("\110\145\154\154\157\40\127\157\162\154\144\41")

Hello World!

\xyy 十六进制数，以\x开头，y代表的字符，例如\x0a代表换行

### 字符串格式化

Python支持格式化字符串的输出，尽管这样可能会用到非常复杂的表达式，但最基本的用法是讲一个值插入到一个有字符串格式符%s的字符串中。在Python中，字符串格式化使用与C中sprintf函数一样的语法。

print ("我叫 %s 今年 %d 岁!" % ('小明', 10))

》》》我叫 小明 今年 10 岁

格式化符号：

%c: 格式化字符及其ascii码

%s: 格式化字符串

%d：格式化整数

%u：格式话无符号整型

%o：格式化无符号八进制数

%x：格式化无符号十六进制数

%X：格式化无符号十六进制数（大写）

%f：格式化浮点数字，可指定小数点后的精度

%e：用科学计数法格式化浮点数

%E：作用同上

%g：%f和%e的简写

%G：%F和%E的简写

%p：用十六进制格式化变量的地址

### Python三引号

python三引号允许一个字符串跨多行，字符串中可以包含换行符、制表符以及其他特殊字符

#### f-String

f-String称之为字面量格式化字符串，是新的格式化字符串的语法

之前一般用的是百分号：

>>> name = 'Runoob'  
>>> 'Hello %s' % name  
'Hello Runoob'

f-String格式化字符串以f开头，后面跟着字符串，字符串中的表达式用大括号包起来，他会将变量或表达式计算后的值替换进去。例如：

>>> name = 'Runoob'  
>>> f'Hello {name}'  # 替换变量  
'Hello Runoob'  
>>> f'{1+2}'         # 使用表达式  
'3'  
  
>>> w = {'name': 'Runoob', 'url': 'www.runoob.com'}  
>>> f'{w["name"]}: {w["url"]}'  
'Runoob: [www.runoob.com'](http://www.runoob.com')

### Python的字符串内建函数

Python字符串常用的内建函数如下：

Capitalize()：将字符串的第一个字符转换为大写

Center(width,fillchar)：返回一个指定的宽度width居中的字符串，fillchar为填充的字符，默认为空格。

Count(str, beg=0,end=len(String))：返回str在string里面出现的次数，如果 beg 或者 end 指定则返回指定范围内 str 出现的次数。

Expandtabs(tabsize = 8) ：把字符串String中的tab 符号转为空格，tab 符号默认的空格数是 8 。

find(str,beg = 0 , end = len(String)) ：检测str是否在字符串中，如果指定范围beg 和end ，则检查是否包含在指定范围内，如果包含返回开始的索引值，否则返回-1。

Isalnum()：如果字符串至少有一个字符并且所有字符都是字母或者数字则返回true，否则返回false

Isalpha()：如果字符串至少有一个字符并且所有字符都是字母或者中文则返回true，否则返回false。

Isdigit()：如果字符串只包含数字则返回true，否则返回false

Isupper()：如果字符串中包含至少一个区分大小写的字符，并且所有这些字符都是大学，泽返回true否则返回false

lenString()：返回字符串长度

Lower()：转换字符串中所有大写字母为小写

Max(str)：返回字符串中最大的字母

Min(str)： 返回字符串中最小的字母

### Python列表

序列是Python中最基本的数据结构。

序列中美搁置都有对应的位置值，称之为索引，第一个索引是0，第二个是1，以此类推。

Python中的6个内置类型，最常见的是列表和元组。

列表常见操作包括索引、切片、加、乘、检查成员。

列表是最常用的Python数据类型，他可以作为一个方括号内的逗号分割值出现。

列表的数据项不需要具有相同的类型。

创建一个列表，只要把逗号分隔的不同数据项使用方括号摞起来即可。

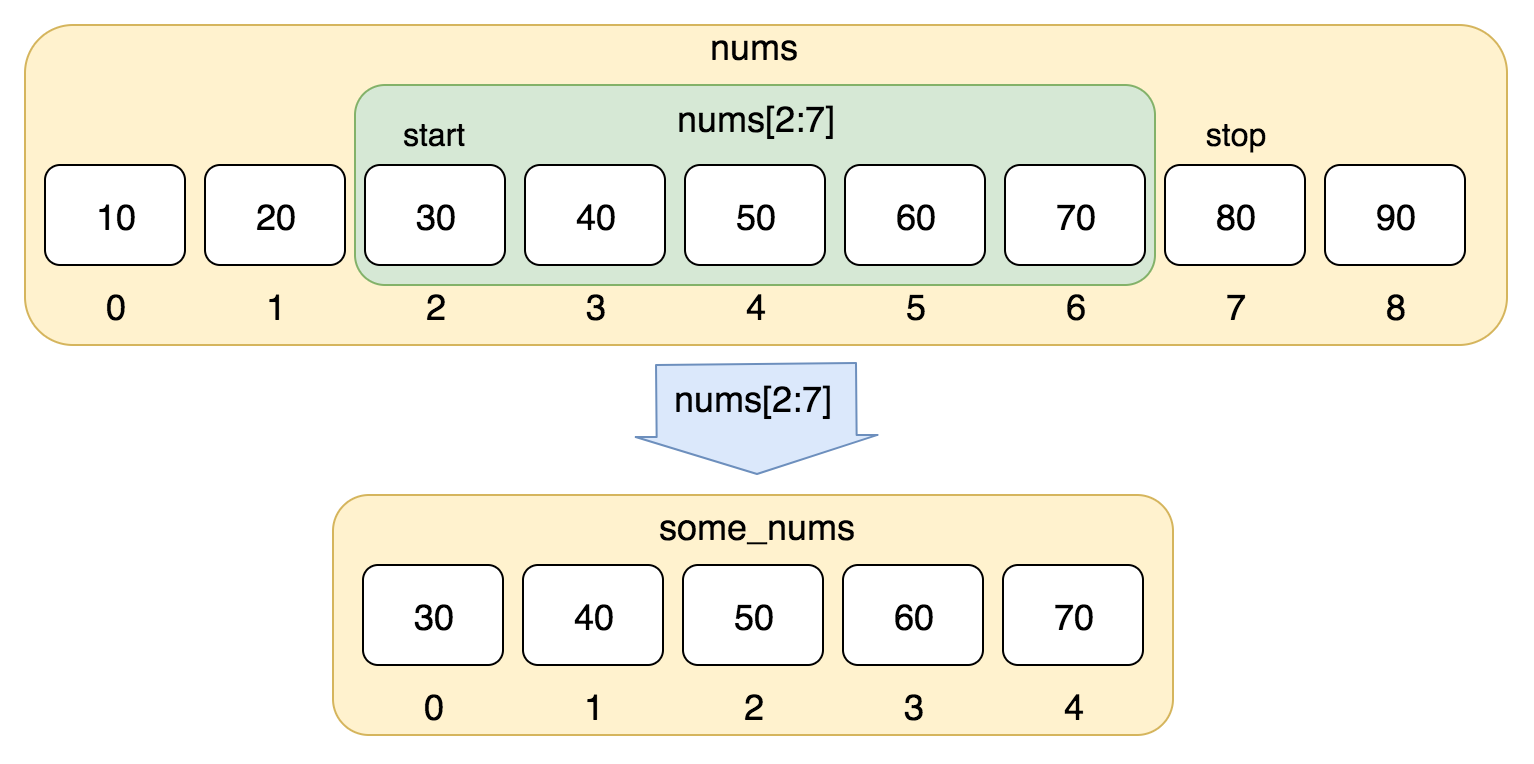
例如：

List1 = [‘Google’ , ‘baidu’ , 12 , 23]

List2 = [“a” , “d” , “c” , “f”]

列表中的索引可以从尾部开始，最后一个元素的索引为-1，往前一位为-2，以此类推。

列表也可以使用下标索引来访问列表中的值，同样也可以使用方括号的形式来截取字符



### 更新列表：

你可以对列表的数据项进行修改或更新，你也可以使用append（）方法来添加列表项

list1 = ['Google', 'Runoob', 'Taobao']

list1.append('Baidu')

print ("更新后的列表 : ", list1)

》》》更新后的列表 : ['Google', 'Runoob', 'Taobao', 'Baidu']

### 删除列表元素：

可以使用del语句来删除列表中的元素，以上例的list1为例

Del list1[2]###删除列表中元素二

### Python 列表脚本操作符： len([1,2,3])

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Python表达式 | | 结果 | | 描述 | |
| len([1,2,3]) | | 3 | | 长度 | |
| [1,2,3]+[4,5] | | [1,2,3,4,5] | | 组合 | |
| [‘HI’]\*4 | | [‘HI’,;HI’,’HI’,’HI’] | | 重复 | |
| 3 in [1, 2, 3] | | True | | 元素是否存在于列表中 | |
| For x in [1,2,3]: Print(x,end=””) | | 1 2 3 | | 迭代 | |
|  |  |  |  |  |  |

### Python 列表截取与拼接

>>>L=['Google', 'Runoob', 'Taobao']

>>> L[2]

'Taobao'

>>> L[-2]

'Runoob'

>>> L[1:]

['Runoob', 'Taobao']

>>>

列表还支持拼接操作：

>>>squares = [1, 4, 9, 16, 25]

>>> squares += [36, 49, 64, 81, 100]

>>> squares

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

>>>

### 嵌套操作：

使用嵌套操作也就是在列表里面创建其他列表。例如：

>>>a = ['a', 'b', 'c']

>>> n = [1, 2, 3]

>>> x = [a, n]

>>> x

[['a', 'b', 'c'], [1, 2, 3]]

>>> x[0]

['a', 'b', 'c']

>>> x[0][1]

'b'

### 列表比较

列表比较需要引入operator模块的eq方法。

### Python列表函数及其方法：

Len(list) ：列表元素个数

Max(list) ：返回列表元素最大值

Min（list)：返回列表元素最小值

List(seq) ： 将元组转换为列表

List.append(obj) : 在列表末尾添加新的对象

List.count(ogj)： 统计某个元素在列表中出现的次数

List.extend(sqe) : 在列表末尾一次性追加另一个序列的多个值

List.index(obj) : 从列表中找出某个值第一个匹配项的索引位置

List,insert(index,obj)：将对象插入列表

List.pop(index = -1)：移除列表中的一个元素，默认是最后一个元素，并且返回该元素的值。

List.remove(obj)：移除列表中某个值的第一个匹配项

List.reverse()：反向列表中的元素

List.sort(key = None,reverse = False)：对原列表进行排序

List.clear()清空列表

List.copy():复制列表

### Python 元组

Python 的元组与列表类似，不同之处在元组的元素不能修改，元组使用小括号（），列表使用的是方括号。

元组的创建非常简单，只需要在括号中添加元素，并使用逗号隔开即可。

样例： tuple = (‘ad’,’ads’,cvb’)

注意：元组的基本操作与列表相似，故直接省略

Tuple(iterable) 将可迭代系列转换为元组

>>> list1= ['Google', 'Taobao', 'Runoob', 'Baidu']

>>> tuple1=tuple(list1)

>>> tuple1('Google', 'Taobao', 'Runoob', 'Baidu')

### 关于元组是不可变的

所谓元组的不可变是指元组所只想的内存中内容不可变

### Python 字典

字典是另一种可变容器模型，且可存储任意类型的对象。

字典的每个键值key =>value对 用冒号分隔，每个对之间用逗号分隔，整个字典包括在花括号{}中，格式如下：

d = {key1 : value1, key2 : value2, key3 : value3 }

注意：dict作为Python的关键字和内置函数，变量名不建议命名为dict

字典中键必须是唯一的，但是值则不必。

值可以去任何数据类型，但是键必须是不可变的，如字符串，数字。

### 创建空字典

使用大括号直接创建空字典：

emptyDict = {}

使用内建函数dict()创建字典：

emptyDict = dict();

### 访问字典里的值

把相应的键放入到方括号中，如下实例嗯：

tinydict = {'Name': 'Runoob', 'Age': 7, 'Class': 'First'}

print ("tinydict['Name']: ", tinydict['Name'])

print ("tinydict['Age']: ", tinydict['Age'])

### 修改字典

向字典添加新内容的方法是增加新的键值对，修改或删除已有键值对如下实例：

Tinydict = {‘name’:’Google’ , ‘Age’:’7’ , ‘Class’:’First’}

Tinydict[‘Age’]=8

Tinydict[‘School’]=”谷歌”

print ("tinydict['Age']: ", tinydict['Age'])

print ("tinydict['School']: ", tinydict['School'])

### 删除字典元素

能删除单一的元素也能清空字典，清空只需要一个操作：

tinydict = {'Name': 'Runoob', 'Age': 7, 'Class': 'First'}

del tinydict['Name'] # 删除键 'Name'

tinydict.clear() # 清空字典

del tinydict # 删除字典

print ("tinydict['Age']: ", tinydict['Age'])

print ("tinydict['School']: ", tinydict['School'])

### 字典键的特性

字典值可以是任意的Python对象，既可以是标准的对象，也可以是用户定义的，但是键不行。

其中两点需要注意：

1. 不允许同一个键出现两次，创建时如果同一个键被复制两次，后一个值会被记住
2. 键必须不可变，所以可以用数字，字符串或者元组充当，但是用列表不行

### 字典内置函数和方法

Len(dict) ：计算字典元素（键值对）个数，既键的总和

Str(dict) ： 输出字典，可以打印字符串的表示