

String

Python 基础入门

回顾

列表

使用 索引

len() max() min() sum()

加法 和 乘法

分片 操作

append() remove() sort()



字符串

引号 中的文本

- ✓ 可以是 单引号、双引号、三引号
- ✓ 字符串中可以包括 任意内容
 - ✓ 字母
 - ✓ 数字
 - ✓ 汉字
 - ✓ 符号
- ✓ 如果不用换行，用 单引号 即可

```
>>> s1 = 'Hello World'
>>> s2 = "Hello World"
>>> s3 = '''Hello
World'''
>>> s4 = """Hello
World"""
```

字符串

字符串 和 列表 有些类似

- ✓ 字符串由很多个 字符 组成
- ✓ 列表由很多个 变量 组成
- ✓ 对字符串使用 `len()`
- ✓ 对字符串使用 索引，只能读 不能改

b	a	n	a	n	a
0	1	2	3	4	5

```
mystr = 'Hello'  
mylist = ['H', 'e', 'l', 'l', 'o']
```

```
print(len(mystr), len(mylist))
```

```
print(mystr[2])  
mystr[2] = 'X'  
print(mystr)
```

```
5 5  
1
```

Traceback (most recent call last):

```
File "/Users/honlan/Desktop/test.py", line 7, in <module>  
    mystr[2] = 'X'
```

TypeError: 'str' object does not support item assignment

查找

使用 `find()` 在字符串中 查找

- ✓ 可以查找某个 字符
- ✓ 如果 找到, 则返回 第一次 出现的 索引
- ✓ 也可以查找某个 更短的字符串
- ✓ 如果 没有找到, 则返回 -1

```
>>> s = 'banana'
>>> pos = s.find('b')
>>> print(pos)
0
>>> pos = s.find('n')
>>> print(pos)
2
>>> pos = s.find('ana')
>>> print(pos)
1
>>> pos = s.find('bab')
>>> print(pos)
-1
```

分片

字符串的 分片 和列表 完全相同

- ✓ 使用 冒号 进行分片
- ✓ 使用 两个 索引
 - ✓ 包括前面的索引
 - ✓ 但不包括后面的索引
- ✓ 只使用 一个 索引
- ✓ 使用 负索引

```
>>> s = 'Hello World'
```

```
>>> s[1:3]
```

```
'el'
```

```
>>> s[:4]
```

```
'Hell'
```

```
>>> s[3:]
```

```
'lo World'
```

```
>>> s[:]
```

```
'Hello World'
```

```
>>> s1 = s[1:-1]
```

```
>>> print(s1)
```

```
ello Worl
```

```
>>> print(s[::-1])
```

```
dlroW olleH
```

字符串

加法 和 乘法

- ✓ 字符串 + 字符串: 连接
- ✓ 字符串 * 整数: 重复

```
>>> 'Hello ' + 'World'
'Hello World'
>>> 'Hello' * 3
'HelloHelloHello'
>>> 3 * 'Hello '
'Hello Hello Hello '
```

练习

查找 并 删除

- ✓ 定义一个长字符串 s1, 今天天气非常非常的好
- ✓ 定义一个短字符串 s2, 天气
- ✓ 在 s1 中查找 s2, 如果包含则删除, 今天非常非常的好
- ✓ 将 s2 改为 非常, 今天天气的好



字符串

分割 和 合并

- ✓ 使用 `split()` 分割 字符串，结果为 列表
- ✓ 使用 `join()` 合并 列表，结果为 字符串
- ✓ 列表中每个元素都应当是 字符串

```
>>> ' '.join([1, 1.2, 'Hello'])
Traceback (most recent call last):
  File "<pyshell#33>", line 1, in <module>
    ' '.join([1, 1.2, 'Hello'])
TypeError: sequence item 0: expected str instance, int found
```

```
>>> s = 'Hello My World'
>>> s.split(' ')
['Hello', 'My', 'World']
```

```
>>> s.split('l')
['He', '', 'o My Wor', 'd']
```

```
>>> s.split('x')
['Hello My World']
```

```
>>> s.split('lo')
['Hel', ' My World']
```

```
>>> ' '.join(['You', 'He', 'She'])
'You He She'
```

```
>>> '*'.join(['You', 'He', 'She'])
'You*He*She'
```

字符串

大小写 转换

- ✓ 使用 `upper()` 将字母 转大写
- ✓ 使用 `lower()` 将字母 转小写
- ✓ 使用 `capitalize()` 将首字母 转大写

```
>>> s = 'Hello 你好'
>>> upper = s.upper()
>>> print(upper)
HELLO 你好
>>> lower = s.lower()
>>> print(lower)
hello 你好
>>> print(lower.capitalize())
Hello 你好
```

■ 加密

使用 加密 保护你的消息

- ✓ 原始消息称为 明文
- ✓ 使用 密钥 进行 加密 和 解密
- ✓ 经过 加密 之后的消息称为 密文
- ✓ 只有自己人才知道 密钥

```
> Please enter a message: Hello world!  
The encrypted message is: mjqqt btwqi!  
>>>
```

加密

Caesar Cipher

将字母 打乱, Cipher 指定了字母之间的 对应关系

Caesar Cipher 通过 平移字母 实现加密



尝试

手动计算

明文为 hello are you ok

- ✓ Key = 1
- ✓ Key = 3
- ✓ Key = 27

手动计算出对应的 密文



A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

实现

加密 单个字母

- ✓ 输入 Key, 可以是 任意整数
- ✓ 输入 单个字母
- ✓ 输出 加密后 的字母

```
alphabet = 'abcdefghijklmnopqrstuvwxyz'
```

```
key = int(input('> Please input the Key: '))  
char = input('> Please input a character: ')
```

```
pos = alphabet.find(char)  
pos = (pos + key) % len(alphabet)
```

```
new_char = alphabet[pos]  
print('The encrypted character is: ', new_char)
```

实现

加密 一段消息

- ✓ 输入 Key，可以是 任意整数
- ✓ 输入 一段消息
- ✓ 输出 加密后 的密文
- ✓ 需要 注意 的问题
 - ✓ 首先将消息统一转为 小写
 - ✓ 处理消息中的 非字母 字符

```
alphabet = 'abcdefghijklmnopqrstuvwxyz'
```

```
key = int(input('> Please input the Key: '))  
msg = input('> Please input a message: ')  
msg = msg.lower()
```

```
new_msg = ''  
for c in msg:
```

```
    pos = alphabet.find(c)  
    if pos == -1:  
        new_msg += c  
    else:  
        pos = (pos + key) % len(alphabet)  
        new_msg += alphabet[pos]
```

```
print('The encrypted message is: ', new_msg)
```

作业

1 解密 消息

- ✓ 在上一页的基础上，添加 解密 的代码
- ✓ 使用 `Key` 和 `new_msg` 得出一开始的明文
- ✓ 在 小写字母 的基础上，增加 大写字母 和 数字

```
# alphabet = 'abcdefghijklmnopqrstuvwxyz'  
alphabet = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'
```


■ 作业

2 你能 看懂 吗

使用 Caesar Cipher 加密的一段消息

*Kyv wLIKyvJK uzJKrEtv zE Kyv NFICu, zJ EFK svKNvvE Czww
rEu uvrKy. sLK NyvE z JKREu zE wIFEK Fw PFL, PvK PFL
uFE'K BEFN KyrK z CFMv PFL*

```
alphabet = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'
```

去实现你想做的任何事情

去实现你想做的任何事情