Πλαν

Plan

A look at the future direction
of programming languages

Every programming language ever:

- "Systems"
- "Low-level"
- "Static"
- Manual memory management
- Compiled
- Fast

- "Scripting"
- "High-level"
- "Dynamic"
- Automatic memory management
- Interpreted
- Slow

Every programming language ever:

- "Systems"
- "Low-level"
- "Static"
- Manual memory management
- Compiled
- Fast

- "Embedded"
- Mid-level?
- Static/Dyamic
- Often, a "latched fence"
- Interpreted or compiled
- Quite fast

- "Scripting"
- "High-level"
- "Dynamic"
- Automatic memory management
- Interpreted
- Slow

Every programming language ever:

- Fortran
- COBOL
- B
- BCPL
- C
- Lua
- JavaScript
- mruby
- Go
- Shell
- sed/awk
- perl
- Python/Ruby

High-level / Low-level

Fast	Slow
Compiled	Interpreted
Static	Dynamic
Manual	Automatic

High-level / Low-level

Fortran

Fast
Compiled
Static
Manual

Slow
Interpreted
Dynamic
Automatic

High-level / Low-level

Fortran

Classical Lisp

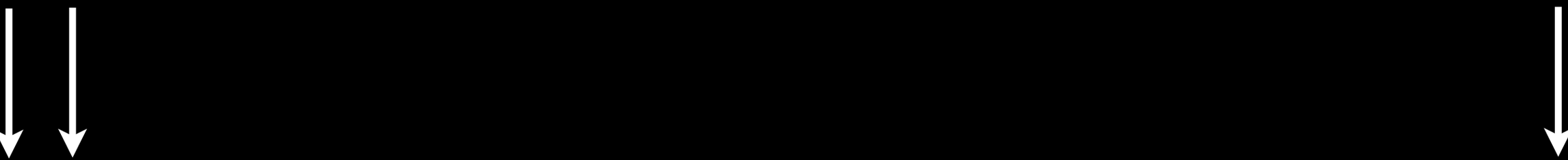
Fast
Compiled
Static
Manual

Slow
Interpreted
Dynamic
Automatic

High-level / Low-level

COBOL
Fortran

Classical
Lisp



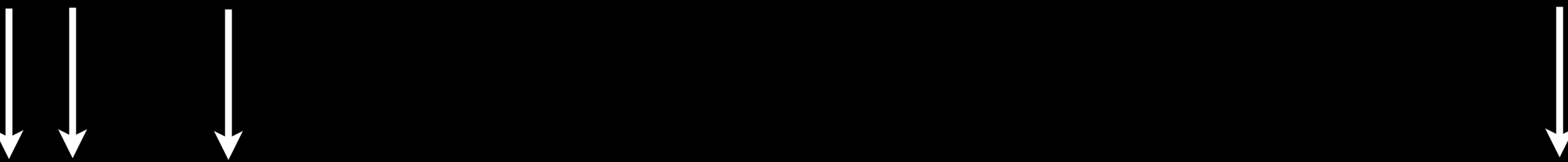
Fast
Compiled
Static
Manual

Slow
Interpreted
Dynamic
Automatic

High-level / Low-level

COBOL
Fortran ALGOL

Classical
Lisp



Fast
Compiled
Static
Manual

Slow
Interpreted
Dynamic
Automatic

High-level / Low-level

Fast
Compiled
Static
Manual

Slow
Interpreted
Dynamic
Automatic

High-level / Low-level

Fast
Compiled
Static
Manual

Slow
Interpreted
Dynamic
Automatic

High-level / Low-level

Fast
Compiled
Static
Manual

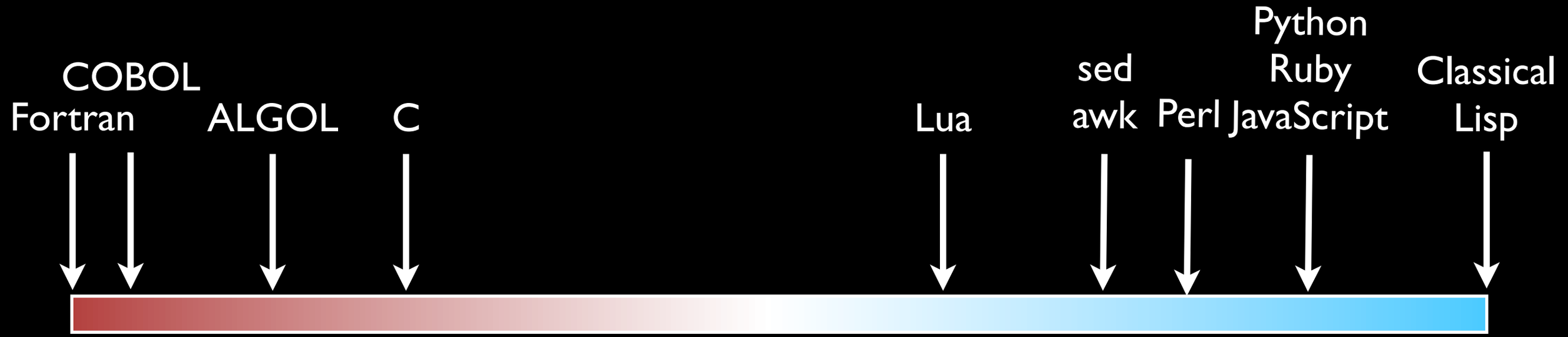
Slow
Interpreted
Dynamic
Automatic

High-level / Low-level

Fast
Compiled
Static
Manual

Slow
Interpreted
Dynamic
Automatic

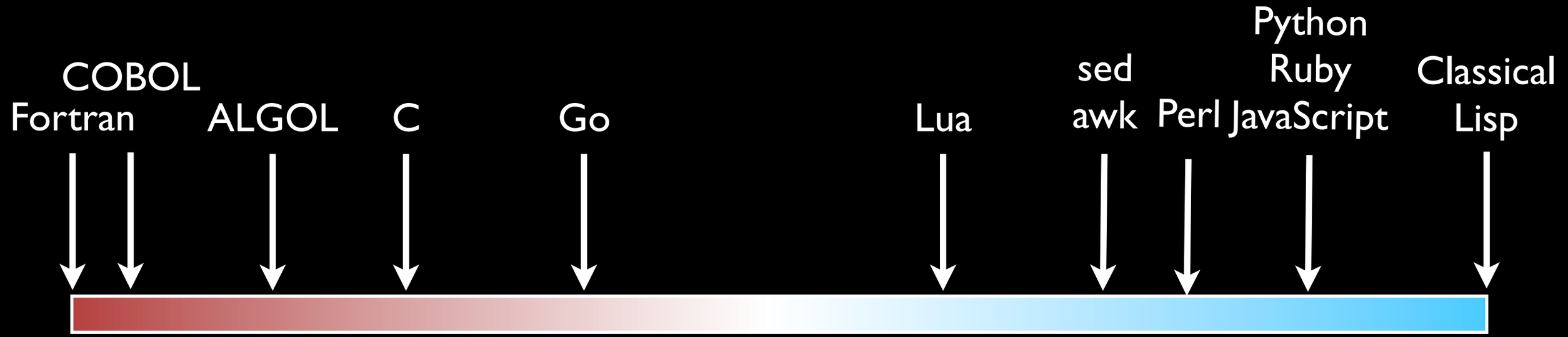
High-level / Low-level



Fast
Compiled
Static
Manual

Slow
Interpreted
Dynamic
Automatic

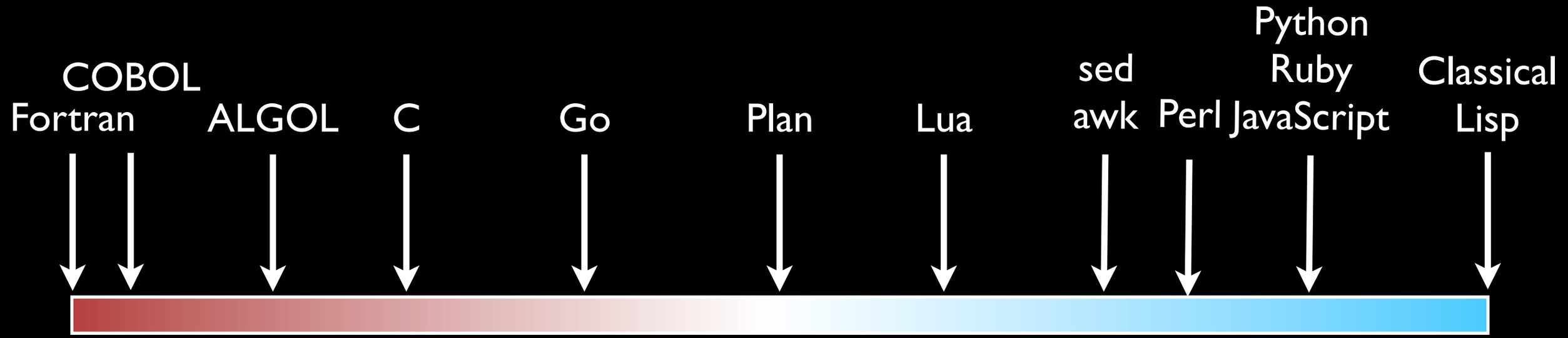
High-level / Low-level



Fast
Compiled
Static
Manual

Slow
Interpreted
Dynamic
Automatic

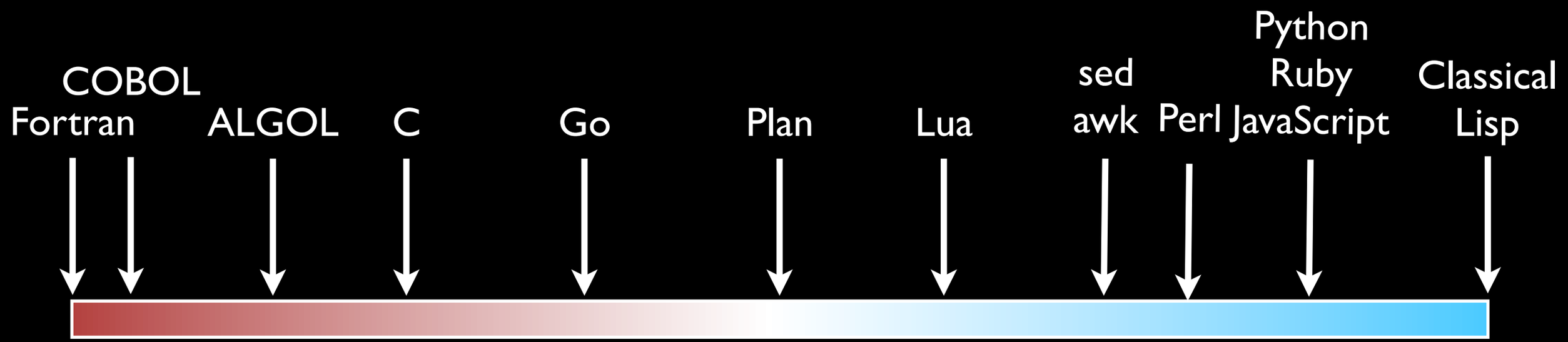
High-level / Low-level



Fast
Compiled
Static
Manual

Slow
Interpreted
Dynamic
Automatic

High-level / Low-level



Fast
Compiled
Static
Manual

Slow
Interpreted
Dynamic
Automatic

(with apologies to Paul Graham)

Saturday, 29 December 12

```
(deffn (map function:f cons:xs)
  ...)
```

```
(deffn (map function:f array:xs)
  ...)
```

```
(deftype list (type? 'cons ':nil))

(set! pair (cons 'a (fn () 'b)))
```

```
(tag pair 'disposition 'stream)

(deftype stream (tag? 'disposition 'stream))
```

Saturday, 29 December 12

```
a[b]
  ⇛ (get-prop a b)

(set a[b] c)
  ⇛ (set (get-prop a b) c)
  ⇛ (set-prop a b c)
```

```
(deffn (flatten list:xs)
  (if (nil? xs) nil
      (cons? (car xs))
        (append (flatten (car xs)) (flatten (cdr xs)))
      (cons (car xs) (flatten (cdr xs)))))
```

```
deffn flatten list:xs
  if nil? xs
       nil
     cons? (car xs)
       append (flatten (car xs)) (flatten (cdr xs))
     ;else
       cons (car xs) (flatten (cdr xs))
```

Thanks

- Alex Payne, Alex Miller, &c.
- Paul Graham (Arc)
- Rich Hickey (Clojure)
- Michael Halls-Moore (photo of me on website)