

2.2 Python的基础对象

Python的基础对象包括一些内建的数据类型和结构，它们是构成Python编程的基本元素

Markdown by 李文铠

(1) 数字

Python支持以下数字类型：

- int (整数)
- float (浮点数)
- complex (复数)

内置的type()函数、isinstance()函数可以分别用来查询、判断变量所指的对象类型

Python 数据类型不需事先声名，变量随定义随用，其类型仅与赋予的对象有关

整数类型

与数学中整数的概念一致，Python中的整数没有取值范围限制，其运算与其它计算机语言类同

四种进制表示形式

- 十进制：1010, 99, -217
- 二进制，以0b或0B开头：0b010, -0B101
- 八进制，以0o或0O开头：0o123, -0O456
- 十六进制，以0x或0X开头：0x9a, -0X89

整数除法运算后是浮点数

```
In [165... print(type(10/9))
```

```
<class 'float'>
```

```
In [166... print(isinstance(10/9,float))
```

```
True
```

type()函数可查看Python对象的类型

浮点数类型

与数学中实数的概念一致

浮点数取值范围和小数精度都存在限制，但常规计算可忽略

取值范围数量级约 -10^{307} 至 10^{308} ，精度数量级 10^{-16} （通过使用Decimal和mpmath模块，甚至可以给出内存允许的最大数字）

Python在进行浮点运算时有时会产生不确定尾数

```
In [167... 0.3/6
```

```
Out[167... 0.049999999999999996
```

```
In [169... from decimal import *
print(Decimal('0.3')/Decimal('6'))
```

0.05

判断两个浮点数是否相等时，要考虑不确定尾数问题。

(2) 字符串

字符串是由单引号（或双引号）括起来的，由字符组成的顺序序列，
如：'Nanjing'，"Beijing"等。

字符串的创建

```
In [170... str1 = 'Hello'
str2 = "World"
str3 = '''This is a
multi-line string'''
print(str1,str2,str3)
```

Hello World This is a
multi-line string

反斜杠\是转义符

```
In [171... print("Nanjing")
```

Nanjing

```
In [172... print("Na\njing")
```

Na
jing

```
In [173... print(r"Na\njing")
```

Na\njing

字符串前加r或R使转义符失效

```
In [174... a = 'Nanjing'
print(dir(a))
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__getstate__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'removeprefix', 'removesuffix', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

```
In [175... print(a.split('n'))
```

['Na', 'ji', 'g']

数字的格式化输出

```
In [176... a='%02i' %(3)
print(a)
```

03

```
In [177... date='%04i-%02i-%02i' %(2025,2,17)
print(date)
```

2025-02-17

```
In [178... pi=3.1415926
print('%4.2f' %(pi))
```

3.14

字符串的长度

使用 `len()` 函数（也可用于列表、元组集合等对象）

```
In [179... str1='Nanjing'
print(len(str1))
```

7

字符串的基本操作

拼接（+），常用于路径字符串的定义

```
In [180... path = r"C:\Windows\Fonts"
path = path+'\\'+ 'simsum.ttc'
print(path)
```

C:\Windows\Fonts\simsum.ttc

重复（*）

```
In [181... result = "Ha " * 5
print(result)
```

Ha Ha Ha Ha Ha

切片式引用（[:]）

字符串的切片（slicing）是一种获取字符串子串的方法，使用方括号 [] 和 冒号 : 来指定起始、结束和步长。

0	1	2	3	n-2	n-1	n
字符1	字符2	字符3	字符n-2	字符n-1	字符n	
-n	-(n-1)	-(n-2)	-3	-2	-1	

基本语法

substring = string[start:stop:step]

start: 起始索引（包含）

stop: 结束索引（不包含）

step: 步长（默认为 1，可为负数）

注意字符串的不可变性

```
In [182... text = "Python"
first_char = text[0] # 'P'
last_char = text[-1] # 'n'
print(first_char)
print(last_char)
```

P
n

```
In [183... substring = text[1:4] # 'yth'
print(substring)
```

yth

反序排列

```
In [184... print(text[::-1])
```

nohtyP

```
In [185... text[:2]='Mara' #字符串具有不可变性
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[185], line 1
----> 1 text[:2]='Mara' #字符串具有不可变性

TypeError: 'str' object does not support item assignment
```

```
In [186... text='Mara'+text[2:]
print(text)
```

Marathon

字符串中进行字符替换，使用字符加法，或使用附带函数。

字符串的属性和函数

```
In [187... print(dir(text))
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__getstate__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'removeprefix', 'removesuffix', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

```
In [188... text = text.replace('Mara', 'Py')
print(text)
```

Python

```
In [189... print(text.split('t'))
```

```
['Py', 'hon']
```

(3) 列表

列表 (List) 是 Python 中最常用的数据结构之一，它是一个**有序、可变、可存储不同类型元素**的集合，使用方括号 `[]` 表示，元素之间用逗号 `,` 分隔。

```
In [190...] my_list = [1, 2, 3, "Python", True, print]
print(my_list)
```

```
[1, 2, 3, 'Python', True, <built-in function print>]
```

直接创建列表

```
In [191...] empty_list = [] # 空列表
numbers = [1, 2, 3, 4, 5]
strings = ["apple", "banana", "cherry"]
mixed = [1, "hello", 3.14, True, strings]
```

使用 `list()` 函数创建

```
In [192...] list_from_string = list("hello") # ['h', 'e', 'l', 'l', 'o']
print(list_from_string)
list_from_tuple = list((1, 2, 3)) # [1, 2, 3]
print(list_from_tuple)
```

```
['h', 'e', 'l', 'l', 'o']
```

```
[1, 2, 3]
```

访问列表元素

与字符串访问方式相同，切片式索引，需注意列表只能一维索引。

```
In [193...] print(list_from_string[0:2])
```

```
['h', 'e']
```

```
In [194...] b = [1,[2,3],4]
print(b[1])
```

```
[2, 3]
```

```
In [195...] print(b[1,1])
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[195], line 1
----> 1 print(b[1,1])

TypeError: list indices must be integers or slices, not tuple
```

```
In [196...] print(b[1][1])
```

```
3
```

列表的可变性1

```
In [201...] list_from_string[2:] = 'avy'
print(list_from_string)
```

```
['h', 'e', 'a', 'v', 'y']
```

```
In [202... print(''.join(list_from_string))
```

heavy

使用空字符串可将字符列表粘成一个字符串。
加法、乘法与字符串中的作用相似。

```
In [203... hr = list_from_string+list(' rainfall')  
print(hr)
```

```
['h', 'e', 'a', 'v', 'y', ' ', 'r', 'a', 'i', 'n', 'f', 'a', 'l', 'l']
```

```
In [204... print(''.join(hr))
```

heavy rainfall

列表的可变性2

```
In [205... hs = hr  
print(hr, ''.join(hr))  
hs[-8:] = list('snowfall')  
print(hr, ''.join(hr))
```

```
['h', 'e', 'a', 'v', 'y', ' ', 'r', 'a', 'i', 'n', 'f', 'a', 'l', 'l'] heavy rain  
fall
```

```
['h', 'e', 'a', 'v', 'y', ' ', 's', 'n', 'o', 'w', 'f', 'a', 'l', 'l'] heavy snow  
fall
```

避免上述问题的两种方式

```
In [206... print(hs, ''.join(hs))  
print(hr, ''.join(hr))  
hr = hs.copy() #或hs[:]
```

```
['h', 'e', 'a', 'v', 'y', ' ', 's', 'n', 'o', 'w', 'f', 'a', 'l', 'l'] heavy snow  
fall
```

```
['h', 'e', 'a', 'v', 'y', ' ', 's', 'n', 'o', 'w', 'f', 'a', 'l', 'l'] heavy snow  
fall
```

```
In [207... hr[-8:] = list('rainfall')  
print(hs, ''.join(hs))  
print(hr, ''.join(hr))
```

```
['h', 'e', 'a', 'v', 'y', ' ', 's', 'n', 'o', 'w', 'f', 'a', 'l', 'l'] heavy snow  
fall
```

```
['h', 'e', 'a', 'v', 'y', ' ', 'r', 'a', 'i', 'n', 'f', 'a', 'l', 'l'] heavy rain  
fall
```

列表附带函数，常用append和extend。

```
In [208... hs = ''.join(hr).split()  
print(hs)
```

```
['heavy', 'rainfall']
```

```
In [209... I = ['I', 'dislike']  
I.append(hs[0])  
print(I)  
I.append(hs[1])
```

```
print(I)
print(' '.join(I))
```

```
['I', 'dislike', 'heavy']
['I', 'dislike', 'heavy', 'rainfall']
I dislike heavy rainfall
```

```
In [210... J = ['I', 'like']
J.extend(hs)
print(J)
print(' '.join(J))
```

```
['I', 'like', 'heavy', 'rainfall']
I like heavy rainfall
```

(4) 元组

元组 (Tuple) 是 Python 中的一种数据结构，它与列表类似，但**元组是不可变的**（即创建后不能修改）。元组使用**圆括号 ()** 表示，元素之间用逗号 **,** 分隔。

```
In [211... tuple1 = (1, 2, 3, "Python", True)
```

单元素元组（注意逗号）

```
In [212... print((5),(5,)) # 需要逗号，否则它是一个整数
```

```
5 (5,)
```

使用 tuple() 建立元组，即将其它类型转换为元组

```
In [213... print(tuple('hello'))
```

```
('h', 'e', 'l', 'l', 'o')
```

元组的索引方式与字符串、列表相同。
注意元组的不可变性。

```
In [214... tuple1[0] = 0
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[214], line 1
----> 1 tuple1[0] = 0

TypeError: 'tuple' object does not support item assignment
```

元组是不可变的，不能直接修改元素，可以通过重新赋值来改变整个元组，或通过变成列表后进行修改。

(5) 字典

字典 (Dictionary) 是 Python 的另一个重要对象。它通过键名 (key，一般是字符串，也可以是整数) 对键值 (keyvalue，可以是 Python 的任意对象) 进行索引，可通过多种方式定义。

```
In [215... dict1 = dict(name="Alice", age=25)
print(dict1)
```

```
{'name': 'Alice', 'age': 25}
```

```
In [216... dict2 = {"name": "Alice", "age": 25}
print(dict2)
```

```
{'name': 'Alice', 'age': 25}
```

```
In [217... keys = ["name", "age"]
values = ["Alice", 25]
dict3 = dict(zip(keys, values))
print(dict3)
```

```
{'name': 'Alice', 'age': 25}
```

zip函数可将不同的顺序序列进行一一配对，生成一个zip对象。注意：字典的键名具有唯一性，相同的键名会被去重。

```
In [218... dict2 = {"name": "Alice", "age": 25, "name": "Ben"}
print(dict2)
```

```
{'name': 'Ben', 'age': 25}
```

字典的索引，键值索引，通过索引可对键值进行修改。

```
In [219... print(dict2['name'])
```

```
Ben
```

字典的附带属性和函数

```
In [220... print(dir(dict2))
```

```
['__class__', '__class_getitem__', '__contains__', '__delattr__', '__delitem__',
 '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__ge
titem__', '__getstate__', '__gt__', '__hash__', '__init__', '__init_subclass__',
 '__ior__', '__iter__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__or_
__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__ror__', '__seta
ttr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'clear', 'cop
y', 'fromkeys', 'get', 'items', 'keys', 'pop', 'popitem', 'setdefault', 'update',
'values']
```

常用到.keys()、.update()和.pop()，分别为字典的所有键名、增加字典元素和删除字典元素。

```
In [221... print(dict2.keys())
```

```
dict_keys(['name', 'age'])
```

```
In [222... dict2.update({'gender': 'male'})
print(dict2)
```

```
{'name': 'Ben', 'age': 25, 'gender': 'male'}
```

```
In [223... dict2.pop('age')
print(dict2)
```

```
{'name': 'Ben', 'gender': 'male'}
```

(6) 集合

集合 (set) 是 Python 内置的数据结构之一，是**无序、不可重复**的元素集合，使用 **大括号 {}** 或 `set()` 函数进行创建。集合是无序的，因此不能使用索引访问。

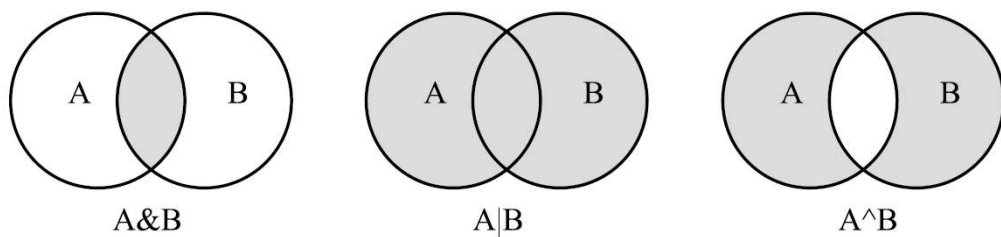
```
In [224...] a = {1,1,1,1,2,3,4,5}
            print(a)
```

{1, 2, 3, 4, 5}

```
In [225...] b = set([3,4,5,6,7,8,9])
            print(b)
```

{3, 4, 5, 6, 7, 8, 9}

集合运算



```
In [226...] print(a & b) #交集
```

{3, 4, 5}

```
In [227...] print(a | b) #并集
```

{1, 2, 3, 4, 5, 6, 7, 8, 9}

```
In [228...] print(a ^ b) #异或
```

{1, 2, 6, 7, 8, 9}

(7) 布尔型

布尔型有 True 和 False 两个值

非0整数也是True, 0是False

比较运算、成员运算和逻辑运算可产生布尔值，布尔值间用逻辑运算

```
In [229...] print(1<2)
```

True

```
In [230...] 'N' in 'Nanjing'
```

Out[230...] True

```
In [231...] a = [1,2]
            b = a #b和a共用了内存
            print(a is not b)
```

False

```
In [232...] print(bool(-1),bool(100),bool(0))
```

True True False

(8) 其他类型

Python 中还包括“None”，即空型。也可使用“class”语句自定义类型，后面学习的各类模块均广泛涉及各类模块的自定义类型，比如：Numpy 的 Narray、Xarray 的 DataArray、Pandas的 DataFrame 等，均属于自定义类型。