

# day2数据类型的操作和方法

## 1.字符串（str）的操作（不可变类型）

### 1.什么是字符

电子计算机或无线电通信中字母、数字和各种符号的统称。

### 2.什么是字符串

由数字、字母、下划线组成的一串字符。

### 3.字符串的作用

记录描述性质的数据，比如人的名字、性别、家庭地址、公司简介

### 4.字符串的定义

在引号内按照从左到右的顺序依次包含一个个字符，引号可以是单引号、双引号、三引号

#### 实例

```
name = '大海'

name1 = "大海1"

info = '''在苍茫的大海上
狂风.....
...'''
```

#### 1.字符串的简单操作

```
print(name)
print(name1)
# 打印2个变量
print(name,name1)

# 字符串里面要有引号
print('my name is "dahai"')
print("my name is 'dahai'")

name4 = 'abcdef'
#      012345
name5 = '大海'
# 索引从0开始，现实中书本的页码从1开始
# 提取单个字符
# 取出第一个英文字符
```

```
print(name4[0])

print(name4[1])
print(name4)

print(name4[5])
print(name4[-1])

print(name5[0])
```

## 5.字符串的操作（总结）

### 1.字符串的操作(查)

#### 1.1.提取单个字符

```
msg = 'hello world'
#      0123456789十
print(msg[0])
print(msg[1])
```

#### 1.2.提取多个字符

```
# 查
# 提取多个字符
#2、切片(顾头不顾尾，步长)查找字符串当中的一段值
# [起始值:终止值:步长]
#   相当于切黄瓜 一节一节

msg = 'hello world'
# #      0123456789十
print(msg[4])
print(msg[5])
# # 默认1
print(msg[0:5])
print(msg[0:5:1])
print(msg[0:5:2])
# # 提取不会改变原值
print(msg)
# # 了解
print(msg[::-1])
print(msg[10:5:-1])
print(msg[-1:5:-1])
```

#### 1.3.长度len方法（查字符串字符个数）

```
# # 3、长度len方法 可以计算长度
print(len(msg))
```

#### 1.4.成员运算

```
# 成员运算in和not in: 判断一个子字符串是否存在于一个大的字符串中
# # 返回布尔类型 True False
#
print('dahai' in 'dahai aaa')
print('dahai' not in 'dahai aaa')
print('xialuo' not in 'dahai aaa')
```

### 1.5.查找子字符串在大字符串的那个位置: find,index (起始索引)

```
# 查找子字符串在大字符串的那个位置 (起始索引)
msga = 'hello dahai is dsb dahai'
print(msga.find('dahai'))
# 没找到会返回-1
print(msga.find('aaaaaaa'))

print(msga.index('dahai'))
# # 没找到会报错
print(msga.index('aaaaaaa'))
```

### 1.6.统计一个子字符串在大字符串中出现的次数: count

```
msga = 'hello dahai is dsb dahai'
# #
print(msga.count('dahai'))
```

### 1.7.判断一个字符串里的数据是不是都是数字: isdigit 返回布尔值

规律带有is开头的方法，返回的是布尔值

```
mun = '1818'
muna = 'a1818'
# # T F
print(mun.isdigit())
print(muna.isdigit())
```

### 1.8.判断每个字符是不是都是字母 isalpha

```
# 判断每个字符是不是都是字母 isalpha
mun2 = 'aaa'
mun3 = '22aaa'
# # T F
print(mun2.isalpha())
print(mun3.isalpha())
```

字符串里面全是数字，是可以转换成整型

```
# 字符串里面全是数字 是可以转换乘整型
n = '111'
n1 = '111aa'
print(type(n))
print(n)
# # 可以变成真正的数字
print(type(int(n)))
print(int(n))
# # 不能转 必须要字符串里面全是数字
int(n1)
```

## 1.9.比较开头/结尾的元素是否相同: startswith,endswith

```
# 比较开头的元素是否相同 startswith
# 比较结尾的元素是否相同 endswith
mm = 'dahai xialuo'
# # T F
print(mm.startswith('dahai'))
print(mm.startswith('d'))
print(mm.startswith('da'))
print(mm.startswith('a'))
print(mm.endswith('o'))
print(mm.endswith('luo'))
print(mm.endswith('xialuo'))
```

## 1.10.判断字符串中的值是否全是小写的/大写的: islower,isupper

```
# 判断字符串中的值是否全是小写的 islower
# 判断字符串中的值是否全是大写的 isupper
letter2 = 'ABC'
letter3 = 'abc'
letter4 = 'aAbc'
# T F
print(letter2.isupper())
print(letter2.islower())
print(letter3.isupper())
print(letter3.islower())
print(letter4.isupper())
print(letter4.islower())
```

## 2.字符串的操作(增)

### 2.1.字符串拼接

```
# # 增
# # 字符串拼接
# # 这个是一个字符串
print('dahai'+dsb')
# 这个是2个字符串 一行打印
print('dahai',dsb')
```

## 2.2.字符串的方法进行增加

### 1.format格式化拼接字符串

```
print('my name is {}'.format('大海'))
print('my name is {},my age is {}'.format('大海',18))
print('my name is {0},my age is {1}'.format('大海',18))
print('my name is {1},my age is {0}'.format('大海',18))
```

### 2.join(它是把一个列表里面的字符串 拼接起来)

```
str1 = '真正的勇士'
str2 = '敢于直面惨淡的人生'
str3 = '敢于正视淋漓的鲜血'
list1 = [str1,str2,str3]
print(list1)
# 可以把列表里面的元素组合成字符串
print(''.join(list1))
print(', '.join(list1))
print('*****'.join(list1))
```

## 3.字符串的操作(删)

```
# 删
name1 = '大海'
del name1
print(name1)
```

## 4.字符串的操作(改)

### 4.1.字符串字母全部变大写和变小写: lower,upper

```
# 改
# 字符串字母全部变大写和变小写 lower,upper
# 不可变类型
msg1 = 'abC'
print(msg1)
print(id(msg1))
# msg1.lower() 它会产生一个新的值 这个值是变的
# print(msg1.lower())
# 新值
msg2=msg1.lower()
```

```
print(msg2)
print(id(msg2))
# 原值
print(msg1)
print(id(msg1))
```

## 不可变类型的理解

```
# 字符串是不可变类型
# 不可变类型      值变  id也变

# 注意字符串进行改变需要重新赋值，所以它也是不可变类型，它的原值的变量不会变，
# 只是做了一个方法改变了它的值，重新赋值给一个新的变量
```

## 4.2.把第一个字母转换成大写：capitalize

```
letter = 'abcd'
#
letter1=letter.capitalize()
#
# # 新值
print(letter1)
#
# # 原值
print(letter)
```

## 4.3.把字符串切分成列表：split 默认空格字符切分

```
# 把字符串切分成列表 split 默认空格字符切分
msgg = 'hello world python'
#
a=msgg.split()
# # 默认以空格切分
print(a)
print(msgg)
# 可以切分你想要的字符 比如*
msgg = 'hello*world*python'
a1=msgg.split('*')
print(a1)
print(msgg)
```

## 切分split的作用

针对按照某种分隔符组织的字符串，可以用split将其切分成列表，从而进行取值

```
msggg = 'root:123456'
#      01234
# print(msggg[0:4])
#
print(msggg.split(':'))
print(msggg.split(':')[0])
print(msggg.split(':')[1])
```

#### 4.4.替换: replace(存在的字符, 新的字符,个数)

```
msgggg = 'dahai你好'
print(msgggg.replace('你','我'))
# 默认是所有
print(msgggg.replace('a','b'))
# 指定个数
print(msgggg.replace('a','b',1))
```

#### 4.5.去掉字符串左右两边的字符: strip 不写默认是空格字符, 不管中间的其他的字符

```
user = '          user          '
print(user)
#
print(user.strip())
```

#### 4.6.添加多余的字符: center,ljust,rjust (了解)

```
# dahai放中间 * 补充 整体 11字符
print('dahai'.center(11,'*'))
# # 1 指的是 dahai 放 那边
print('dahai'.ljust(11,'*'))
print('dahai'.rjust(11,'*'))
```

### 5.字符串的转义

有些\字母的字符带有含义

\n的意义是竖向换行符

\t的意义是横向换行符 相当于一个tab

```
# 字符串的转义
# 字符串的转义 加了 \ 字符不再表示它本身的含义
# 常用的 \n \t
# \n 竖向换行符

print('hello\npython')
# # \t 横向换行符 相当于一个tab
print('hello\tpython')
```

## 字符串的反转义

```
# # 反转义
# # 第一种 加了 \ 字符不再表示它本身的含义 每个都要加 \
print('hello\\npython hello\\tpython')
# # 第二种 加了 r 字符不再表示它本身的含义 只要在前面写一次
print(r'hello\npython hello\tpython')
```

## 总结

字符串是不可变类型，不能强行改它的原值

所有的方法全是产生了新值

## 2.数字类型的操作 (int/float) (不可变类型)

### 注意

所有的加了引号的数据类型都是字符串类型

```
print(type('17'))
```

### 2.1.整数类型 (int)

#### 2.1.2.作用

记录年龄，等级，QQ号，各种号码

#### 2.1.2.定义

```
age = 18
print(age)
print(type(age))
```

### 2.2.浮点型 (float)

#### 2.2.2.作用

记录身高、体重weight、薪资

#### 2.2.2.定义

```
weight = 151.2
print(type(weight))
print(weight)
```



## 数字类型是不可变类型

```
# 不可变类型
age = 18
print(age)
print(id(age))
age = 20
print(age)
print(id(age))
```

## 2.3.赋值运算

```
# 赋值运算
# 普通赋值 =
# 加法赋值 +=
# 减法赋值 -=
# 乘法赋值 *=
# 除法赋值 /=
# 取余赋值 %=
# 乘方赋值 **=
# 地板除赋值 //=

# 语法 n = n + XXX 等同于 n += XXX
n = 2
n = n + 3
# 等价
n += 3
# print(n)
n -= 1 #等价于 n=n-1
print(n)

n /= 1 #等价于 n=n/1
print(int(n))
```

## 3.布尔类型(bool)(不可变)

### 3.1.布尔类型的作用

用来作为判断的条件去用

### 3.2.布尔值的含义

布尔值，一个True一个False  
计算机俗称电脑，即我们编写程序让计算机运行时，  
应该是让计算机无限接近人脑，或者说人脑能干什么，  
计算机就应该能干什么，  
人脑的主要作用是数据运行与逻辑运算，此处的布尔类型就模拟人的逻辑运行，  
即判断一个条件成立时，用True标识，不成立则用False标识

### 3.3.代码实例

```
tag = False
print(tag)
print(type(tag))
```

### 3.4.布尔类型的重点知识

所有的数据类型都自带布尔值

```
# 重点知识
# 不仅仅是真假 还有True 无False
# 所有的数据类型都自带布尔值
# 1.None,0,空(空字符串, 空列表, 空字典,)三种情况下布尔值为False,
# 2.其余均为真
tag1 = ' asdasda'
# print(type(tag1))
# print(bool(tag1))

# True满足 False不满足
# if 如果
if tag1:
    print('数据类型自带True')
else:
    print('数据类型自带False')
```

## 4.列表的操作和方法 (list) (可变类型)

### 4.1.对比之前的数据类型

字符串, 数字, 布尔, 复数 都是一个值 (全都是不可变类型)

### 4.2.列表作用

记录/存多个值, 可以方便地取出来指定位置的值, 比如人的多个爱好, 一堆学生姓名

### 4.3.列表定义

在[]内用逗号分隔开多个任意类型的值

```
L = ['dahai',1,1.2,[1.22,'小海']]
#      0      1 2      3
#              0      1
print(L)
print(type(L))
```

### 4.4.列表使用

## 4.4.1.列表的操作（查）

### 1.列表的取个值（单个索引取值）

```
# # 索引从0开始 相当于我们书的页码
print(L[0])
print(L[1])
print(L[0][0])
# 反向取
print(L[-1])
print(L[-1][1])
print(L[-1][1][0])
dahai=L[0]
print(dahai)
```

### 2.列表的取个值（切片，索引范围）

2、切片(顾头不顾尾，步长)

查找列表当中的一段值 [起始值:终止值:步长]

和字符串提取字符一样,只不过字符串取的是字符，列表取的是一个数据类型/元素

```
name = 'dahai'
print(name[0])
L = ['dahai',1,1.2,[1.22,'小海']]
#     0      1  2   3
# 默认步长为1
print(L[0:3])
print(L[0:3:1])
print(L[0:3:2])
print('-----了解')
print(L[::-1])
print(L[3::-1])
print(L[3:0:-1])
```

### 3.查看列表元素有多少个：len

```
print(len(L))
```

### 4.成员运算：in和not in

```
L = ['dahai',1,1.2,[1.22,'小海']]
print('红海' in L)
print('红海' not in L)
print('dahai' not in L)
print('dahai' in L)
```

### 5.查看列表某个元素的个数：count

```
# 查看列表某个元素的个数 count
print(L.count('dahai'))
print(L)
```

## 6.查看列表的某个值所在的下标/索引: index

```
# 在列表中从左至右查找指定元素,
# 找到了返回该值的下标/索引
print(L.index('dahai'))
# 没找到报错
print(L.index('dahaaai'))
```

## 4.4.2.列表的操作(改)

### 1.索引赋值修改

列表是在原值上面修改，所以是可变类型

```
# # 原值 索引改
print(L)
L[0] = '红海'
# # 原值
print(L)
```

字符串做不到在原值上修改所以是不可变类型

```
# name = 'dahai'
# print(name[0])
# 但是字符串不能索引改值 不可变类型
# name[0]='fff'
```

### 可变和不可变的id和值的关系

```
# 值修改, id变 是 不可变类型
# 一个值 字符串, 数字, 布尔, 复数 不可变类型

# 值修改, id不变是可变类型
# 列表是可变类型
# 多个值容器 有序的 列表 可变类型
```

### 规律

```
# 规律列表的修改和增加都不需要重新复制, 直接改变了原值, 所以是可变类型
# 字符串, 数字, 布尔, 复数 都是一个值, 改变需要重新赋值, 都是不可变类型
```

### 2.反序: reverse

```
L.reverse()
print(L)
```

### 3.对列表中的数字排序: sort

```
# sort 排序 对数字
list_num = [1,3,2,5]
# 不写默认是正序
# 默认参数 reverse=False
list_num.sort()
print(list_num)
# 等价
list_num.sort(reverse=False)
print(list_num)
# reverse=True参数是倒序
list_num.sort(reverse=True)
print(list_num)
```

## 4.4.3.列表的操作(增)

### 1.往列表末尾追加一个元素: append

```
L = ['dahai',1,1.2,[1.22,'小海']]
# 原值
print(L)
print(id(L))
# append(元素) 往列表末尾追加一个元素
L.append('蓝海')
# 原值
print(L)
print(id(L))
L.append('蓝海')

print(L)
L.append('蓝海')
print(L)
```

### 2.往列表当中添加多个元素 括号里放列表 也是末尾追加: extend(列表)

```
# extend(列表) 往列表当中添加多个元素 括号里放列表 也是末尾追加
L.extend(['绿海','紫海'])
print(L)
L.extend(['绿海','紫海'])
print(L)
```

### 3.往指定索引位置前插入一个元素: insert(索引, 元素)

```
# insert(索引, 元素)
L.insert(1, '黄海')
print(L)
```

### 4.4.3.列表的操作(删)

#### 1.变量全部删掉

```
# 删除
# 变量全部删掉
del L
print(L)
```

#### 2.索引删除: del

```
# 索引删除
del L[0]
print(L)
```

#### 3.指定删除: remove

```
# # 指定删除
L.remove('紫海')
L.remove('紫海')
print(L)
```

#### 4.从列表里面拿走一个值 (通过索引) : pop

```
# pop # 从列表里面拿走一个值
# # 按照索引删除值
# # 默认是删除最后一个
res1=L.pop()
print(L)
# # 返回值最后一个元素
print(res1)

res2=L.pop(0)
print(L)
# # 返回值第一个元素
print(res2)
```

#### 5.清空列表: clear

```
# 清空列表clear
L.clear()
print(L)
```

## 5.元组的操作 (tuple) (不可变类型)

### 5.1.元组作用

记录多个值，当多个值没有改的需求，此时用元组更合适

### 5.2.元组定义

在()内用逗号分隔开多个任意类型的值

```
t = (1,2,'大海',(2,3),['红海',2,3])
#   0 1 2      3      4
#                   0      1 2
print(t)
print(type(t))
```

### 5.3.元组的简单使用

#### 5.3.1.取值方式和列表一样

```
# 简单使用
print(t[0])
```

#### 5.3.2.不能修改元组内的元素

```
# t[0]= 5
# t[4] = 2
```

#### 5.3.3.元组里面列表的值可以修改 (因为列表可以修改)



```

# # 能不能修改要看数据类型所在的容器是否允许修改
# ['红海', 2, 3] 它属于这个元组 整体不能改
# ['红海', 2, 3] 它里面的数据可以改
print(t[4])
t[4][0]='黄海'
print(t)

```

## 例子

```

L = ['西施', '顾安', '木木', ({'name': [4, 5, '大海']}), '你好丫']]
# #      0      1      2      3
# # 提取大海
print(L[3][0]['name'][2])
# '你好丫'是否可以修改
print(L[3][1])
# 能不能修改要看数据类型所在的容器是否允许修改
# 错误
L[3][1] = '我好'

```

### 5.3.4.元组是不能修改和添加所以元组是不可变类型

```

# 元组是不能修改和添加所以元组是不可变类型
# 如果想修改可以转换成列表
t = (1, 2, '大海', (2, 3), ['红海', 2, 3])
t1 = list(t)
print(t1)
t1[0] = 8
print(t1)
t=tuple(t1)
print(t)

```

### 5.3.5.总结

查

与列表一样索引，切片，长度len，count个数，index查找元素所在索引，成员运算

不能修改(增，改，删)不可变类型

## 6.字典的操作 (dict) (可变类型)

### 6.1.字典作用

记录多个key:value值,优势是每一个值value都有其对应关系/映射关系key,而key对value有描述性的功能

### 6.2.字典定义

在{}内用逗号分隔开多个key:value元素,其中value可以是任意的数据类型,而key通常应该是字符串类型



```
info = {'name': '大海', 'age': 18}
print(info)
print(type(info))
```

## 6.3.字典简单使用

```
# 简单使用
# 通过key进行取值
print(info['name'])
print(info['age'])
```

## 6.4.列表和字典的区别

```
# 列表是依靠索引
# 字典是依靠键值对 # key描述性的信息
```

## 6.5.字典的操作(增)

### 1.直接赋值一个不存在的key和value

```
# 字典的增加操作
info = {'name': '大海', 'age': 18}
# 原值
print(info)
print(id(info))
# # 直接赋值一个 不存在 的key和value
# # 可变类型
info['addr'] = '长沙'
# # 原值
print(info)
print(id(info))
```

字典我在添加的时候也没有进行重新赋值，所以字典是可变类型

列表却不行,添加和修改必须是操作 存在 的索引

### 2.高级添加键值对方法

setdefault(添加，有就不添加，添加的那个字典里面没有就添加)

```

info = {'name': '大海', 'age': 18}
# 有则不动/返回原值, 无则添加/返回新值
# 字典中已经存在key则不修改, 返回已经存在的key对应的value
res=info.setdefault('name', 'xxx')
print(info)
print(res)
# # 字典不存在key则添加"sex": "male", 返回新的value
res2=info.setdefault("sex", "male")
print(info)
print(res2)

```

## 6.6.字典的操作(删)

### 1.清空字典: clear

```

# # clear 清空字典
info.clear()
print(info)

```

### 2.指定删除键值对: del

```

# # del
del info['name']
print(info)

# # 不存在的key会报错
# del info['xx']

```

### 3.pop 删除 返回值是value 实际上就是拿走了字典的value

```

# pop 删除 返回值是value 实际上就是拿走了字典的value
info = {'name': '大海', 'age': 18, 'addr': 'changsha'}
# # 必须指定key
print(info)
res=info.pop('addr')
print(info)
print(res)
# # # 不存在的key会报错
info.pop('xxx')
# # popitem 最后一对键值对删除 字典无序 返回的是一个元组
res1=info.popitem()
print(info)
print(res1)
print(type(res1))

```

## 6.7.字典的操作(改)

### 1.赋值添加键值对

```
# # 改
info = {'name': '大海', 'age': 18}
print(info)
info['name'] = '红海'
print(info)
```

## 2.update添加键值对

```
info.update({'name': 'xiaohai'})
print(info)
```

## 6.8.字典的操作(查)

### 1.通过键值对直接进行查找

```
# 查
print(info['name'])
# # 查一个不存在的key会报错
# print(info['xxx'])
```

### 2.get方法查找value

```
print(info.get('name'))
# # 没有key就返回None, 不会报错
print(info.get('xxx'))
```

### 3.查看所有的key

```
# 取出所有的key
print(list(info.keys()))
```

### 4.查看所有的value

```
# 取出所有的值
print(list(info.values()))
```

### 5.查看所有的键值对

```
# 取出所有的键值对
print(list(info.items()))
```

## 6.成员运算

```
# # 成员运算in和not in:字典的成员运算判断的是key 返回值是布尔类型
print('name'in info)
print('大海'in info)
print('大海'not in info)
```

## 7.字典 len 查看的是键值对的个数

```
print(len(info))
```

## 7.集合操作和方法 (set) (可变类型)

### 7.1.集合的作用

关系运算

### 7.2.集合定义

在{}内用逗号分开个的多个值

#### 注意

- 1.元素不能重复(定义不能这样写相同的)
- 2.集合里面的元素是无序

```
s = {1,2,3}
print(s)
print(type(s))
```

### 7.3.集合的使用

#### 1.关系运算

```
s1 = {'a','b','c'}
s2 = {'b','c','d'}
# # 拿2个集合相同的元素 shift + 7交集符号 交集
print(s1 & s2)
# 拿2个集合所有的元素 并集
print(s1 | s2)
# s1 去 抵消它们的交集 差集
print(s1 - s2)
# s2 去 抵消它们的交集 差集
print(s2 - s1)
```

补充集合里面的每一个值都必须是不可变类型

```
# # 补充
# # 3 每一个值都必须是不可变类型
sss = {'a', 1, 'aa', ('oldname', 'ccc')}
print(sss)
```

## 2.集合的操作(增)

```
s = {1, 2, 3}
# print(s)
# print(id(s))
```

### 1.添加一个值: add

```
# # 增
# # 集合可变类型 add
s.add('小海')
# print(s)
# print(id(s))
```

### 2.添加多个值: update

```
# # # update
s.update(['蓝海', '紫海'])
print(s)
```

## 3.集合的操作(删)

### 1.删 pop 看你的pycharm是怎样无序排列的，从第一个元素删除

```
# # 删 pop 看你的pycharm是怎样无序排列的，从第一个元素删除
s.pop()
print(s)
```

### 2.指定删除remove

```
# # # 指定删除remove
s.remove(2)
print(s)
```

## 4.集合的操作(改)

### 集合转换成列表进行修改

```
# 集合转换成列表
s1 = list(s)
print(s1)
s1[0] = 8
print(s1)
s=set(s1)
print(s)
print(type(s))
```

## 5.集合去重

```
# 集合去重
# 局限性
#1、无法保证原数据类型的顺序
#2、当某一个数据中包含的多个值全部为不可变的类型时才能用集合去重
names = ['dahai', 'xialuo', 'xishi', 'dahai', 'dahai', 'dahai']
s=set(names)
print(s)
l = list(s)
print(l)
# 2、当某一个数据中包含的多个值全部为不可变的类型时才能用集合去重 错误示范
names = ['dahai', 'xialuo', 'xishi', 'dahai', 'dahai', {'name': 'dahai'}, {'name': 'dahai'}]
# 不行
# set(names)
# 要用for循环和if判断去重就可以保证顺序和对可变类型去重
# 总结
# 字符串，数字，布尔，复数 都是一个值，改变需要重新赋值，都是不可变类型
# 容器元组是不可变类型，字典，列表，集合都是可变类型
```

## 8.可变和不可变总结

```
# id变 值也变 是不可变类型
# 完全变成了另一个人 喝了孟婆汤 投胎
# id不变 值变 是可变类型
# 孙悟空 七十二变 孙悟空id没变（灵魂）
```