

第一章 为什么要学 Python

1.1 为什么要学编程

1.1.1 算法与编程

算法是信息技术学科精炼出来的学科大概念，它贯穿整个信息技术学科的始终。那么，什么是算法呢？通俗地说，算法就是解决问题的步骤和方法。它是指解决问题方案的准确而完整的流程描述，是一系列解决问题的清晰指令。它代表着用系统的方法描述解决问题的策略机制。精确的算法是计算工具有效计算从而解决问题的前提条件。

算法可以用自然语言、流程图、伪代码等方法描述。但是计算机并不能直接执行这些描述符号。为了让计算机能按照算法的意图解决问题，就需要将算法“翻译”成计算机能懂的代码，这就是计算机编程。编程是编写程序的中文简称，即是人和计算体系之间交流的过程。为了使计算机能够理解人的意图，人类就必须将需解决问题的算法翻译成计算机能够理解的形式，使得计算机能够根据人的指令一步一步去工作，完成某种特定的任务。编程过程是以算法为基础，以某种程序设计语言为工具，以解决问题为目的，它包括分析问题、设计算法、编写代码、测试运行等不同阶段。

1.1.2 编程教育的国家战略

随着人工智能行业的高速发展，国家十分重视在中小学开设人工智能相关课程，编程教育成为我国社会和学校普遍关注的教育领域。《国务院关于印发〈新一代人工智能发展规划〉的通知》（国发〔2017〕35号）明确提出“实施全民智能教育项目，在中小学阶段设置人工智能相关课程，逐步推广编程教育，鼓励社会力量参与寓教于乐的编程教学软件、游戏的开发和推广。”《教育部关于印发〈教育信息化2.0行动计划〉的通知》（教技〔2018〕6号）明确要求“完善课程方案和课程标准，充实适应信息时代、智能时代发展需要的人工智能和编程课程内容。”最新的《信息技术课程标准》也明确定位了“信息技术课程是一门旨在全面提升高中学生信息素养，帮助学生掌握信息技术基础知识与技能、增强信息意识、发展计算思维、提高数字化学习与创新能力、树立正确的信息社会价值观和责任感的基础课程。”的课程性质。因此，学习编程既是国家发展战略，也是教育发展必然，更是信息技术课程改革之必需。2016年9月，《中国学生发展核心素养》发布，以培养“全面发展的人”为核心。学生发展核心素养是指学生应具备的、能够适应终身发展的需要的必备品格和关键能力。它是通过学科教育培养，落实在学生身上最有价值的核心素养，是跨学科的综合素质。而信息技术学科核心素养则包括信息意识、计算思维、数字化学习与创新和信息社会责任。其中，信息技术学科本质核心素养是计算机思维，简称计算思维。计算机思维不是说计算机有思维，用脑思考，而是因为这种思维方式是伴随着计算机出现的。计算思维是个体运用计算机科学领域的思想方法，在形成问题解决方案的过程中产生的一系列思维活动。“计算思维的本质是翻译，也就是把人想要做的具体事情，翻译成计算机能够懂得的程序语言”清华大学史元春教授如是说。

1.1.3 学习编程的意义

在后信息时代，或者是即将进入的智能时代，所有人都需要升级自己的思维方式，让自己的思维方式跟上这个时代。学习编程不只是为了教孩子敲代码，更重要的是让他们明白人工智能时代科技背后的原理，通过编程课程学习达到计算思维训练的目的。计算思维能力已成为人们有意识地使用计算机科学思想、方法、技术、工具、资源、环境去思考和实践的一种基本技能。学会编程，方能真正揭开计算机解决问题的神秘面纱，并以此提高自己分析问题和解决问题的能力，进而培养与拓展人类大脑思维能力。在中小学信息技术教学中，顺应时代发展需求，以学生计算思维培育为切入点，基于项目活动的学习方式，开展编程

教育培养。通过方法习得、工具应用、思维迁移三个层面的落实，促进学生核心素养的养成、内化和拓宽，形成学生应具备的、能够适应终生发展需要的必备品格和关键能力。

1.2 为什么学 Python

1.2.1 Python 语言的发展

随着计算机应用的需求剧增，高级语言层出不穷，Fortran、Basic、Pascal 逐渐远离，如今，让你从数百种编程语言中选择一门入门语言，是应用率最高、长期霸占排行榜的常青藤 JAVA？还是易于上手、难以精通的 C？还是在游戏和工具领域仍占主流地位的 C++？抑或是占据 Windows 桌面应用程序半壁江山的 C#？殊不知，另有一门异军突起的 Python 语言。

Python 是一种面向对象、解释型计算机程序设计高级语言，由 Guido van Rossum 于 1989 年开发创建，是纯粹的自由软件，源代码和解释器 CPython 遵循 GPL (GNU General Public License) 协议。1989 年圣诞节期间，在阿姆斯特丹，Guido 为了打发圣诞节的无趣，决心开发一个新的脚本解释程序，作为 ABC 语言的一种继承。之所以选中 Python（大蟒蛇的意思）作为该编程语言的名字，是因为他是一个叫 Monty Python 的喜剧团体的爱好者。就 Guido 本人看来，ABC 这种语言非常优美和强大，是专门为非专业程序员设计的。但是 ABC 语言并没有成功，Guido 决心在 Python 中避免不足，同时还想实现在 ABC 中闪过过但未曾实现的东西。就这样，Python 在 Guido 手中诞生了。

1.2.2 Python 语言的特点与优势

Python 已经成为最受欢迎的程序设计语言之一，其主要优势有：

首先，Python 是面向对象的、动态数据类型的解释型语言，省去了变量声明的过程，程序运行的过程中自动决定对象的类型。在 Python3 后，变量可以存放任意大小的整数，只有内存不够，没有数据溢出，不会像其他语言那样受到溢出问题的困扰，降低了学习门槛。

其次，Python 使用缩进语法格式，使得语法简单、风格清晰、严谨易学，它能让用户编写出更易读、易维护的代码，能让开发者、分析人员和研究人员在项目中更好地合作。

第三，Python 代码效率高，10 行 Python 代码就能完成 C++20 行代码的工作。

第四，Python 拥有丰富的扩展库，常被昵称为胶水语言，能够把用其他语言制作的各种模块很便捷地联结在一起，可以轻易完成各种高级任务。

第五，Python 完全免费，众多开源的科学计算库都提供了 Python 的调用接口，用户可以在任何计算机上免费安装 Python 及其绝大多数扩展库。在国内外及各领域，如卡耐基梅隆大学的编程基础、麻省理工学院的计算机科学及编程导论都在使用 Python 语言讲授；如著名的计算机视觉库 OpenCV、三维可视化库 VTK、医学图像处理库 ITK 等众多开源的科学计算软件包也都提供了 Python 的调用接口；又如 NumPy、SciPy 和 Matplotlib 等十分经典的科学计算扩展库，它们分别为 Python 提供了快速数组处理、数值运算以及绘图功能。

总之，Python 作为一门面向对象的高级编程语言，其魅力和影响力已经远超 C#、C++ 等编程语言前辈，被程序员誉为“最美丽的”编程语言。这也许是 Python 成为人工智能、大数据科研人员首选语言的原因之一吧。从云端、客户端，到物联网终端，再到人工智能，Python 应用无处不在，高中新课标所有模块都可以 Python 为基础实现（图 1）。在人工智能普及的当下，选用 Python、学习 Python，不仅可以培养信息技术学科核心素养，也定将为学生终身发展提供无限可能！

数据与计算	信息系统与社会
数据与大数据 数据处理、分析与可视化 (Python + Pandas) 编程与算法 (Python) 人工智能简介 (Python + Baidu)	认识信息系统 设备、网络与软件 (Raspberry Pi + Python) 传感与控制 (Raspberry Pi + Python) 信息社会：伦理与法规
数据与数据结构	数据管理与分析
Python 实现	Python + Pandas Python + Matplotlib
人工智能初步	网络基础
Python + scikit-learn Raspberry Pi + TensorFlow App Inventor + TensorFlow App Inventor + BATK	Windows + Python Raspberry Pi + Python Android + App Inventor
三维设计与创意	算法初步
Minecraft + Python Python + Vpython	Python + NumPy + SciPy
移动应用设计	开源硬件项目设计
Android + App Inventor 2 Python + Django	Raspberry Pi + Python App Inventor + Arduino MicroPython + IoT

图 1 高中信息技术新课标模块结构

第二章 Python 环境安装

2.1 Python 下载

Python 是跨平台的，它可以运行在 Windows、Mac OS 和各种 Linux/Unix 系统上，这里我们以 Windows 操作系统为例。下载之前需要确认你的系统是 64 位还是 32 位，查看方法为：打开计算机属性面板，如图：

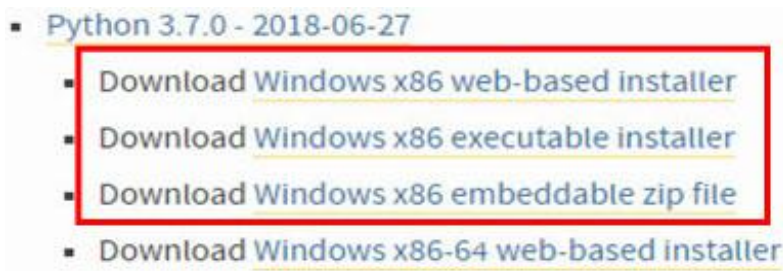


在此以 32 位系统为例，下载 Python 安装包：

首先打开 Python 官网首页 (www.Python.org)，依次选择“Downloads”->“windows”，如图：



选择最新的版本 Python 3.7.0



矩形框内为 32 位 Python 安装包，这里的 embeddable zip file 是可以嵌入到其它应用的版本，web-based install 是需要联网安装的版本，executable installer 是一个可执行文件，可以直接安装（推荐！），你可以根据需要选择。

2.2 Python 安装与运行

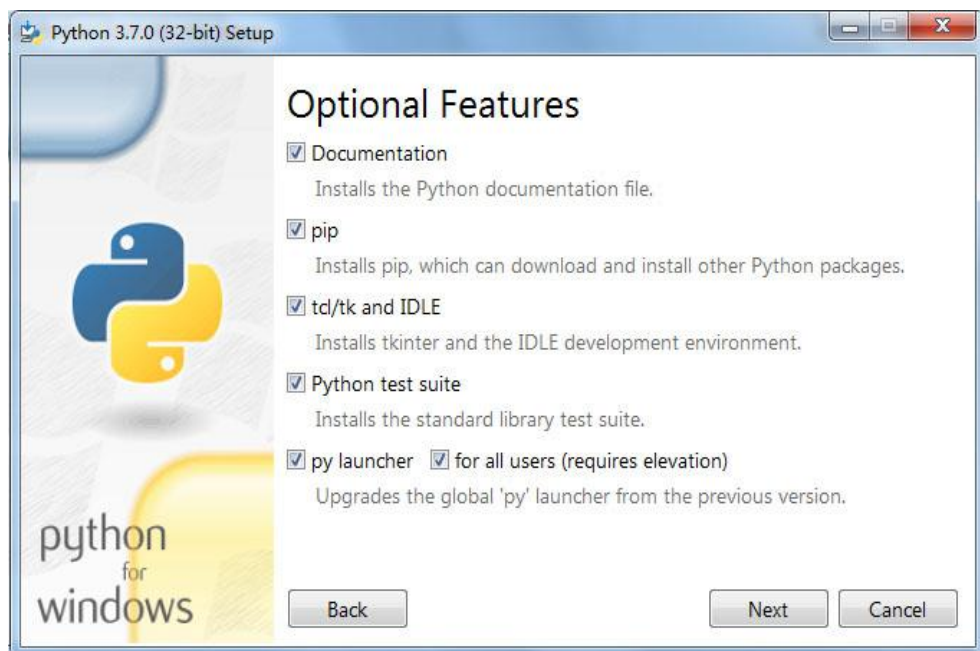
2.2.1 Python 安装

双击下载好的安装包（这里下载的是 Windows x86 executable installer），出现如下安装界面，**注意：这里一定要选择“Add Python3.7 to path”。**

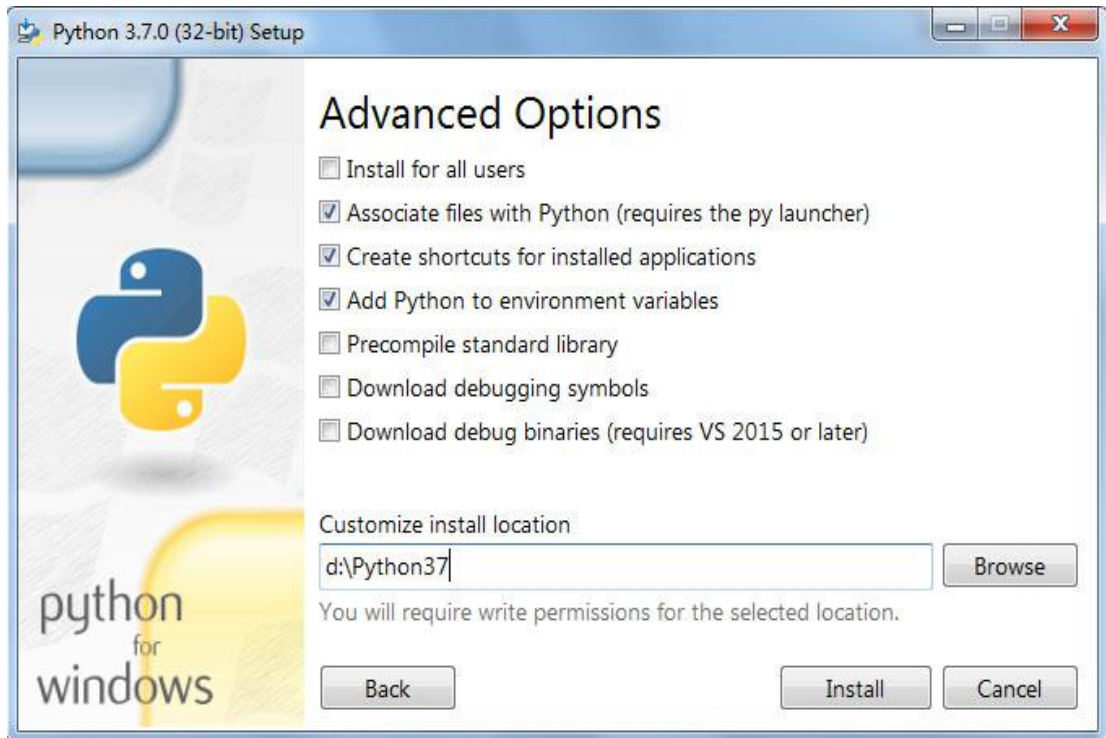
进入程序安装界面



将“添加到环境变量”前面的复选框勾选，然后点击自定义安装



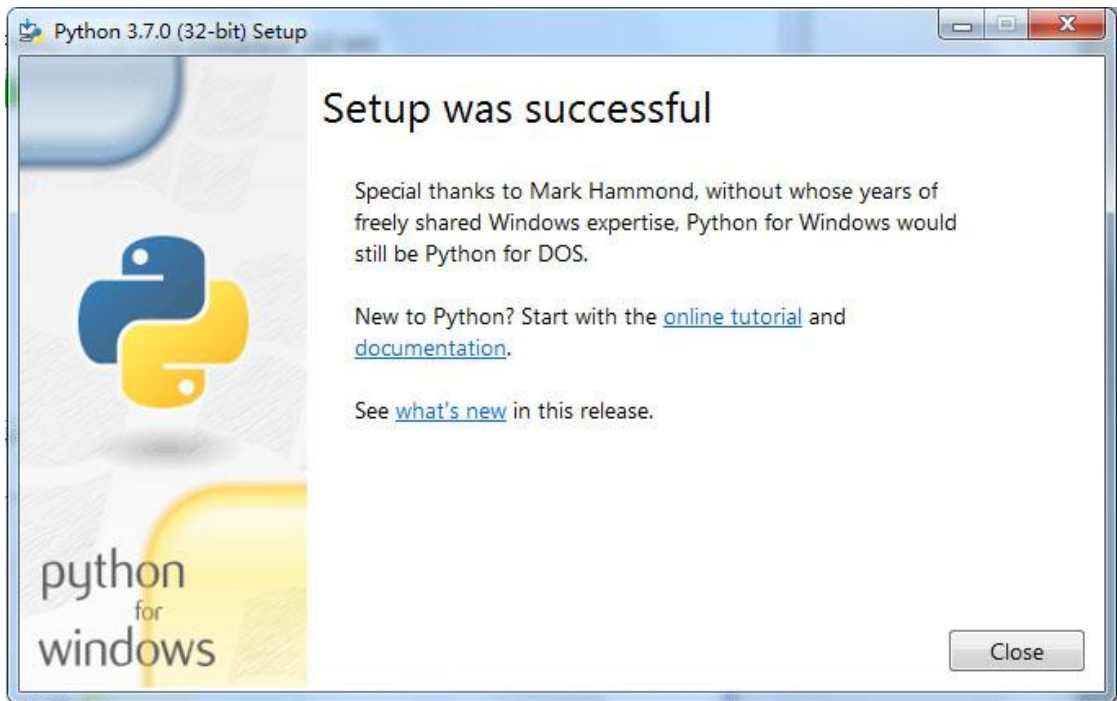
上图是选择安装选项，Documentation 是安装 Python 帮助文档，pip 是用来安装第三方模块的工具，tcl/tk and IDLE 是 Python 官方自带的简单 GUI 模块与基于 tcl/tk 编写的一个简单的 IDLE 开发环境，Python test suite 是安装 Python 的标准库和测试套件，py launcher 为是否系统关联 py 文件，for all users(requires elevation) 为是否使所有用户都关联 py 文件。我们这里全部勾选，单击“Next”



单击 Browse 按钮将安装路径更改为你希望的实际路径（如：D:\Python37）。该界面的一些选项如不理解，默认即可。下面是这些选项的简单注释。

- ◆ Install for all users 为所有用户安装
- ◆ Associate files with Python (requires the py launcher) 将文件与 Python 关联（需要 py 启动器）
- ◆ Create shortcuts for installed applications 为已安装的应用程序创建快捷方式
- ◆ Add Python to environment variables 将 Python 添加到环境变量
- ◆ Precompile standard library 预编译标准库
- ◆ Download debugging symbols 下载调试符号
- ◆ Download debug binaries (requires VS 2015 or later) 下载调试二进制文件（需要 VS 2015 或更高版本）

根据需要选择安装选项后，单击 **install** 按钮开始安装，安装完成后，将出现成功安装界面。单击 **Close** 按钮关闭对话框完成安装，如下图所示：



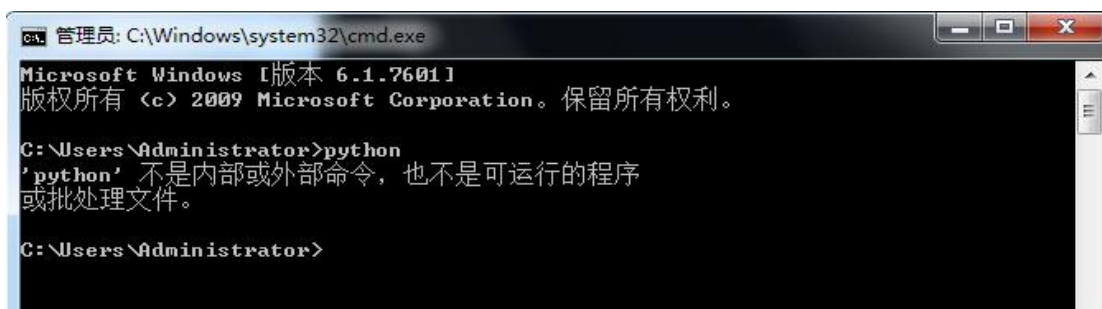
2.2.2 运行 Python

Win+R, 打开命令行窗口或“开始菜单”->“运行”, 输入CMD, 确定, 进入DOS环境, 输入“python”, 如果出现下图界面 (Python 的交互式操作界面), 说明安装成功, 至此完成 Python3.7.0 安装。



当看到提示符“>>>”就表示已经在 Python 交互式环境中了, 可以输入任何 Python 代码, 回车后会立刻得到执行结果。现在, 输入 `exit()` 并回车, 就可以退出 Python 交互式环境。

注: 如果出现错误: ‘python’ 不是内部或外部命令, 也不是可运行的程序或批处理文件。

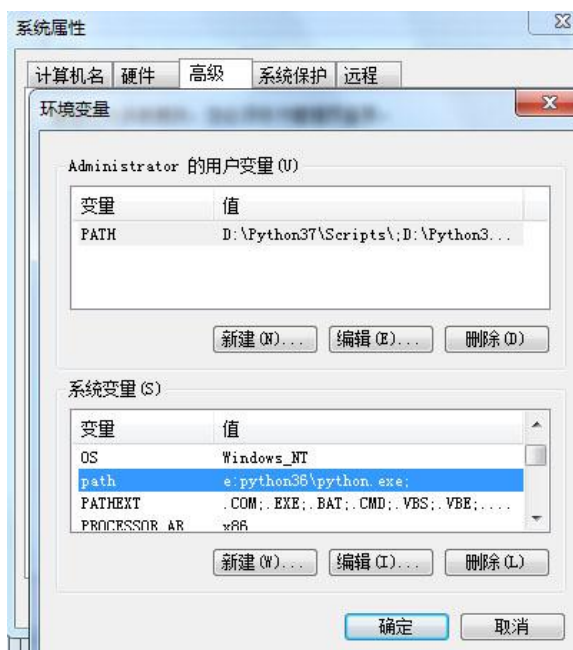


这是因为 Windows 在环境变量设定的路径中去查找文件 `python.exe` 时, 没有找到 `Python.exe`, 可能的原因是在安装过程中没有选中 `Add Python 3.7.0 to Path`。

解决办法为：右击“计算机”选择“属性”



在左侧栏中单击“高级系统设置”，选择对话框中的“高级”选项卡，单击“环境变量”按钮。



打开环境变量对话框，在系统变量中双击 Path 变量，或单击 Path 变量后单击下面的编辑按钮。



最后，在编辑系统变量对话框中的变量值的最后添加“;D:\Python37”（注意 D 前面的分号），确认即完成环境变量的添加。

注：如果你还是不知道怎么修改环境变量，那么就把 Python 安装程序重新运行一遍，务必记得选中 Add Python 3.7 to Path。

2.2.3 Python 程序初体验

现在，了解了如何启动和退出 Python 的交互式环境，我们就可以正式开始编写 Python 代码了。

在交互模式下，直接输入代码，按回车，就可以立刻得到代码执行结果。现在，试试输入 $300+500$ ，看看计算结果是不是 800：

```
>>>300+500
800
```

很简单吧，任何有效的数学计算都可以算出来。

如果要让 Python 打印出指定的文字，可以用 `print()` 函数，然后把希望打印的文字用单引号或者双引号括起来作为参数即可，例如：

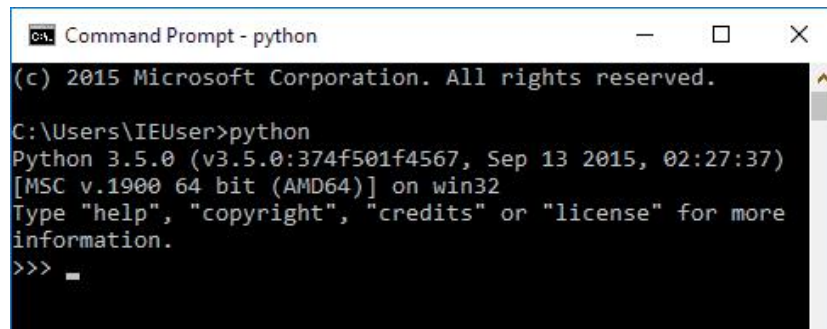
```
>>>print('hello, world' )
hello, world
```

这里，我们把用单引号或者双引号括起来的文本叫做字符串。

最后，用 `exit()` 退出 Python。我们的第一个 Python 程序完成！唯一的缺憾是没有保存下来，下次运行时还要再输入一遍代码。

在命令行模式下，可以执行 Python 进入 Python 交互模式，也可以执行 `Python hello.py` 运行一个 .py 文件。

看到“>>>”提示符，说明在 Python 交互式环境下，如下图：



在 Python 交互模式下，只要输入 Python 代码就立刻执行。

例如，在 Python 交互式环境下，输入：

```
>>> 100 + 200 + 300
600
```

直接可以看到结果 600。

小结

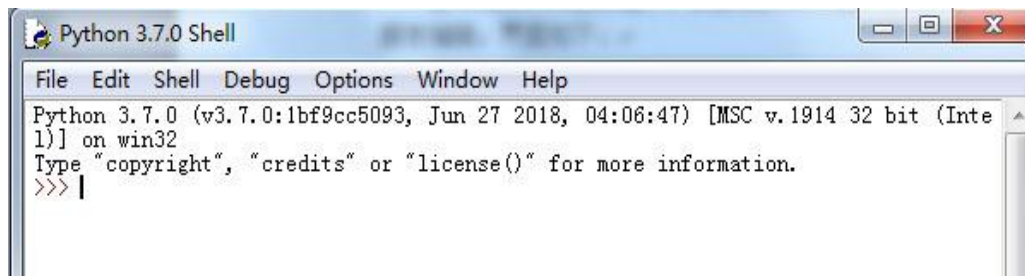
- ①在 Python 交互模式下，可以直接输入代码，然后执行，并立刻得到结果；
- ②完整演示了在 Windows 操作系统下 Python3.7.0 的安装过程以及运行 Python 的操作；
- ③启动 Python 只需在命令行输入 Python 即可。

2.3 IDLE 工具的使用

我们已经安装了 Python 开发环境，写的代码能够执行但不能保存，每次都需要重新输入，如果希望把编写的代码保存起来，还需要一个编程的专业工具，就象广告设计师使用 Photoshop 处理图像一样，编程也需要相应的工具，叫做集成开发环境 (IDE)。由于在交互模式下是不能开发大型项目的，因此，Python 的开发工具就此诞生。以下介绍几款常用的 Python 开发工具。

2.3.1 IDLE

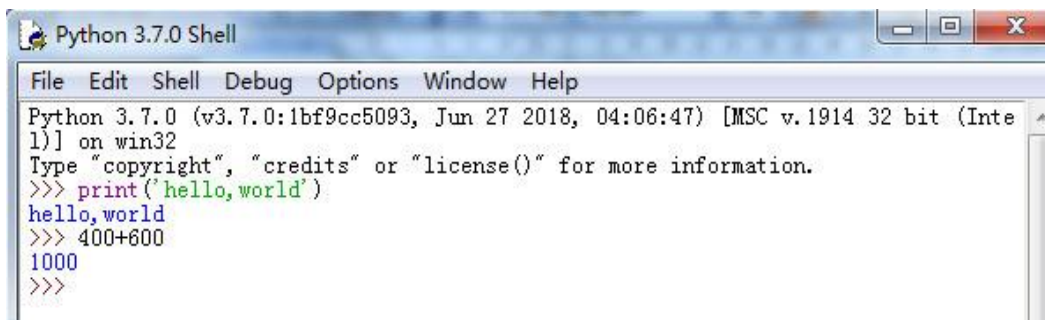
在 Windows 平台下，安装 Python 时自动安装了一个 Python 自带的 IDLE，可以进行脚本编辑；界面如下：



IDLE 是 Python 软件包自带的一个集成开发环境，我们可以利用它方便地创建、运行和调试 Python 程序。

➤ IDLE 交互编辑

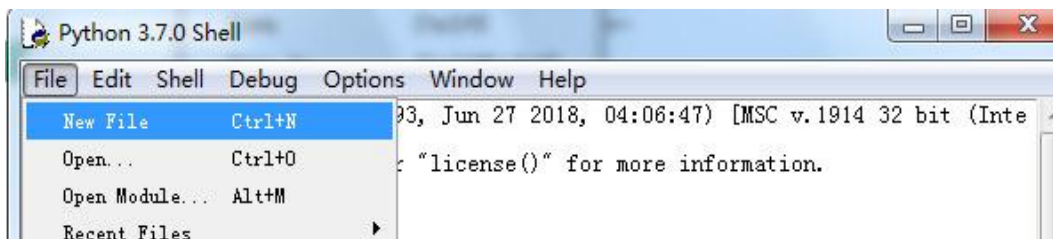
在交互模式下，可以进行一些简单代码的测试，如打印 hello, world 字符串；400+600 的运算等：



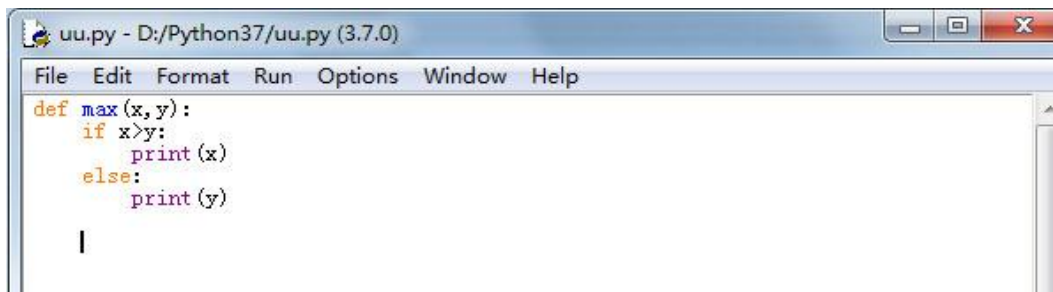
在“>>>”符号后键入 print('hello, world')，然后按回车，即可看到此行代码运行的结果，输入 400+600 回车，即可看到加法运算结果。

➤ IDLE 文件编辑

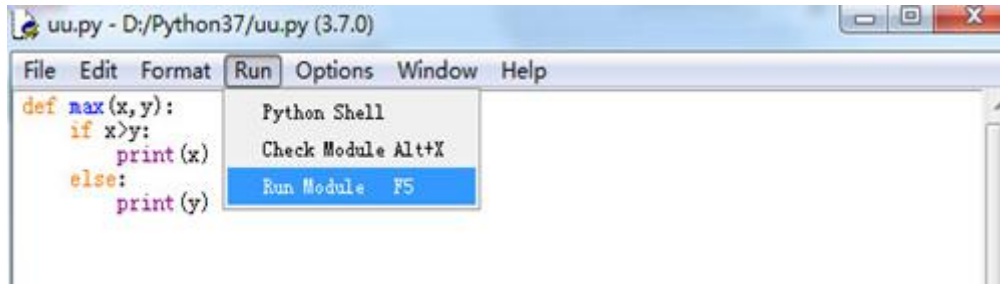
如果需要编写大段的 Python 程序并重复使用，可以使用 IDLE 提供的文件编辑功能，在 IDLE 的菜单 File 中选择 New File，或者按下 Ctrl+N 快捷键，如下图所示：



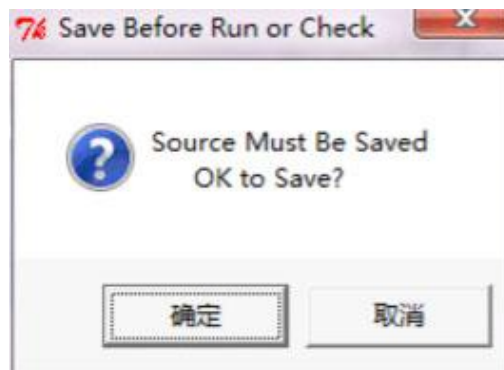
IDLE 会打开一个新的空白窗口，在此窗口中即可书写代码，如下所示：



当完成编辑后，执行 Run 菜单中的 RunModule 菜单项运行程序，或者按 F5 键，如下图所示：



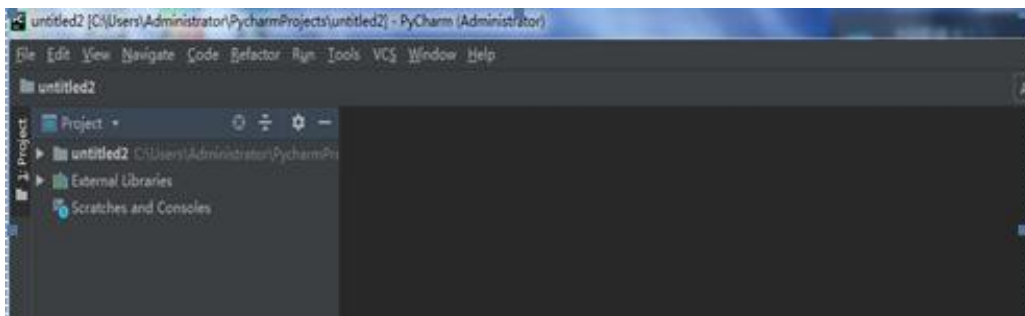
系统会提示保存文件，选择确定即可：



2.3.2 Pycharm

Pycharm 是公认的最好用的 Python IDE 之一，支持 Windows 用户，有社区版（免费）和专业版（付费），官网地址：<http://www.jetbrains.com/pycharm/>，初学者使用社区版即可。

界面如下：



具体使用方法请参阅其它资源。

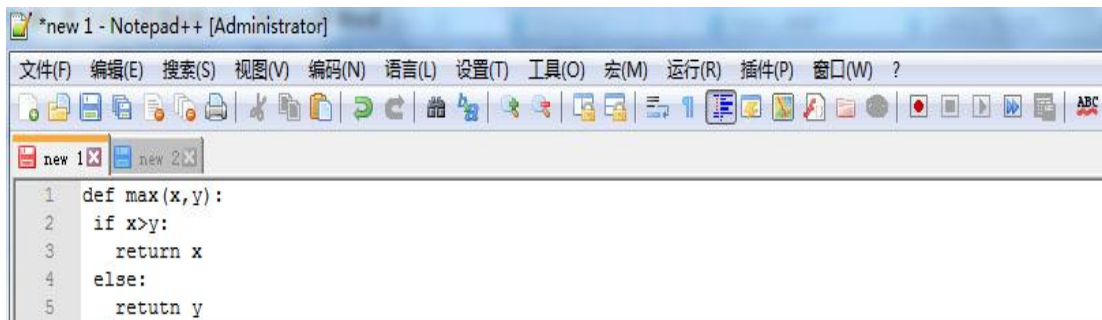
2.3.3 Notepad++

Notepad++是 Windows 操作系统下的一款文本编辑器，支持多国语言编写功能(UTF8 技术)。

Notepad++功能比 Windows 中的 Notepad(记事本)强大，除了可以用来制作一般的纯文字说明文件，也十分适合编写计算机程序代码。Notepad++ 不仅有语法高亮度显示，也有语法折叠功能，并且支持宏以及扩充基本功能的外挂模组。

Notepad++是免费软件，可以免费使用，自带中文，支持众多计算机程序语言

界面如下：



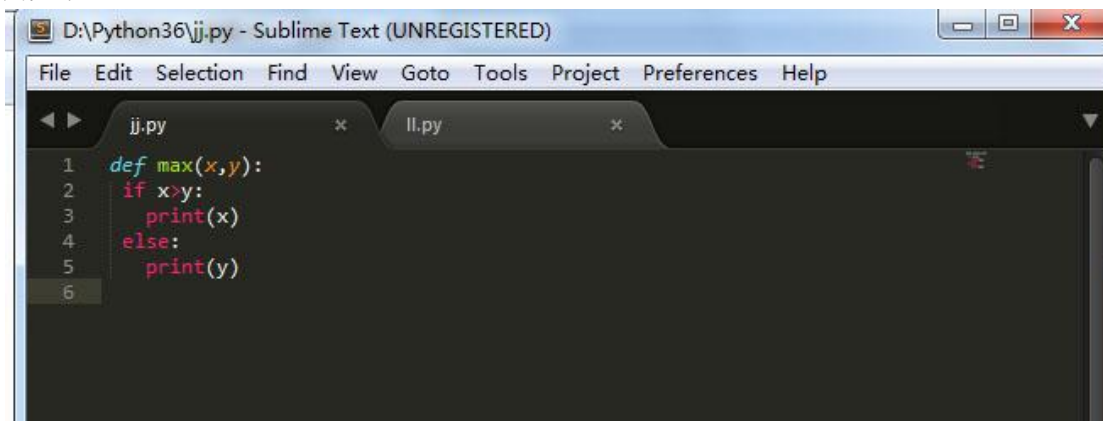
A screenshot of the Notepad++ application window. The title bar reads '*new 1 - Notepad++ [Administrator]'. The menu bar includes '文件(F)', '编辑(E)', '搜索(S)', '视图(V)', '编码(N)', '语言(L)', '设置(T)', '工具(O)', '宏(M)', '运行(R)', '插件(P)', and '窗口(W)'. The toolbar contains various icons for file operations and editing. Two tabs are open: 'new 1' and 'new 2'. The main text area contains the following Python code:

```
1 def max(x,y):
2     if x>y:
3         return x
4     else:
5         retutn y
```

2.3.4 Sublime Text

Sublime Text 是一个轻量、简洁、高效、跨平台的编辑器，一款具有代码高亮、语法提示、自动完成且反应快速的编辑器软件，具体使用方法请参阅其它资源。

界面如下：



A screenshot of the Sublime Text application window. The title bar reads 'D:\Python36\jj.py - Sublime Text (UNREGISTERED)'. The menu bar includes 'File', 'Edit', 'Selection', 'Find', 'View', 'Goto', 'Tools', 'Project', 'Preferences', and 'Help'. Two tabs are open: 'jj.py' and 'll.py'. The main text area shows the same Python code as in the Notepad++ screenshot, but with syntax highlighting: 'def' is blue, 'if' is red, 'else' is red, 'print' is red, and 'return' is blue. The code is as follows:

```
1 def max(x,y):
2     if x>y:
3         print(x)
4     else:
5         print(y)
6
```