

数据类型-数字(重点)

```
1 Python3 中有六个标准的数据类型 :
2 Number ( 数字 )
3 String ( 字符串 )
4 #以下后面专门讲解
5 List ( 列表 )
6 Tuple ( 元组 )
7 Sets ( 集合 )
8 Dictionary ( 字典 )
9
10 Number类型:整型和浮点型
11 注意:
12 1.如果是除法运算,总是返回一个浮点型
13 10/2==5.0
14 2.如果要取整,可以使用//
15 10//3==3
16 3.取余用%
17 10%3==1
18 print(10/2)
19 print(10//3)
20 print(10%3)
21
22
23 #案例:定义三个数字类型的变量a,b,c.分别赋值10,20.22,30,然后计算b/a+c的值
24
```

数据类型-字符串介绍

```
1 String类型:
2 str='abc'
3 str2="abc"
4 注意:单引号或双引号包含的都是字符串
5 Python 没有单独的字符类型,一个字符就是长度为1的字符串
6 有时候,字符串中需要包含单引号或双引号,这时候怎么办?
7 单包双,双包单
8 print("我在学习'python'")
9 print('我在学习"python"')
10
```

获取字符串中的字符(重点)

```
1 通过下标获取字符串中的字符
2 mystr="hello"
3 print(mystr[4])#注意不能超过最大下标,不然报错
4
5 案例:定义一个变量,变量赋值'中华人民共和国',取出中国来分别赋值给a,b变量,并打印出来
```

```

6  mystr='中华人民共和国'
7  a=mystr[0]
8  b=mystr[6]
9  print(a)
10 print(b)
11 #注意,下标可以是负数,-1表示最后一个字符
12 b=mystr[-1]
13 print(b)
14
15 #案例:有字符串str='北京,上海,广州,深圳的房价都很高,但是咸鱼都有梦想,超市冷藏柜的50元螃蟹都知道往
    100元一斤的
16 #柜子爬,我们呢?加油!梦想在不远的地方,面包就在眼前!',
17 # 请从上述字符串获取字符组成'我们在深圳,有100的梦想',打印出来

```

转义(重点)

```

1  转义字符不是普通的字符,是有特殊含义的字符,比如\n表示换行,并不会打印出\n字符串
2  print("我是\n程序员")
3  我是
4  程序员
5
6  #如果在转义字符前面加个\,那么转义字符就是字符串的一部分
7  print("我是\\n程序员")
8
9  反斜杠可以用来转义,使用r可以让反斜杠不发生转义
10 print(r'我是\n程序员')#我是\n程序员
11
12
13 #案例:打印字符串'he\tlloworld',一次让\t有效果,一次没有效果
14

```

字符串的连接和重复(重点)

```

1  字符串可以用+运算符连接在一起,用*运算符重复
2  mystr="hello"
3  mystr2="world"
4  print(mystr+mystr2)#helloworld
5
6  print(mystr*3)#hellohellohello
7
8  #案例:定义一个变量,变量的值是'你好,约吗?',再定义一个变量,变量的值是'叔叔,我们不约',然后把这两个变
    量的值连接起来打印出来,然后使用*打印三次
9
10 #案例:从终端输入用户名的值,密码的值,把这两个信息储存到2个变量,然后整合成
11 #'用户名:葫芦娃&密码:xiaohudie',打印出来
12
13 #案例:从终端输入2次123,一个是整型,一个是字符串,分别乘以3

```

字符串不能改变

```
1 字符串中的字符不能改变,如:myst="hello"
2  myst[0]="k"#报错
3
4
5
6  #案例:定义一个变量,变量的值为你的名字,然后尝试修改你的姓,看看有什么反应
```

截取字符串(切片)(重点)

```
1 1.语法:str[起始:结束:步长]
2     说明:
3         起始-结束是区间,步长是截取字符串时跳过的字符数量=步长-1
4         起始,结束,步长都是可写可不写,选取区间属于左闭右开(包头不包尾),结束不包括结束本身
5
6     #记住:切片方向主要看步长是否是正数据还是负数,如果是正数(默认),那么就从左到右
7     #如果是负数,那么就是从右到左
8     如果要切到数据,那么切片方向和起始和结束方向一致
9
10    myst="abcdefg"
11    print(myst[0:6:2])#ace
12
13    myst="abcdefg"
14    print(myst[2:])#从下标为2的位置切到最后
15
16    print(myst[::-1])#gfedcba
17
18    #案例:给定一个字符串aStr,请反转它
19    #案例:有字符串'123456789',打印单数出来
20    #案例:有字符串'hello world 123 abc',打印出world和abc
21
22
```

判断字符串

```
1  print(a.isdigit())#a为纯数字返回True,否则返回False
2  print(a.isupper())#a有大写字母且没有小写字母返回True,否则返回False
3  print(a.islower())#a有小写字母且没有大写字母返回True,否则返回False
4  print(a.isspace())#a为纯空格时返回True,否则返回False
5  print(a.isalpha())#a为纯字母时返回True,否则返回False
6  print(a.isalnum())#a为字母或数字时返回True,否则返回False
7  注意:
8  isupper或islower在isalnum前面判断的话,如果输入的是字母和数字,那么会进入到isupper或islower判断中
9  #案例:从终端输入一个密码,要求只能是字母或数字,如果符合要求打印True,不符合要求打印False
10 #案例:从终端输入电话号码,判断输入是否合法,是打印True,不合法打印False
```

多行字符串(重点)

```
1 使用三引号(''或''')可以指定一个多行字符串
2
3 如果多行注释前面有变量来接,那么就变为多行字符串
4
5 mystr=''
6 print("xxx")
7 print("xxxxxx")
8 '''
9
10 mystr=''sfsfdsf
11 dfasfd
12 sdfasdfa
13 fdafsf
14 '''
15
16 print(mystr)
```

级联字符串(重点)

```
1 按字面意义级联字符串,如"this " "is " "string"会被自动转换为this is string
2 print("this" "is" "string")#thisisstring
3
4 #字符串综合案例:有字符串'192.168.88.66:8808?
5 id=01&user=maker&passwd=123abc&ipthon=1386688666&
6 email=7634773@qq.com&qq=7634773&wx=weixi\\n123abc'
7 1. 请用多行字符串方式定义变量
8 2. 请获取ip地址和端口号,并验证是否为纯数字,然后拼接起来,用冒号分开ip和端口号,,并打印出来
9 3. 获取wx的值,并打印
10 4. 获取user和passwd的值,并拼接好打印出来
11 5. 获取电话号码,并判断是否为纯数字,并打印
```

字符串的处理

```
1 字符串处理函数有
2 1. 获取字符串长度
3 2. 获取子串出现的次数
4 3. 替换
5 4. 截取
6 5. 获取子串的位置
7 6. indxe
8 7. 修剪字符串
9 8. 修饰字符串
10 9. 检查字符串
11 10. 对齐字符串
```

获取字符串长度(重点)

```
1 1.len(str):通过len函数计算字符串的长度（重点）
2 print(len("hello"))#5
3
4 #案例:获取字符串'软件测试,自动化测试'的长度,结合切片把自动化测试截取出来赋值给一个变量,然后打印
5
```

获取子串出现的次数(重点)

```
1 2.mystr.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
2 mystr="helloworld123abchelloabcworld"
3 cstr="abc"
4 print(mystr.count(cstr))#2
5
6 mystr="sfauserflsajfsafsfuserlfjlasjfsafsaf"
7 print(mystr.count("user",4,8))#0
8
9 #案例:有字符串'dsafasuserfasfafspasswdkxdfesfafasuserfdsfa',判断该字符串中是否有user和
   passwd
10
```

替换(重点)

```
1 3.newstr=mystr.replace(str1, str2 [, max]):把 将字符串中的 str1 替换成 str2,如果 max 指
   定,则替换不超过 max 次（重点）
2 mystr="123456abcfs1fjlsabcfdasfsaaafdsf"
3 newstr=mystr.replace('a','k')#把字符串中的a替换为k
4 print(newstr)#123456kbcfs1fjlskbcfdksfskkkfdsf
5
6 mystr="123456abcfs1fjlsabcfdasfsaaafdsf"
7 newstr=mystr.replace('a','k',2)#把字符串中的a替换为k
8 print(newstr)#123456kbcfs1fjlskbcfdasfsaaafdsf
9
10
11 #案例:获取字符串str="曾今有一段真诚的爱情摆在我面前,我没有珍惜,我很后悔,如果上天再给我一次机会",
12 #替换字符串中所有的'我'为'俺',但保留最后一个'我'
```

截取(重点)

```
1 4.split(str="", num=string.count(str)):以 str 为分隔符截取字符串,如果 num 有指定值,则仅截
   取成num+1 个子字符串
2
3 mystr="fsdsafjlsjfkflsjjsfjslfjs user=maker sdfasfsdfdsafasfk1j1"
4
5 print(mystr.split())#['fsdsafjlsjfkflsjjsfjslfjs', 'user=maker',
   'sdfasfsdfdsafasfk1j1']
6 mylist=mystr.split()#默认是以空隔为标记
```

```

7  print(mylist[1])#user=maker
8
9  mystr="sfsafsafsaf:user=maker:fsasfsfsa"
10 print(mystr.split(":",1))#['sfsafsafsaf', 'user=maker:fsasfsfsa']
11
12 #案例:有字符串'192.168.1.22:8808?id=88&user=maker&passwd=123456'
13 #获取id,user,passwd的值,并拼接打印
14 mystr='192.168.1.22:8808?id=88&user=maker&passwd=123456'
15 mylist=mystr.split("=")
16 ps=mylist[3]
17 mylist2=mylist[1].split('&')
18 id=mylist2[0]
19 mylist3=mylist[2].split('&')
20 user=mylist3[0]
21 print(id+user+ps)

```

获取子串的位置(重点)

```

1  5.str.find(子串,开始位置,结束位置):从字符串中查找子串所在位置, 没找到返回-1
2  mystr="helloworlduserpassworldidpost"
3  print(mystr.find('user'))#10,是u的下标
4  n=mystr.find('user')
5  print(mystr[n:14])#user
6
7  #案例:有字符串'192.168.88.66:8808?id=01&user=maker&passwd=123abc&ipthon=13866888666&
8  email=7634773@qq.com&qq=7634773&wx=weixi\n123abc',获取ipthon的值

```

indxe

```

1  6.index(str, beg=0, end=len(string)):跟find()方法一样, 只不过如果str不在字符串中会报一个异常
2  #案例:有字符串'百度,华为,腾讯,阿里,华子,京东,美图秀秀',查询是否有华子.然后截取出华子来,打印出来
3
4
5  print(mystr.index('user'))#10
6

```

修剪字符串(重点)

```
1 7.获得字符串修剪左右两侧空格,strip(要删除的字符)
2 str3 = '    jbb@126.com    '
3 print(str3.strip())
4
5 mystr="    maker    "
6 print(mystr.strip())#maker
7
8 mystr="xxxxxxmakerxxxx"
9 print(mystr.strip('x'))#maker
10
11 #案例:获取'xxxuser=makerxxxxxx'中的user=maker
12
```

修饰字符串(重点)

```
1 8.join: 返回通过指定字符连接序列中元素后生成的新字符串
2 sep = "-";
3 str = ("a", "b", "c"); # 字符串序列
4 print(sep.join(str));#a-b-c
5 #案例:用户输入了字符串' 安妮 abc123 女 ',请获取'安妮','abc123','女'
6 #并最终打印用户名:安妮-密码:abc123-性别:女
7
8 seq=':'
9 # mystr=("20",'54','10')
10 mystr=["20","54","10"]
11 print(seq.join(mystr))#20:54:10
12
```

检查字符串

```
1 9.检查字符串
2 # 检查字符串是否以指定的字符串开头
3 print(str1.startswith('He')) # True
4 # 检查字符串是否以指定的字符串结尾
5 print(str1.endswith('!')) # True
6 # 获得字符串首字母大写
7 print(str1.capitalize()) # Hello, world!
8 # 获得字符串变大写
9 print(str1.upper()) # HELLO, WORLD!
10 #mystr中所有大写转小写
11 print(str1.lower())
12 #把字符串的每个单词首字母大写
13 print(str1.title()) #Hello,world
14 #案例:把字符串'HELLO,WORLD!'变为小写,然后把单词的首字母变为大写
15 #案例:有字符串'192.168.18.22:8080',检测该字符串是否以192.开头,80结尾
```

对齐字符串

```
1 10. 对齐字符串
2 # 将字符串以指定的宽度居中并在, 两侧填充指定的字符
3 print(str1.center(50, '*'))
4 # 将字符串以指定的宽度靠左放置, 左侧填充指定的字符
5 print(str1.ljust(50, '*'))
6 # 将字符串以指定的宽度靠右放置, 左侧填充指定的字符
7 print(str1.rjust(50, '*'))
8
9 str1="hello"
10 print(str1.center(20, '&'))#&&&&&&hello&&&&&&
11 print(str1.ljust(20, '&'))#hello&&&&&&&&&&&&&&&&
12 print(str1.rjust(20, '&'))#&&&&&&&&&&&&&&&&hello
13
14 #案例:打印以下图形
15     **
16 **
17     **
18     **
19 #案例:打印左边50个x,右边50个x,中间是hello
```