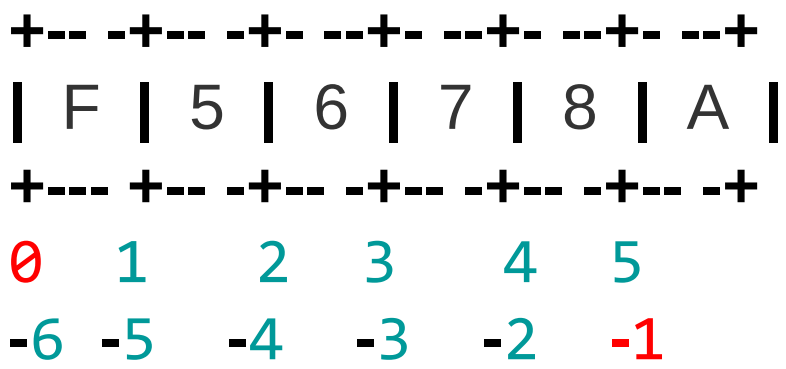


字符串切片是指对操作的对象截取一部分的操作。

切片的语法：[起始:结束:步长]

注意：从"起始"位开始，到"结束"位的前一位结束（不包含结束位本身）。

str[x:y]表示，
从索引x开始取，直到索引y为止，但不包括索引y。

- 由于Python源代码也是一个文本文件，所以，当你的源代码中包含中文的时候，在保存源代码时，就需要务必指定保存为UTF-8编码。当Python解释器读取源代码时，为了让它按UTF-8编码读取，我们通常在文件开头写上这两行：
- *#!/usr/bin/env python3*
- *# -*- coding: utf-8 -*-*
- 第一行注释是为了告诉Linux/OS X系统，这是一个Python可执行程序，Windows系统会忽略这个注释；
- 第二行注释是为了告诉Python解释器，按照UTF-8编码读取源代码，否则，你在源代码中写的中文输出可能会有乱码。

字符编码

- 本着节约的精神,,出现了把Unicode编码转化为“可变长编码”的UTF-8编码。

- UTF-8编码把一个Unicode字符根据不同的数字大小编码成1-6个字节，常用的英文字母被编码成1个字节，汉字通常是3个字节，只有很生僻的字符才会被编码成4-6个字节。如果你要传输的文本包含大量英文字符，用UTF-8编码就能节省空间：

字符	ASCII	Unicode	UTF-8
A	01000001	00000000 01000001	01000001
中	x	01001110 00101101	11100100 10111000 10101101

- 从上面的表格还可以发现，UTF-8编码有一个额外的好处，就是ASCII编码实际上可以被看成是UTF-8编码的一部分，所以，大量只支持ASCII编码的历史遗留软件可以在UTF-8编码下继续工作。

输入函数

<变量>=input('提示信息：')

返回结果是字符串

如何实现输入？input()函数

格式： input（ ）

作用：等待用户用键盘输入数据

常见用法：

	显示内容
input（“请输入购买纪念品的数量：”）	请输入购买纪念品的数量：
a=input（“请输入购买纪念品的数量：”）	从键盘输入一个字符串，赋值给a

如何计算？数据类型转换

格式： int（表达式 ）

作用：把变量转换成整型

常见用法：

```
a=int(input("请输入购买纪念品的数量："))
```

如何实现输出？print()函数

格式： print（表达式）

作用：输出表达式的值

常见用法：

若b=10	显示内容
print（b ）	10
print（ “您购买的纪念品共消费：” ）	您购买的纪念品共消费：
print（ “您购买的纪念品共消费：” ，b）	您购买的纪念品共消费：10

```
>>> print('100+200','=',100+200)
100+200 = 300
```

输出函数

print([object, …][, sep=' '][,end='\n'])

(1)方括号中的项是可选的，可以省略，如省略则取系统的默认值。

(2) object是要输出的对象，可以是常量、变量或表达式等。

(3) sep后面的空格**(**可以指定为其他字符**)**表示每个输出对象之间的分隔符，如果缺省的话，默认值是一个单个的空格。separate[‘sepəreɪt]中文:v. 分隔开

```
>>>print( 5,6,7 )
>>>print( 5,6,7, sep= ' ')
```

(4) end后面的字符串含义为输出文本尾的一个字符串，如果缺省的话，默认值是一个**\n**换行符。如果设为其他字符，如**end= ' '**， 则输出当前行的所有内容后，在末尾加一个空格，不换行接着输出下一个**print()**的输出对象。

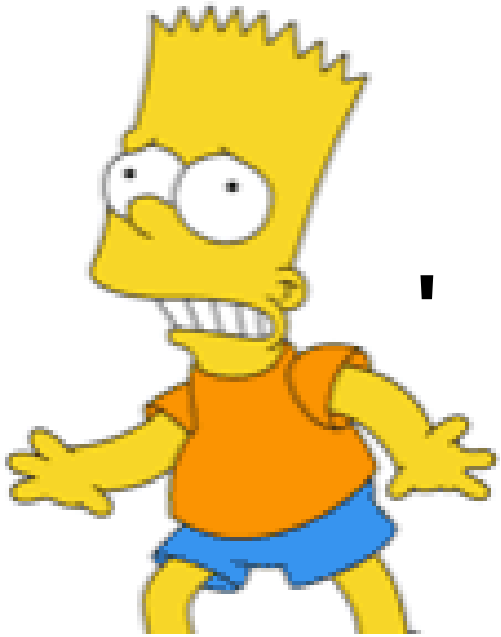
```
>>>print(“hello“, end= ' ' )
>>>print(“world“)
```

(5) 函数支持参数格式化。

经常会输出类似'**亲爱的xxx你好！你的得分是xxx。**'之类的字符串，而xxx的内容都是根据变量变化的，所以，需要一种简便的**格式化字符串的方式**。

'Hi %s, your score is %d.' % ('Bart', 59)

Hi Bart,your score is 59.



' % ' % ( )

print()函数格式化输出

占位符

‘ % ’ % 替换内容

%f　格式化浮点数字，可指定小数点后的精度。

%.2f　是指保留小数点后两位。

% y　是指格式化的对象是y。

```
print("购买的笔记本1和笔记本2的数量均为：",n,"本")
print("可节省的费用为：","%.2f"%y,"元")
```

占位符	替换内容
%d	整数digital
%f	浮点数float
%s	字符串string

```
>>>import math
>>>math.pi
```

```
>>>'%.2f' % math.pi
```

‘3.14’

```
>>> print('%.2f' % math.pi)
```

```
>>> ' %d-%02d' % (3, 1)
```

‘3-01’

%2d是将数字按宽度为2，采用右对齐方式输出，若数据位数不到2位，则左边补空格

拓　展

字符串里面的%是一个普通字符怎么办？这个时候就需要转义，用%%来表示一个%：

```
>>> ' %d  %%'   %  20

'20 %'
```

\n 和 \r都是以前的那种打字机传承来的。

\n代表换行，就是走纸，下一行。

\r 代表回车，也就是打印头归位，回到某一行的开头。

\ 是转义字符

字符串前加r，使得字符串中的\用作普通字符

```
>>> print(r'C:\some\name')
C:\some\name
```

对话中的引号？
” My age is 18.”

单引号　\’
双引号　\”

（3）顺序控制结构的程序执行过程是（　　）依次执行的。

A．自上而下　　B．自下而上　　C．自外到内　　D．由内到外

（4）input( )函数的返回结果是（　　）。

A．数值型　　B．集合　　C．列表　　D．字符串

（5）算术运算符、赋值运算符和关系运算符的运算优先级按从高到低依次为（　　）。

A．算术运算、赋值运算、关系运算

B．算术运算、关系运算、赋值运算

C．关系运算、赋值运算、算术运算

D．关系运算、算术运算、赋值运算

小　结

```
n=int(input("输入购买的笔记本1和笔记本2的本数均为："))
```

50

```
a = 3 * n
b = 5 * n
y = (a+b)*(1-0.8)
print("购买的笔记本1和笔记本2的数量均为：",n,"本")
print("可节省的费用为：","%.2f"%y,"元")
```