

## 03.万恶之源-基本数据类型(int, bool, str)

本节主要内容:

1. python基本数据类型回顾
2. int---数字类型
3. bool---布尔类型
4. str--- 字符串类型

### 一.python基本数据类型

1. int ==> 整数. 主要用来进行数学运算
2. str ==> 字符串, 可以保存少量数据并进行相应的操作
3. bool==>判断真假, True, False
4. list==> 存储大量数据.用[]表示
5. tuple=> 元组, 不可以发生改变 用()表示
6. dict==> 字典, 保存键值对, 一样可以保存大量数据
7. set==> 集合, 保存大量数据. 不可以重复. 其实就是不保存value的dict

### 二. 整数(int)

在python3中所有的整数都是int类型. 但在python2中如果数据量比较大. 会使用long类型. 在python3中不存在long类型

整数可以进行的操作:

bit\_length(). 计算整数在内存中占用的二进制码的长度

十进制	二进制	长度bit_length()
1	1	1
2	10	2
3	11	2
4	100	3
5	101	3
6	110	3
7	111	3

8	1000	4
---	------	---

### 三. 布尔值(bool)

取值只有True, False. bool值没有操作.

转换问题:

```
str => int    int(str)
int => str    str(int)
```

```
int => bool   bool(int). 0是False 非0是True
bool=>int     int(bool)  True是1, False是0
```

```
str => bool   bool(str) 空字符串是False, 不空是True
bool => str    str(bool) 把bool值转换成相应的"值"
```

### 四. 字符串(str)

把字符连成串. 在python中用', ', '"', '"""'引起来的内容被称为字符串.

#### 4.1 切片和索引

1. 索引. 索引就是下标. 切记, 下标从0开始

```
#      0123456 7 8
s1 = "python最牛B"
print(s1[0])    # 获取第0个
print(s1[1])
print(s1[2])
print(s1[3])
print(s1[4])
print(s1[5])
print(s1[6])
print(s1[7])
print(s1[8])
# print(s1[9])    # 没有9, 越界了. 会报错
print(s1[-1])    # -1 表示倒数.
print(s1[-2])    # 倒数第二个
```

2. 切片, 我们可以使用下标来截取部分字符串的内容

语法: str[start: end]

规则: 顾头不顾腩, 从start开始截取. 截取到end位置. 但不包括end

```
s2 = "python最牛B"
```

```

print(s2[0:3]) # 从0获取到3. 不包含3. 结果: pyt
print(s2[6:8]) # 结果 最牛
print(s2[6:9]) # 最大是8. 但根据顾头不顾腚, 想要取到8必须给9
print(s2[6:10]) # 如果右边已经过了最大值. 相当于获取到最后
print(s2[4:]) # 如果想获取到最后. 那么最后一个值可以不给.

print(s2[-1:-5]) # 从-1 获取到 -5 这样是获取不到任何结果的. 从-1向右数. 你怎么数也数不到-5
print(s2[-5:-1]) # 牛b, 取到数据了. 但是. 顾头不顾腚. 怎么取最后一个呢?
print(s2[-5:]) # 什么都不写就是最后了
print(s2[:-1]) # 这个是取到倒数第一个
print(s2[:]) # 原样输出

```

### 跳着截取

```

# 跳着取, 步长
print(s2[1:5:2]) # 从第一个开始取, 取到第5个, 每2个取1个, 结果: yh, 分析: 1:5=>ytho => yh
print(s2[:5:2]) # 从头开始到第五个. 每两个取一个
print(s2[4::2]) # 从4开始取到最后. 每两个取一个
print(s2[-5::2]) # 从-5取到最后. 每两个取一个
print(s2[-1:-5]) # -1:-5什么都没有. 因为是从左往右获取的.
print(s2[-1:-5:-1]) # 步长是-1. 这时就从右往左取值了
print(s2[-5::-3]) # 从倒数第5个开始. 到最开始. 每3个取一个, 结果oy

```

步长: 如果是整数, 则从左往右取. 如果是负数. 则从右往左取. 默认是1

切片语法:

str[start:end:step]

start: 起始位置

end: 结束位置

step: 步长

## 4.2 字符串的相关操作方法

切记, 字符串是不可变的对象, 所以任何操作对原字符串是不会有影响的

### 1. 大小写转来转去

```

s1.capitalize()
print(s1) # 输出发现并没有任何的变化. 因为这里的字符串本身是不会发生改变的. 需要我们重新获取

ret1 = s1.capitalize()
print(ret1)

# 大小写的转换

```

```

ret = s1.lower()    # 全部转换成小写
print(ret)

ret = s1.upper()    # 全部转换成大写
print(ret)

# 应用，校验用户输入的验证码是否合法
verify_code = "abDe"
user_verify_code = input("请输入验证码:")
if verify_code.upper() == user_verify_code.upper():
    print("验证成功")
else:
    print("验证失败")

ret = s1.swapcase()    # 大小写互相转换
print(ret)

# 不常用
ret = s1.casefold()    # 转换成小写，和lower的区别：lower()对某些字符支持不够好。
                        # casefold()对所有字母都有效。比如东欧的一些字母
print(ret)

s2 = "БВБ" # 俄美德
print(s2)
print(s2.lower())
print(s2.casefold())

# 每个被特殊字符隔开的字母首字母大写
s3 = "alex eggon,taibai*yinwang_麻花藤"
ret = s3.title()    # Alex Eggon,Taibai*Yinwang_麻花藤
print(ret)

# 中文也算是特殊字符
s4 = "alex老男孩wusir"    # Alex老男孩Wusir
print(s4.title())

```

## 2. 切来切去

```

# 居中
s5 = "周杰伦"
ret = s5.center(10, "*")    # 拉长成10，把原字符串放中间。其余位置补*
print(ret)

# 更改tab的长度
s6 = "alex wusir\teggong"

```

```

print(s6)
print(s6.expandtabs())    # 可以改变\t的长度，默认长度更改为8

# 去空格
s7 = "  alex wusir  haha "
ret = s7.strip()    # 去掉左右两端的空格
print(ret)

ret = s7.lstrip()    # 去掉左边空格
print(ret)

ret = s7.rstrip()    # 去掉右边空格
print(ret)

# 应用，模拟用户登录。忽略用户输入的空格
username = input("请输入用户名:").strip()
password = input("请输入密码: ").strip()
if username == 'alex' and password == '123':
    print("登录成功")
else:
    print("登录失败")

s7 = "abcdefgabc"
print(s7.strip("abc"))    # defg 也可以指定去掉的元素，

# 字符串替换
s8 = "sylar_alex_taibai_wusir_eggon"
ret = s8.replace('alex', '金角大王')    # 把alex替换成金角大王
print(s8)    # sylar_alex_taibai_wusir_eggon 切记，字符串是不可变对象。所有操作都是产生新字符串返回
print(ret)    # sylar_金角大王_taibai_wusir_eggon

ret = s8.replace('i', 'SB', 2)    # 把i替换成SB，替换2个
print(ret)    # sylar_alex_taSBbaSB_wusir_eggon

# 字符串切割
s9 = "alex,wusir,sylar,taibai,eggon"
lst = s9.split(",")    # 字符串切割，根据,进行切割
print(lst)

s10 = """诗人
学者
感叹号
渣渣"""
print(s10.split("\n"))    # 用\n切割

# 坑

```

```
s11 = "银王哈哈银王呵呵银王吼吼银王"
lst = s11.split("银王") # [' ', '哈哈', '呵呵', '吼吼', ' '] 如果切割符在左右两端. 那么一
定会出空字符串. 深坑请留意
print(lst)
```

### 3. 格式化输出

```
# 格式化输出
s12 = "我叫%s, 今年%d岁了, 我喜欢%s" % ('sylar', 18, '周杰伦') # 之前的写法
print(s12)
s12 = "我叫{}, 今年{}岁了, 我喜欢{}".format("周杰伦", 28, "周润发") # 按位置格式化
print(s12)
s12 = "我叫{0}, 今年{2}岁了, 我喜欢{1}".format("周杰伦", "周润发", 28) # 指定位置
print(s12)
s12 = "我叫{name}, 今年{age}岁了, 我喜欢{singer}".format(name="周杰伦", singer="周润
发", age=28) # 指定关键字
print(s12)
```

### 4. 查找

```
s13 = "我叫sylar, 我喜欢python, java, c等编程语言."
ret1 = s13.startswith("sylar") # 判断是否以sylar开头
print(ret1)
ret2 = s13.startswith("我叫sylar") # 判断是否以我叫sylar开头
print(ret2)

ret3 = s13.endswith("语言") # 是否以'语言'结尾
print(ret3)
ret4 = s13.endswith("语言.") # 是否以'语言.'结尾
print(ret4)

ret7 = s13.count("a") # 查找"a"出现的次数
print(ret7)

ret5 = s13.find("sylar") # 查找'sylar'出现的位置
print(ret5)

ret6 = s13.find("tory") # 查找'tory'的位置, 如果没有返回-1
print(ret6)

ret7 = s13.find("a", 8, 22) # 切片找
print(ret7)

ret8 = s13.index("sylar") # 求索引位置. 注意. 如果找不到索引. 程序会报错
print(ret8)
```

### 5. 条件判断

```

# 条件判断
s14 = "123.16"
s15 = "abc"
s16 = "_abc!@"
# 是否由字母和数字组成
print(s14.isalnum())
print(s15.isalnum())
print(s16.isalnum())
# 是否由字母组成
print(s14.isalpha())
print(s15.isalpha())
print(s16.isalpha())
# 是否由数字组成, 不包括小数点
print(s14.isdigit())
print(s14.isdecimal())
print(s14.isnumeric()) # 这个比较牛B. 中文都识别.
print(s15.isdigit())
print(s16.isdigit())

# 练习. 用算法判断某一个字符串是否是小数
s17 = "-123.12"
s17 = s17.replace("-", "") # 替换掉负号
if s17.isdigit():
    print("是整数")
else:
    if s17.count(".") == 1 and not s17.startswith(".") and not s17.endswith("."):
        print("是小数")
    else:
        print("不是小数")

```

## 6. 计算字符串的长度

```

s18 = "我是你的眼, 我也是a"
ret = len(s18) # 计算字符串的长度
print(ret)

```

注意: len()是python的内置函数. 所以访问方式也不一样. 你就记着len()和print()一样就行了

## 7. 迭代

我们可以使用for循环来便利(获取)字符串中的每一个字符

语法:

for 变量 in 可迭代对象:

pass

可迭代对象: 可以一个一个往外取值的对象

```
s19 = "大家好，我是VUE，前端的小朋友们。你们好么?"
# 用while循环
index = 0
while index < len(s19):
    print(s19[index])    # 利用索引切片来完成字符的查找
    index = index + 1

# for循环，把s19中的每一个字符拿出来赋值给前面的c
for c in s19:
    print(c)

'''
in有两种用法：
    1. 在for中。是把每一个元素获取到赋值给前面的变量。
    2. 不在for中。判断xxx是否出现在str中。
'''
print('VUE' in s19)

# 练习，计算在字符串"I am sylvia, I'm 14 years old, I have 2 dogs!"
s20 = "I am sylvia, I'm 14 years old, I have 2 dogs!"
count = 0
for c in s20:
    if c.isdigit():
        count = count + 1

print(count)
```



