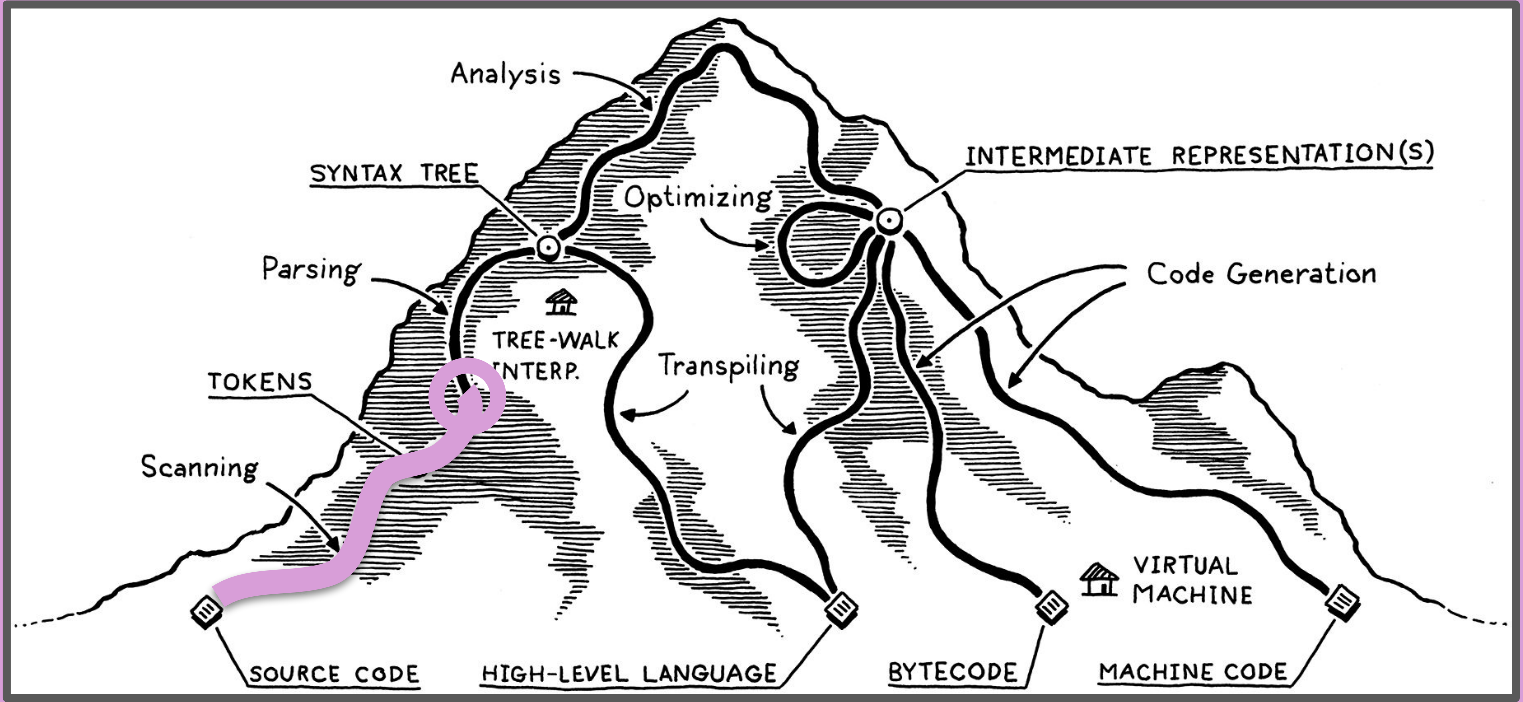
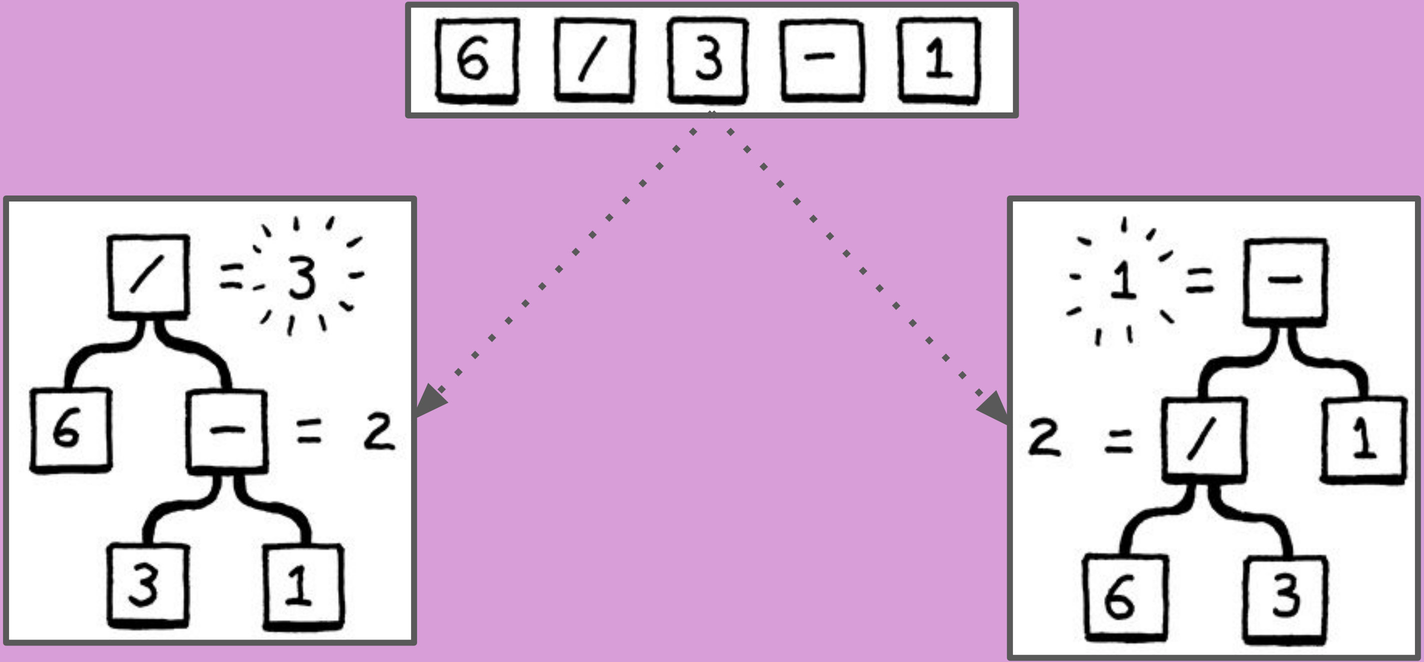
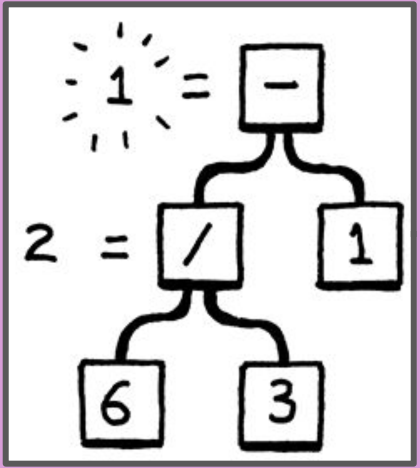
CMPSC 201: PROGRAMMING LANGUAGES


Analysis
SYNTAX TREE
INTERMEDIATE REPRESENTATION(S)
Optimizing
Parsing
Code Generation
TREE-WALK INTERP.
Transpiling
TOKENS
Scanning
VIRTUAL MACHINE
SOURCE CODE
HIGH-LEVEL LANGUAGE
BYTECODE
MACHINE CODE

Precedence



Precedence

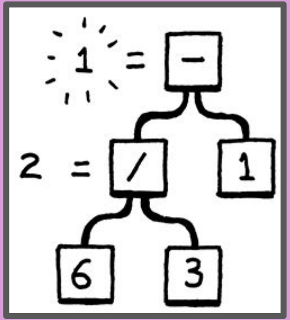


(6 / 3) - 1

Agreed upon order of operations; higher-order operators happen first.

Precedence vs. Associativity

Precedence



Agreed upon order of operations; higher-order operators happen first.

Associativity

5 - 3 - 1

(5 - 3) - 1

How operators of the *same precedence* are grouped in the absence of clear grouping

Precedence vs. Associativity

Precedence

```
8 / 2 / ( 2 / 2 )
```

Associativity

```
8 / 2 / ( 2 / 2 )
```

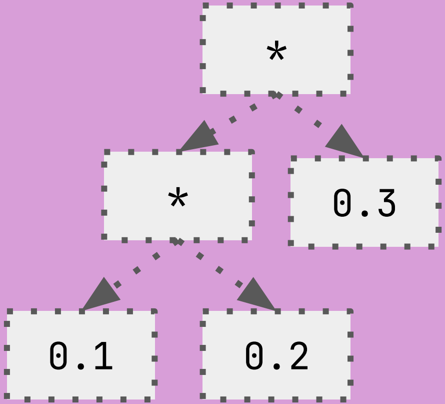
```
8 / 2 / 1
```

Associativity?

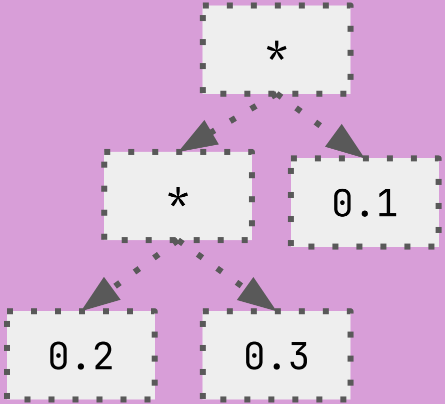
Lox is left-associative. But does it matter?

Consider:

```
print 0.1 * 0.2 * 0.3;
```

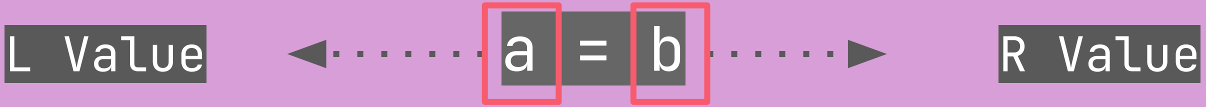


`(0.1 * 0.2) * 0.3`



`0.1 * (0.2 * 0.3)`

Associativity vs. Assignment



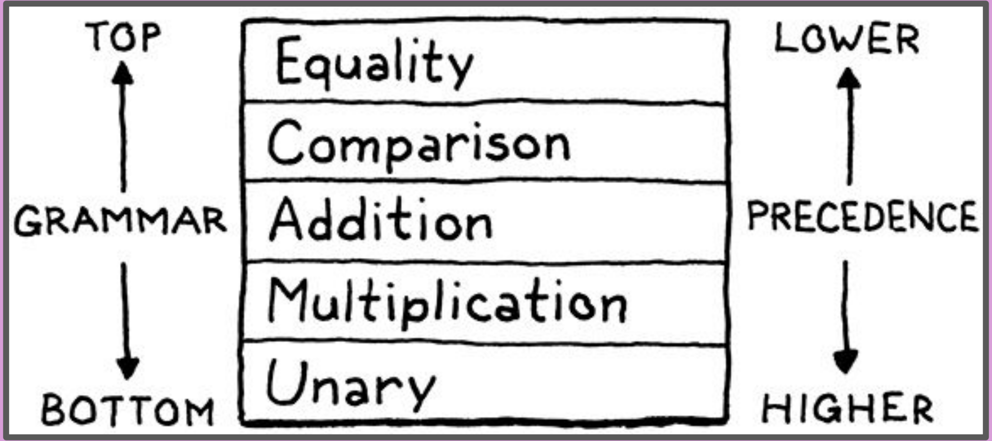
Associativity vs. Assignment

Expression	Steps		Valid?
a = b = c	b = c	a = b	✓
(a = b) = c	?	?	✗
a < b < c	b < c	a < b	✓
	(a < b) and (b < c)		

Precedence, Associativity, Assignment

Precedence	Associativity	Assignment
Agreed upon order of operations; higher-order operators happen first	How operators of the same precedence are grouped in the absence of clear grouping	Order in which expressions are evaluated such that a production rule can only ever produce one unambiguous outcome
Implemented at the level of the grammar	Implemented by the order of the grammar productions	Implemented language-wide

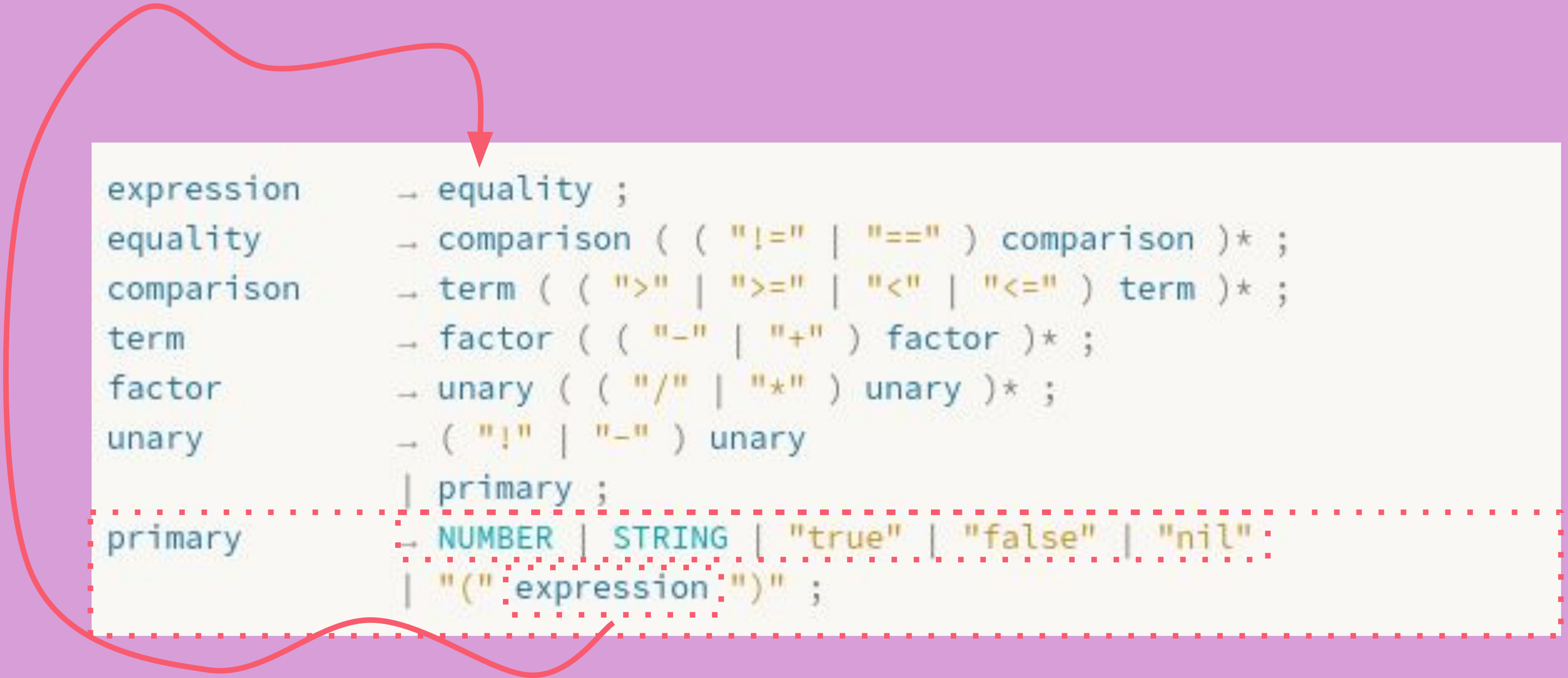
Effect of precedence on Lox



Lox and precdence

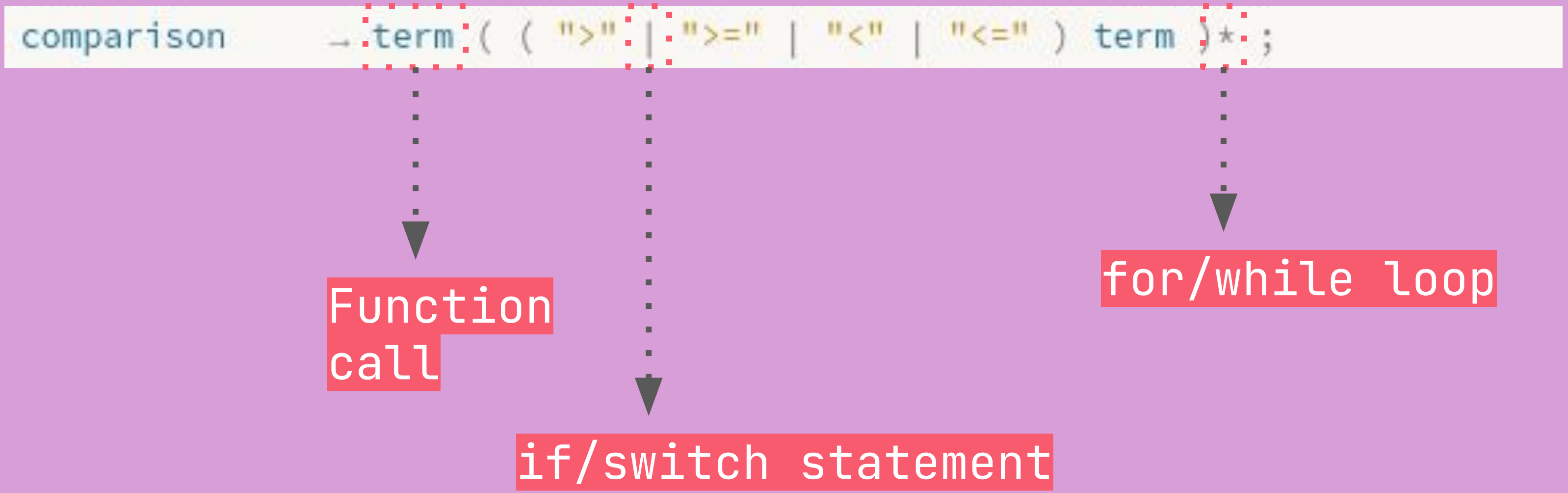
Functions	Function bodies
expression	→ equality ;
equality	→ comparison (("!=" | "==") comparison)* ;
comparison	→ term ((">" | ">=" | "<" | "<=") term)* ;
term	→ factor (("-" | "+") factor)* ;
factor	→ unary (("/" | "*") unary)* ;
unary	→ ("!" | "-") unary
	| primary ;
primary	→ NUMBER | STRING | "true" | "false" | "nil"
	| "(" expression ")" ;

Lox and precdence



```
expression     → equality ;
equality       → comparison ( ( "!=" | "==" ) comparison )* ;
comparison     → term ( ( ">" | ">=" | "<" | "<=" ) term )* ;
term           → factor ( ( "-" | "+" ) factor )* ;
factor         → unary ( ( "/" | "*" ) unary )* ;
unary          → ( "!" | "-" ) unary
               | primary ;
primary        → NUMBER | STRING | "true" | "false" | "nil"
               | "(" expression ")" ;
```

Lox, associatvity and precedence



```
comparison     → term ( ( ">" | ">=" | "<" | "<=" ) term )* ;
```