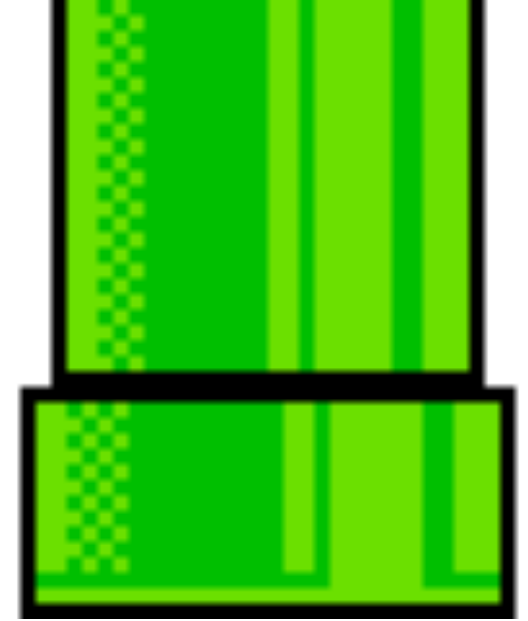






# Interpreters



$\lambda$   
42

Matt Might  
University of Utah  
[matt.might.net](http://matt.might.net)

```
function id(x)
{
    return x ;    // comment
}
```

FUNCTION

IDENT(id)

LPAR

IDENT(x)

RPAR

LBRACE

RETURN

IDENT(x)

SEMI

RBRACE

FUNCTION

LPAR

RPAR

LBRACE

IDENT ( )

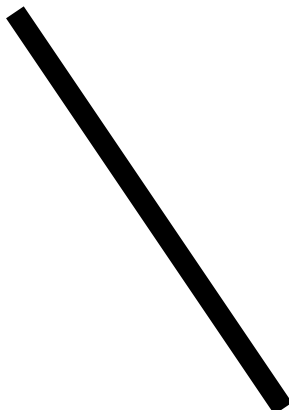
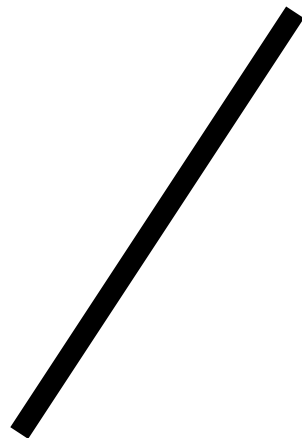
IDENT ( )

RETURN

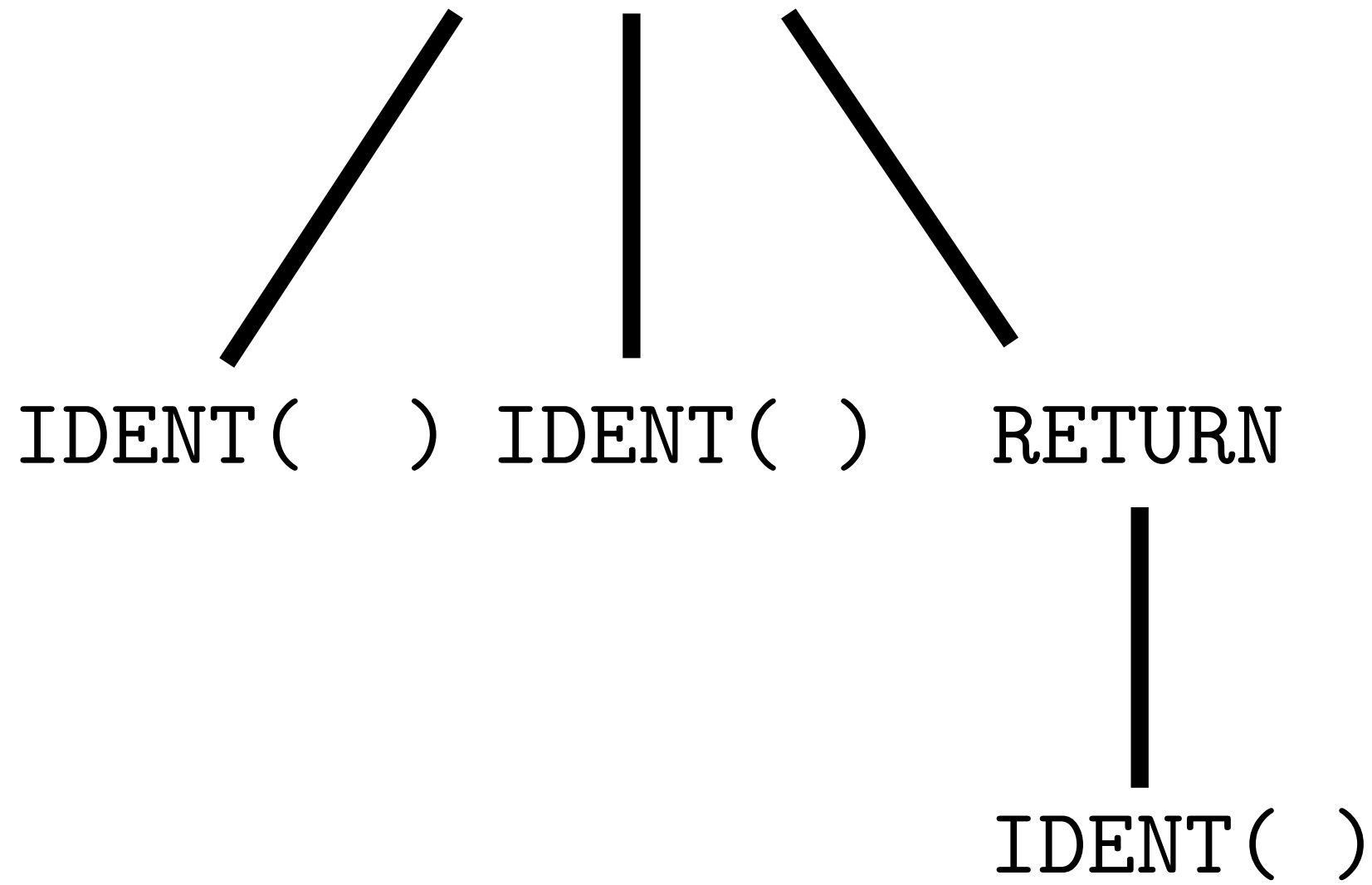
IDENT ( )

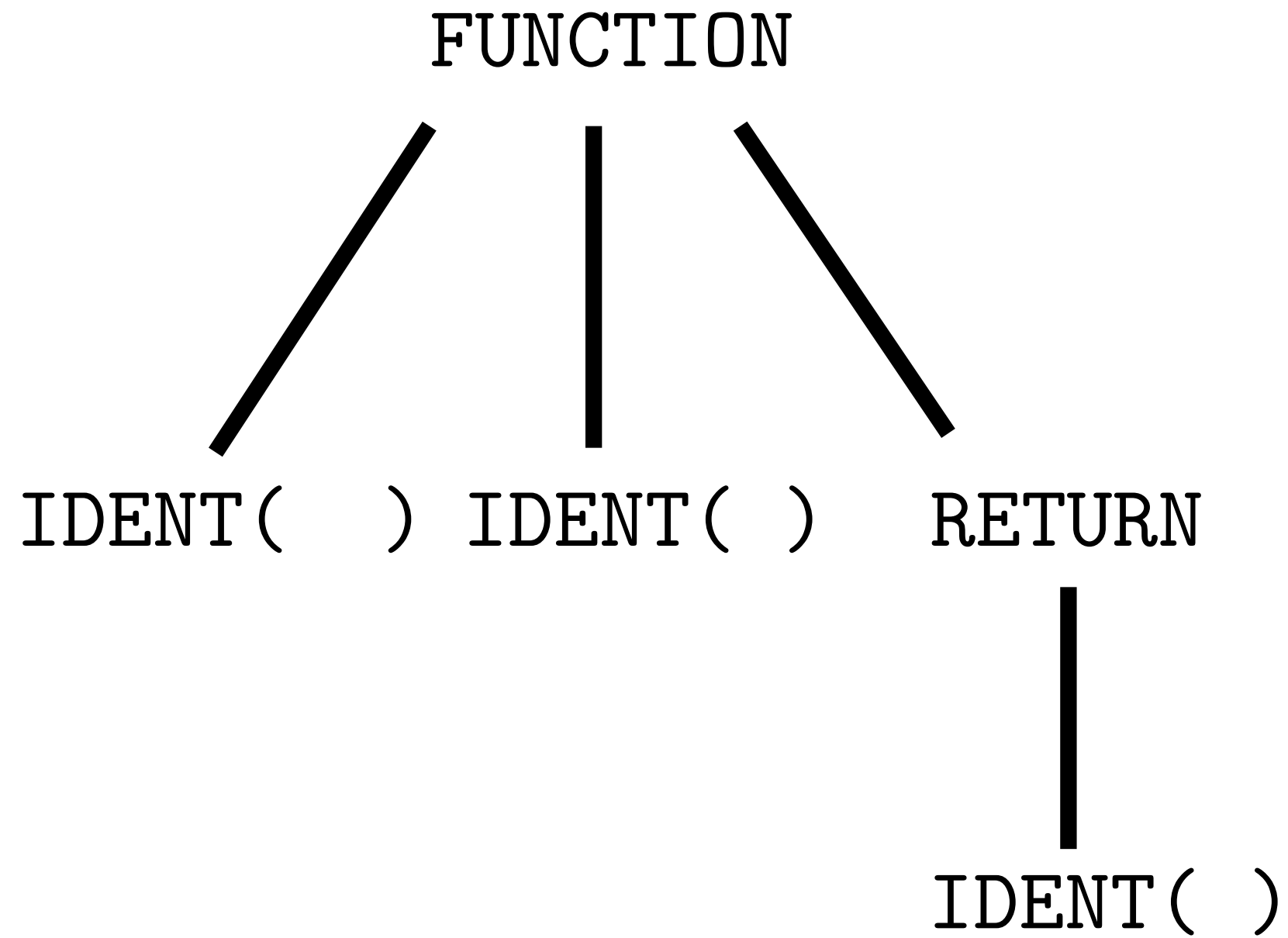
SEMI

RBRACE



FUNCTION

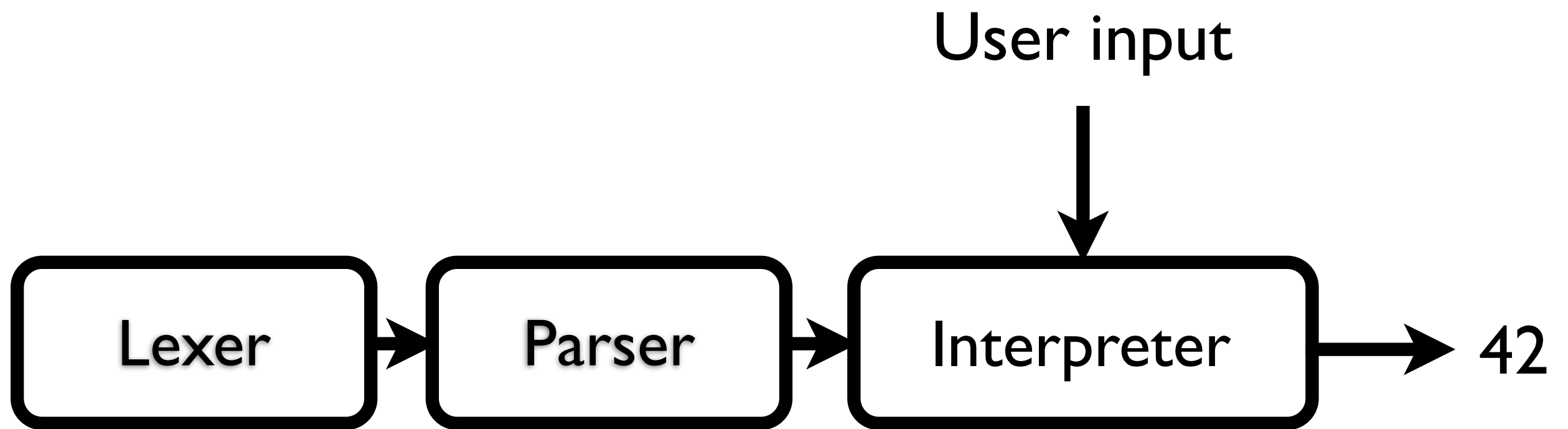






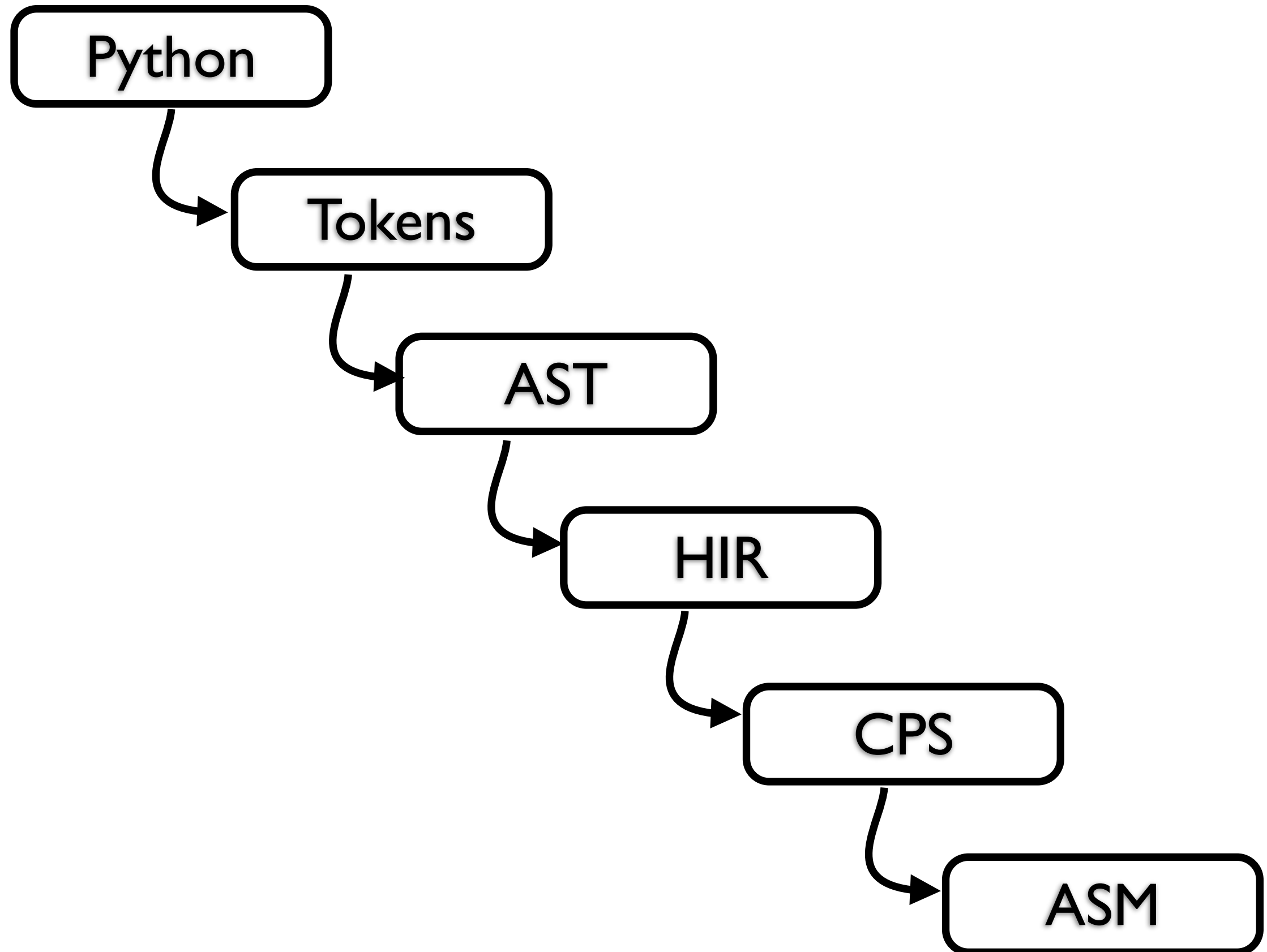


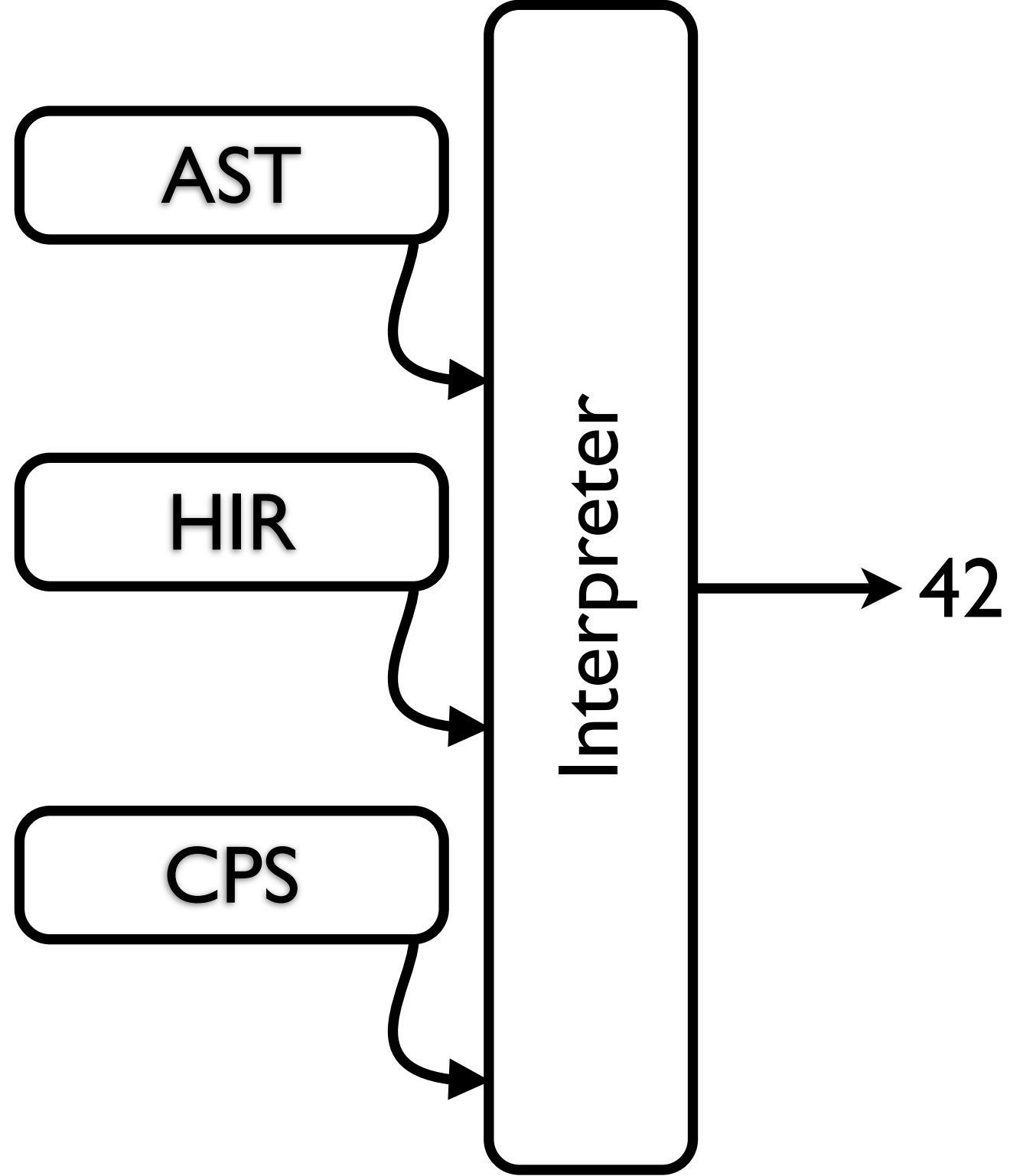




**Why study interpreters?**







**It's a model.**



**It's easier to write.**

**It's easier to port.**

**Compilers > Interpreters?**

**Interpreters > Compilers?**

Neither.

**They're equivalent.**

# How to write interpreters

# Syntactic reduction





program

program

program  $\Rightarrow$  program

am  $\Rightarrow$  program  $\Rightarrow$  program

am  $\Rightarrow$  program  $\Rightarrow$  program

$$(1 + 2) * (4 + 5)$$

$$(3) * (4 + 5)$$



$$3 * (4 + 5)$$

$$3 * (9)$$

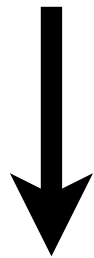
$$3 * 9$$

27

Code

(if *cond* then *else*)

Reduce first



(if *cond* then *else*)

*then*



*else*

(let *var exp body*)

*{exp/var}body*

$$\{e/v\}n$$

*n*

$$\{e/v\}v$$

*e*

$$\{e/v\}v'$$



$v'$

$\{e/v\} (+ \ a \ b)$

$(+ \{e/v\} a \{e/v\} b)$

$\{e/v\}$  (if  $a \ c \ o$ )

(if  $\{e/v\} a$   
     $\{e/v\} c$   
   $\{e/v\} o$ )

$\{e/v\}(\text{let } v \text{ } a \text{ } b)$

$(\text{let } v \{e/v\} a \ b)$

$\{e/v\}(\text{let } v' \text{ } a \text{ } b)$



$(\text{let } v' \{e/v\} a \{e/v\} b)$

Code

$(\lambda \ v \ . \ b)$

$((\lambda \ v \ . \ b) \ a)$

$\{a/v\}b$



(set! *var val*)

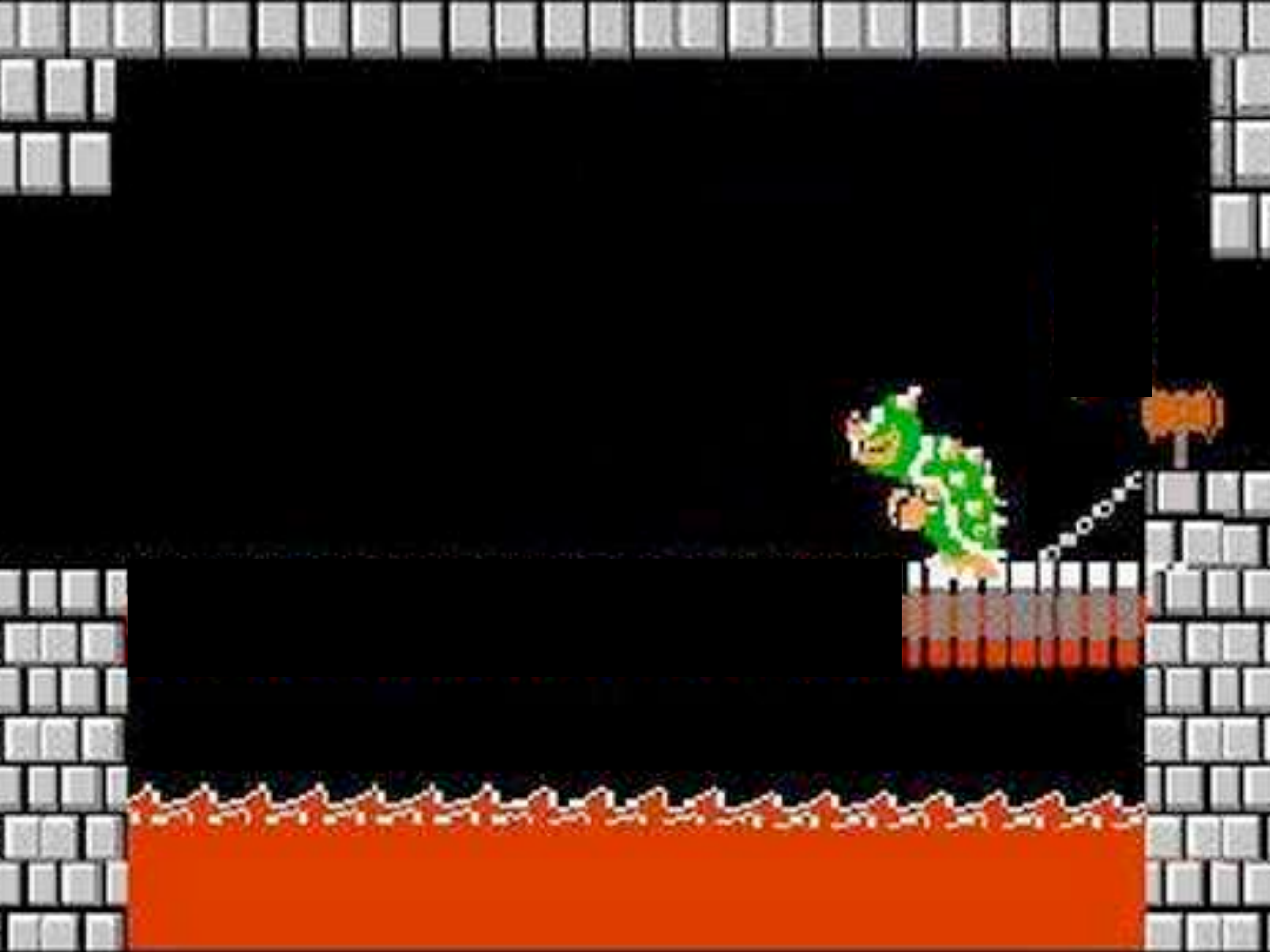


(set! *var val*)









**Environment-based**

**What's an environment?**

**variable => value**

**hash-table, balanced tree**

eval & apply

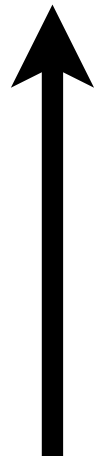
$$\text{eval} : \text{Exp} \times \textit{Env} \rightarrow \textit{Value}$$

eval & apply



`apply` :  $Value \times Value \rightarrow Value$

$\text{apply} : \textit{Value} \times \textit{Value} \rightarrow \textit{Value}$



Expects procedure

$$Value = Clo + \mathbb{Z}$$

$$Clo = \text{Lambda} \times Env$$

$$\mathit{Env} = \mathbf{Var} \rightarrow \mathit{Value}$$

Code

(set! *var exp body*)

(call/cc *exp*)



MARIO  
058900

● X 37

WORLD  
1-4

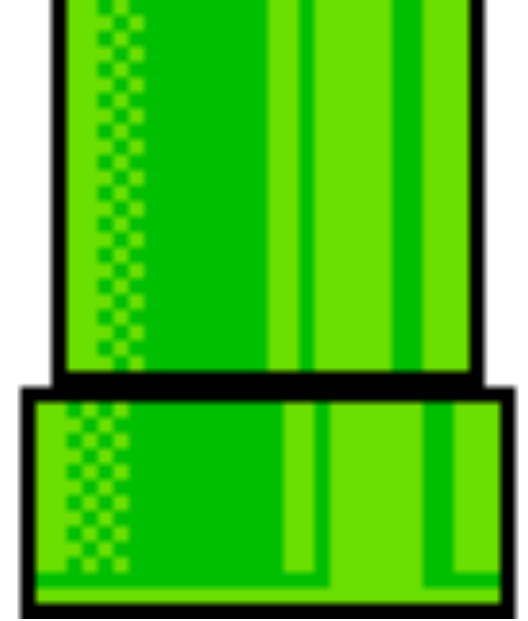
TIME  
220

THANK YOU MARIO!

BUT OUR PRINCESS IS IN  
ANOTHER CASTLE!







# Questions?

# PHP

