

Python输入与输出

基本输入和输出

1. 使用print()函数输出

在Python中，使用内置函数可以将结果输出到控制台上。（对，就是那个黑色的窗口。）

print()函数的基本语法格式如下：

```
print(输出内容)
```

其中，输出的内容可以是数字，文字列（字符串），字符串需要用单引号或者双引号括起来。此类内容将直接输出到控制台，如果是表达式，那么输出的是运行结果。

```
a = 1024 #变量a，值为1024
b = 5 #变量b，值为5
print(9) #输出数字9
print(a * b) #输出a * b 的结果，* 为乘法运算
print("go big or go home") # 输出"go big or go home"，注意引号为英文引号
```

```
PS F:\笔记相关\Python基第二次分享\day01> python .\day01.py
9
5120
go big or go home
PS F:\笔记相关\Python基第二次分享\day01> █
```

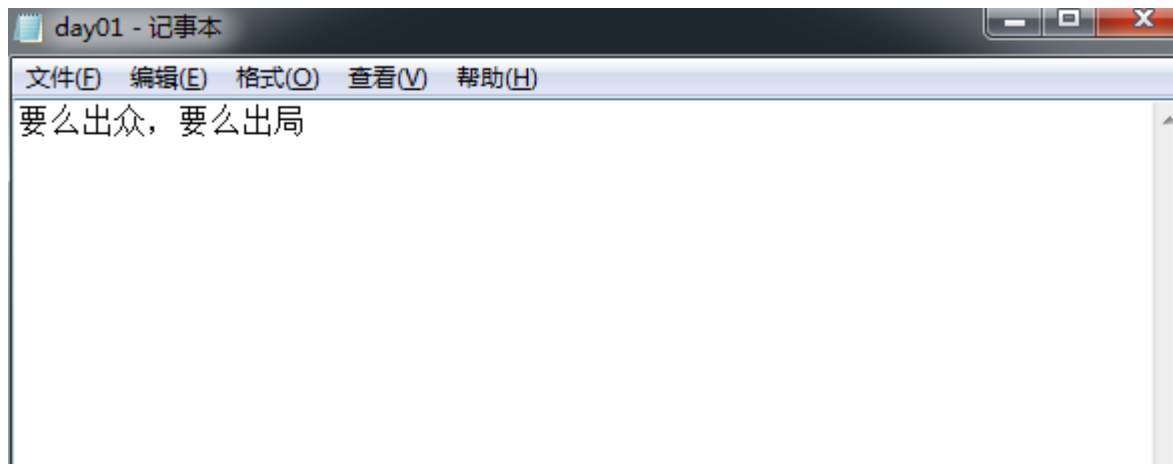
在Python中默认情况下，一个print()函数输出后，会自动换行，如果想要一次输出多个内容而且不换行，可以将要输出的内容用逗号分开

```
print(a, b, "go big or go home") #一次输出多个内容
```

```
1024 5 go big or go home
PS F:\笔记相关\Python基第二次分享\day01> █
```

使用print()函数不但可以输出到屏幕，还可以输出到指定文件中。

```
fp = open(r'F:\day01.txt', 'a+') #打开要写入的文件
print("要么出众，要么出局", file = fp) #写入内容
fp.close() #关闭文件
```



注意点，由于我为了方便管理电脑内容，有些文件夹的路径有中文，所以运行时，会报错。大家联系时，路径要避免中文和空格。

输出当前的年份和日期时间

```
import datetime # 引入Python的日期模块，后面会具体分析

print('当前年份: ' + str(datetime.datetime.now().year)) #输出当前的年，str()是将内容转换成字符串
#输出当前时间，注意代码中的引号，大小写
print('当前时间: ' + datetime.datetime.now().strftime('%y-%m-%d %H:%M:%S'))
```

```
PS F:\笔记相关\Python基第二次分享\day01> python .\printDate.py
当前年份: 2018
当前时间:18-12-27 22:01:55
PS F:\笔记相关\Python基第二次分享\day01> █
```

2. 使用input()函数输入

在Python中，使用input()内置函数，可以接收用户的键盘输入。

基本用法

```
变量名 = input("提示文字")
```

```
tip = input("请输入文字") #获取用户输入的内容，并保存到tip这个变量中
```

在Python3.x版本中，无论你输入的是文字还是数字，最终都是字符串类型，我们可以用type()函数来查看变量的类型。

```
print('tip:', tip, 'tip的类型:', type(tip))
```

```
PS F:\笔记相关\Python基第二次分享\day01> python .\day01_input.py
请输入文字wq
tip: wq tip的类型: <class 'str'>
PS F:\笔记相关\Python基第二次分享\day01> python .\day01_input.py
请输入文字123
tip: 123 tip的类型: <class 'str'>
PS F:\笔记相关\Python基第二次分享\day01> |
```

将str类型的数字转换成数值类型,可以用python的int()函数。

```
tip = int(tip)
```

```
请输入文字123
tip: 123 tip的类型: <class 'str'>
转换后tip: 123 tip的类型: <class 'int'>
PS F:\笔记相关\Python基第二次分享\day01> |
```

作业练习

根据用户输入的出生年份,判断是否成年。关于if判断语句,后面会讲,大家可以先在网上查一下if的用法。很多时候,编程是需要在百度谷歌等网站上进行搜索的。

注释

1. 单行注释

在Python中使用#号来注释,井号后面的内容不会被编译。语法:

```
# 注释的内容
```

```
# 要求输入的年份,必须是4位。如1988
year = int(input('请输入你的出生年份,4位数:'))
```

还可以写成这样

```
year = int(input('请输入你的出生年份,4位数:')) # 要求输入的年份,必须是4位。如1988
```

直接写在语句后面。

2. 多行注释

在Python中并没有单独的多行注释标签,而是使用三个单引号或双引号开始,并以三个单引号或双引号结束。语法:

```
'''
注释的内容
注释的内容注释的内容
'''
或
"""
注释内容
"""
```

注释内容

```
''''
```

这里的引号，开始和结束一定要保持一致。不可以开始用单引号，结束用双引号。

```
'''
Python基础知识分享
作者：Tango
日期：2018年12月27日
'''
```

3. 中文编码声明注释

在Python编写代码时，如果用到指定字符编码类型的中文编码，需要在文件开头加上声明注释。

语法：

```
# -*- coding:编码 -*-
```

或者

```
# coding=编码
```

例如保存文件编码格式为UTF-8，可以使用下面的中文编码注释

```
# -*- coding:utf-8 -*-
```

代码缩进

在Python中不像其他语言用{}来区分代码块。而是用空格来区分，通常是4个空格。如果你用IDE开发的情况下，也可以用tab，因为大部分IDE已经将4个空格绑定在了Tab键上。如果是纯文本如vim等编写代码时，不建议用tab键，因为windows和linux等系统的tab不一样。

代码规范

Python是一个有自己格调的语言。它有自己的编程规范，常见的PEP8.

常见的：

1. 每个import语句只导入一个模块
2. 不要在句末加；号，虽然能编译。
3. 建议每行代码不超过80个字，超过的情况可以使用“\”进行拼接

```
print("这是一段没有意义的文字，dfsdf发抖动对的啊人都是额个人身上个的\
这里有个斜线用来拼接")
```

```
PS F:\笔记相关\Python基础第二次分享\day01> python -i comment.py
请输入你的出生年份，4位数：1234
请输入你的出生年份，4位数：222
这是一段没有意义的文字，dfsf发抖动对的啊人都是额个人身上个的这里有个斜线用来拼接
PS F:\笔记相关\Python基础第二次分享\day01> █
```

运行后，发现并没有换行，而是一起打印出来了。同时也发现我们注释掉的内容也没有打印出来。

命名规范

1. 模块名尽量短小
2. 包名尽量短小
3. 类名使用首字母大写
4. 模块内部的类采用下划线+首字母大写
5. 常量全部大写
6. 变量名尽量要有意义