

计算机语言和程序设计

第一讲 Python概述

计算机和程序设计基本概念



张 华

WHU

计算机和程序设计基本概念

计算机基础

- ✿ 计算机的硬件
- ✿ 计算机的组成和工作原理
- ✿ 计算机中信息的表示
- ✿ 计算机的软件
- ✿ 计算机程序和语言

程序设计基本概念

- ✿ 程序的基本编写方法
- ✿ 程序的开发过程
- ✿ 掌握计算机编程的价值
- ✿ 怎样学习计算机编程

计算机

计算工具的发展

从算盘到电子计算机



500B.C.



1642



1673



1822



1946



1981

计算机 (Computer)

在存储的指令 (instructions) 集 (程序 (programs)) 的控制下，接受输入，处理数据，存储数据，并产生输出的设备。

- 计算机是根据指令操作数据的设备。
- 可编程性：根据一系列指令自动地、可预测地、准确地完成操作者的意图。

计算机

计算机的分类

✿ 按计算机的功能可分类：

- 专用计算机
- 通用计算机

✿ 按计算机的综合性能指标（运算速度、存储容量、输入输出能力、规模大小、软件配置）分类：



巨型机



大型机



小型机



微型机（PC）



手持智能设备

计算机

计算机的特点

✿ 运算速度快

- 巨型机已达到每秒千万亿次

✿ 计算精度高

- 一般的**PC**可以表示十几位或几十位有效数字，计算精度达到百万分之几

✿ 超强的记忆能力

- 一般的**PC**配置**4~32GB**内存，**320~1000GB**外存（硬盘）

✿ 准确的逻辑判断能力

- 借助逻辑运算模拟人类思维

✿ 自动控制能力

- 自动执行所存储的程序

计算机

计算机的应用

- ✿ 科学计算（数值计算）
- ✿ 数据处理（信息处理）
- ✿ 自动控制
- ✿ 计算机辅助设计和辅助教学
- ✿ 人工智能方面的研究和应用
- ✿ 计算机网络的应用
- ✿ 多媒体技术的应用

计算机

计算机的发展

✿ 计算机的发展参照摩尔定律，表现为指数方式。

➤ 摩尔定律 **Moore's Law**

- 计算机发展历史上最重要的预测法则。
- Intel公司创始人之一戈登·摩尔在**1965**年提出。
- 单位面积集成电路上可容纳晶体管的数量约每两年翻一番。
- **CPU/GPU**、内存、硬盘、电子产品价格等都遵循摩尔定律。

➤ 计算机硬件所依赖的集成电路规模参照摩尔定律发展。

➤ 计算机运行速度因此也接近几何级数快速增长。

➤ 计算机高效支撑的各类运算功能不断丰富发展。

✿ 当今世界，唯一长达**50**年有效且按照指数发展的技术领域。

✿ 计算机深刻改变人类社会，甚至可能改变人类本身。

✿ 可预见的未来**30**年，摩尔定律还将持续有效。

计算机系统

计算机系统包括硬件和软件。

硬件 (Hardware)

- 组成计算机的各种物理部件
- 键盘、鼠标、显示器、内存、硬盘、光驱等等

软件 (Software)

- 计算机中运行的程序和数据，保存在外存上的程序和数据

计算机的组成

计算机包括五大部件

输入单元 (Input unit)

- 从输入设备（键盘、鼠标）获得数据

输出单元 (Output unit)

- 通过输出设备（显示器、打印机）显示或打印计算结果

内存储器 (Memory unit)

- 临时存储正在处理的数据和运行的程序
- 可以高速地访问
- 容量不大



计算机的组成

中央处理单元 (Central processing unit, CPU)

算术逻辑单元 (Arithmetic and logic unit, ALU)

➤ 进行算术和逻辑运算

控制器 (Control unit)

➤ 管理和协调计算机的其它部件



辅助存储器 (Secondary storage unit)

保存需长期存储的数据和非活动的程序

访问速度慢

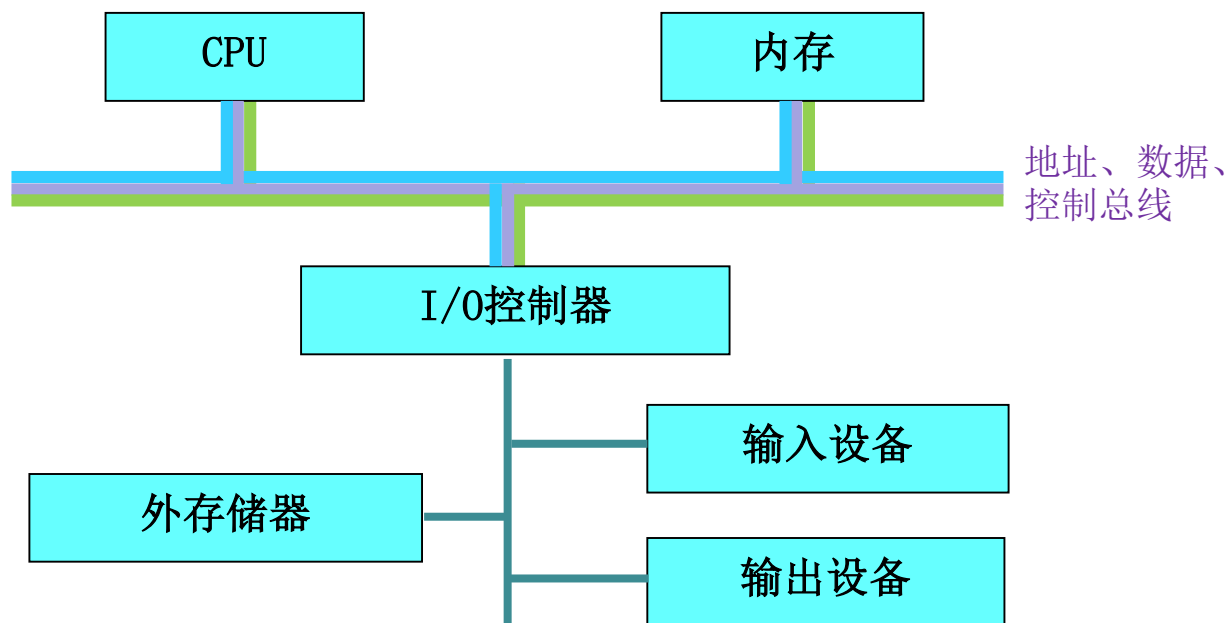
价格便宜、容量大

例如硬盘、光盘、磁带



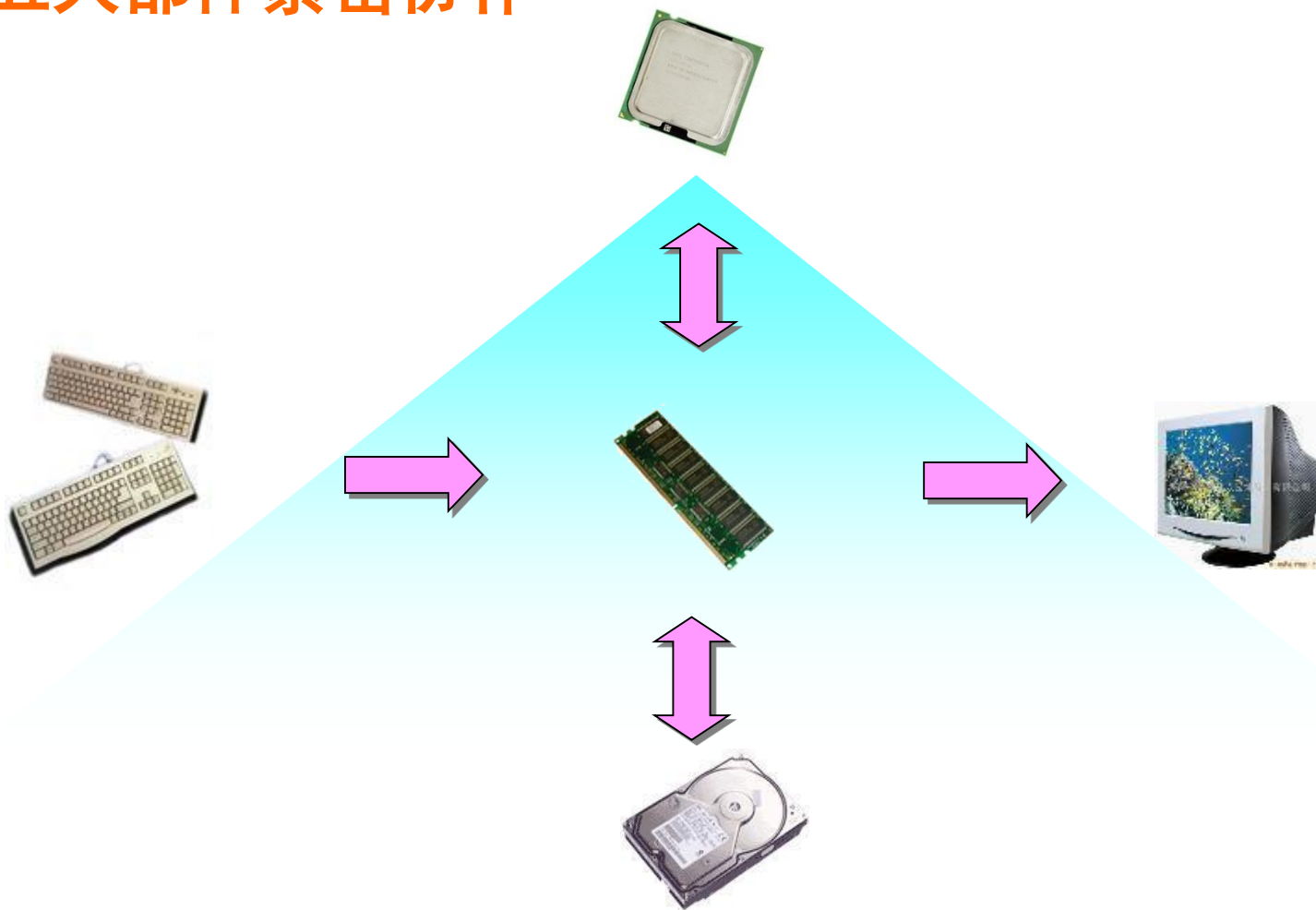
计算机的工作原理

计算机的结构示意图



计算机的工作原理

五大部件紧密协作



计算机的工作原理

■ CPU负责执行程序

- ✿ 从内存中读取一条指令
- ✿ 解析并执行该指令
- ✿ 读取下一条指令，解析并执行
- ✿
- ✿ 一个主频为1GHz的CPU可以在一秒钟内进行大约一亿次这样的操作



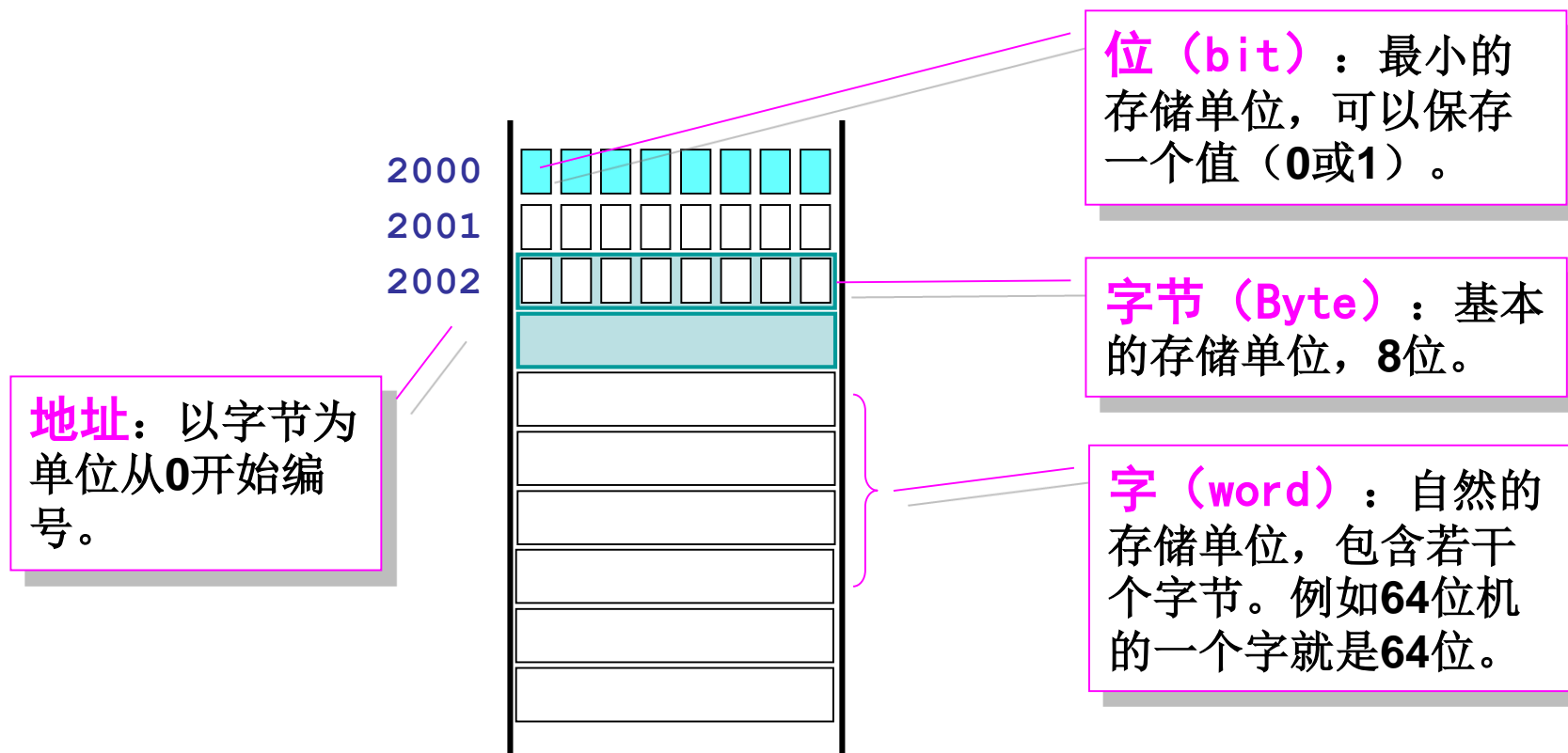
■ CPU有自己的工作区：包括若干个寄存器（Registers）

■ CPU只能理解有限的指令（CPU的指令集）

- ✿ 指令能够完成的任务是具体的
- ✿ 例如，把一个数从内存移动到某一个寄存器

位、字节和字

内存存储器的组织



计算机中的存储容量单位

常见的存储容量单位

- ✿ 1Kilo字节 (1KB) = 1024B (2^{10} 字节) (约 10^3 字节)
- ✿ 1Mega字节 (1MB) = 1024KB (2^{20} 字节) (约 10^6 字节)
- ✿ 1Giga字节 (1GB) = 1024MB (2^{30} 字节) (约 10^9 字节)
- ✿ 1Tera字节 (1TB) = 1024GB (2^{40} 字节) (约 10^{12} 字节)
- ✿ 1Peta字节 (1PB) = 1024TB (2^{50} 字节) (约 10^{15} 字节)
- ✿ 1Exa字节 (1EB) = 1024PB (2^{60} 字节) (约 10^{18} 字节)
- ✿ 1Zetta字节 (1ZB) = 1024EB (2^{70} 字节) (约 10^{21} 字节)
- ✿ ...
- ✿ 举例：百度每天处理的数据量将近100PB。
 - 相当于5000个国家图书馆的信息量总和。
 - 中国国家图书馆，预估藏书3510万册以上，世界排名第10位。

计算机中信息的表示

■ 存储在计算机中的一切内容都是二进制的数字形式

- ✿ 数据：数、字符、图片、声音、动画等等
- ✿ 指令

■ 常用数制

- ✿ 二、十、八、十六进制
- ✿ 不同进制数之间的转换

■ 计算机内信息的表示

- ✿ 数的表示
- ✿ 字符的表示
- ✿ 汉字的表示
- ✿ ...

```
011000110000
011001011100
0000110000011011
0110001100000110
0110010111001100
0000110000011011
```



计算机程序和语言

计算机程序

- ✿ 确切告诉计算机如何完成某项任务的指令序列。

软件

- ✿ 用来完成某种特定任务的一个或多个计算机程序以及附加的文件。
- ✿ 至少包含一个让用户运行的可执行文件。

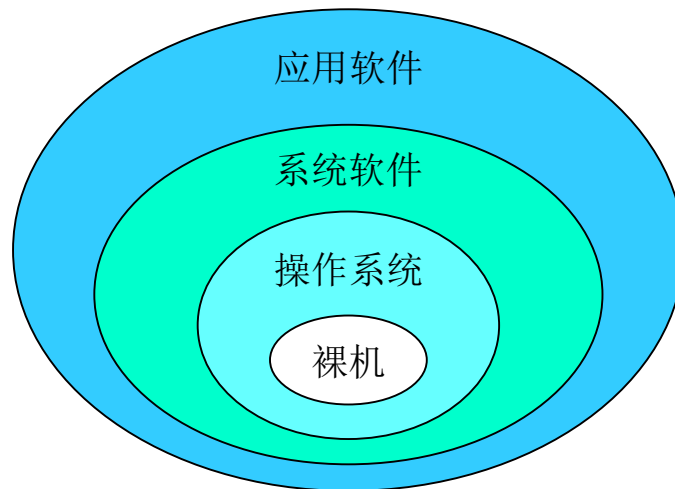
软件 的分类

系统软件

- 帮助计算机完成基本操作功能的软件。
- 包括操作系统、设备驱动程序、语言处理程序等。

应用软件

- 帮助人们使用计算机完成各种任务的软件。
- 例如文档制作、电子表格、数据管理、图形处理程序等等。



操作系统

操作系统 (Operating System, OS)

- ✿ 最基本，最重要的系统软件，是系统软件的核心。
- ✿ 负责管理计算机系统的全部软件资源和硬件资源，合理地组织计算机各部分协调工作，为用户提供操作和编程界面。

常用的操作系统

- ✿ Windows
- ✿ Unix/Linux
- ✿ Mac OS



计算机语言

计算机语言

- ✿ 亦称为程序设计语言、编程语言
- ✿ 用来编写程序
- ✿ 用于人和计算机之间进行交互
- ✿ 发展
 - 机器语言
 - 汇编语言
 - 高级语言

```
#include <stdio.h>
#include <math.h>
void main()
{
    float a,b,c,p,disc,q,x1,x2;
    scanf("%f%f%f",&a,&b,&c);
    p=-b/(2*a);
    disc=b*b-4*a*c;
    q=sqrt(disc)/(2*a);
    x1=p+q; x2=p-q;
    printf("x1=%f,x2=%f",x1,x2);
}
```



机器语言

■ 数字形式的指令码就是机器语言 (Machine language)

- ✿ 用机器语言编写的程序能够被计算机直接理解和执行
- ✿ 但是，编写程序非常费力
- ✿ 例如，编写程序完成两个数相加的任务需要以下几步
 - 把地址为**20**的内存单元中的数复制到寄存器**1**；
 - 把地址为**24**的内存单元中的数复制到寄存器**2**；
 - 把寄存器**2**中的数与寄存器**1**中的数相加，结果保留在寄存器**1**中；
 - 把寄存器**1**中的数复制到地址为**28**的内存单元中。

对应的机器语言程序（部分）

```
0110001000010100
0110001100011000
0000110000000011
0111001100011100
```

指令的结构

指令码	操作数
-----	-----

汇编语言

■ 符号化的指令码就是汇编语言 (Assembly language)

✿ 例如，完成两个数相加的汇编语言程序（部分）

```
ldreg n1, r1  
ldreg n2, r2  
add r1, r2  
store r1, sum
```

把变量n1的值复制到寄存器1 (r1) ;
把变量n2的值复制到寄存器2 (r2) ;
把r2中的数与r1中的数相加，结果保留在r1中；
把r1中的数复制到变量sum。

✿ 汇编语言程序必须被翻译成机器语言程序才能被执行

➤ 汇编程序 (Assembler) 充当“翻译”

✿ 用汇编语言编写程序还是比较费力

高级语言

高级语言 (High-level language) 使用数学符号和类似英语的单词

- 例如，完成两个数相加的C语言程序（部分）

```
sum = n1 + n2;
```

- 从多方面提高了编程效率

- 不必考虑CPU的指令集
- 不必考虑CPU实现特定任务的精确步骤
- 采用接近人类思考问题的方式去书写程序

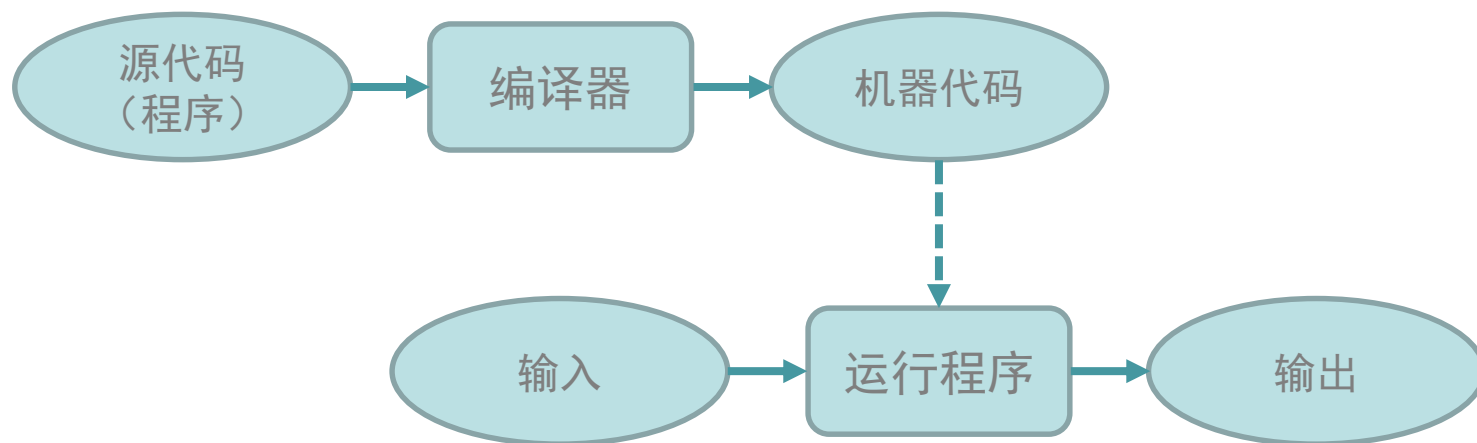
- 高级语言程序必须被翻译成机器语言程序才能被执行

- 由两种专门的程序来完成：编译器程序和解释器程序

编译器和编译执行

编译器 (Compiler)

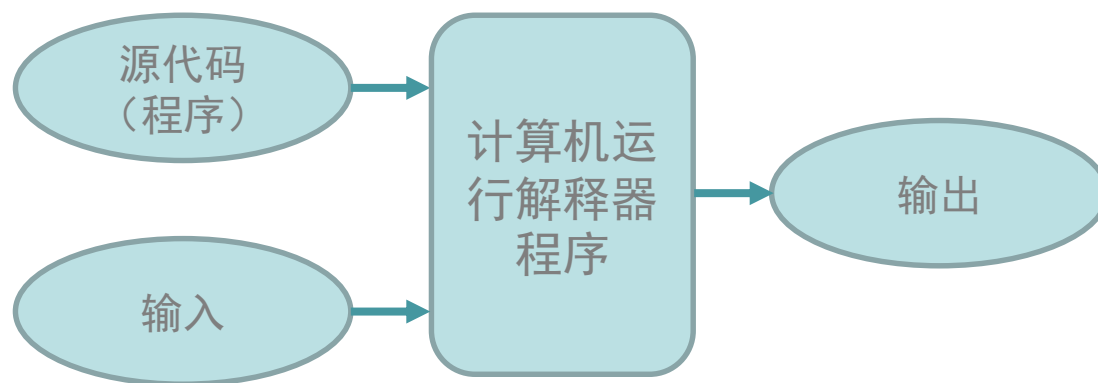
- ✿ 用编译器把用高级语言写好的程序翻译成机器语言程序，生成可执行文件，该文件可以独立运行。
- ✿ 这种运行程序的方式被称为**编译执行**。
 - C/C++, C#, Java



解释器和解释执行

解释器 (Interpreter)

- 一旦用高级语言写好程序后就可以运行，由解释器逐条翻译程序中的语句。
- 称为解释执行方式。
 - Python, Visual Basic (VB)



静态语言和脚本语言

根据执行方式不同，编程语言分为两类

✿ 静态语言：使用编译执行的编程语言

- C/C++语言、Java语言
- 编译器一次性生成目标代码，优化更充分。
- 程序运行速度更快。

✿ 脚本语言：使用解释执行的编程语言

- Python语言、JavaScript语言、PHP语言
- 执行程序时需要源代码，维护更灵活。
- 源代码维护灵活、跨多个操作系统平台。

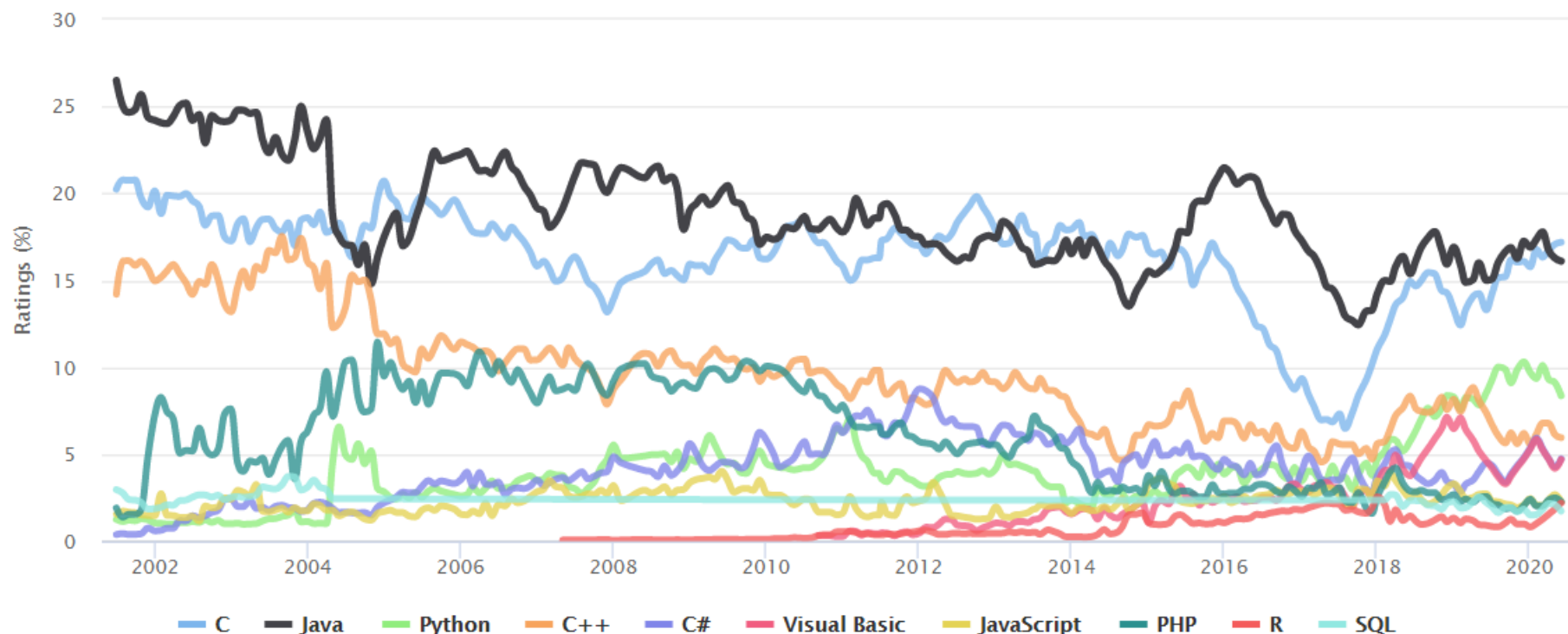
主流的编程语言

编程语言的流行趋势

🌿 <https://www.tiobe.com/tiobe-index>

TIOBE Programming Community Index

Source: www.tiobe.com



计算机程序设计

■ 程序设计是计算机可编程性的体现

- ✿ 程序设计，亦称编程，是深度应用计算机的主要手段。
- ✿ 程序设计已经成为当今社会需求量最大的职业技能之一。
- ✿ 很多岗位都将被计算机程序接管，程序设计将是生存技能。

■ 计算机只能解决计算问题，即问题的计算部分

- ✿ 问题的计算部分是一个待解决问题中可以用程序辅助完成的部分。
 - 可计算问题的一种解释：可用有限步骤计算出来的数学问题，即有明确解的数学问题。（感兴趣的同学可以课外阅读可计算理论）
- ✿ 一个问题可能有多种角度的理解，产生不同的计算部分。
- ✿ 问题的计算部分一般都有输入、处理和输出过程。

程序的基本编写方法

IPO

✿ Input: 输入，程序的输入

- 文件输入、网络输入、控制台输入、交互界面输入、内部参数输入等。
- 输入是一个程序的开始。

✿ Process: 处理，程序的主要逻辑

- 处理是程序对输入数据进行计算产生输出结果的过程。
- 处理方法统称为**算法**，它是程序最重要的部分。
- **算法是一个程序的灵魂。**

✿ Output: 输出，程序的输出

- 控制台输出、图形输出、文件输出、网络输出、操作系统内部变量输出等。
- 输出是程序展示计算/处理结果的方式。

求解计算问题的精简步骤

编程的三个步骤

- ✿ 确定IPO：明确计算部分及功能边界
- ✿ 编写程序：将计算求解的设计变成现实
- ✿ 调试程序：确保程序按照正确逻辑能够正确运行

程序的开发过程

❏ 复杂软件的开发是一项工程（软件工程），一般至少包括六个步骤

- ❁ 需求分析，定义程序目标（想清楚，规划IPO）
- ❁ 设计程序（关注算法，以及程序结构）
- ❁ 编写代码
- ❁ 构建程序（对编译执行的语言）
- ❁ 运行、测试和调试程序
- ❁ 维护和升级（适应问题的升级维护，更新完善程序）

程序的开发过程

需求分析，定义程序目标

- ✿ 在开始解决问题之前，程序的开发人员必须彻底了解问题是什么，对即将要创建的程序要做的事情有个清晰的认识。
- ✿ 这需要仔细考虑以下问题
 - 程序需要什么信息
 - 程序需要执行哪些计算和操作
 - 程序应该报告什么信息
- ✿ 例如，计算任意一个圆环的面积
 - 提出问题，明确目标
 - 圆环是什么？如何计算面积？需要什么数据？计算精度？
- ✿ 例如，为某航空公司开发一个机票订购系统
 - 试试站在航空公司的角度提需求

程序的开发过程

设计程序

- ✿ 决定程序如何去解决问题。
- ✿ 需要考虑
 - 如何给程序提供必要的数据
 - 程序中如何表示数据
 - 用什么方法来处理数据
 - 程序如何组织
 - 程序怎样报告结果
 - 完成这个程序需要多长时间
- ✿ 用一般的概念考虑设计中的问题，而不必考虑具体的语言和代码。

程序的开发过程

编写代码

- ✿ 编写代码来实现前面的设计，即用某一门语言来表示程序设计。
- ✿ 需要开发人员熟练掌握该语言的知识。
- ✿ 可以在草稿纸上勾画自己的想法或书写代码，但最终必须将代码输入计算机。
- ✿ 输入代码所采用的机制则取决于具体的编程环境。
 - 一般来说，需要使用文本编辑器（例如**Windows**的记事本程序）来创建一种文件，称为**源文件**
 - 源文件包含程序设计的计算机语言表示形式（称为**源程序**或**源代码**）

程序的开发过程

构建程序

✿ 把源程序转换成机器语言表示的程序。

- 这种程序可以直接被计算机理解和执行，所以称之为可执行程序，放在可执行文件中。
- 构建的具体细节取决于使用的编程环境。

✿ C/C++语言分两步完成这一工作：编译和链接。

- 编译器将源代码转换为目标代码，并存在目标文件中。
- 链接器将目标代码与其他代码结合起来生成可执行文件。
- 这种把编译和链接分开来做的方法便于程序的模块化。
 - 可以分别编译程序的各个模块，然后用链接器把编译过的模块结合起来。
 - 这样，如果需要改变一个模块，则不需要重新编译所有其他模块。

程序的开发过程

运行、测试和调试程序

- ✿ 运行包含可执行程序的文件，观察运行的结果。
 - 在不同的系统中运行程序的方式可能不同。
 - 例如，**Windows**的控制台和资源管理器。
- ✿ 应该对程序进行仔细的检查，看程序是否在做该做的事，结果是否正确或满足要求。
 - 比较好的做法是为验证程序的正确性设计一个测试计划。
 - 越早做越好，因为它有助于理清程序员的思路。
- ✿ 程序中的错误被称为**bug**，调试（**Debug**）就是要发现并修正错误。

掌握计算机编程的价值

编程能够训练思维

- ✿ 编程体现了一种抽象交互关系、自动化执行的思维模式。
- ✿ **计算思维**：区别逻辑思维和实证思维的第三种思维模式。
 - 计算思维（**Computational Thinking**）是运用计算机科学的基础概念进行问题求解、系统设计、以及人类行为理解等涵盖计算机科学之广度的一系列思维活动，由周以真于**2006年3月**首次提出。
 - 课外查阅相关资料。
- ✿ 能够促进人类思考，增进观察力和深化对交互关系的理解。

掌握计算机编程的价值

编程能够增进认识

- ✿ 编程不单纯是求解计算问题。
- ✿ 不仅要思考解决方法，还要思考用户体验、执行效率等。
- ✿ 能够帮助程序员加深用户行为以及社会和文化认识。

编程能够提高效率

- ✿ 能够更好地利用计算机解决问题。
- ✿ 显著提高工作、生活和学习效率。
- ✿ 为个人理想实现提供一种借助计算机的高效手段。

掌握计算机编程的价值

编程带来就业机会

- ✿ 程序员是信息时代最重要的工作岗位之一。
- ✿ 国内外对程序员岗位的缺口都在百万以上规模。
- ✿ 计算机已经渗透于各个行业，就业前景非常广阔。

编程能够带来乐趣

- ✿ 编程能够提供展示自身思想和能力的舞台。
- ✿ 让世界增加新的颜色、让自己变得更酷、提升心理满足感。
- ✿ 在信息空间里思考创新、将创新变为现实。

怎样学习计算机编程

合适的方法

- ✿ 首先，掌握编程语言的语法，熟悉基本概念和逻辑。
- ✿ 其次，结合计算问题思考程序结构，会使用编程套路。
- ✿ 最后，参照案例多练习多实践，学会举一反三。

小结

- 计算机是根据指令操作数据的设备，具有可编程性。
- 计算机语言是人和计算机交互的人造语言，有编译执行和解释执行两种方式。
- 计算机只能解决计算问题，问题的计算部分由程序来完成，一般包括输入、处理和输出（IPO）。
- 完整的程序开发过程包括六个步骤。
- 掌握计算机编程具有很高的价值，要采取合适的学习方法。