

概述

编程语言

什么是编程语言

编程语言的发展

Python的介绍

诞生

发展

特点

用途

概述

「该节主要了解一下Python的非代码层面的基本知识和概念，只需要留个印象就可以了！」

本小节没有重点知识，但是作为Python程序员，对诸如Python历史之类的内容应该尽量了解一些！

建议将该小节作为故事了解，阅读一下基本就可以了！

编程语言

什么是编程语言

编程语言（programming language）：**可以简单的理解为一种计算机和人都能识别的语言** 作用是：让程序员能够准确地定义计算机所需要使用的数据，并精确地定义在不同情况下所应当采取的行动

简而言之，编程语言就是计算机和人都可以识别的语言，是人去告诉计算机做什么的一门语言



编程语言的发展

编程语言的发展，一般我们认为经历了三个时代

- 低级语言

主要包括被称为“天书”的机器语言以及汇编语言等。

- 高级语言

数十年来，全世界涌现了2500+种高级语言，有一些则流传至今，还有一些则逐渐消失，代表有C语言等。

- 面向对象

从上个世纪90年代发展至今，以对象为核心，在当代程序设计中尤为重要，代表语言有Python、Java、C++等

结论：编程语言，不论如何发展，其实就是计算机和人可以识别的语言 主要是人去告诉计算机做什么的一门语言。

Python的介绍



人生苦短，我用Python ——Life is short, I need Python

诞生

Python的创始人为吉多·范·罗苏姆（Guido van Rossum），江湖人称“龟叔”。

- 1989年的圣诞节期间，吉多·范罗苏姆为了在阿姆斯特丹打发时间，决心开发一个新的**解释程序**，作为ABC语言的一种继承
- ABC是由吉多参加设计的一种教学语言，就龟叔本人看来，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，究其原因，吉多认为是**非开放**造成的。吉多决心在Python中避免这一错误，并获取了非常好的效果

- 之所以选中Python（蟒蛇）作为程序的名字，是因为他是BBC电视剧蒙提·派森的飞行马戏团（Monty Python's Flying Circus）的爱好者
- 1991年，第一个Python**解释器**诞生，它是用C语言实现的，并能够调用语言的库文件



CSDN @ 超团团团团狗

发展

- **Python 0.9.0 (1991年)**：Python的第一个公开版本是在1991年发布的0.9.0。这个版本包含了诸多基本的语言特性，如模块、异常处理、函数以及核心数据类型（字符串、列表等）。
- **Python 1.0 (1994年)**：Python 1.0于1994年发布，引入了一些现在看来很基本的特性，如lambda、map、filter和reduce。此时Python已经有了一个相对完整的语言结构。
- **Python 2.0 (2000年)**：Python 2.0引入了重要的特性，包括垃圾回收机制和Unicode支持。在此之后，Python逐渐成为一门适用于多种领域的编程语言，从Web开发到科学计算。
- **Python 3.0 (2008年)**：为了解决Python 2版本中的一些设计缺陷和不一致性，Python 3.0（也被称为Python 3000或简称为Py3k）发布。这个版本引入了不兼容的语法和库变化，以提高语言的一致性和清晰度。尽管切换到Python 3花费了一些时间，但它为未来的发展奠定了基础。
- **Python 2的结束**：随着Python 2于2020年正式终止支持，官方推动用户升级到Python 3，以获得更好的性能、安全性和新特性。
- **Python的未来**：Python继续迎来新的发展。PEP（Python Enhancement Proposal）流程通过社区的力量推动语言的发展。随着新的特性的不断加入，Python在未来将继续发挥其在软件开发领域的重要作用。

特点

Python语言在设计之初就是专门为非专业程序员设计的。

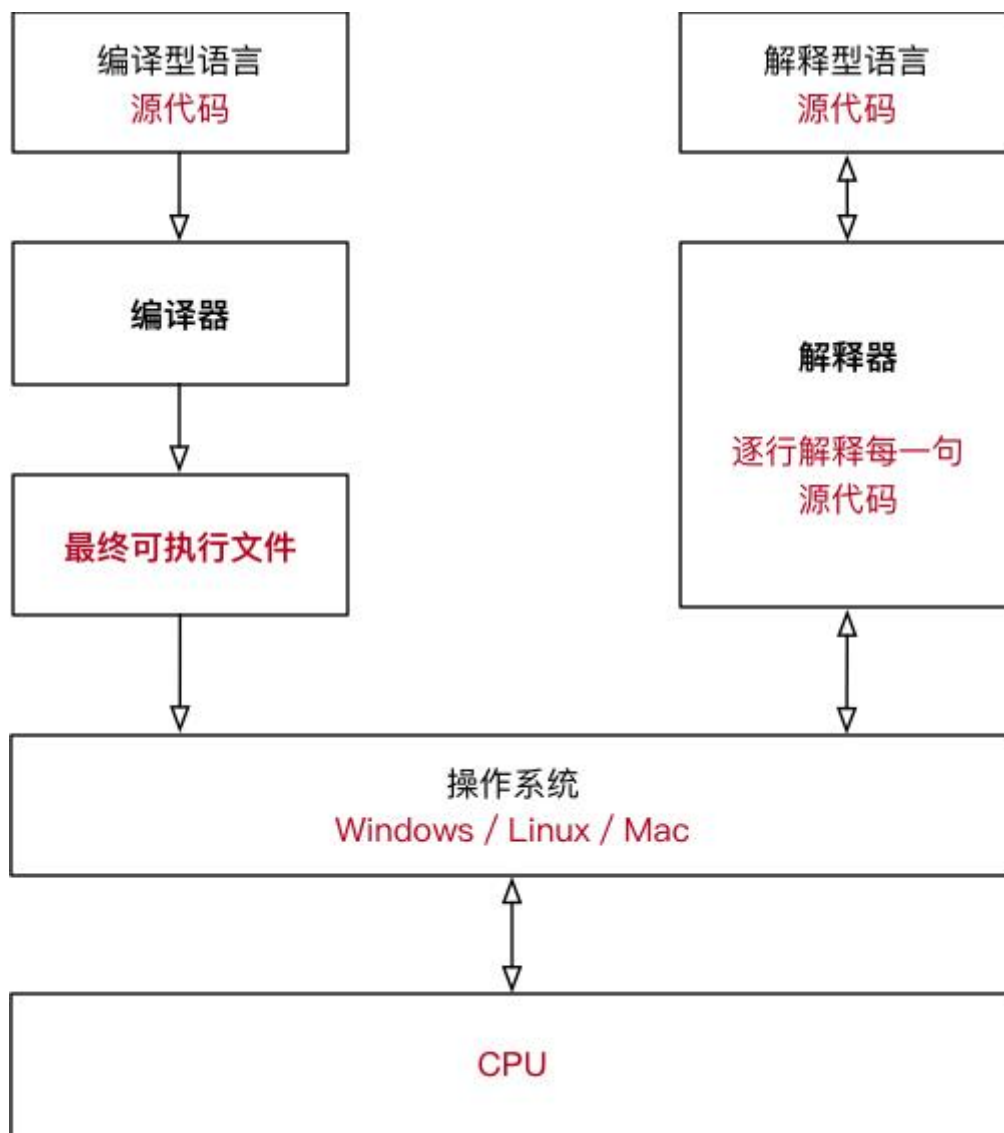
Python语言设计的初衷是让编程人员把精力放到解决问题上而不是编程语言本身。

优点：

- **简单：**Python是一种代表简单主义思想的语言。阅读一个良好的Python程序就感觉像是在读英语一样，Python的这种代码本质是它最大的优点之一。它使你能够专注于解决问题而不是去搞明白语言本身。
- **易学：**就如同你即将看到的一样，Python极其容易上手。前面已经提到了，Python有极其简单的语法。
- **开源：**Python是开源的。简单地说，你可以自由地阅读它的源代码、对它做改动、这是为什么Python如此优秀的原因之一——它是由一群希望看到一个更加优秀的Python的人创造并经常改进着的。
- **可移植性：**由于它的开源本质，Python已经被移植在许多平台上（经过改动使它能够工作在不同平台上）。如果你小心地避免使用依赖于系统的特性，那么你的所有Python程序无需修改就可以在下述任何平台上面运行。
- **丰富的库：**Python标准库确实很庞大。它可以帮助你处理各种工作，包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、CGI、FTP、电子邮件、XML、XML-RPC、HTML、WAV文件、密码系统、GUI（图形用户界面）、Tk和其他与系统有关的操作。记住，只要安装了Python，所有这些功能都是可用的。这被称作Python的“功能齐全”理念。

缺点：

- Python语言非常完善，没有明显的短板和缺点，唯一的缺点就是执行效率慢。
这个是解释型语言所通有的，同时这个缺点也将被计算机越来越强大的性能所弥补。



- 编译型语言：

程序员编写的源代码，在执行之前，需要先经过编译器，生成最终可执行的文件，然后再由操作系统去执行该文件

这些编译得到的可执行文件，在程序再次需要执行时，不需要重复编译，可以直接执行
- 解释型语言：

程序员编写的源代码，在运行的时候，不需要先编译，而是由解释器，逐行对源代码，进行解释翻译并交给操作系统执行。

对于解释型语言，看起来在第一次运行的时候少了道工序，实际在后续反复多次运行的时候，每一次都需要先解释再执行。

用途

为什么学习Python?

- 技术趋势

Python自带明星属性，热度稳居编程语言界前三，目前热度稳居世界第一



TIOBE编程语言热度排行榜

- 简单易学

开发代码少，精确表达需求逻辑；33个关键字，7种基本数据类型；语法规则简单，接近自然语言



- 引用广泛

Python语言涉及IT行业 70%以上的技术领域



后续就业方向

- **数据分析**
- 大数据开发
- 网络爬虫
- **人工智能**
- WEB开发
- 桌面软件
- 运维测试

其实目前很多知名的机器学习、人工智能以及深度学习框架也都是基于Python语言进行开发的：

- Facebook开源深度学习框架：Pytorch
- Google开源深度学习框架：TensorFlow
- 开源社区主推学习框架：Scikit-learn
- 百度开源深度学习框架：Paddle