

第一章 基础知识

1.1 第一个程序

1.创建并进入该文件夹

```
fray_hao$ mkdir -p basic-tutor/basic  
fray_hao$ cd basic-tutor/basic
```

2.创建python源文件：

```
fray_hao$ touch 01-HelloPython.py
```

3.用vim打开该文件：

```
fray_hao$ vim 01-HelloPython.py
```

输入以下内容：

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

最后保存文件。

4.在shell中通过python解释器执行源代码

```
fray_hao$ python 01-HelloPython.py  
Hello Python
```

1.2 Python执行的三种方式

1.2.1 解释器执行

用python解释器执行源代码：

```
# 使用 python 2.x 解释器  
python xxx.py  
  
# 使用 python 3.x 解释器  
python3 xxx.py
```

Python 的解释器 如今有多个语言的实现，包括：

- CPython —— 官方版本的 C 语言实现
- Jython —— 可以运行在 Java 平台
- IronPython —— 可以运行在 .NET 和 Mono 平台
- PyPy —— Python 实现的，支持 JIT 即时编译

1.2.2 交互式运行

(1). python shell

直接在终端中运行解释器进入交互shell，在shell中直接输入 Python 的代码，会立即看到程序执行结果

```
fray_hao$ python

>>> print("Hello World")
Hello World
```

缺点： 代码无法保存，不适合运行太大的程序

退出python shell：

```
exit()
```

(2). ipython

IPython 中的 “I” 代表 交互 interactive,是一个 python 的 交互式 shell，比默认的 python shell 好用得多

- 支持自动补全：按tab键
- 自动缩进
- 支持 bash shell 命令
- 内置了许多很有用的功能和函数

```
fray_hao$ ipython

In [1]: print("hello World")
hello World
```

1.2.3 集成开发环境（IDE）

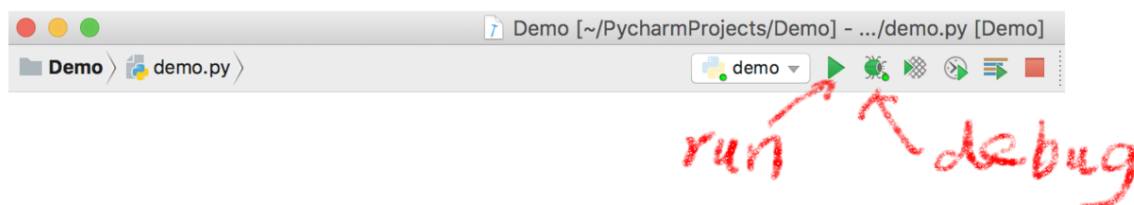
集成开发环境（IDE，Integrated Development Environment）—— 集成了开发软件需要的所有工具，一般包括以下工具：

- 图形用户界面
- 代码编辑器（支持 代码补全 / 自动缩进）

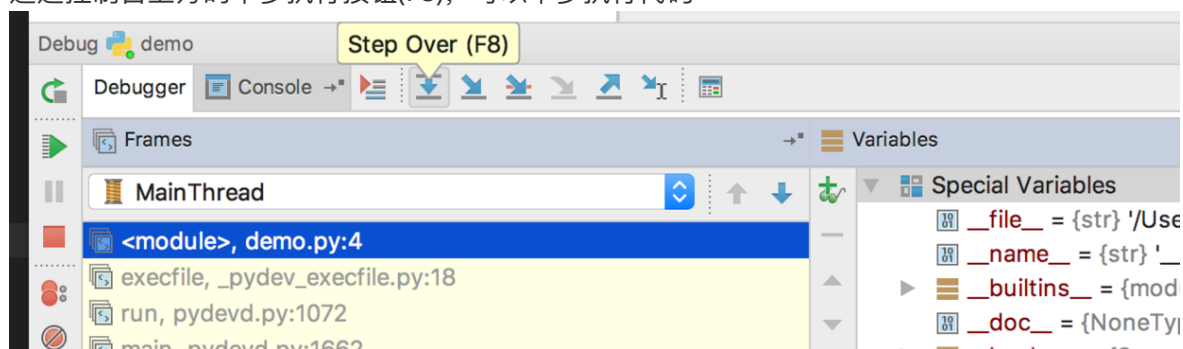
- 编译器 / 解释器
- 调试器 (断点 / 单步执行)

PyCharm 是 Python 的一款非常优秀的集成开发环境.PyCharm 除了具有一般 IDE 所必备功能外, 还可以在 Windows、Linux、macOS 下使用,适合开发大型项目。

1. 在文件编辑区域编写代码
2. 右上角的工具栏 能够 执行(SHIFT + F10) / 调试(SHIFT + F9) 代码



3. 通过控制台上方的单步执行按钮(F8), 可以单步执行代码



1.3 注释

注释可以增强代码的可读性

1.3.1 单行注释

以 # 开头, # 右边的所有东西都被当做说明文字, 而不是真正要执行的程序, 只起到辅助说明作用

```
# 这是第一个单行注释
print("hello python")
```

单行注释可以写到代码后面:

```
print("Hello python") # 这是一个单行注释
```

为了代码的规范性, 根据PEP8, 代码和注释之间需要至少两个空格

1.3.2 多行注释 (块注释)

如果编写的注释信息很多, 一行无法显示, 就可以使用多行注释。

要在 Python 程序中使用多行注释, 可以用一对连续的三个引号(单引号和双引号都可以)

```
'''
这是多行注释
可以写很多的信息
'''

print("Hello python")
```

1.3.3 什么时候需要使用注释？

1. 注释不是越多越好，对于一目了然的代码，不需要添加注释
2. 对于 复杂的操作，应该在操作开始前写上若干行注释
3. 对于 不是一目了然的代码，应在其行尾添加注释（为了提高可读性，注释应该至少离开代码 2 个空格）
4. 绝不要描述代码，假设阅读代码的人比你更懂 Python，他只是不知道你的代码要做什么

在一些正规的开发团队，通常会有 代码审核 的惯例，就是一个团队中彼此阅读对方的代码

1.4 关于代码规范

Python 官方提供有一系列 PEP（Python Enhancement Proposals）文档。- 其中第 8 篇文档专门针对 Python 的代码格式 给出了建议，也就是俗称的 PEP 8。

- 文档地址：<https://www.python.org/dev/peps/pep-0008/>
- 谷歌有对应的中文文档：http://zh-google-styleguide.readthedocs.io/en/latest/google-python-styleguide/python_style_rules/