

步骤一

Python 基础知识

步骤介绍

◆ python初识

◆ 流程控制语句

◆ 可变类型的数据集合

◆ 变量和字符串

◆ 运算符

◆ 非可变类型的数据集合

第一课 Python初识

课时介绍

- ◆ 介绍Python语言的背景与特点
- ◆ 讲解Python的执行原理
- ◆ 演示Python运行与开发环境搭建

课程目标

- ◆ 理解Python的运行原理
- ◆ 掌握安装Python解释器
- ◆ 掌握PyCharm的使用技巧

Python编程基础

课程介绍

- ◆ 了解Python的特点与优势
- ◆ 掌握Python3基本语法
- ◆ 掌握Python3的核心特性

课程介绍

- ◆ 介绍Python 3
- ◆ 介绍Python 3的特点
- ◆ 重点讲解解释器的作用

为什么要学Python

- ◆ Python十分简洁，上手轻松
- ◆ Python是最好大数据、人工智能语言
- ◆ 庞大的用户群体，成熟的技术体系

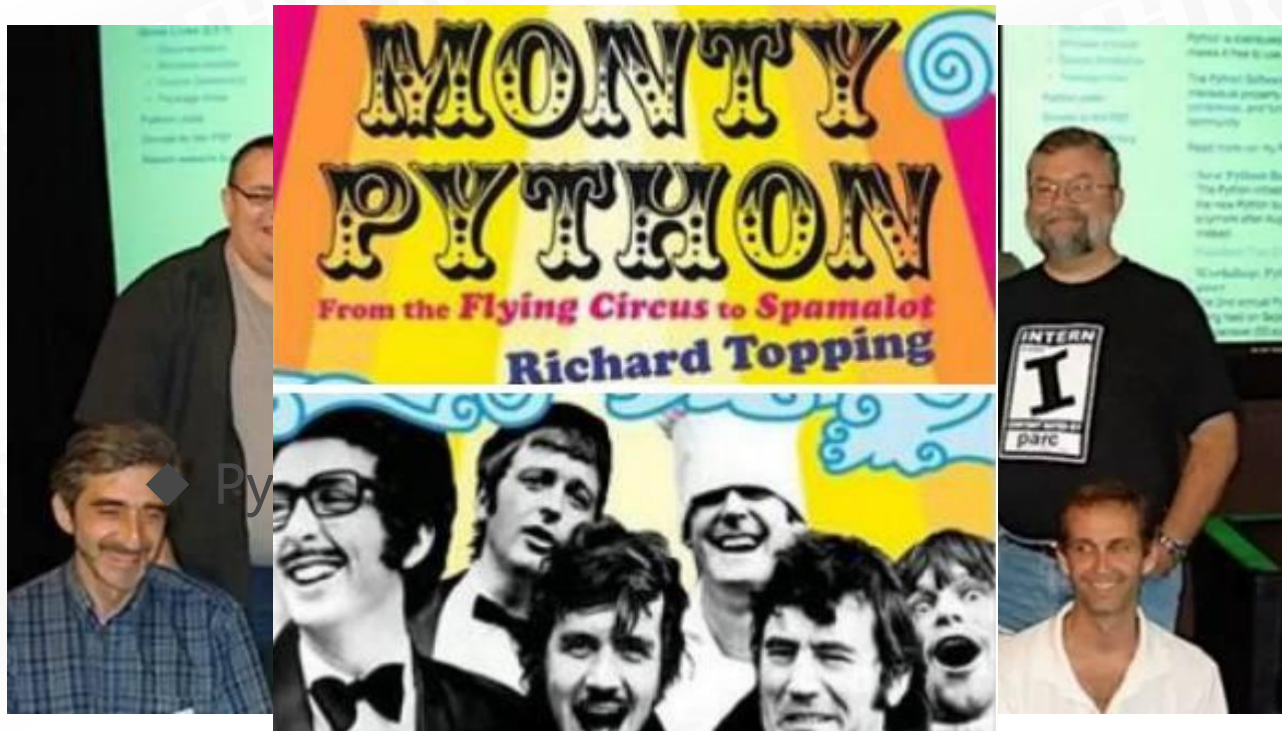
编程语言的发展史

◆ 第一代：机器语言

◆ 第二代：汇编语言

◆ 第三代：高级语言

Python的由来



Python的版本

- ◆ Python主要版本：Python 2 与Python 3
- ◆ Python 3 目前最新版本为3.7
- ◆ Python 2 已经停止更新，2020年退出历史舞台

Python的设计目标

- ◆ 简单优雅的语言，像自然语言一样容易理解
- ◆ Python是**开源**的，全世界程序员都在为之添砖加瓦
- ◆ Python适用于**短平快**的日常任务

Python的特点

- ◆ Python是完全面向对象的语言
- ◆ Python拥有强大的标准库，代码量极少
- ◆ Python拥有海量第三方模块

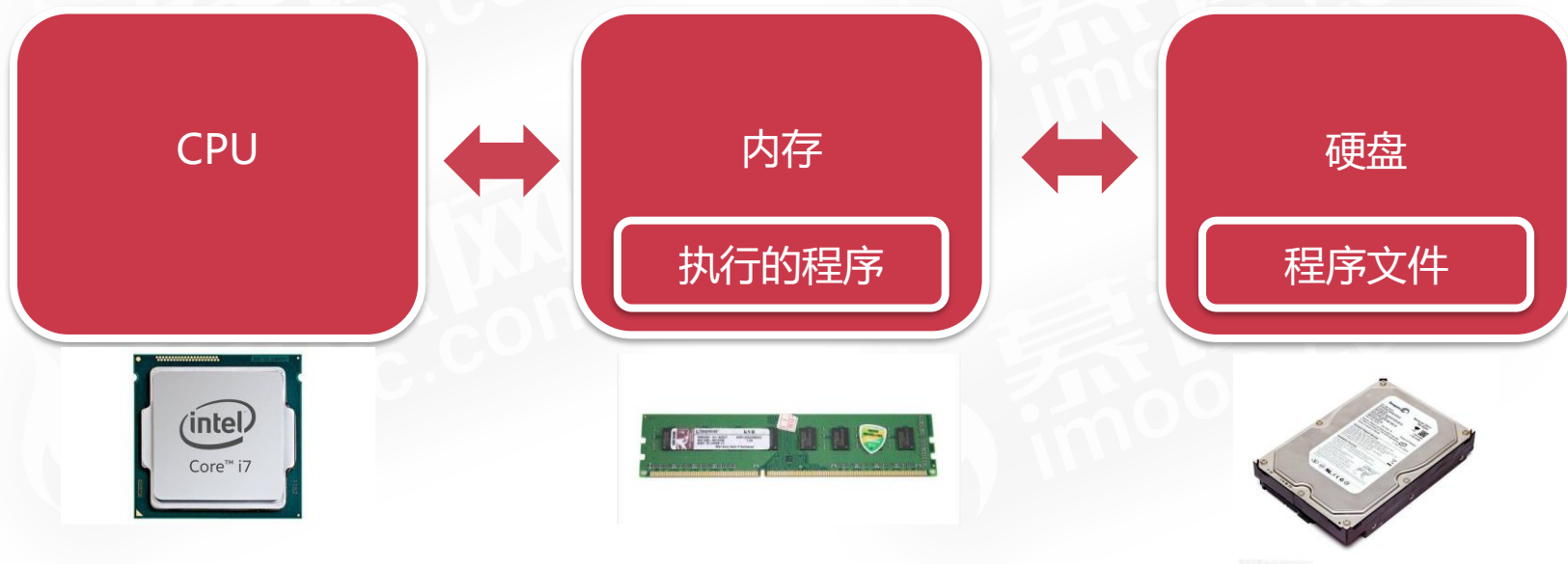
Python的优缺点

优点	缺点
简单、易学	运行速度慢
免费、开源	国内刚刚起步
丰富的库、可扩展性	中文资料匮乏

解释型语言

```
#自己飞机发射子弹
@classmethod
def shoot(cls):
    cls.hero.shoot()
    #子弹打到敌机让敌机从列表中消失
    enemyIndex = 0
    for i in cls.g_enemyList:
        enemyRect = pygame.Rect(i.image.get_rect())
        enemyRect.left = i.x
        enemyRect.top = i.y
        bulletIndex = 0
        for j in cls.hero.bulletList:
            bulletRect = pygame.Rect(j.image.get_rect())
            bulletRect.left = j.x
            bulletRect.top = j.y
            if enemyRect.colliderect(bulletRect):
                #判断敌机的宽度或者高度, 来知道打中哪种类型的敌机
                if enemyRect.width == 39:
                    cls.score += 1000    #小中大飞机分别100,500,1000分
                elif enemyRect.width == 60:
                    cls.score += 5000
                elif enemyRect.width == 78:
                    cls.score += 10000
                cls.g_enemyList.pop(enemyIndex)    #敌机删除
                cls.hero.bulletList.pop(bulletIndex)    #打中的子弹删除
            bulletIndex += 1
        enemyIndex += 1
```


程序的执行原理



解释器的执行流程



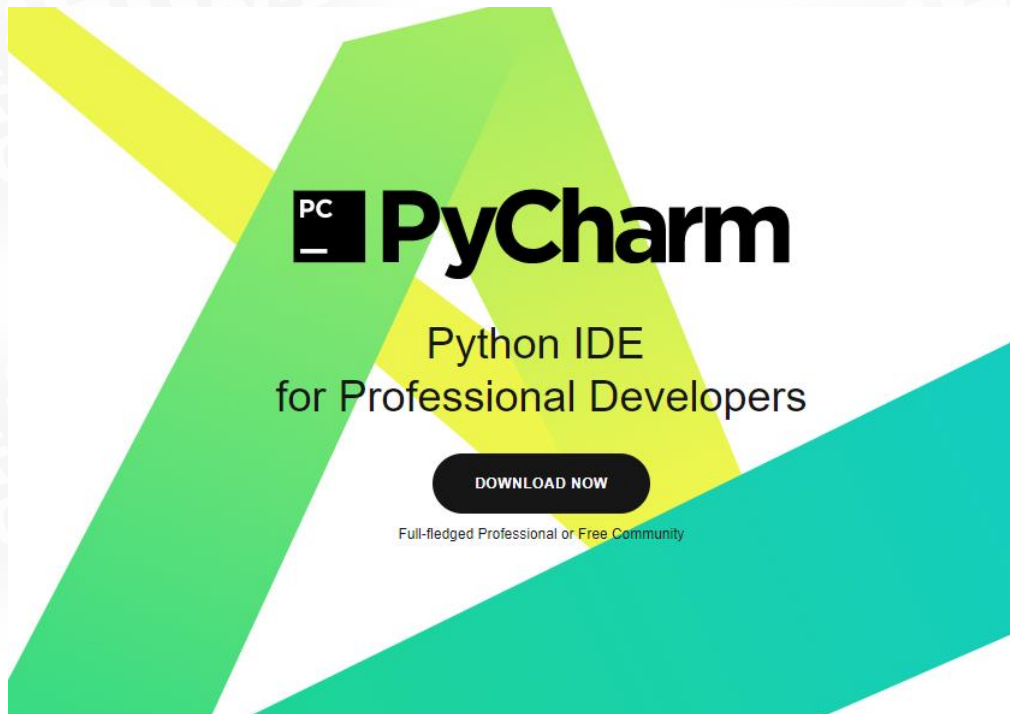
安装Python 3解释器



IDE与PyCharm

- ◆ 集成开发环境 (Integrated Development Environment)
- ◆ PyCharm是Python最好的IDE之一
- ◆ Pycharm是JetBrains公司的产品，具备跨平台特性

IDE与PyCharm



安装PyCharm



PyCharm项目

- ◆ PyCharm采用“项目（Project）”的方式管理源代码
- ◆ 一个项目通常就是一个完整的程序，包含多个源代码文件
- ◆ 通常项目是以目录的方式保存

熟悉PyCharm

- ◆ 了解PyCharm的界面布局
- ◆ 设置PyCharm的解释器
- ◆ 调整PyCharm的界面主题与样式

初次使用PyCharm



第一课 Python初识

课程总结

- ◆ 了解Python的特点
- ◆ 解释器就是Python的核心
- ◆ 掌握Python与PyCharm的安装配置