

X

P 6 q. 5 MR ①

② Comp P. Ls. 2000

Smalltalk

p 13 q 6

body  $\rightarrow$  | name ... name | E, E ... E

E $\rightarrow$	name	2	2
	# E	# green	
	number	132	
	string	my 'hello'	
	( E )		
	^ E	return E	
	[ E, E, ... E ]	a block	
	[ name name   E, E, ... E ]	anonymous func.	
	E name	method calls	
	E name E name E ...		
	E name E ; name E ...		

Type bool has a method if True: [ ] if False: [ ]

if sent to True causes B<sub>1</sub> to be executed  
 False " B<sub>2</sub> to be executed

WHILE

[ - ] while: [ - ]

↑ evaluate to a bool

if true executes indefinitely

add the method fact to Int.

(2)

fact  
body  $\left[ \begin{array}{l} \text{self} = 0 \text{ if True } [1, 1] \\ (self \neq 1) \text{ fact} \end{array} \right]$

5 fact  
evaluates to 120

The inefficiency is due to runtime lookup of all variables and methods and type checking.

using just-in-time compilation lookup can be done once. Type checking can be done. BUT we must notice when the environment changes to make the code invalid. Such changes are rare. Speed of 450 has been reported.