

# Python 编程基础

# HelloWorld程序

# HelloWorld程序



```
1  """
2  This is my first Python program
3  Author: Kai Wang
4  Create Date: 07/29/2018
5  """
6  print("Hello World!") #在屏幕上输出 "Hello World!"
```

# 运行方式



## 交互式运行

```
>>> """
... This is my first Python program
... Author: Kai Wang
... Create Date: 07/29/2018
... """
'\nThis is my first Python program\nAuthor: Kai Wang\nCreate Date: 07/29/2018\n'
>>> print("Hello World!") #在屏幕上输出"Hello World!"
Hello World!
```



## 脚本式运行

```
C:\Users\admin>python d:/pythonsamplecode/01/helloworld.py
Hello World!
```

# 中文编码



## HelloWorld代码修改

```
6      print("你好，世界！") #在屏幕上输出 "你好，世界！"
```

### 注意

使用Python 3.x环境创建Python脚本文件时，需要将文件编码格式设置为UTF-8，否则运行脚本时可能会报错。

例如，如果在使用ANSI编码的Python脚本文件中输入代码并运行，则会出现如下错误信息提示：

SyntaxError: Non-UTF-8 code starting with '\xcd' in file  
d:/pythonsamplecode/01/helloworld.py on line 4, but no encoding declared; see  
<http://python.org/dev/peps/pep-0263/> for details

# 注释



## 单行注释

```
6      print("Hello World!") #在屏幕上输出"Hello World! "
```



## 多行注释

- 1、 `"""` #也可使用三个连续的双引号
- 2、 `This is my first Python program`
- 3、 `Author: Kai Wang`
- 4、 `Create Date: 07/29/2018`
- 5、 `"""` #也可使用三个连续的双引号

# 注释

## 注意

在编写程序时是否对代码添加注释不会影响程序的实际运行结果，但良好的注释将有助于增强程序的可读性、从而提高程序的可维护性



## 书写规范

➤ Python语言通过缩进方式体现各条语句之间的逻辑关系

➤ Python语言中的强制缩进代码示例

```
1      bPrint = True #为变量bPrint赋值为True
2      if bPrint: #如果bPrint的值为True , 则执行bPrint=False和print("Yes")
3          bPrint=False #将bPrint设置为False
4          print("Yes") #输出"Yes"
5      print(bPrint) #输出bPrint的值
```



# 关于Python中的代码缩进



Python语言对于行首缩进的方式没有严格限制，既可以使用空格也可以使用制表符（Tab键），常用对代码进行一个层次缩进的方式有：1个制表符，2个空格，或者4个空格。



对于同一层次的代码，必须使用相同的缩进方式，否则会报错

例如，如果同一层次的代码有的缩进2个空格，而有的缩进4个空格，则会报如下错误：

`IndentationError: unexpected indent`

再如，如果同一层次的代码有的缩进1个制表符，而有的缩进4个空格，则会报如下错误：

`IndentationError: unindent does not match any outer indentation level`