

Python语言程序设计

第一章 计算机和Python程序



Part One

...

第一讲 课程内容和教学安排

□东北大学计算机1级课程

- 目标：建立程序设计的概念，掌握程序设计的基本方法和技巧，养成良好的程序设计风格，具备使用Python程序设计语言解决相关专业领域内实际问题的基本能力。
- 学时：64学时（32理论+32实践）
- 学分：3学分

课程涵盖如下内容，并包含各类实际应用案例

- 程序设计与Python概述
- Python程序基础语法和基本数据类型
- 程序的控制结构
- 函数
- Python组合数据类型
- 文件和异常

□ 离线和在线学习

- 课堂+实践

- 在线学习平台

□ 成绩组成

- OJ平台作业 (20分)

- 平时测试 (2+次 , 30分)

- 期末测试 (50分)

□ 跟上进度

- 跟随课程进度，完成课程要求的学习内容

□ 重视实践

- 重复课堂中的所有实践内容
- 完成额外程序设计练习内容

□ 多思考、勤练习

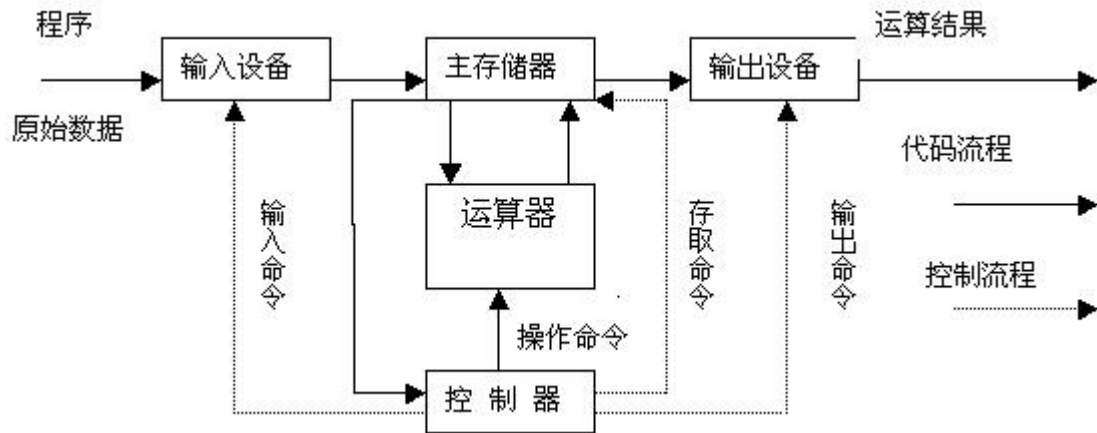
The only way to learn a new programming language is by writing programs in it.

Part Two

...

第二讲 计算机和程序

- 计算机是可根据一系列指令操作数据的设备。
 - 计算机可以操作数据，包括数据计算、输出输出处理和数据存储等；
 - 计算机是根据一系列指令自动地完成操作。



- 软件（程序）主宰硬件（机器）。
- 创建软件的过程称为“编程”。
- 学习编程的理由：
 - ✓ 编程是计算机科学的基础
 - ✓ 编程有助于深入理解计算机这个工具
 - ✓ 编程可以帮助你解决生活中和专业领域内遇到的问题
 - ✓ 编程包含很多乐趣

- 机器语言：由0和1组成的二进制代码，是内嵌在计算机内的原始指令集。

- 1001 0010 0011

- 计算机只能执行机器语言程序。

- 机器语言因计算机的种类不同而不同。

- 汇编语言：使用简短的描述性单词（助记符）表示机器语言指令
 - ADD 2, 3, result
- 计算机无法理解汇编语言，需要通过“汇编器”将汇编程序翻译成机器代码。
- 汇编语言本质上接近机器语言。

- 高级语言：向自然语言靠近的语言

- `sum = 2 + 3`

- 高级语言独立于平台

- 高级语言编写的程序称为“源程序”或“源代码”

□常用的高级程序设计语言：

□Python

□C

□C++

□C#

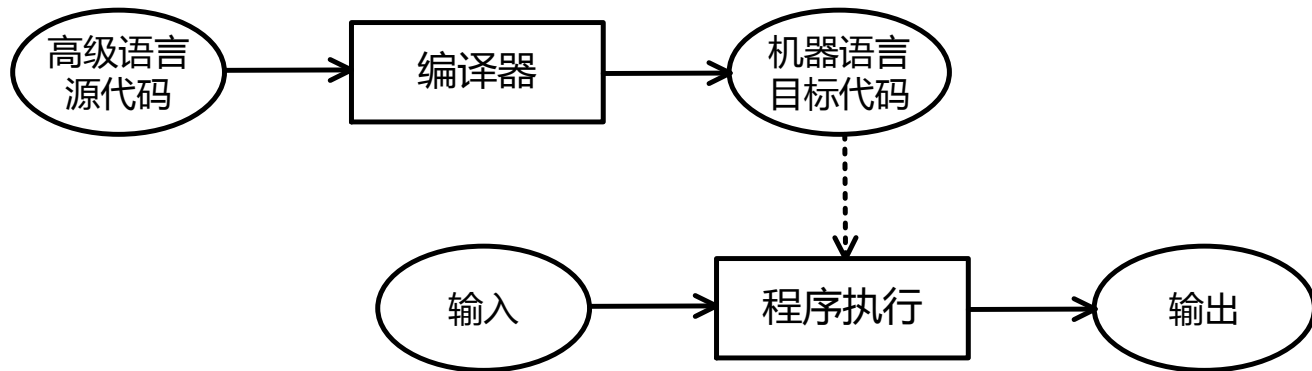
□Java

□JavaScript

□PHP

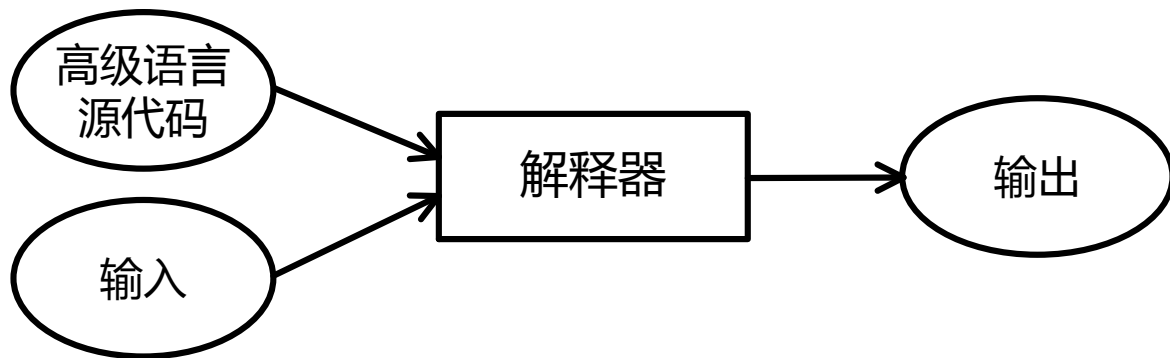
□第一种方式：编译

- 将高级语言源代码转换成目标代码（机器语言）
- 程序可以直接运行
- 目标代码执行速度更快



□第二种方式：解释

- 将高级语言源代码逐条转换成目标代码，边转换边执行
- 每次运行程序需要源代码和解释器
- 良好的跨平台可移植性



□编译和解释方式的区别：

- 编译是一次性的翻译。一旦程序被编译，它可以重复运行而不需要编译器或源代码。
- 解释方式每次程序运行都需要解释器和源代码。
- 编译的优势在程序执行速度快；解释的优势在编程灵活，允许交互式开发和运行。

Part Three

...

第三讲 Python语言概述

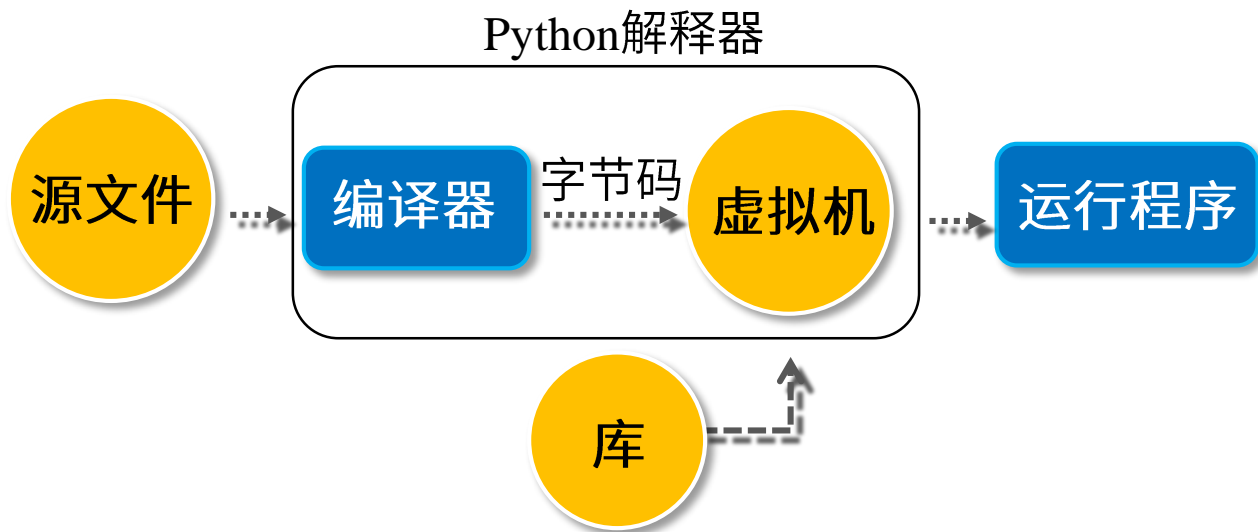


Guido van Rossum, 1989年圣诞节开发Python语言

1990年正式诞生

“Python”的命名源于Guido喜爱的一部英剧“Monty Python's Flying Circus”

- 从编程的视角：Python是解释型的语言
- 从Python的视角：Python的解释器中包含着编译的部分



- 免费
- 可移植
- 可混合
- 简单易用
- 简单易学
- 面向对象和函数式
- 功能强大
 - 动态类型
 - 自动内存管理
 - 大型程序支持
 - 内置对象类型
 - 内置工具
 - 库工具
 - 第三方库

- Python唯一的缺点是：在目前的实现方式下，与C或C++这类编译语言相比，Python的执行速度还不够快。
- Python本身在不断优化，在绝大多数应用领域运行的速度足够快。
- Python的开发速度带来的效益往往比执行速度带来的损失更为重要。
- 对很多程序而言，速度不够快是因为代码编写的质量问题。

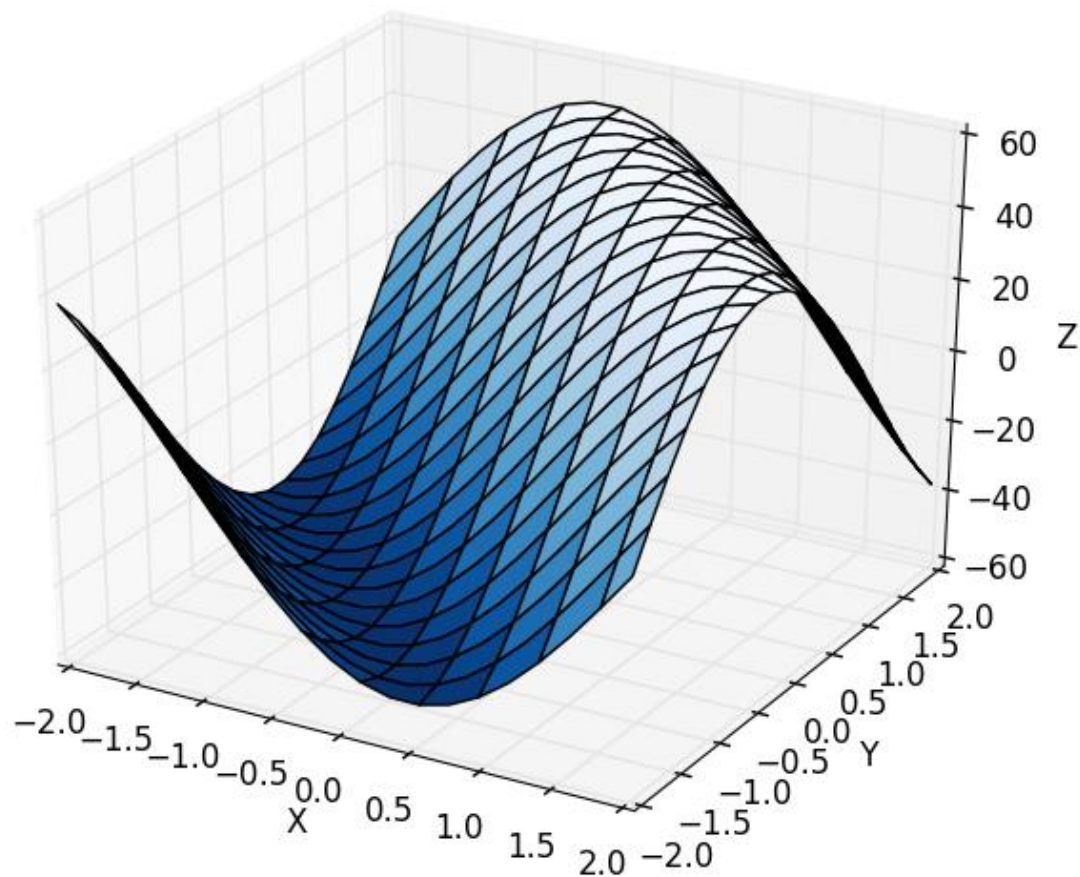
- 2000年10月，Python2.0发布。
- 2010年，Python2.x发布终结版本，主版本号为Python2.7，并不再进行重大改进。
- 2008年3月，Python3.0发布。
- 目前最新版本是Python3.7.1（2018年10月20日）
- 几个说明
 - Python2和Python3不兼容
 - Python2已经是Python的遗产，Python3是Python的现在和将来
 - 除非必须使用Python2编写的程序，不必学习Python2语法
 - Python3的最新版本更新频繁，一般采用Python3.6以上版本即可

- Python主流应用
 - 科学计算
 - 数据分析
 - 人工智能
 - 云计算
 - web开发
 - 自动化运维
 - 网络爬虫
 - 游戏开发
 - 常规桌面应用开发
 - 网络软件开发.....



Python能做什么





Python能做什么



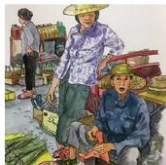
热点内容 ····· (更多)



旧上海的艺术 65张照片



朔尔曼斯亮, 新年北极圈的第一缕阳光 42张照片



旱市系列 58张照片



最近时差 94张照片

天太冷, 去南洋小岛吹暖风吧

偷妮斯由的日记

入冬之后, 我仿佛每天都被冻住了一样, 整个人麻木的。想念夏天, 想去海边, 却只是...

在人言走过漫长白日

“照亮一隅的光芒”

奇幻童年和独居老人

杭州 | 晚来天欲雪, 太匆匆

冬的东京, 颇真哪, 维米尔

不过年主义 / 暴力美学

大航海时代

韩松: 家是一个令人眩晕的洞 | 科幻春晚

北京山野半貌: 爬山有什么意思?



热门话题 ····· (去话题广场)

第91届奥斯卡

1408020人浏览

收集全世界的彩虹

82659人浏览

猫咪“骑马”大赛

新话题

写给你爱的人的情诗

620082人浏览

你理想中的两性关系

2763039人浏览

你在恋爱中听过最寒心的话是什么?

1164人浏览



- 软件质量
- 开发效率
- 可移植性
- 标准库的支持
- 享受开发乐趣

- Python语言是应用最广泛的计算机语言之一
- Python语言是通用语言
- Python语言是解释型语言
- Python语言是开源语言
- Python语言是跨平台语言

人生苦短，我用Python！

Part Four

...

第四讲 Python开发环境

□常见的Python集成开发环境（IDE）

□**IDLE**（Python官方自带工具）

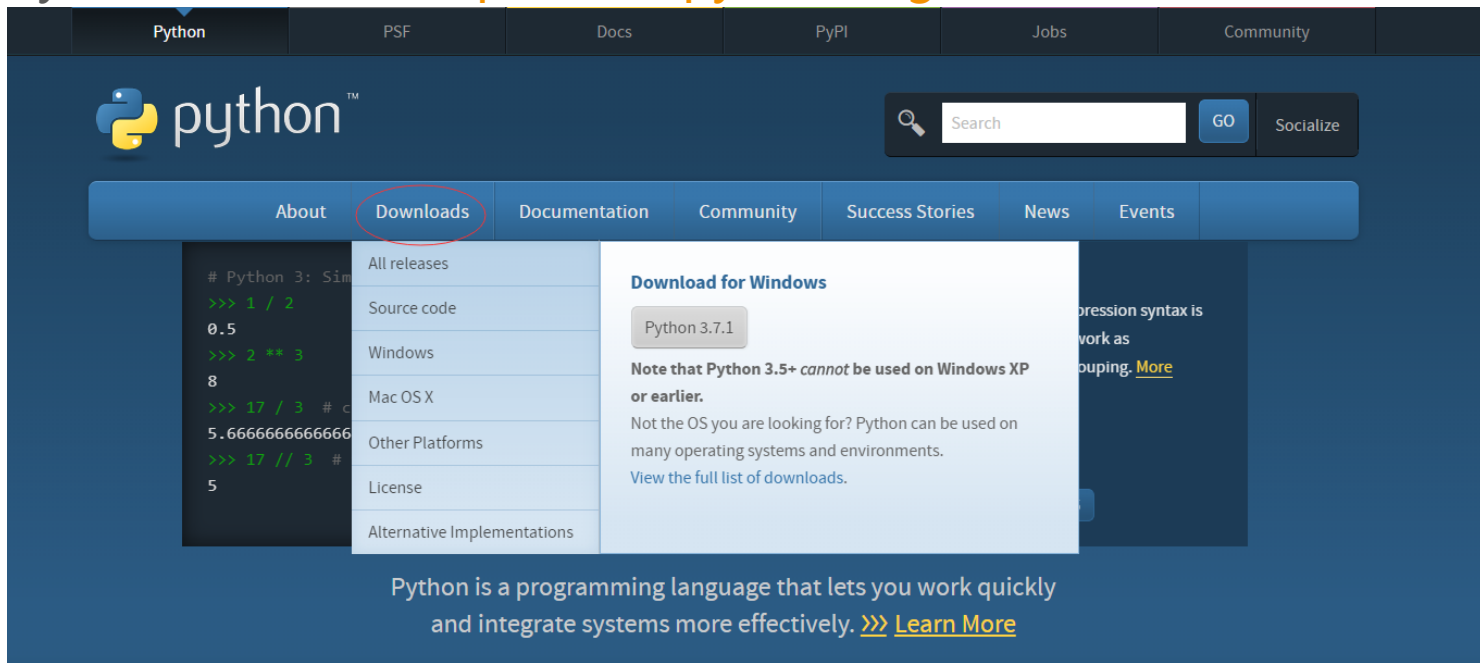
□PyCharm（商业软件，Python开发专业工具）

□Anaconda+Spyder（免费Python集成开发环境，适用于科学计算等）

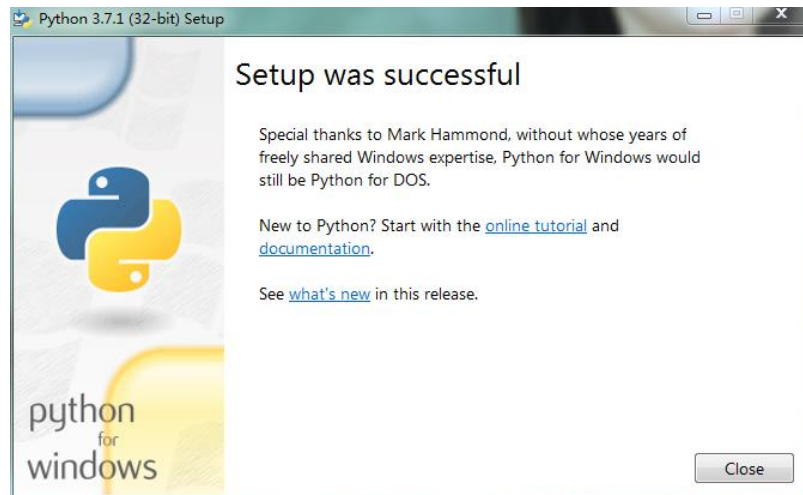
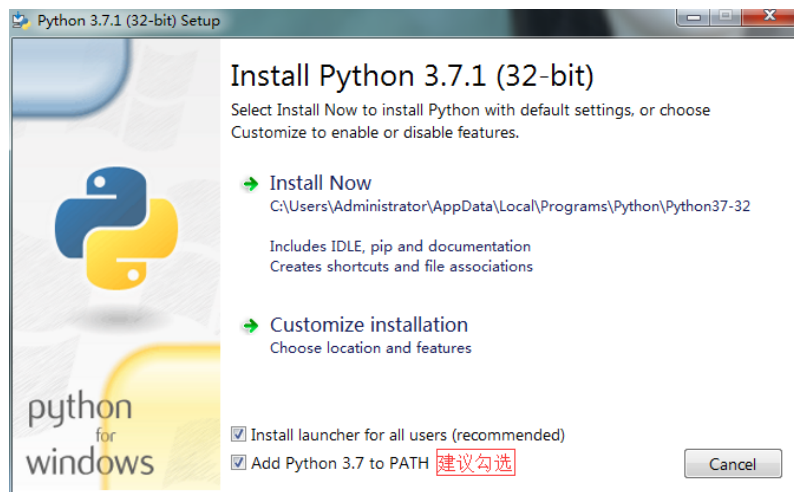
□Eclipse+PyDev（通用流行Eclipse环境配合Python插件，免费）

□.....

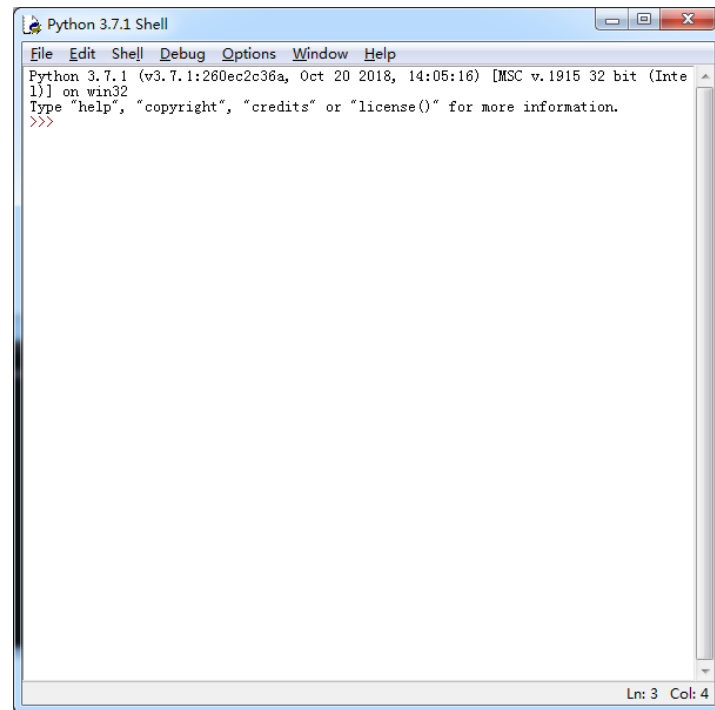
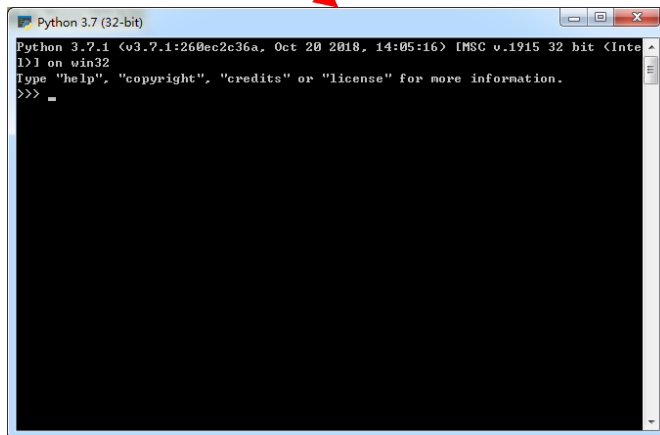
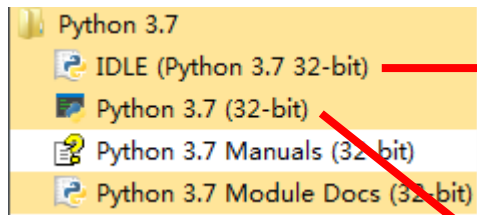
Python官方网站：<http://www.python.org>



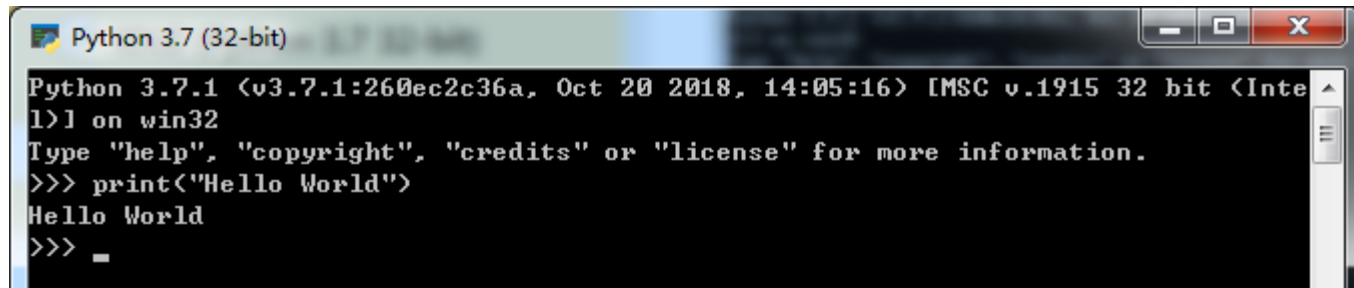
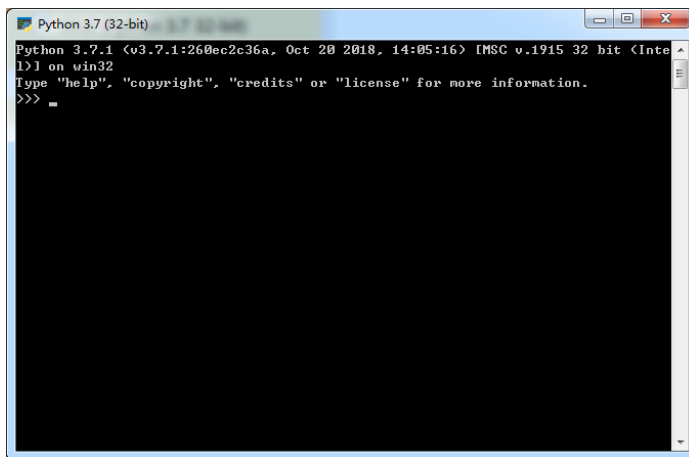
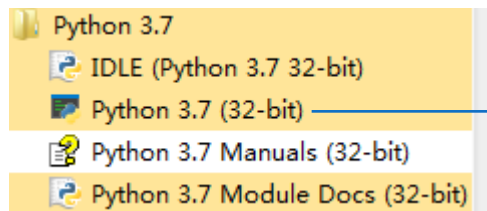
Python安装



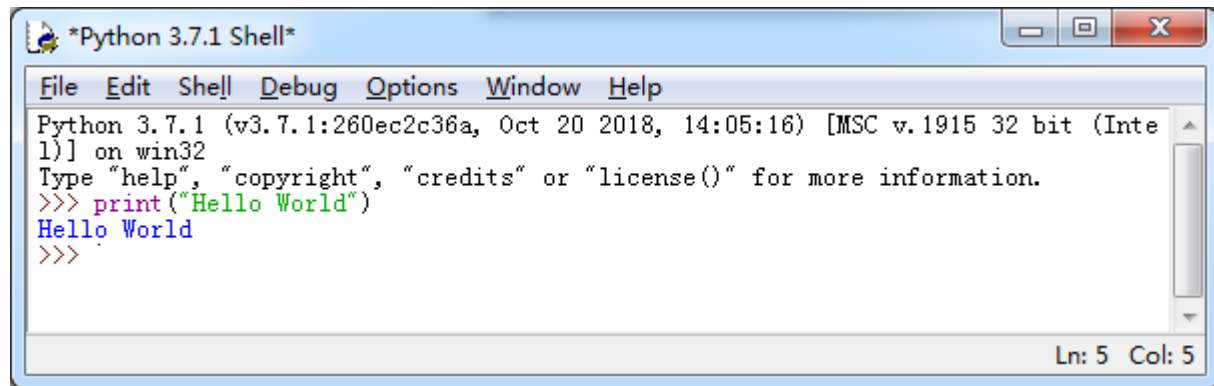
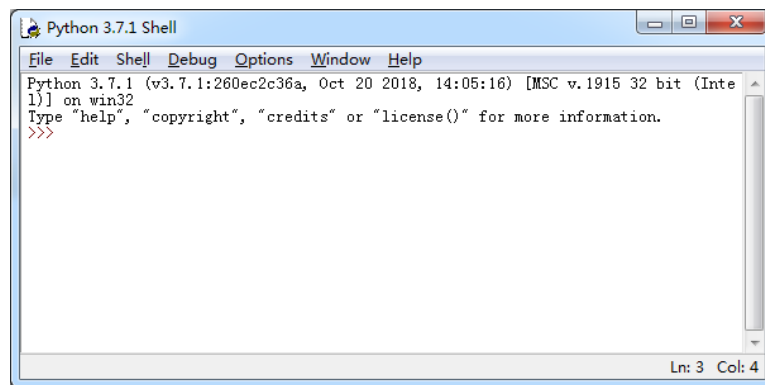
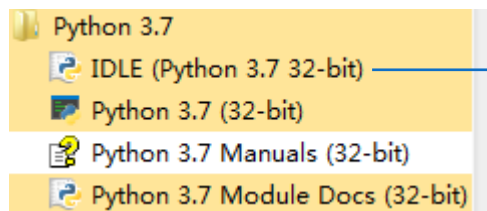
Python安装



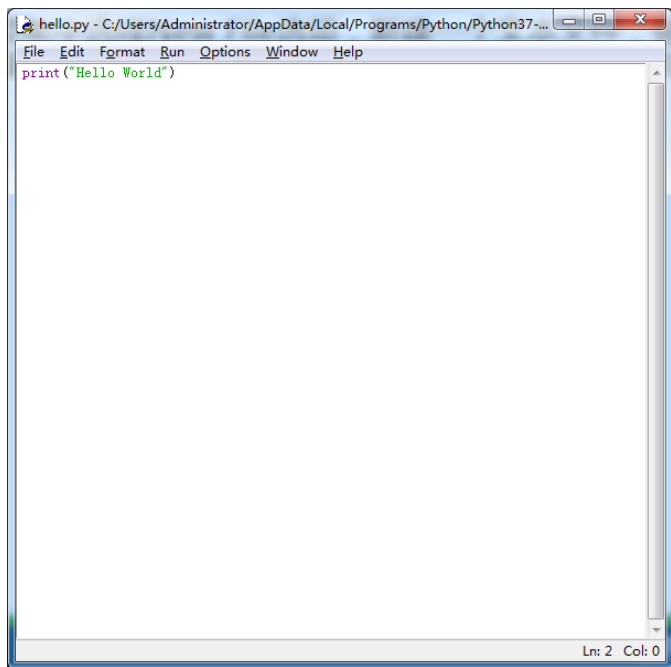
- 交互式方式运行程序（使用Python命令终端）



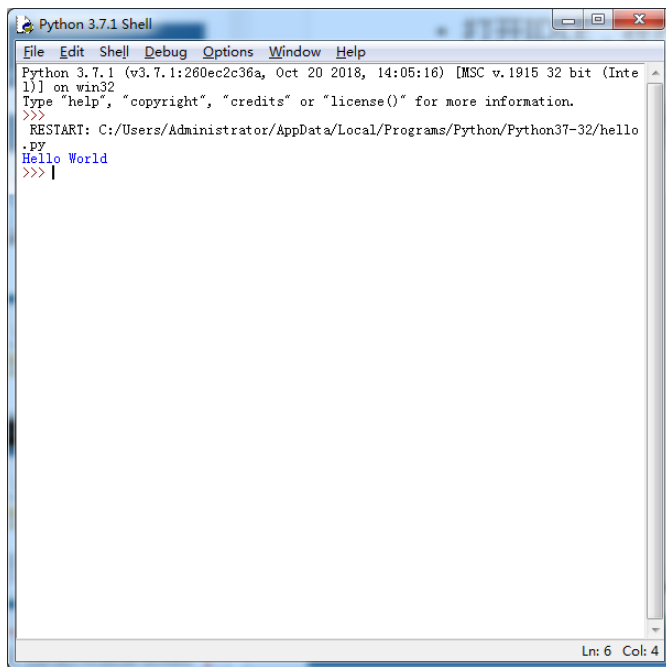
- 交互式方式运行程序（使用IDLE交互环境）



- 文件方式运行程序（使用IDLE创建文件）
 - 打开IDLE，在File菜单中选择New File选项，新建一个文本文件，输入代码并保存为hello.py，然后选择Run菜单中Run Module选项运行该程序，结果在交互环境中显示。



The screenshot shows the Python IDLE editor window. The title bar reads 'hello.py - C:/Users/Administrator/AppData/Local/Programs/Python/Python37-...'. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code editor contains a single line of Python code: `print('Hello World')`. The status bar at the bottom right indicates 'Ln: 2 Col: 0'.



The screenshot shows the Python 3.7.1 Shell window. The title bar reads 'Python 3.7.1 Shell'. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell displays the following text: 'Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32', 'Type "help", "copyright", "credits" or "license()" for more information.', and a prompt 'RESTART: C:/Users/Administrator/AppData/Local/Programs/Python/Python37-32/hello.py'. The output 'Hello World' is displayed, followed by a new prompt '>>>>'.

□Python程序两种运行方式的比较

□交互方式可以快速执行代码，但无法保存代码。建议在实验和测试一些简单的语法和小程序段时使用。

□文件方式可以保存代码。建议在编写正式的程序时采用。

Part Five

...

第五讲 第一个Python程序


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

这是Python最常用的一个函数，用来向屏幕输出。

这是print函数的参数，要输出什么，就把什么写在print后的小括号里。
注意：这里输出一串字符，需要用双引号把这串字符括起来！

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

- 被引号限定的这个字符序列叫做字符串。
- 必须把字符串中的内容包含在引号里来明确表示这是一个字符串。
 - 例如`print("print")`中带引号的`print`表示这是一个字符串，不带引号的`print`则是一个函数。
- 字符串可以放在一对双引号或一对单引号中。

```
print(100)
```

- `print`函数可以直接输出数值（显示100）。

```
print(1+2)
```

- `print`函数也可以计算一个表达式的值并输出（显示3）。

```
print("the answer is", 3 * 4)
```

- `print`函数可以接收多个参数并按给定顺序输出（显示
the answer is 12）。

print语句语法


```
print()
```

```
print(value1, value2, ..., valuen)
```

```
print()
```

print函数所有参数都是可选的，不指定参数，就输出一个空行。

```
print("the answer is", 3 * 4, "!")
```



这是要输出的值，按顺序一个接一个输出，输出时使用空格分隔。

- 三类程序错误

- 编译时错误（语法错误）

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

SyntaxError: EOL while scanning string literal

- 运行时错误（异常）

```
print(1/0)
```

Traceback (most recent call last):
File "<pyshell#1>", line 1, in <module>
print(1/0)
ZeroDivisionError: division by zero

- 逻辑错误

```
print("Hello Word")
```

- 开发程序很难避免错误。
 - 语法错误——编译器报告错误。
 - 运行时错误——解释器报告错误。
 - 逻辑错误——解释器无法标记，需要程序设计者自己来检查和避免。

Part Six

...

第六讲 图形化程序设计入门



这些漂亮的作品都是用Python的turtle库编程产生的！

- turtle是Python内嵌的图形绘制模块，可理解为一个绘图函数库。
- turtle绘图的基本框架思想：
 - 一个小海龟（画笔）在一个直角坐标系中爬行，其爬行的轨迹构成了图形。
 - 小海龟有前进、后退、旋转等爬行行为。
 - 小海龟的头部可对应前进方向、后退方向、左侧方向、右侧方向等方位。
 - 小海龟初始位于画面正中央，坐标为(0,0)。

- Python里有成千上万个库（模块），都可以采用import关键字来引用。

```
import 库名
```

- 通常情况下，库中会提供很多函数。如果想使用这些函数，可以采用如下的形式去调用。

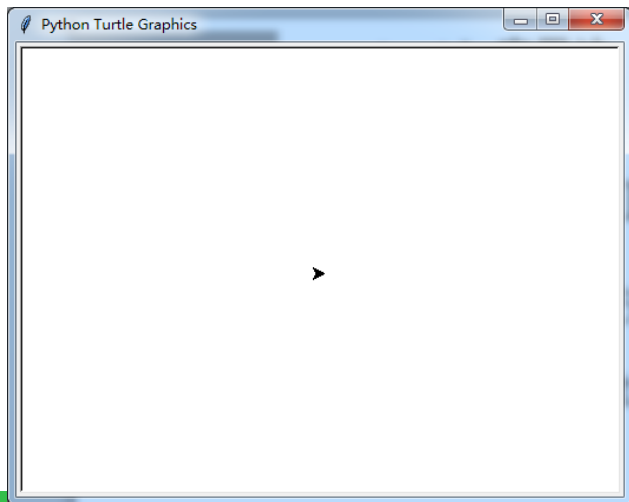
```
库名.函数名(函数参数)
```

- 导入turtle模块

```
>>> import turtle
```

- 显示turtle当前的位置和方向

```
>>> turtle.showturtle()
```



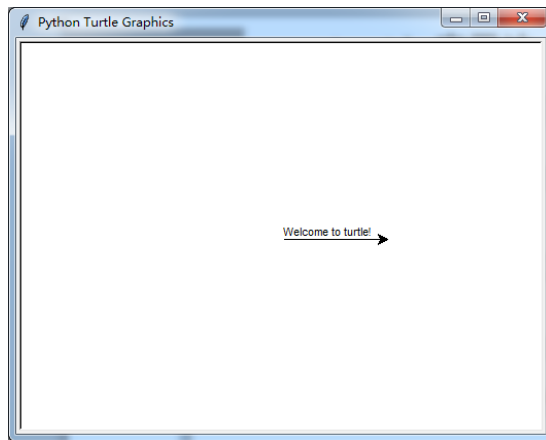
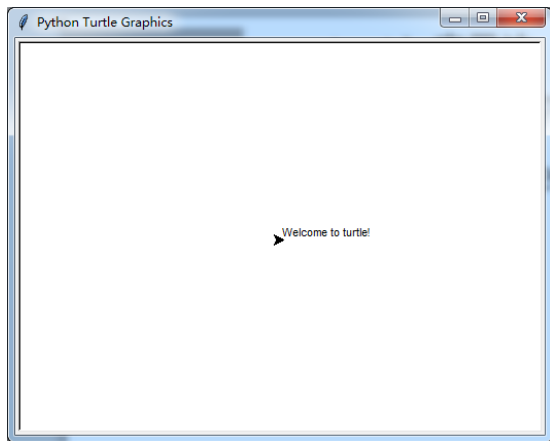
turtle库用法示例

- 输入命令绘制一个文本字符串

```
>>> turtle.write("Welcome to turtle!")
```

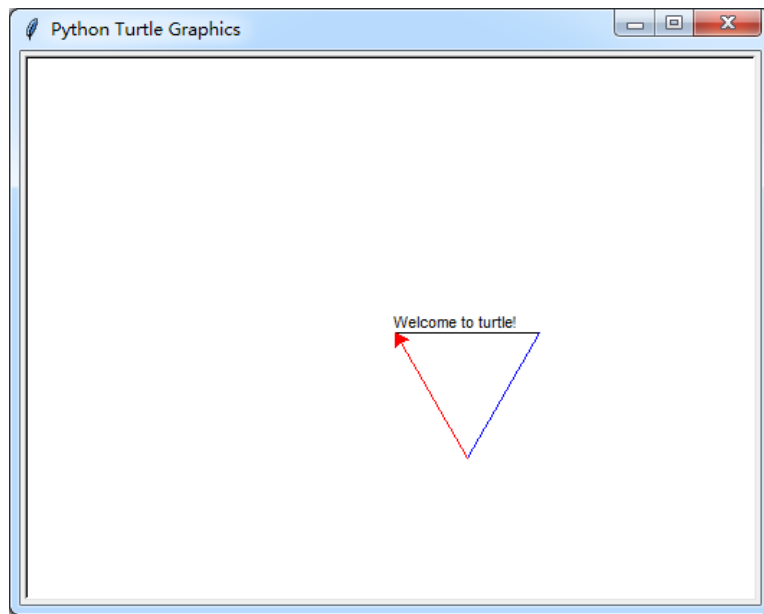
- 输入命令将箭头向前移动100像素，同时绘制出一条直线

```
>>> turtle.forward(100)
```



- 连续命令：右转120度，换成蓝色，移动100像素，右转120度，换成红色，移动100像素

```
import turtle
turtle.showturtle()
turtle.write("Welcome to turtle!")
turtle.forward(100)
turtle.right(120)
turtle.color("blue")
turtle.forward(100)
turtle.right(120)
turtle.color("red")
turtle.forward(100)
```



函数名	功能
<code>turtle.forward(distance)</code>	向当前画笔方向移动distance像素长度
<code>turtle.backward(distance)</code>	向当前画笔相反方向移动distance像素长度
<code>turtle.right(degree)</code>	顺时针移动degree°
<code>turtle.left(degree)</code>	逆时针移动degree°
<code>turtle.pendown()</code>	移动时绘制图形，缺省时也为绘制
<code>turtle.penup()</code>	提起笔移动，不绘制图形，用于另起一个地方绘制
<code>turtle.goto(x,y)</code>	将画笔移动到坐标为x,y的位置(画笔未提起时按直线移动并绘制)
<code>turtle.hideturtle()</code>	隐藏画笔的turtle形状
<code>turtle.showturtle()</code>	显示画笔的turtle形状
<code>turtle.write(s, [font])</code>	写文本，s为文本内容，font是字体的参数

- 练习一：在屏幕正中心画一个长度为200像素的十字。
- 练习二：画出一个五角星（提示：五角星的每个内角都是36度）。

Python语言程序设计



THANK YOU !

