第2章 内存与变量

2.1 Python的内存管理

2.2 Python的变量命名

2.3 Python的变量类型

语法信息参见“菜鸟教程”：
https://www.runoob.com/python3/python3-tutorial.html

2.1 Python的内存管理

- 工作空间（分配给变量用的内存）
  - √ 查询工作空间中的变量：dir()
  - √ 获取变量保存的地址(指针)：id()
  - √ 移除变量释放空间：del，%reset
  - √ Python的变量实质上是指向存储内容的一个引用（标签），如：a=1; b=1;执行后两者的指针是相同的！

- C语言给每个变量分配空间
- Python给每个值分配空间

```
from sys import getrefcount
a=1.2345
print(getrefcount(a)) #显示引用数
b=a
print(getrefcount(a))
print( id(a), id(b) ) #显示指针
a=a+1
print(getrefcount(a))
print( id(a), id(b) ) #显示指针
```

```
3

4
2610761245872 2610761245872

2
2610762793776 2610761245872
```

2.1 Python的内存管理

➢ 工作空间（分配给变量用的内存）

√ 内存回收：
```
import gc        #garbage collector
del  Avar
gc.collect()     #清除引用计数为0的内存空间
```

Python的内存管理参见：

https://blog.csdn.net/jiangjiang_jian/article/details/79140742

https://www.cnblogs.com/vamei/p/3232088.html

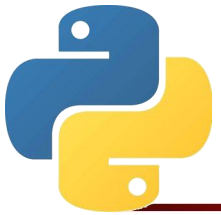
2.2 变量命名

_开头的变量作为系统内部变量，在Spyder变量观察窗看不到，用dir()可以看到

➢ 变量的命名规则

√ 以字母、汉字或下划线开头；
√ 后接字母、汉字、数字或下划线的字符序列；
√ 最多?个字符（允许大于128）；
√ 区分字母大小写；
√ 不能有除了“_”以外的其它特殊字符，如“@#$^&”等；
√ 不能用语法中的保留字作为变量名，如“if、for”。

正确变量名	错误变量名
_2test	1abc
floatMyName	aver#
M630_2a	$abcd
测试a	zqf@303
For	class



2.2 变量命名

- 良好的变量命名习惯——匈牙利命名法
  - √ 变量名＝属性_类型＋描述
  - √ 描述中的多个单词以首字母大写来分隔

属性	类型	
g_ 全局 p_保留 c_常量	i 整形； s 字符串； p 指针； ce 元胞变量。	f 浮点数； st 结构变量； h 句柄；

全局用global声明，保留和常量只是君子协定

举例：g_iDispLength；　　p_sDispName

c_fTransCoef;

2.3 变量类型

Python3.x中所有int都为长整数，且为“任意长”，可以字符方式保存和运算

numpy支持

字段名对应值，类似于数据库

成员可为任意异构变量

无序、元素不重复的列表

C语言	字节数	Python	字节数	举例
short	2	——		32767，-32768
long	4	int	≥28	2147483647，2**65-1
float	4	——		3.4028e38，3.14e-10
double	8	float	24	1.79769313e+308
complex	16	complex	32	1j，2.4 + 65.3J
char		str		’Abc’，“汉字”
logical	1	bool	28	b= 3>1
struct		dict	字典	x={’name’:’zqf’,’age’:52}
		list tuple	列表 元组	x=[1,’zqf’,0.5] x=(1,’zqf’,0.5)
		set	集合	x={1, ’zqf’, 0.5, 1}

2.3 变量类型

整数

Python对不同大小的整数采用不同的存储方式，但操作都一样，对用户是透明的。

sys.getsizeof()

数值	2**1	2**29	2**30	2**59	2**60	2**89	2**90
存储字节数	28	28	32	32	36	36	40

常用函数：

处理函数	作用	举例
%	求余	100%3=1
//	整除（除后取整）	100//3=33
hex	转为16进制字符串	hex(1000)='0x3e8'
bin	转为二进制字符串	bin(100)='0b1100100'
int	各种进制字符串转为整数	int('0x3e8', base=16)=1000 int('0b1100100', base=2)=100

2.3 变量类型

浮点数

Python对浮点数都采用双精度，即8个字节表示一个浮点数。

```
>>>import sys
>>>sys.float_info
sys.float_info(max=1.7976931348623157e+308,
max_exp=1024, max_10_exp=308,
min=2.2250738585072014e-308, min_exp=-1021,
min_10_exp=-307, dig=15, mant_dig=53,
epsilon=2.220446049250313e-16, radix=2, rounds=1)
```

双精度

IEEE 754-2008
浮点数表示

性质（有限精度）：

浮点数1.0的最小增量

```
eps=sys.float_info.epsilon
(1+eps)-1=2.2204e-16          (1+eps/2)-1=0
eps/100=2.2204e-18            10+eps-10=0
```

用途举例：y=sin(x+eps)/(x+eps);

2.3 变量类型

浮点数

➢ 常用函数

处理函数	作用	举例
round	就近取整	x=3.14159，round(x)=3， round(x,3)=3.142
int	取整数部分	int(4.9999)=4，int(-4.9999)=-4
abs	取绝对值	abs(-567.4)=567.4
float.hex(x) 或x.hex()	浮点数转为 16进制字符串	x=64.0 x.hex()='0x1.0000000000000p+6'
float.fromhex (str)	16进制字符串 转为浮点数	float.fromhex('0x1.0p+6')

float.hex(x) 类成员函数作用于变量
x.hex() 引用变量(对象)所在类的成员函数✔

2.3 变量类型

复数

复数是基于**float**的对象，由实部和虚部组成，虚数的单位**1j**。

举例：**c=1+2j**

j可以是一个变量名
这里不能用i

常用函数：

对象的成员变量

处理函数	作用	举例
c.real	取实部	c.real=1.0
c.imag	取虚部	c.imag=2.0
complex.conjugate(c) c.conjugate()	取共轭	complex.conjugate(c)=1-2j c.conjugate()=1-2j
complex.__abs__() abs()	取模	abs(c)=2.2360679774997 9

对象基类的成员函数
重载了

对象的成员函数
（method方法）

多选题 1分

设置

下列变量命名中，不正确的有：

A A3b6

B _姓名

C 电压

D 3test

E _12

F abc$

G x￥

H d_23

提交

2.3 变量类型

字符串

Python的字符串常量可以有6种构建方式：

（1）单引号。例如：'I love ”中国”！'

（2）双引号。例如：”西工大\n缩写为NWPU”。

（3）两组三个单引号或两组三个双引号。例如：

```
'''海军的胸怀像大海，
空军的胸怀像蓝天'''
```

常用转义符：

转义符	作用	举例
\	续行符，放行末表示未完待续	
\”	双引号自身作为字符串的一部分	
\n	换行	
\t	制表符	
\???	三位8进制数字表示的ASCII码	'\101'=='A'
\x??	两位16进制数字表示的ASCII码	'\x41'=='A'

2.3 变量类型

字符串

Python的字符串常量可以有6种构建方式：

（4）原始字符串（转义符也作为普通字符）

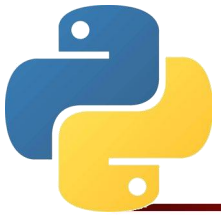
```
>>> str1=r'e:\zqf\本科教学\n课程建设'
>>>str2='e:\zqf\本科教学\n课程建设'
```

名称	类型	大小	值
str1	str	17	e:\zqf\本科教学\n课程建设
str2	str	16	e:\zqf\本科教学 课程建设

（5）f格式字符串（变量或表达式的值转为字符串）

```
>>> x=5
>>>print(f'x+5={x+5}')
x+5=10
>>>print(f'{x+5=}')
x+5=10
```

含“=”的表达式



2.3 变量类型

字符串

Python的字符串常量可以有6种构建方式：

（6）%格式字符串（变量或表达式的值转为指定格式的字符串）

```
>>> x=3.14159
>>>print('x的值是%.2f'%x)
x的值是3.14
>>>y=200;
>>>print('%d的十六进制为%x'%(y,y))
200的十六进制为c8
```

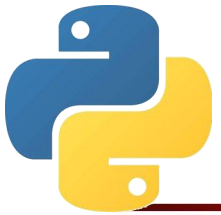
每个print语句输出后默认换行，例如：

```
>>>print('我'); print('海院')
我
海院
```

除非指定结束符，例如：

```
>>>print('我',end='@'); print('海院')
我@海院
```

格式符	说明
%s	字符串（采用str()的显示）
%r	字符串（采用repr()的显示）
%c	单个字符
%b	二进制整数
%d, %i	十进制整数
%o	八进制整数
%x, %X	十六进制整数
%e, %E	指数（基底写为e或E）
%f, %F	浮点数
%g, %G	指数(e)或浮点数（根据显示长度）
%%	字符"%"



2.3 变量类型

字符串

012345

- 提取部分内容，假设s='abcdef'，则：

操作	结果	说明
s[2:4]	'cd'	下标从0开始，止于后者（不含）
s[:2]	'ab'	从头开始止于2（不含）
s[3:]	'def'	下标从3开始到最后
s[1:5:2]	'bd'	起点：终点(不含)：步长
s[::2]	'ace'	从头到尾以步长2取值
s[-2:]	'ef'	倒数第二个开始到最后
s[::-1]	'fedcba'	从尾部到头部反向取值
s[1]='B'	TypeError	字符串中的字符不能单独修改！！

- 拼接与复制

运算符	作用	举例
+	拼接两个字符串	'abc' +'123' = 'abc123'
*	复制字符串	'abc'*3 = 'abcabcabc'

2.3 变量类型

字符串

➢ 常用函数：假设s='abCDa'，则：

函数名	作用	举例
s.upper( ) str.upper(s)	把字符串中的字符转为大写	s.upper()='ABCDA'
s.lower( ) str.lower(s)	把字符串中的字符转为小写	s.lower()='abcda'
len( )	获得字符串的字符数	len('张abc\n') = 5
s.count( )	统计字符串中特定字符的个数	s.count('a')=2
in	字符串中是否有特定字符(串)	'ab' in s = True 'aB' in s = False
s.find( )	在字符串中查找特定字符(串)	s.find('bC')=1
s.replace( )	替换字符串中的特定字符(串)	s.replace('CD','E')='abEa’
s.join( )	将本字符串重复插入到指定字符串的各个字符之间	c=',' c.join(s)='a,b,C,D,a'

2.3 变量类型

字符串

➢ 字符与数字的转换

函数名	作用	备注
int(s[,base])	把其它进制字符串转换为整数	int('F0',base=16)
hex(x)	将整数转换为十六进制字符串	hex(240) = '0xf0'
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	合并到str()
eval(str)	计算在字符串中的表达式	
chr(x)	将Unicode码转换为一个字符	chr(65)=’A’
unichr(x)	将整数转换为Unicode字符	将包含在chr()中
ord(x)	将字符转换为它的Unicode码	ord(‘A’)=65

2.3 变量类型

列表list

Python的列表变量是一个有序的容器，里面可以包括整数、浮点数、复数、字符串等各种量，甚至可以包含列表本身（嵌套）。

➢ 构建列表的方式：

使用方括号[] 或list() 创建，元素间用逗号分隔。各元素类型可以不同，无长度限制。

➢ 举例：

```
ls = ["cat", "dog","tiger",1024]
ln = list(range(2,21,2))
ls[1] = 'Dogs'  ✔
```

列表的元素值可以单独修改

注意：

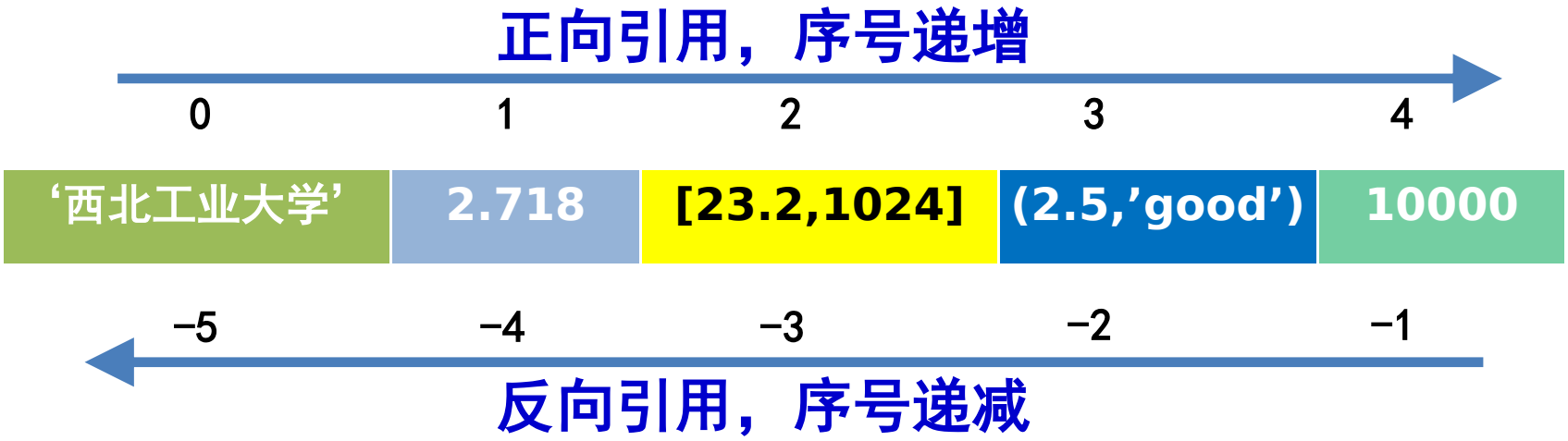
列表只有一维（可以嵌套），不能形成**3×3**这样的数组。

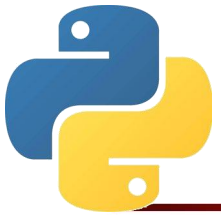
2.3 变量类型

列表list

Python的字符串、列表list及元组tuple都属于“序列”，它们有共同属性和操作。

➢ 序列中序号的定义





2.3 变量类型

列表list

➢ 序列类型的通用操作

序列操作符	含义
x in s	如果x是序列s的元素，返回True，否则返回False
x not in s	如果x是序列s的元素，返回False，否则返回True
s + t	连接两个序列s和t
s * n	将序列s复制n次
s[i]	索引，返回s中的第i个元素，i是序列的序号
s[i:j] 或s[i:j:k]	切片：返回序列s中第i到j个元素子序列 返回序列s中第i到j以k为步长的元素子序列
len(s)	返回序列s的长度，即元素个数
min(s)	返回序列s的最小元素，s中元素需要可比较
max(s)	返回序列s的最大元素，s中元素需要可比较
s.index(x) 或 s.index(x,i,j)	返回序列s中第一次出现元素x的位置 返回序列s从i开始到j位置中第一次出现元素x的位置
s.count(x)	返回序列s中出现x的总次数

2.3 变量类型

列表list

➢ 列表的专用操作(假设ls为列表变量)

列表专用操作符	含义
ls.append()	在末尾添加元素，例如：ls.append('abc')
ls.insert()	在指定位置插入一个元素，例如：ls.insert(2, 'abc')
del ls[?]	删除列表内的若干个元素，例如：del ls[2:4] 而del ls是在从内存中删除变量
ls.reverse()	把列表中所有元素倒序重排，等价于ls=ls[::-1]
ls.clear()	清空列表，使其值为[ ]，但ls的指针不变

2.3 变量类型

元组tuple

元组是跟列表一样的一种序列类型，使用小括号() 或tuple()创建，元素间用逗号分隔，采用前一种方式时小括号可省略。

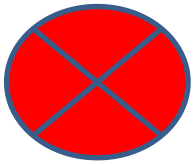
```
>>>creature = "cat", "dog","tiger","human"
>>>color = (0x001100, "blue", creature)
>>>number = tuple(range(2,21,2))
```

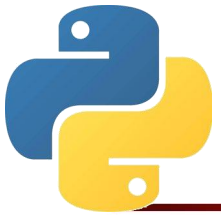
可以嵌套

名称	类型	大小	值
color	tuple	3	(4352, 'blue', ('cat', 'dog', 'tiger', 'human'))
creature	tuple	4	('cat', 'dog', 'tiger', 'human')
number	tuple	10	(2, 4, 6, 8, 10, 12, 14, 16, 18, 20)

元组的索引、切片等操作与列表相同，但其元素不能被独立修改或扩充。

```
>>>color[1] = "red"
>>>color.append("red")
```





2.3 变量类型

元组tuple

- 元组与列表元素的遍历

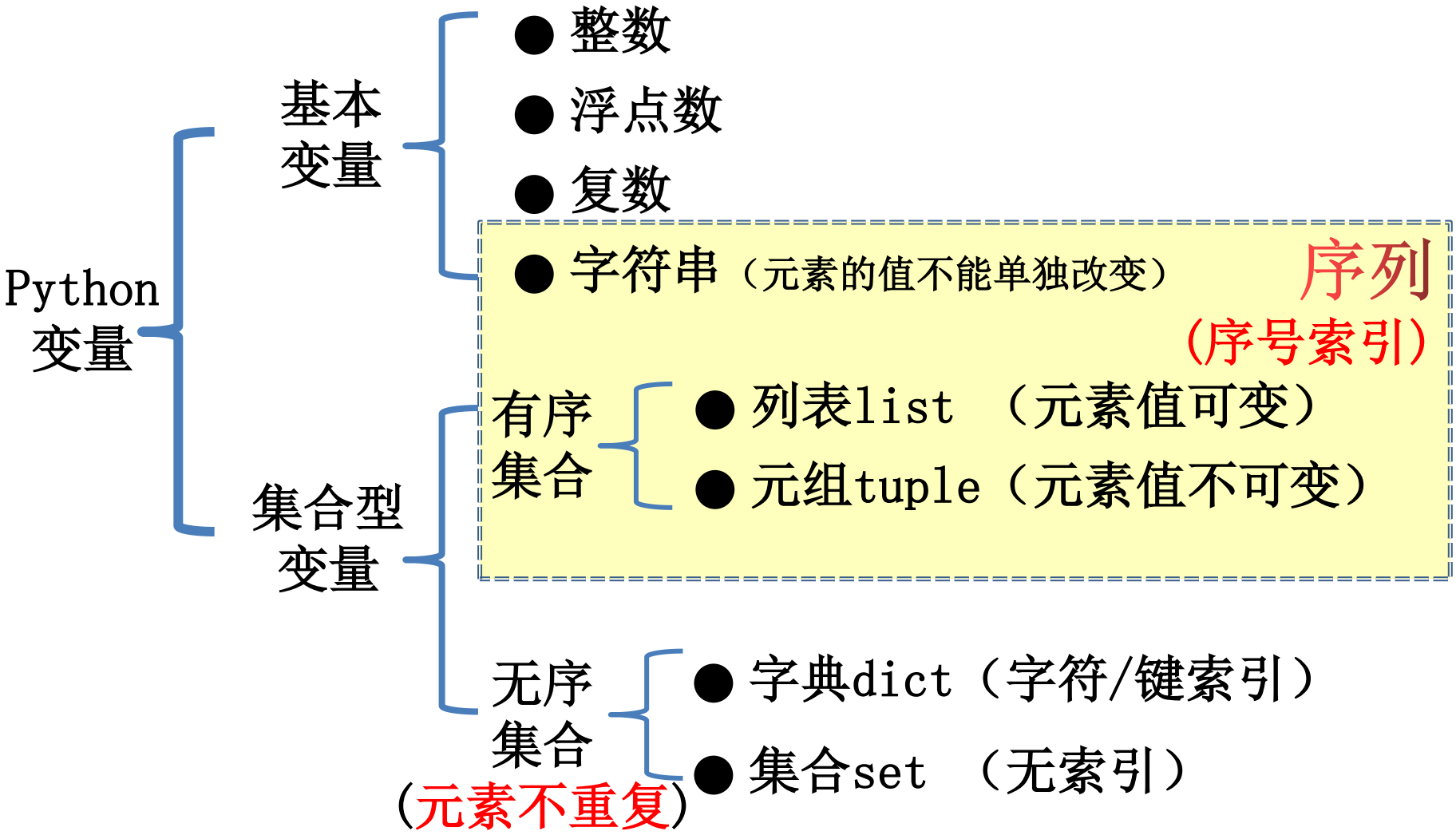
```
for item in list:
    <语句块>
animals = ['tiger','horse','dolphin','penguin']
for animal in animals[:3]:
    print("I love " + animal)
```

```
for item in tuple:
    <语句块>
animals = 'tiger','horse','dolphin','penguin'
for animal in animals[:3]:
    print("I love " + animal)
```

二者输出都是：

```
I love tiger
I love horse
I love dolphin
```

2.3 变量类型


Python
变量
基本
变量
整数
浮点数
复数
字符串（元素的值不能单独改变）
序列
(序号索引)
集合型
变量
有序
集合
列表list （元素值可变）
元组tuple（元素值不可变）
无序
集合
(元素不重复)
字典dict（字符/键索引）
集合set （无索引）

2.3 变量类型

集合set

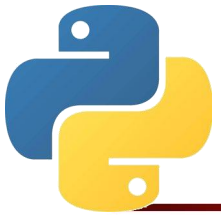
集合(set)是存储无序、不可变数据的一种集合类型。

➢ 创建 **mySet = {value01,value02,...}** 或
mySet = set(...)

举例：

创建集合的方式	说明
s='西工大'; a={1, 2.5, s}	用｛｝包容常数或不可变的变量类型，如：字符串、元组。
t=(1,2,3); a={1, 2.5, t}	
c=[1,2,3]; a=set(c)	取变量里的所有元素值作为集合的成员
c=[1,2,3]; b={c,'a'}	错误！列表变量c是可变的。
se={1,2,3}; b={se,'a'}	错误！集合变量se是可变的，集合不能嵌套。

集合变量是可变的（元素可增减）
但集合内的元素是不可变的



2.3 变量类型

集合set

➢集合的常用运算

（1）去除重复元素

```
>>>a = {1,2,3,1,2,3}
>>>a
{1,2,3}
```

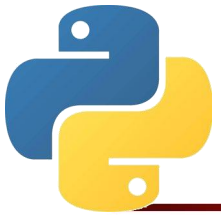
```
>>>a = [1,2,3,1,2,3]
>>>a = list(set(a))
>>>a
[1,2,3]
```

（2）添加元素：set.add(element)

```
>>>mySet = set(["a", "b", "c"])
>>>mySet.add("d")
>>>mySet
{"a", "b", "c","d"}
```

（3）删除元素：set.remove(element)

```
>>>mySet = set(["a", "b", "c", "d"])
>>>mySet.remove("d")
>>>mySet
{"a", "b", "c"}
```



2.3 变量类型

集合set

➢集合的常用运算

（4）交集运算:

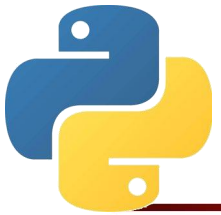
setA.intersection(setB)

```
>>>a = {1,2,3,"nwpu"}
>>>b =
{1,2,"nwpu","pek"}
>>>c = a.intersection(b)
>>>print(c)
{1, 2, 'nwpu'}
```

（5）并集运算:

setA.union(setB)

```
>>>a = {1,2,3,"nwpu"}
>>>b =
{1,2,"nwpu","peking"}
>>>c = a.union(b)
>>>print(c)
{1, 2, 3, 'peking', 'nwpu'}
```

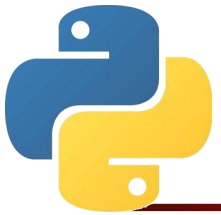


2.3 变量类型

集合set

➢集合的常用运算

集合操作方法	含义
in	检测某值是否在集合内，如：’d’ in mySet
for ... in ...	遍历集合内的所有成员，如：for item in mySet
a.issubset(b)	判断集合a是否为集合b的子集
a.issuperset(b)	判断集合a是否为集合b的超集（包含b）
a.difference(b)	返回多个集合的差集
a.clear	移除集合中的所有元素
a.update(b)	把集合b合并到集合a中，类似于add
b=a.copy()	获得一个集合的副本，a的元素变化不会影响到b
b=a	赋值使b获得a的引用，此后a的变化也导致b变化



2.3 变量类型

字典dict

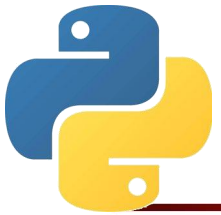
字典是存储无序数据的一种数据结构，其存储元素必须是键值对(key:value)。键必须是不可变的数据类型，值可以是任意类型。

➢字典的创建

```
myDict={key01:value01,key02:value02,...}
```

例如：

```
>>>myDict = {'姓名':'张三', '身高':170, "考研初试成绩":[85,122,68,138]}
>>>myDict
 {'姓名': '张三', '身高': 170, '考研初试成绩': [85, 122, 68, 138]}
```



2.3 变量类型

字典dict

➢ 字典的访问和操作

- 通过键索引取值：dict[key]

```
>>> myDict['身高']
170
```

- 通过get方法取值：dict.get(key)

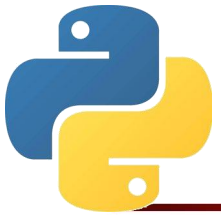
```
>>> myDict.get('身高')
170
```

- 通过键索引修改：dict[key]=value

```
>>> myDict['身高']=185
>>> myDict
 {'姓名': '张三', '身高': 185, '考研初试成绩': [85, 122, 68, 138]}
```

- 通过键索引删除键值：del dict[key]

```
>>> del myDict['身高']
>>> myDict
 {'姓名': '张三', '考研初试成绩': [85, 122, 68, 138]}
```



2.3 变量类型

字典dict

➢ 字典的访问和操作

- 获得所有键：dict.keys()

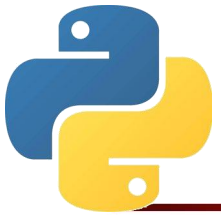
```
>>> myDict.keys()
dict_keys(['姓名', '身高', '考研初试成绩'])
```

- 获得所有值：dict.values()

```
>>> myDict.values()
dict_values(['张三', 170, [85, 122, 68, 138]])
```

- 获得所有键和值：dict.items()

```
>>> myDict.items()
dict_items([('姓名', '张三'), ('身高', 170), ('考研初试成绩', [85, 122, 68, 138])])
```



2.3 变量类型

字典dict

➢字典的常用操作函数

字典操作方法	含义
dict.clear	删除字典内所有元素
dict.has_key(key)	如果键在字典dict里返回true，否则返回false
dict.update(dict2)	把字典dict2的键/值对更新或加到dict里
dict.values()	以列表返回字典中的所有值
dict.popitem()	返回并删除字典中的最后一对键和值
dict.setdefault(key, default=None)	如果指定的键存在，和get()功能类似； 如果键不存在于字典中，将会添加键并将值设为default指定的默认值

多选题 5分

设置

以下的字典变量赋值中哪个是有错的？

A myDict = {1:"number", 'nwpu':"string", (2,3):"tuple"}

B myDict = {'1':"num", 'nwpu':"string", "myD":{"myD":"tuple"}}

C myDict = {1:"number", 'nwpu':list(), "myD":{"myD":"tuple"}}

D myDict = {1:"number", "nwpu":list, set:{"myDict":"tuple"}}

E myDict = {1:"number", "nwpu":list(), list():{"myDict":"tuple"}}

提交