

# 编程入门基础

——编程语言综述

主讲教师：耿宇航

# 编程语言的多样性

- c c++ c# Java Basic Python Perl Javascript Ruby Delphi
- PL/SQL Lisp Pascal Matlab Bash Assembly PHP  
Transact-SQL
- Ada SAS Lua R COBOL Fortran Scheme D Elang Scala  
Haskell
- Logo Prolog Scratch TCL F# Smalltalk APL NXT-G
- Go ML Forth ActionScript RPG Groovy Foxpro ABC
- Alice Awk Clean Clojure Dart Factor OCaml Curl xBase
- .....

# 机器语言 vs 高级语言

- 面向机器的，直接处理硬件指令的语言
  - 宏汇编
- 面向人的，以人容易理解的方式描述等价的逻辑
  - Fortran Alogo-60 Pascal
- 优缺点
  - 机器语言快速高效，不便于阅读、理解、维护、移植。
  - 高级语言易于理解维护，需要一个“翻译”过程，效率不如机器语言
- Java是高级语言
- C 是“准高级”语言
  - 有人戏称为：高级汇编语言
  - 能够深入到内存、指令一级的控制。对程序员的驾驭能力要求高

# 编译过程 vs 解释过程

- 编译过程
  - 源代码 → 可执行代码
  - 脱离源代码，直接运行 “可执行代码”
- 解释过程
  - 读入一行源代码 → 翻译为机器指令 → 执行机器指令
  - 再读入一行源代码 .... 循环
- C 语言是编译型
- Javascript 是解释性
- Java是半编译半解释型
  - Java源代码 → java字节码 （编译过程）
  - Java虚拟机 解释 java字节码 （解释过程）

# 命令式 vs 函数式

- 命令式
  - 大多数高级语言都是命令式
  - 明确指示计算机运算步骤，先做什么...再做什么...如果...
- 函数式
  - 描述量间的函数依赖关系
  - 由执行程序 自行决定运算顺序 便于并发处理（尤其多cpu, 分布式环境等）
  - Lisp Haskell Clojure
  - Scala JavaScript
- C Java 是命令式语言

# 第X代语言

- 大多数流行的高级语言是：第三代。
  - 结构化，描述解决问题的过程
- 第4代
  - 面向问题，只描述问题是什么，需要什么，不描述如果求解。
  - SQL
  - 依据现代软件理论而构建 smalltalk-80
- Java c 是第三代

# 通用语言 vs 专用语言

- 通用语言
  - 不绑定特定领域，为通用目的而设计的
  - Pascal Ruby Python Ada
- 专用语言
  - 为某个应用领域而专门设计
  - Fortran 科学计算 数值应用
  - PowerBuilder 数据库前端开发
- Java C 都是通用语言

# 语言是不断演化的

- 流行语言版本不断升级
- 每种语言都吸收其它语言的合理成分
- 语言间的界限不断模糊

# 语言间相互调用

- 为了扩大应用范围，语言会提供接口
  - 调用其它语言
  - 被其它语言调用
- 语言间的互操作性

# 谢 谢 !