

Python-2025-1-9

变量类型

```
1 intT=1      # int整型
2 floatT=1.0  # float浮点型 多了小数点的整型
3 strT="1"    # str字符串
4 listT=[1]   # list列表
5 tupleT=(1,) # tuple元组 一旦创建，就不能修改其内容。你不能添加、删除或修改元组中的元素
6 dictT={1:1} # dict字典
7 setT={1}    # set集合 set的元素只能出现一次
8 boolT=True  # bool布尔型
9 #python的类型大小是内存分块储存，所以没有int/short/longlong的区别，float也是，没有double
```

数组

```
1 # 数组的下标从0开始计数，假设一个数组大小为n那么该数组：
2 #      [1,2,3,4,5,6,...,n]
3 # 那么他的下标则为
4 #      [0,1,2,3,4,5,...,n-1]
5 # 倒序的下标从-1开始，-1表示最后一个元素，-2表示倒数第二个元素，以此类推
6 #      [-n,-(n-1),-(n-2),-(n-3),...,-3,-2,-1]
7 arr=[] #定义一个空列表
8 arr.append(1) #向arr列表中添加一个元素1 添加元素是在尾部添加
9 # arr: [1]
10 arr.clear() #清空arr列表
11 # arr: []
12 arr.extend([1,2,3]) #向arr列表中添加三个元素1,2,3 将1, 2, 3直接加入arr尾部
13 # arr: [1,2,3]
14 arr.index(2) #返回arr列表中元素2的索引
15 # 返回 1
16 arr.count(2) #返回arr列表中元素2的个数
17 # 返回 1
18 arr.insert(1,4) #在arr列表下标为1中插入元素4 之后的下标往后移1
19 # arr: [1,4,2,3]
20 arr.remove(2) #删除arr列表中的元素2
21 # arr: [1,4,3]
22 arr.pop() #删除arr列表中的最后一个元素 无参数默认是最后一个
```

```

23 # arr: [1,4]
24 arr.pop(1) #删除arr列表中的下标为1的元素 有参数则按参数来
25 # arr: [1]
26 arr.extend([1234,54,765,3,2,1,4,5,6,7,8,9,0]) #向arr列表中添加13个元素
27 # arr: [1, 1234, 54, 765, 3, 2, 1, 4, 5, 6, 7, 8, 9, 0]
28 arr.sort() #对arr列表进行排序
29 # arr: [0, 1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 54, 765, 1234]
30 arr.reverse() #对arr列表进行逆序 等同于 arr=arr[::-1]
31 # arr: [1234, 765, 54, 9, 8, 7, 6, 5, 4, 3, 2, 1, 1, 0]

```

切片/区间

区间有三个元素：[起始下标:终止下标:步长] 如果不写步长，则默认为1

如果步长为正数，则从左到右遍历数组，如果步长为负数，则从右到左遍历数组

终止下标取不到，即不包含终止下标对应的元素，但包含起始下标对应的元素（前闭后开）

如果步长为正数，则起始下标默认为0，终止下标默认为数组的长度

如果步长为负数，则起始下标默认为数组的长度，终止下标默认为-1

如果起始下标大于等于终止下标，则返回空数组

```

1 arr=[1,2,3,4,5,6,7,8,9,0] #定义了一个长度为10的数组
2 print(arr[0:5]) #输出数组的前5个元素
3 # arr: [1, 2, 3, 4, 5]
4 print(arr[5:10]) #输出数组的后5个元素
5 # arr: [6, 7, 8, 9, 0]
6 print(arr[0:10:2]) #输出数组的偶数位置的元素
7 # arr: [1, 3, 5, 7, 9]
8 print(arr[1:10:2]) #输出数组的奇数位置的元素
9 # arr: [2, 4, 6, 8, 0]
10 print(arr[::-2]) #输出数组的偶数位置的元素
11 # arr: [1, 3, 5, 7, 9]
12 print(arr[1::-2]) #输出数组的奇数位置的元素
13 # arr: [2, 4, 6, 8, 0]
14 print(arr[::-1]) #输出数组的逆序
15 print(arr[-1:-11:-1]) #输出数组的逆序
16 # arr: [0, 9, 8, 7, 6, 5, 4, 3, 2, 1]

```

range的格式为 range(起始下标, 终止下标, 步长)

range的规则跟切片的规则一样，包含起始下标，不包含终止下标

range(10)表示从0到9，步长为1

range(1, 10)表示从1到9，步长为1

range(1, 10, 2)表示从1到9，步长为2

range(10, 1, -1)表示从10到2，步长为-1

range的返回值类型是range类型，可以通过list()函数转换为列表类型，如list(range(10))则返回：

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

集合

set是一个无序的不重复元素序列

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

```
1 set1 = set()
2 print(set1)    # set()
3 # 创建一个非空集合
4 set2 = {1, 2, 3, 4}
5 print(set2)    # {1, 2, 3, 4}
6 # 通过set()函数创建集合
7 set3 = set("hello")
8 print(set3)    # {'e', 'l', 'o', 'h'} 随机顺序
9 # set的基本操作
10 # 1.添加元素
11 set1.add(1)
12 print(set1)    # {1}
13 set1.add(2)
14 print(set1)    # {1, 2}
15 set1.update([2, 3, 4])
16 print(set1)    # {1, 2, 3, 4}
17 # 2.删除元素
18 set1.remove(2)
19 print(set1)    # {1, 3, 4}
20 # set1.remove(5) # 不存在会发生错误
21 set1.discard(5) # 不存在不会发生错误
22 print(set1)    # {1, 3, 4}
23 set1.pop()
24 print(set1)    # {3, 4}
25 # 3.计算集合元素个数
26 print(len(set1)) # 2
27 # 4.清空集合
28 set1.clear()
29 print(set1)    # set()
30 # 5.判断元素是否在集合中
31 print(1 in set2) # True
32 print(1 not in set2) # False
33 # 集合的交集、并集、差集、对称差运算
```

```

34 # 1.交集
35 set4 = {1, 2, 3, 4, 5}
36 set5 = {4, 5, 6, 7, 8}
37 print(set4 & set5) # {4, 5}
38 print(set4.intersection(set5)) # {4, 5}
39 # 2.并集
40 print(set4 | set5) # {1, 2, 3, 4, 5, 6, 7, 8}
41 print(set4.union(set5)) # {1, 2, 3, 4, 5, 6, 7, 8}
42 # 3.差集
43 print(set4 - set5) # {1, 2, 3}
44 print(set4.difference(set5)) # {1, 2, 3}
45 # 4.对称差运算
46 print(set4 ^ set5) # {1, 2, 3, 6, 7, 8}
47 print(set4.symmetric_difference(set5)) # {1, 2, 3, 6, 7, 8}
48 # 判断子集和超集
49 # 1.判断子集
50 set6 = {1, 2, 3}
51 set7 = {1, 2, 3, 4, 5}
52 print(set6 <= set7) # True
53 print(set6.issubset(set7)) # True
54 # 2.判断超集
55 print(set7 >= set6) # True
56 print(set7.issuperset(set6)) # True
57 # 集合的遍历
58 for i in set7:
59     print(i, end=" ") # 1 2 3 4 5
60 # 集合的生成式
61 set8 = {x for x in "hello" if x not in "abc"}
62 print(set8) # {'e', 'l', 'o', 'h'}

```

字典

字典的形式：{键1:值1,键2:值2,...}

字典的键是唯一的，值不唯一

字典的键可以是数字、字符串、元组，但不能是列表、字典

字典是无序的

字典的值可以是任意数据类型，键只能是不可变数据类型

```

1 dic={}
2 # 字典的增加：字典[键]=值
3 dic['name']='张三'
4 # dic={'name':'张三'}
5 # 字典的删除：del 字典[键]

```

```
6 del dic['name']
7 # dic={}
8 # 字典的修改: 字典[键]=新值
9 dic['name']='李四'
10 # dic={'name':'李四'}
11 # 字典的查找: 字典.get(键)
12 print(dic.get('name'))
13 # 李四
14 # 字典的遍历: for k,v in 字典.items():
15 for k,v in dic.items():
16     print(k,v)
17 # name 李四
18 # 字典的长度: len(字典)
19 print(len(dic))
20 # 1
21 # 字典的清空: 字典.clear()
22 dic.clear()
23 # dic={}
24 #=====
    =====
25 dic['name']='王五'
26 dic['age']=18
27 # dic={'name':'王五','age':18}
28 # 字典的复制: 字典.copy()
29 dic1=dic.copy()
30 # dic1={'name':'王五','age':18}
31 # 字典的键: 字典.keys()
32 dic.keys()
33 # ['name', 'age']
34 # 字典的值: 字典.values()
35 dic.values()
36 # ['王五', 18]
37 # 字典的键值对: 字典.items()
38 dic.items()
39 # [('name', '王五'), ('age', 18)]
40 # 字典的排序: sorted(字典.items(),key=lambda x:x[0])
41 sorted(dic.items(),key=lambda x:x[0])
42 # [('age', 18), ('name', '王五')] #按键排序
43 # 字典的反转: {v:k for k,v in 字典.items()}
44 tmp={v:k for k,v in dic.items()}
45 # {'王五': 'name', 18: 'age'}
46 # 字典的更新: 字典1.update(字典2)
47 dic1.update(tmp)
48 # dic1={'name': '王五', 'age': 18, '王五': 'name', 18: 'age'}
49 # 字典的推导式: {k:v for k,v in 字典.items()}
50 {k:v for k,v in dic.items()}
51 # {'name': '王五', 'age': 18}
```

```
52 # 字典的推导式: {k:v for k,v in 字典.items() if 条件 else}
53 {k:v for k,v in dic.items() if v=='王五'}
54 # {'name': '王五'}
```

控制转移/函数/基础类

```
1 # if else
2 if True:
3     print('True')
4 elif False:
5     print('False')
6 else:
7     print('1')
8 # 循环
9 for i in range(10):
10     print(i)
11 while True: # 条件为真即循环 该循环无限循环
12     print('True')
13     break
14 # 函数
15 def func1(): # 定义函数
16     print('func1')
17 func1() # 调用函数
18 # 类
19 class Student:
20     name = ""
21     mscore = 0
22     cscore = 0
23     escore = 0
24     def __init__(self, name): # 构造函数, 初始化对象
25         self.name = name
26     def getSum(self): # 定义方法
27         return self.escore + self.cscore + self.mscore
28 name="Tom"
29 stu = Student(name)
30 stu.mscore = 90
31 stu.cscore = 80
32 stu.escore = 70
33 print(stu.getSum())
34 # 240
```