

Sec 1.4

The end of history..

languages are accepted and evolve socially

languages minimal resources

simple "performance model"

easy to know fast vs slow operations

alternately: what is a primitive operation in Big-O sense.

Non trivial to determine is many langs:

Sorting 10,000,000 items

Go:	array of structs by number:	4.7
	array of ints	4.7
	array of structs by string:	9.0
KT:	array of obs sort by number:	6.9
	array of ints	1.0
	array of obs by string:	15.7

so what is the speed difference between Kotlin and Go?

KT: ~40% slower
~66% slower
80% faster????

easy to understand

ALSO for a new lang:

widely available

local experts

"minimally acceptable"

similar to existing langs

reason to move

Semicolon wars

What is the purpose of the semi-colon?

why have it at all? Do you need it right before a } in java — why?

statement ender vs statement separator

how do you count — from 0, from 1 ... Why?

```
java/c a=2+3
      a=a+a
      System.out.println(String.format("%d", a))
lisp  (let ((a (+ 2 3)))
      (message "%d value" a))
```

imperitive, functional, Object-oriented, logic

Fortran "an infantile disorder"

PL1 "A fatal disease"

Cobol "mutilated beyond hope"

" write in XXXX — not because it is the best language, but because it is the language I know best"

This happens all of the time — examples

WHAT IS A COMPILER?

what does a compiler do?

translate from high-level language into machine language

2 aspects

thorough analysis

non-trivial transformation

eg. tail-recursion to iteration

page 17 picture

What is alternative to compiler?

interpreter

page 17 picture

why have a compiler vs interpreter?

tradeoffs between

interpret: flexibility, diagnostics (at run time)

LISP: program can write code that it executes

late binding

compile: speed!!!!

“Interpreted languages” — Java

picture on pg 18

why bother???

Multi-step “compilation” in C — preprocess, compile, link (pg 19 figure for Fortran)