

Python一本通

信息技术学考教程

第1章 Python语言入门

第一节 Python语言简介

第一节 Python语言简介

Python是一门快乐的语言，学习和使用Python也是一个快乐的过程。与C语言系列和Java等语言相比，Python 更加容易学习和使用，但这并不意味着可以非常轻松地掌握Python。用户熟练掌握和运用Python仍需要通过大量的练习来锻炼自己的思维和熟悉Python 编程模式，同时还要经常关注Python社区的最新资源。

Python是一门优雅的语言。Python语法简洁清晰，提供大量内置对象和内置函数，编程模式非常符合人类的思维方式和习惯。在有些编程语言中需要编写大量代码才能实现的功能，在Python中仅需要调用内置函数或内置对象的方法即可实现。

一、Python语言的特点

1. 简单易学

Python 是一种代表简单主义思想的语言，给人的感觉像是在读英语段落一样。

2. 开源

Python 是FLOSS（自由/开源源码软件）之一，简单地理解就是，用户使用 Python 进行开发和发布自己编写的程序，不需要支付任何费用，也不用担心版权问题。

3. 高级语言

Python 是高级语言，使用 Python 语言编写程序时，无需再考虑一些底层细节方面的问题。

4. 解释型语言

Python 语言写的程序不需要完全编译成二进制代码后才能运行，而是直接从源代码边解释边运行程序，代码与平台无关。因此，Python 程序更加易于移植。

一、Python语言的特点

5.可移植性

Python 是开源的，它已经被移植到许多平台上。避免使用依赖系统的特性，所有 Python 程序都无需修改就可以在好多平台上运行，Python 作为一门解释型的语言，只要为平台提供了相应的 Python 解释器，Python 就可以在该平台上运行。

6.面向对象

Python 既支持面向过程编程，也支持面向对象编程。与其他编程语言（如 C++ 和 Java）相比，Python 是以一种非常强大，而又简单的方式实现的面向对象编程。

7.强大的功能

Python 强大的功能也许才是很多用户支持 Python 的最重要的原因，从字符串处理到复杂的 3D 图形编程，Python 借助扩展模块都可以轻松完成。

8.可扩展性

Python 的可扩展性体现为它的模块，Python 具有脚本语言中最丰富和强大的类库，这些类库覆盖了文件 I/O、GUI、网络编程、数据库访问等绝大部分应用场景。Python 可扩展性一个最好的体现是，需要一段关键代码运行的更快时，可以将其用 C 或 C++ 语言编写，然后在 Python 程序中使用它们即可。

二、简单的Python语言程序实例

在介绍Python语言程序的结构之前，我们先来看一个简单的例子：

例1.1 Hello Python^{浙版}

【题目描述】

编写一个能够输出 Hello Python的程序，这个程序常常作为一个初学者接触一门新的编程语言所写的第一个程序，也经常用来测试开发、编译环境是否能够正常工作。

【输入格式】

无。

【输出格式】

Hello Python

【样例输入】

(无)

【样例输出】

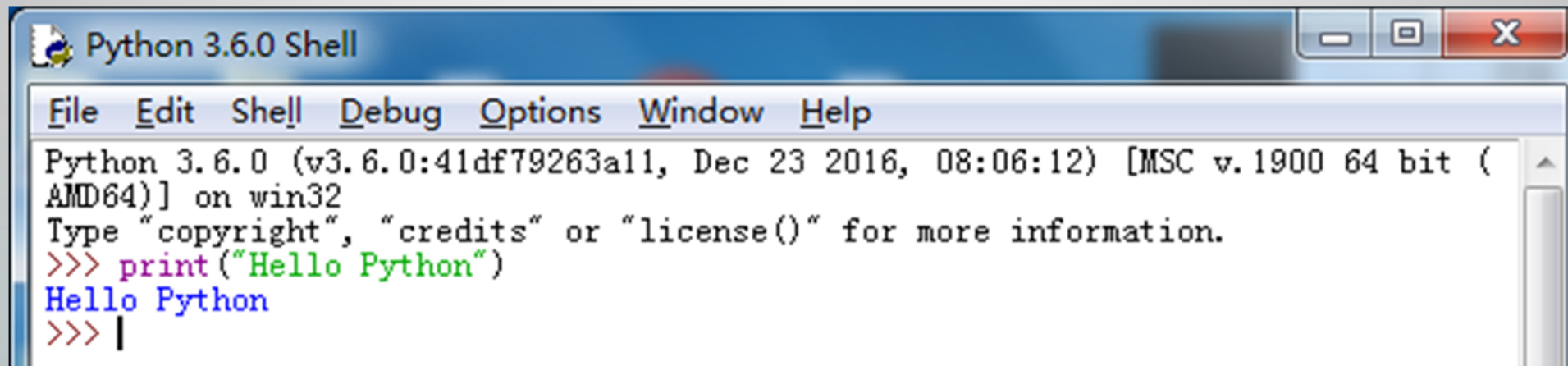
Hello Python

【操作步骤】

1. 在 Shell 中输出 Hello Python 语句。

```
>>> print("Hello Python")
```

```
Hello Python
```



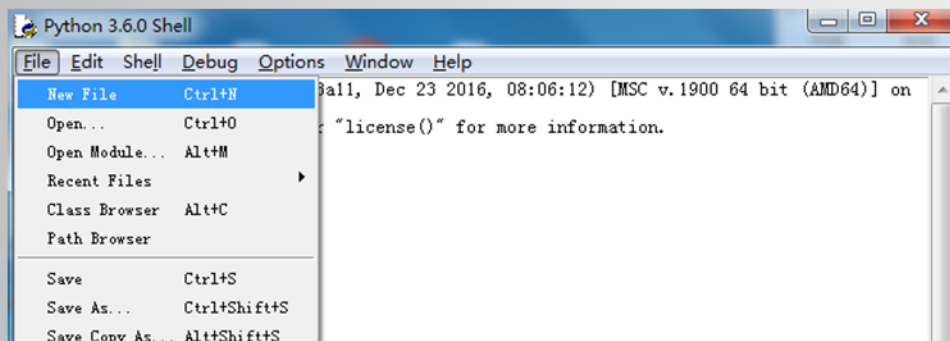
The screenshot shows a window titled "Python 3.6.0 Shell" with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). The main text area displays the following content:

```
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Hello Python")
Hello Python
>>> |
```

【操作步骤】

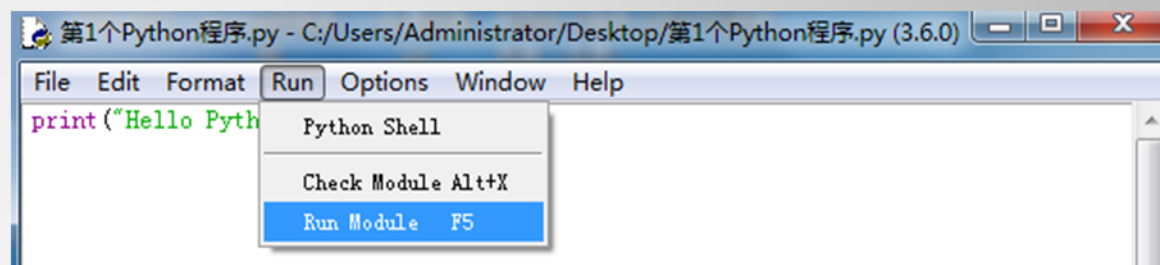
2. 在python文件中输出Hello Python语句。

1) Shell中, New File

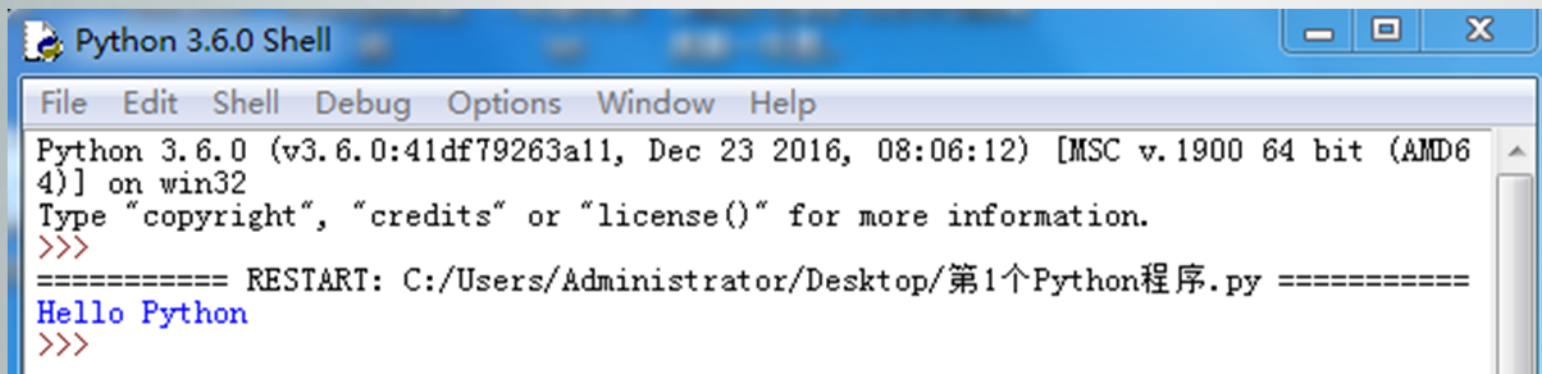


2) 在py文件中写好代码

```
print("Hello Python")
```



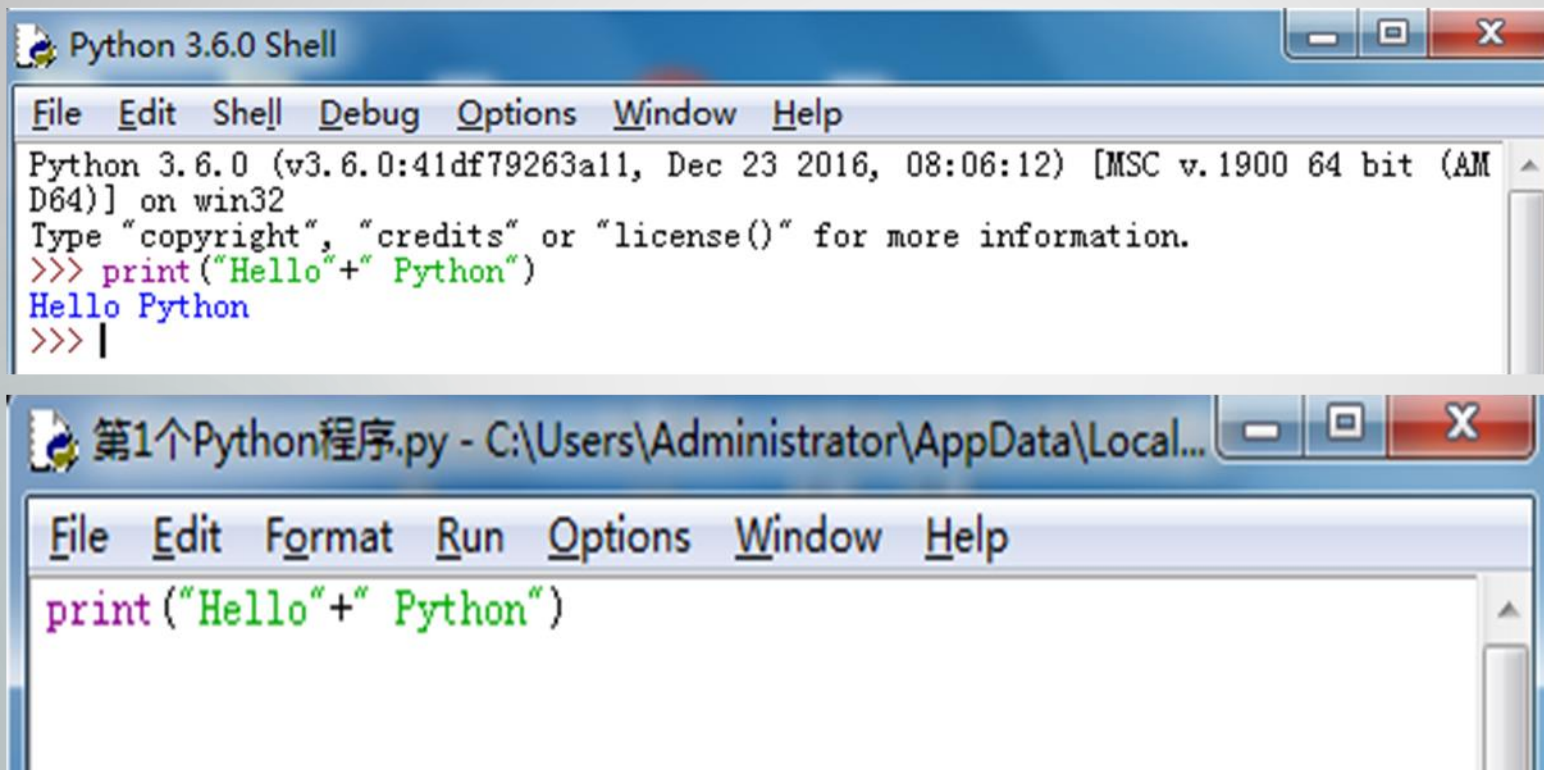
3) 结果



【操作步骤】

3. 输出“Hello Python!”第2种方法。

在上面中可以输入：`print("Hello"+" Python")` #注意Python前有1个空格



The image contains two screenshots. The top screenshot shows a 'Python 3.6.0 Shell' window. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The text area shows the following content: 'Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32', 'Type "copyright", "credits" or "license()" for more information.', and a prompt '>>>'. The user has entered the command `print("Hello"+" Python")` and the output 'Hello Python' is displayed. The bottom screenshot shows a Python IDE window titled '第1个Python程序.py'. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The text area contains the code `print("Hello"+" Python")` with syntax highlighting: 'print' is purple, the opening quote is green, 'Hello' is green, the closing quote is green, '+' is green, ' Python' is green, and the closing parenthesis is purple.

编写程序时，编辑器会以不同的颜色显示程序的不同部分，如紫色表示函数input和print、绿色表示字符串、红色表示注释等。这种功能称为语法高亮，对编写程序很有帮助。

三、Python语言系统的使用

在安装Python后，会自动安装一个IDLE，它是一个Python Shell(用于开发人员与Python交互的“壳”)，程序开发人员可以在Python Shell上使用python语言与Python交互。

本节将以IDLE为例，简单介绍如何使用IDLE编写Python程序。Windows 10系统里，点击开始菜单，找到P->Python 3.6->IDLE(Python 3.6 64bit)

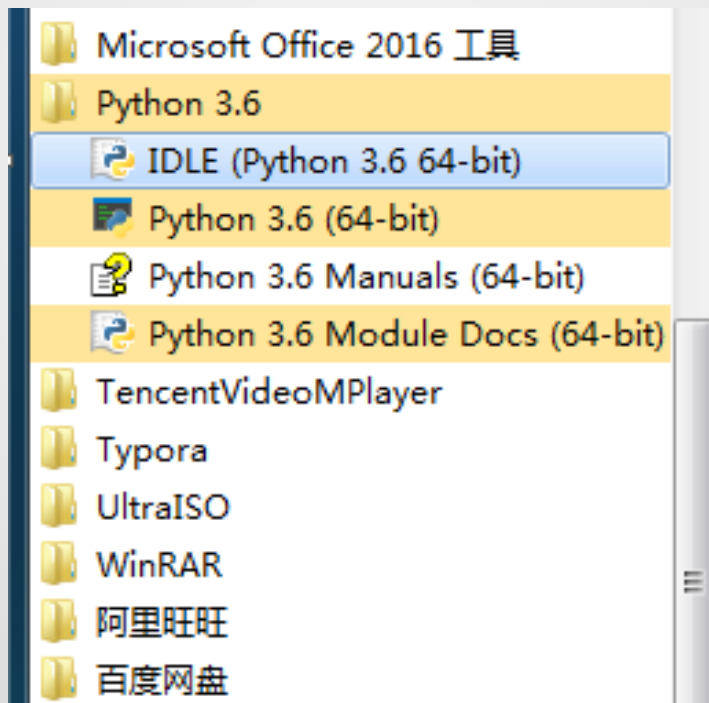


图1 打开IDLE

Windows 7系统里，可以单击系统的开始菜单，然后依次选择“所有程序-> Python3.6 -> IDLE(Python3.6 64-bit)”菜单项，即可打开IDLE窗口，如图2所示。

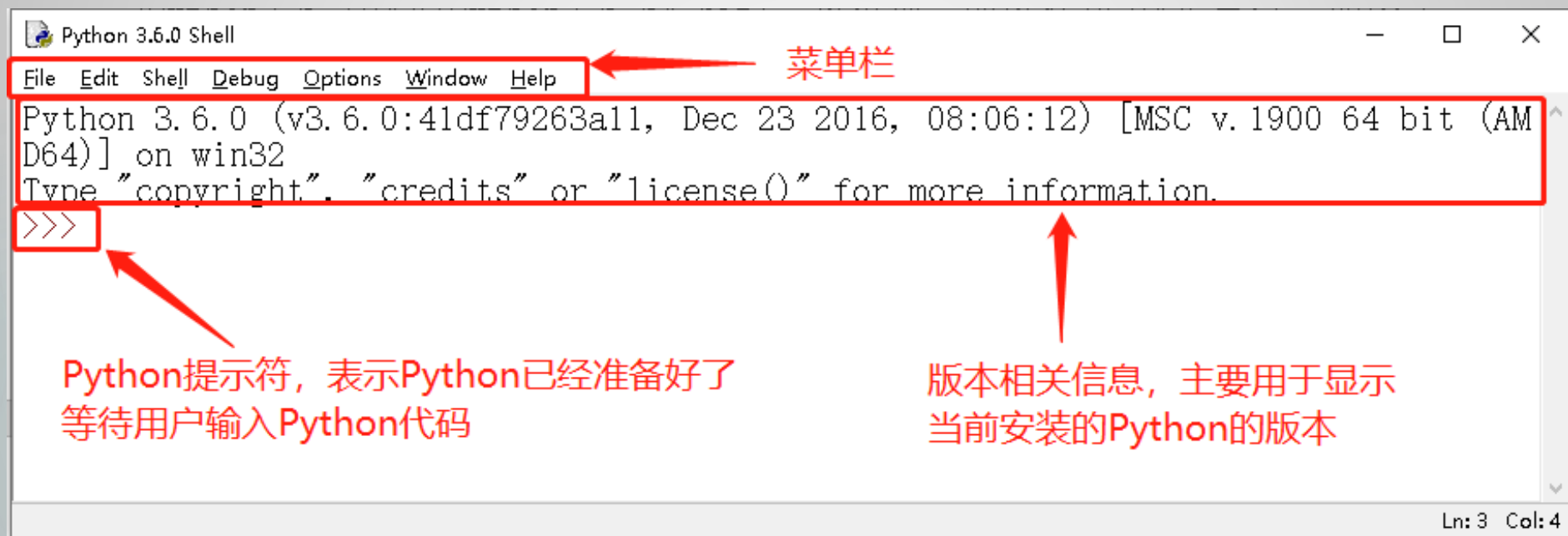
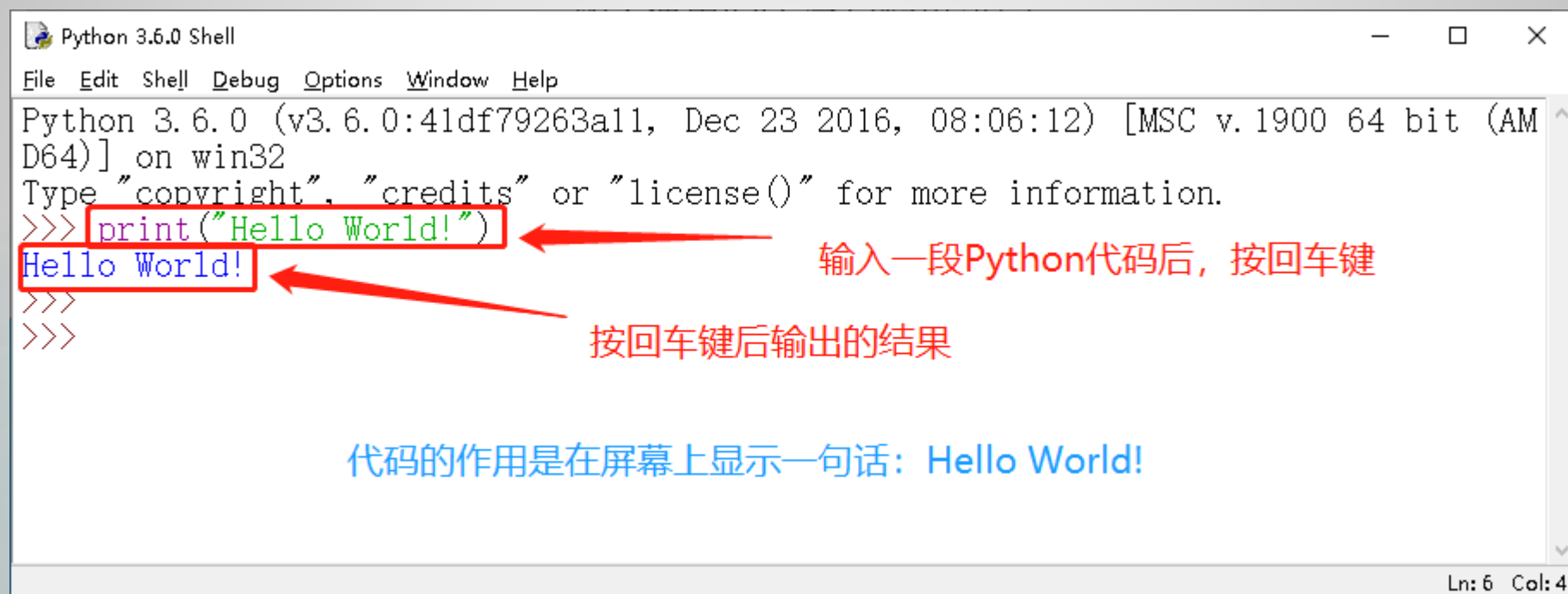


图2 IDLE主窗口

使用**IDLE**输出简单的语句的方法：



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Hello World!")
Hello World!
>>>
>>>
```

输入一段Python代码后，按回车键

按回车键后输出的结果

代码的作用是在屏幕上显示一句话：Hello World!

Ln: 6 Col: 4

图3 使用**IDLE**运行输出语句

实际开发中，通常不能只包含一行代码，当需要编写多行代码时，可以单独创建一个文件保存这些代码，在全部编写完成后一起执行。具体方法如下：

1. 在IDLE主窗口的菜单栏上，选择“File->New File”菜单项（图4），将打开一个新窗口，在该窗口中，可以直接编写Python代码（图5）。

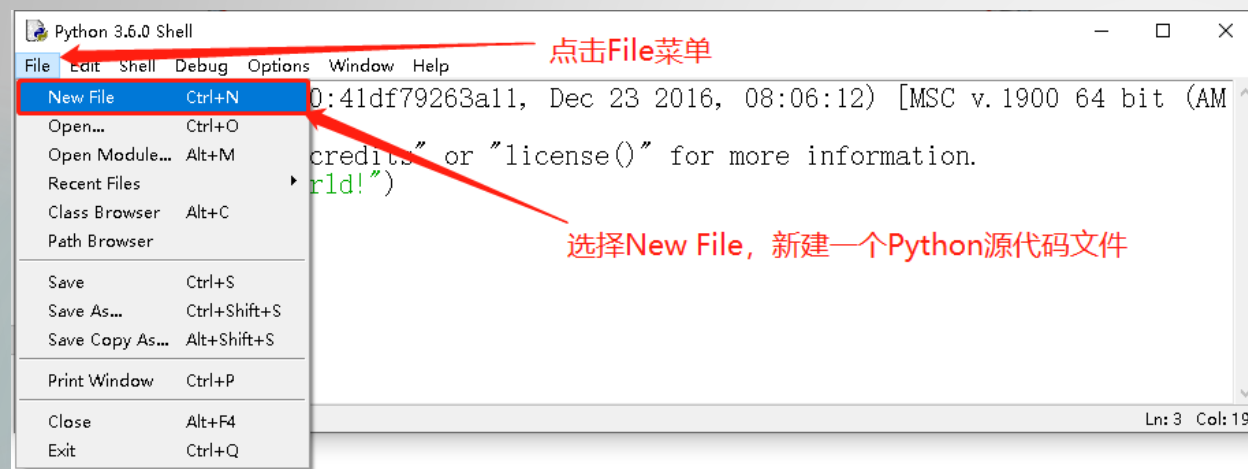


图4 新建Python 源代码文件方法

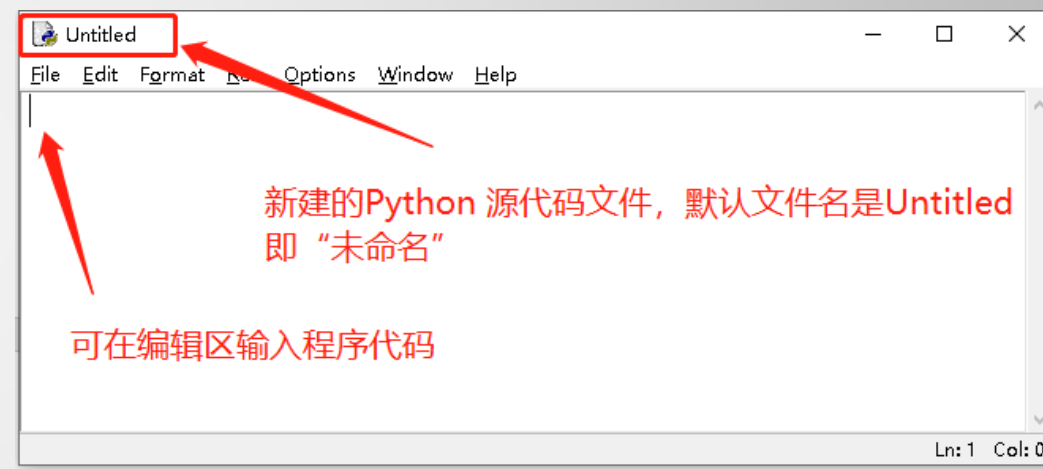


图5 新建Python 源代码文件方法

2. 在输入一行代码后再按下<Enter>键，将自动换到下一行，等待继续输入，如图6所示。

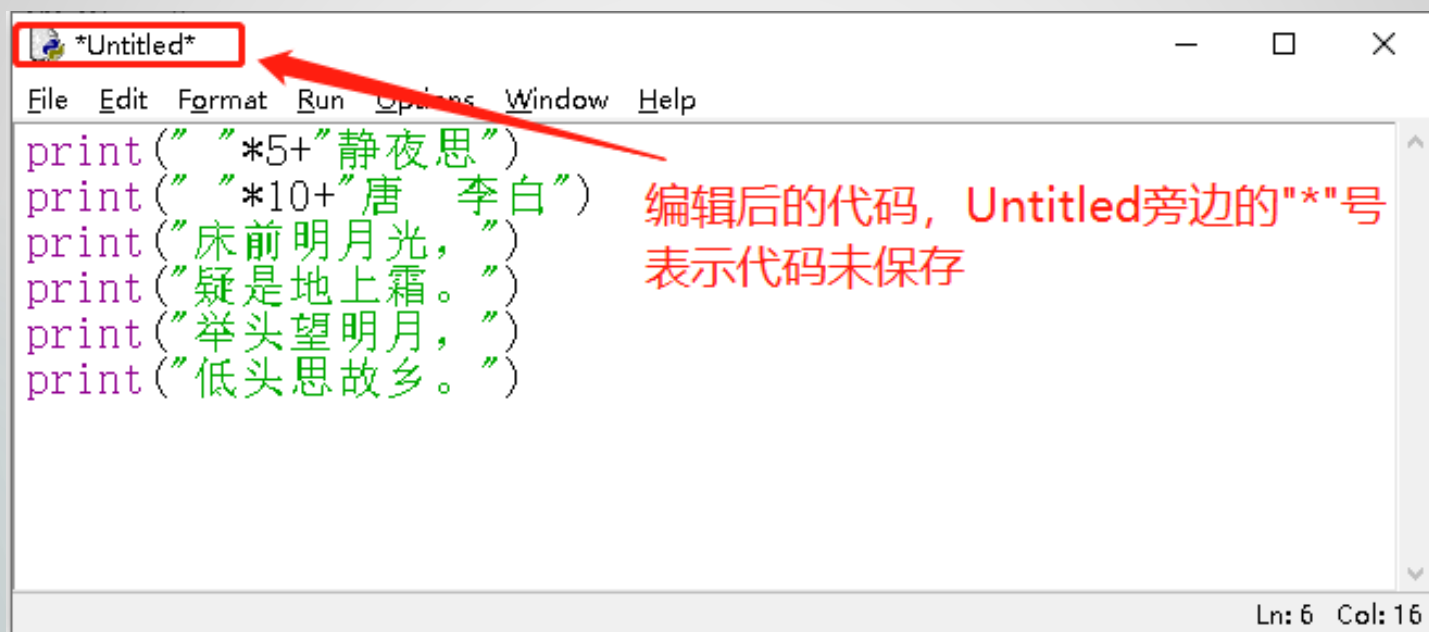


图6 输入多行代码

3. 按下快捷键<Ctrl+S>保存文件，这里将文件名称设置为“静夜思.py”。其中，“.py”是Python文件的扩展名。在菜单栏中选择“Run->Run Module”菜单项（也可以直接按下快捷键<F5>），运行程序，如图7所示。

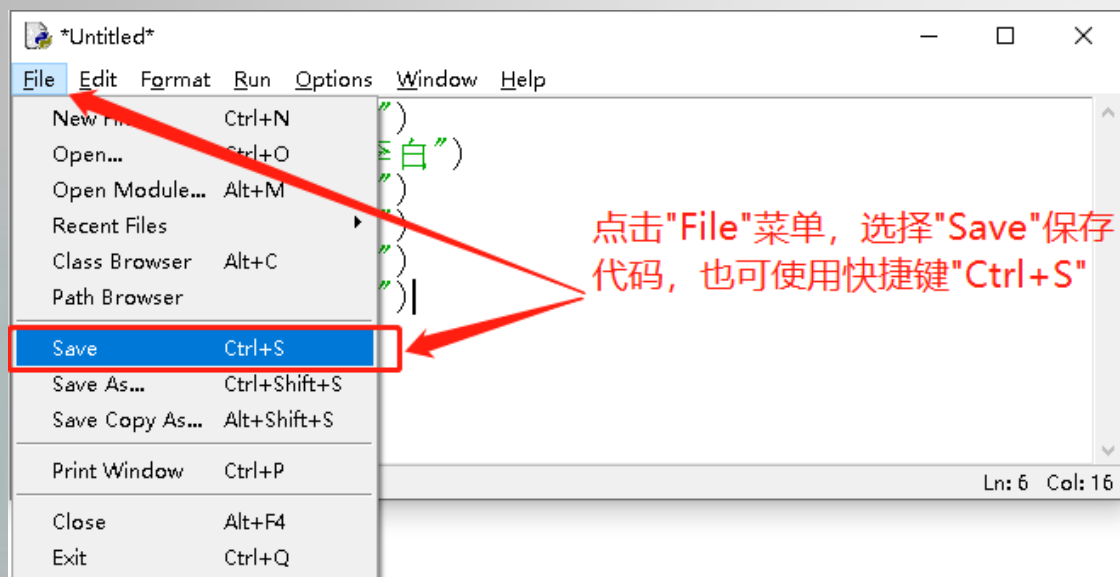


图7-1 保存源文件

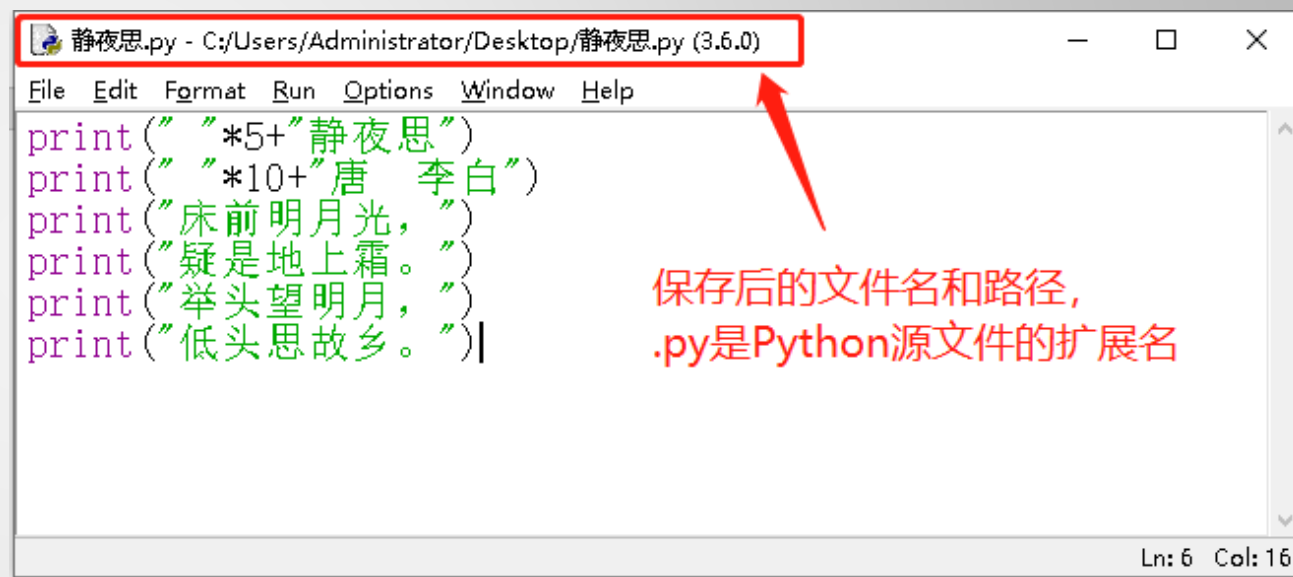


图7-2 保存后的源文件

4. 选择“Run”菜单中的“Run Module”命令，或按“F5”快捷键运行程序，如图8所示：

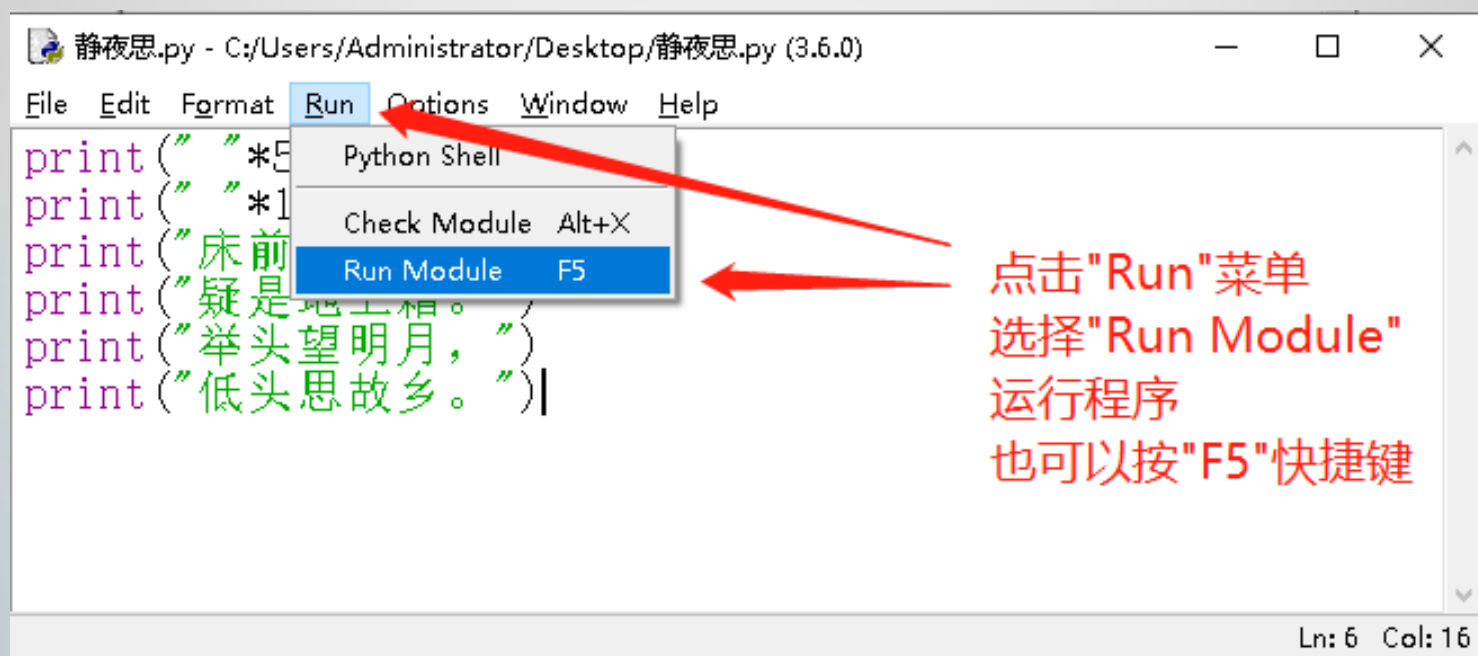
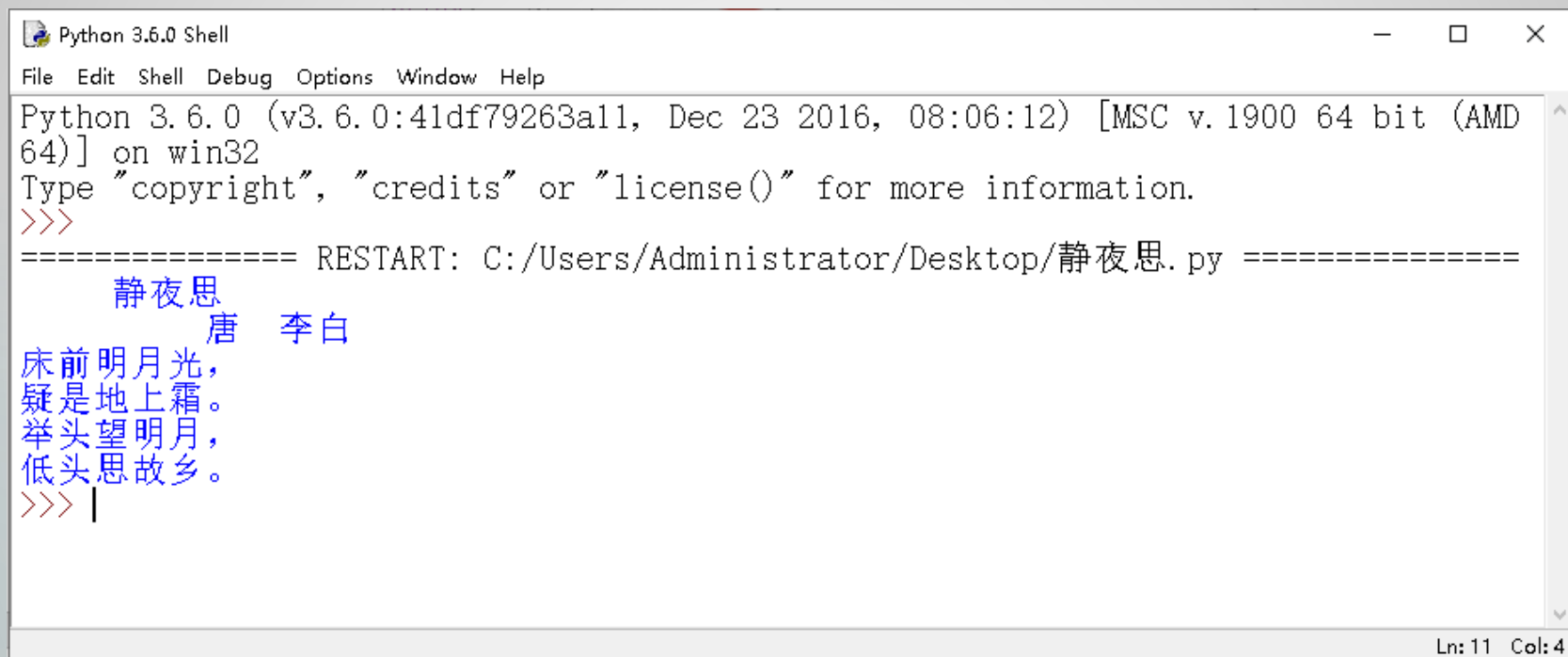


图8 运行程序

5.运行程序后，将自动打开Python Shell窗口显示运行结果，如图9所示。



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD
64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Administrator/Desktop/静夜思.py =====
    静夜思
      唐 李白
床前明月光，
疑是地上霜。
举头望明月，
低头思故乡。
>>> |
Ln: 11 Col: 4
```

图9 运行结果

四、Python IDLE的常用快捷键

在程序开发过程中，合理使用快捷键不但可以减少代码的错误率，而且可以提高开发效率。在IDLE中，可通过选择“Options->Configure IDLE”菜单项，在打开的“Settings”对话框的“Keys”选项卡中查看，但是该界面是英文的，不便于查看。为方便读者学习，下表列出了IDLE中一些常用的快捷键。

快捷键	说明	适用范围
Alt+P	浏览历史命令（上一条）	仅PythonShell窗口可用
Alt+N	浏览历史命令（下一条）	仅PythonShell窗口可用
Alt+/ 	自动补全前面曾经出现过的单词，如果之前有多个单词具有相同前缀，可以连续按下该快捷键，在多个单词中间循环选择	Python文件窗口和Shell窗口均可用
Alt+3	注释代码块	仅Python文件窗口可用
Alt+4	取消代码块注释	仅Python文件窗口可用
Ctrl+Z	撤销一步操作	Python文件窗口和Shell窗口均可用
Ctrl+Shift+Z	恢复上一次的撤销操作	Python文件窗口和Shell窗口均可用
Ctrl+S	保存文件	Python文件窗口和Shell窗口均可用
Ctrl+]	缩进代码块	仅Python文件窗口可用
Ctrl+[	取消代码块缩进	仅Python文件窗口可用
Ctrl+F6	重新启动PythonShell	仅PythonShell窗口可用

IDLE提供的常用快捷键

第二节 Python语言程序结构

第二节 Python语言程序结构

无论做任何事情，都要有一定的方式方法与处理步骤，所谓“无规矩无以成方圆”。计算机程序设计比日常生活中的事务处理更具有严谨性、规范性和可行性。为了使计算机有效地解决实际问题，必须将处理步骤编排好，用计算机能理解的计算机语言编写成“序列”，让计算机自动识别并执行这个“序列”，达到解决实际问题的目的。将处理问题的步骤编排好，用计算机语言组成序列，就是常说的编写程序。在Python语言中，执行每条语句都是由计算机完成相应的具体操作，编写程序是利用Python语句的功能来实现预定的处理要求。“千里之行，始于足下”，我们从简单程序学起，逐步了解和掌握怎样编写程序。

在学习Python语言之前，让我们绕过那些繁琐的语法规则细节，通过一些简单的例题，来熟悉程序的基本组成和基本语句的用法，选手刚接触编程时，多动手模仿是一条捷径。

一、计算机程序

计算机程序就是一组计算机能够识别和执行的指令或语句的序列，是算法的一种描述。

二、程序设计语言

1. 机器语言：机器语言是由二进制的**0**、**1**代码指令构成，能够被计算机直接识别和执行，不同的CPU具有不同的指令系统。但理解和记忆机器语言非常困难，并且容易出错，编程效率极低。

2. 汇编语言：汇编语言是符号化的机器语言，采用英文助记符代替机器指令，比机器语言容易识别和记忆，从而提高了程序的可读性。但汇编语言同样是一种低级语言，需要对寄存器或存储器进行直接操作，开发难度大，效率低。

3. 高级语言：高级语言更接近自然语言，并不特定某一种语言，也不依赖于特定的计算机系统，因而更容易掌握和使用，通用性也更好。比较流行的高级语言有**C/C++**、**Java**、**Python**等。

三、主流程序设计语言

高级语言并不是特指某一种具体的语言，其种类繁多，如FORTRAN、Basic、Visual Basic、C、C++、C#、Java、Python等等。

四、程序的基本结构

根据程序执行的流程，程序可以分为以下3种基本结构，这3种基本结构可以组成各种复杂的程序。

- 顺序结构：按照程序的语句顺序执行，是最简单的程序结构。
- 选择结构：也称为分支结构，根据给定的条件选择执行的程序语句。
- 循环结构：根据给定的条件反复执行相同的程序语句。

五、简单的入门例题

例1.2 A+B问题^{浙版}

【题目描述】

给定两个整数A和B，输出A+B的和。

【输入格式】

两行，每行一个整数。

【输出格式】

输出一个数，即他们的和。

【样例输入】

1

2

【样例输出】

3

【题目分析】

思路很简单，`print`就能实现，其中`a`、`b`用来输入的2个整数，`s`用来存储`a+b`的和，最后输出`s`的值。

【代码实现】

```
a = int(input())    #用input("请输入整数a: ")，输入时有提示
b = int(input())    #用input("请输入整数b: ")，输入时有提示
s = a + b           #计算a+b的和
print(s)            #输出s的值
```

【说明】

(1) `"#"`后面的内容为注释，在程序运行时不执行。注释主要用于对程序代码进行说明，便于程序的理解和维护，目的是提升代码的可读性。

(2) 程序中的`int`、`input`、`print`等都是Python的内建函数。其中，`input`函数的功能是获取用户通过键盘输入的数据，`int`函数是将字符串转换为整数。

(3) `input`函数主要用来接收键盘的输入，返回值为字符串型数据。通常，在输入时可以给出提示信息，例如：提示"请输入整数`a`: "，输入合适的数据并按回车键。

(4) `print`函数主要用于在屏幕上输出一个或多个输出项的值，多个输出项中间用逗号隔开，例如：`print(x, "是奇数")`。

例1.3 温标换算^{沪版}

【题目描述】

利用公式 $C=5 \times (F-32) \div 9$ (其中C表示摄氏温度, F表示华氏温度) 进行计算转化, 输入华氏温度F, 输出摄氏温度C, 要求精确到小数点后2位。

【输入格式】

输入一行, 包含一个实数F, 表示华氏温度。(F $\geq$ -459.67)

【输出格式】

输出一行, 包含一个实数, 表示对应的摄氏温度, 要求精确到小数点后2位。

【样例输入】

41

【样例输出】

5.00

【题目分析】

用程序解决问题的过程是: 首先接收用户输入的华氏温度, 然后使用换算公式计算出相应的摄氏温度, 最后输出摄氏温度的值。

【代码实现】

```
#用input("请输入一个华氏温度:"), 输入时有提示
f = float(input())

#计算出对应的摄氏温度
c = 5*(f-32)/9

#输出结果, 精确到小数点2位
print("%0.2f"%c)
```

【说明】

输出语句中的格式设定为%0.2f, f表示float实数, .2表示小数点后保留2位, 四舍五入。

【上机练习】

Python在线评测OJ: <http://py.ssoier.cn:8088>

例1.4 字符三角形

【题目描述】

给定一个字符，用它构造一个底边长5个字符，高3个字符的等腰字符三角形。

【输入格式】

输入只有一行， 包含一个字符。

【输出格式】

该字符构成的等腰三角形，底边长5个字符，高3个字符。

【样例输入】

#

【样例输出】

#

###

#####

【题目分析】

用print来实现，其中a用来存储输入的字符，注意输出a字符前的空格个数。

【代码实现】

#输入1个字符

a = input()

"先输出2个空格，然后输出1个字符

print(" " + a)

"先输出1个格，然后输出3个字符

print(" " + a + a + a)

#输出5个字符

print(a * 5)

例1.5 购买电影票

【题目描述】

已知一位小朋友的电影票价是10元，计算x位小朋友的总票价是多少？

【输入格式】

输入小朋友人数x。

【输出格式】

人数和电影票总价，中间用一个空格隔开。

【样例输入】

2

【样例输出】

2 20

【题目分析】 假设总票价用y来表示，则这个问题可以用以下几个步骤来实现：

- (1)输入小朋友的数目x；
- (2)用公式 $y=10*x$ 计算总票价；
- (3)输出总票价y的值。

【代码实现】

#输入小朋友人数

```
x = int(input())
```

#计算总票价

```
y = 10 * x
```

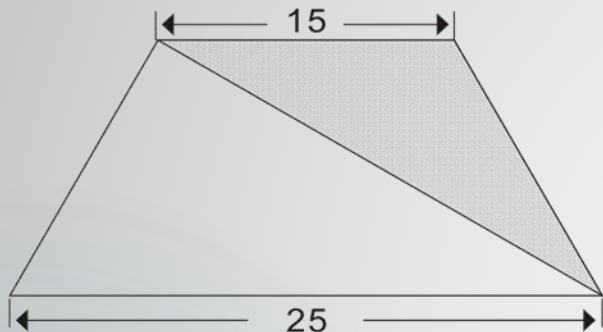
#输出人数和总票价，注意它们之间有空格

```
print(x,y)
```

例1.6 梯形面积

【题目描述】

在梯形中阴影部分面积是**150**平方厘米，求梯形面积。



【输入格式】

(无)

【输出格式】

输出梯形面积(保留两位小数)。

【样例输入】

(无)

【样例输出】

400.00

【题目分析】

已知梯形上、下底长为**15**和**25**。令梯形的高为**h**，则由已知三角形面积为**150**平方厘米，有 $150 = (15 * h) / 2$ ，得**h**为**20**，然后根据梯形面积公式算出梯形面积。

【代码实现】

```
up = 15           #上底长度
down = 25         #下底长度
h = 2 * 150 / up  #求出梯形高度
s = (up + down) * h / 2  #求出梯形面积
print("%.2f"%s)   #输出梯形面积
```