

# How to implement a Python-like programming language

by Rodrigo Girão Serrão



Rodrigo 🐍🚀  
🐦 @mathsppblog



Rodrigo 🐍🚀

🐦 @mathsppblog

Textualize  
[mathspp.com](http://mathspp.com)

# Setup

# Empty folder

Empty folder  
& Python

Empty folder

& Python

& venv



<https://github.com/mathsp/bpc-ws>



<https://mathspp.com/blog/tag:bpci>

# The four components

# Tokenizer

Tokenizer

Parser

Compiler

Tokenizer

Parser

Compiler

Tokenizer

Parser

Interpreter

SOURCE CODE



Compiler

Tokenizer

Parser

Interpreter

SOURCE CODE

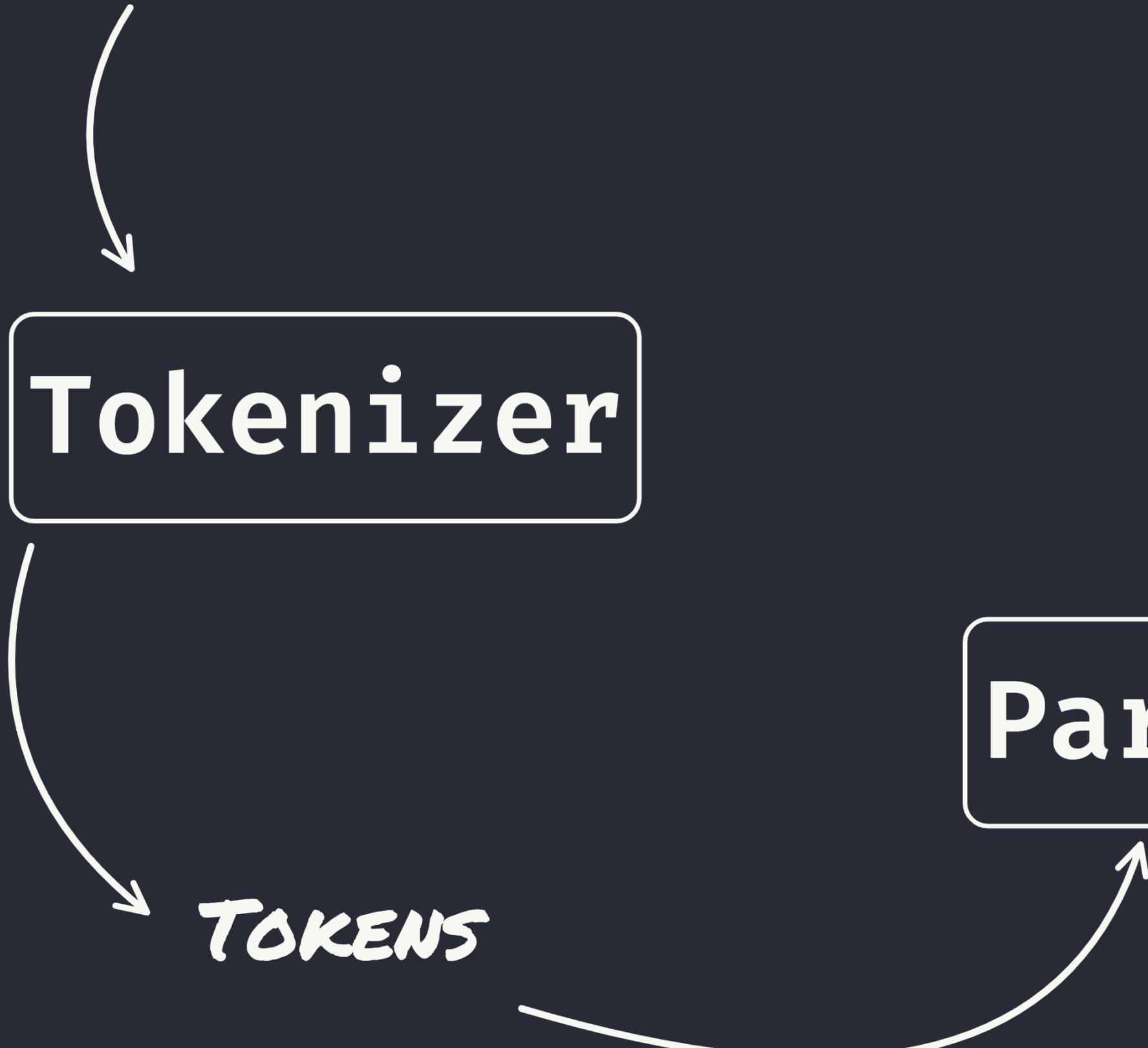
Compiler

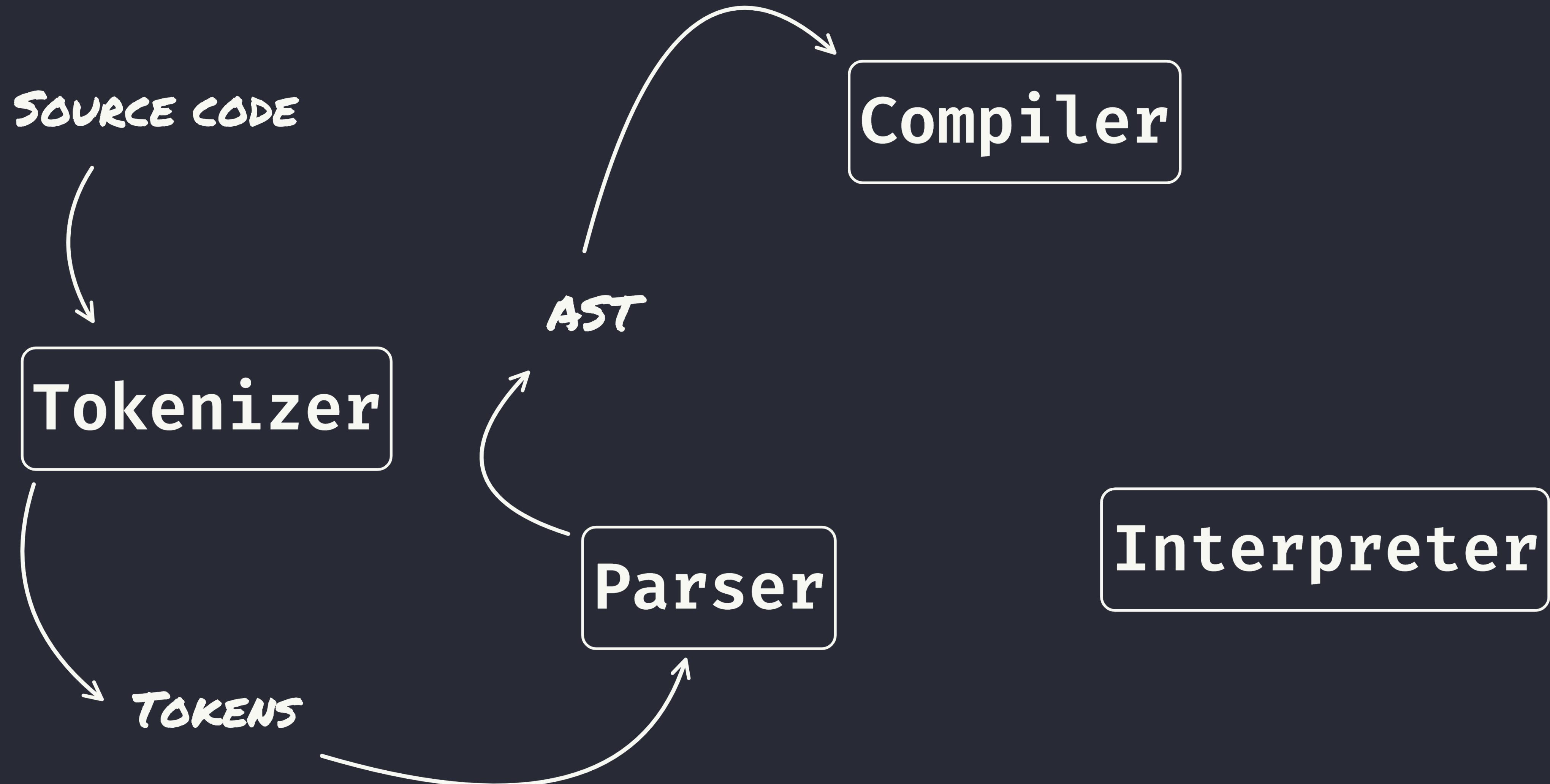
Tokenizer

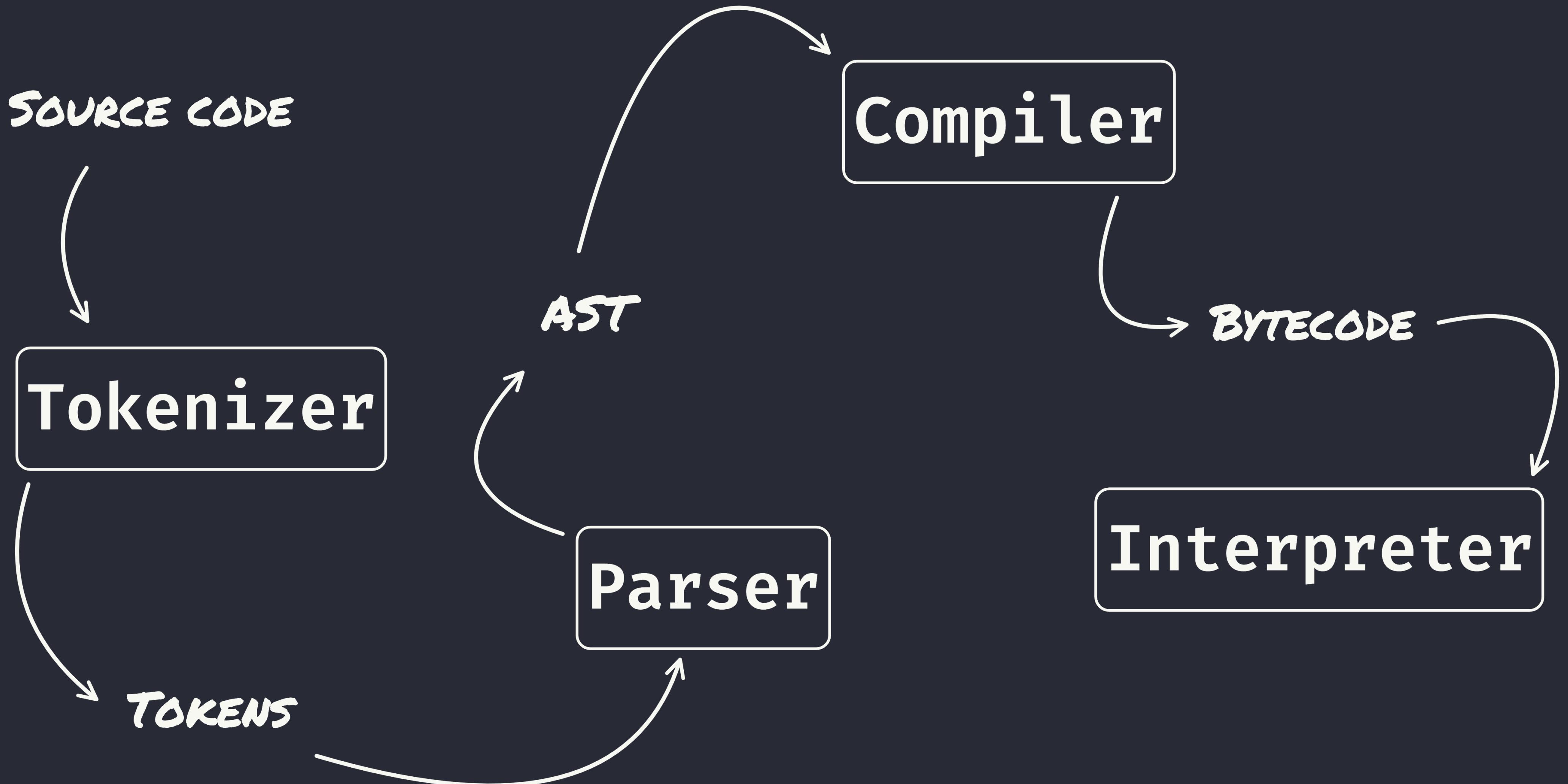
Interpreter

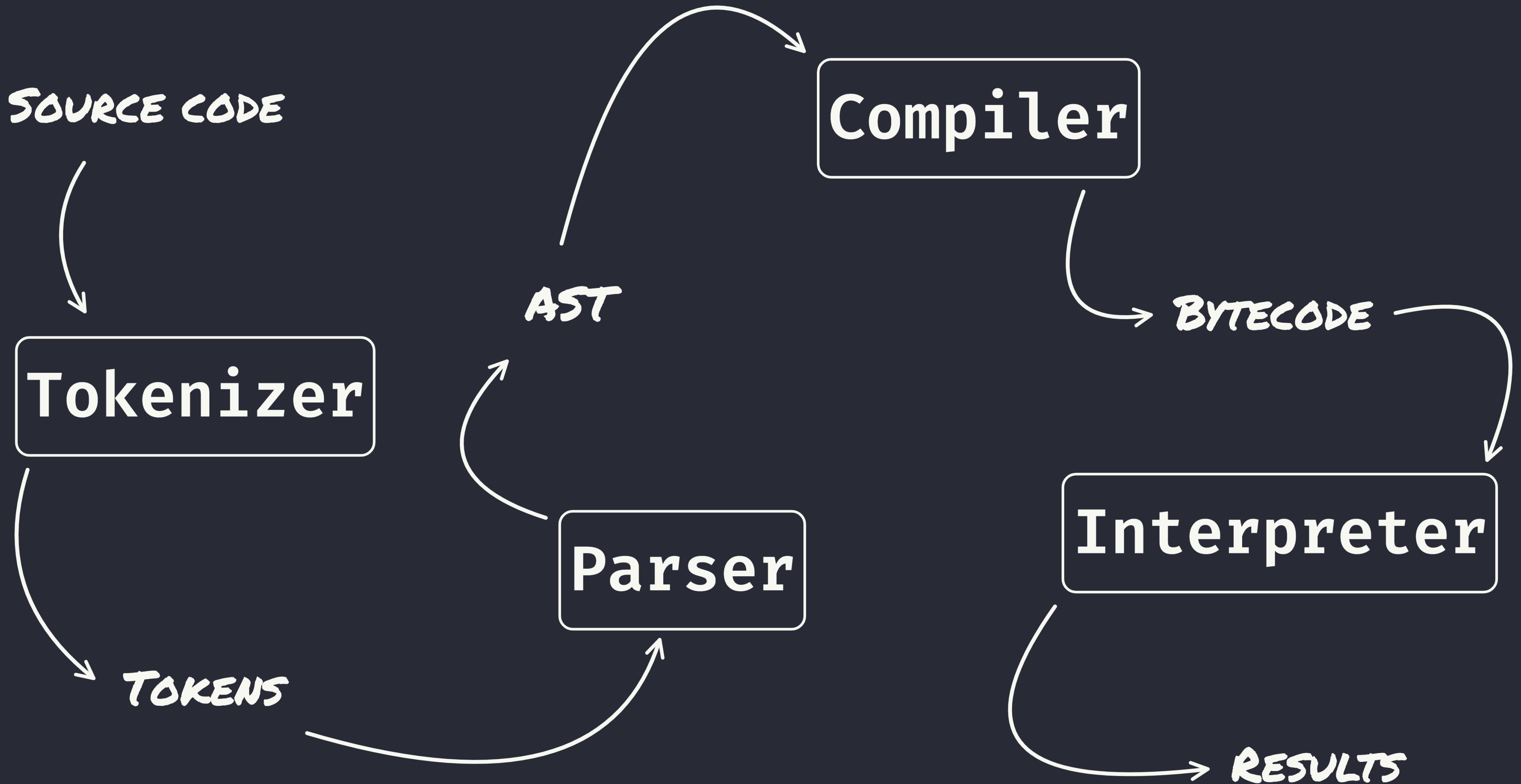
Parser

TOKENS









# Example pass

$$3 + 5$$

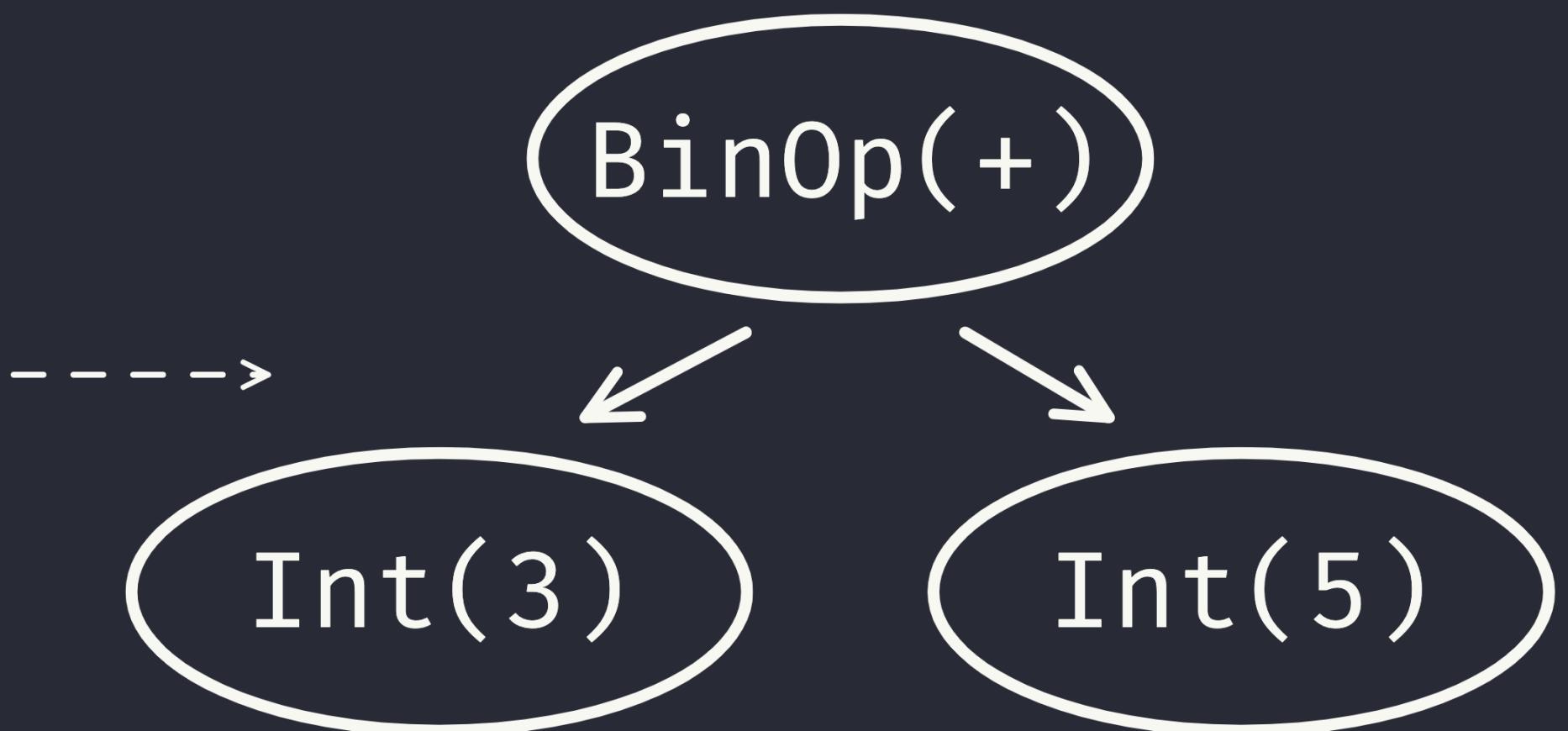


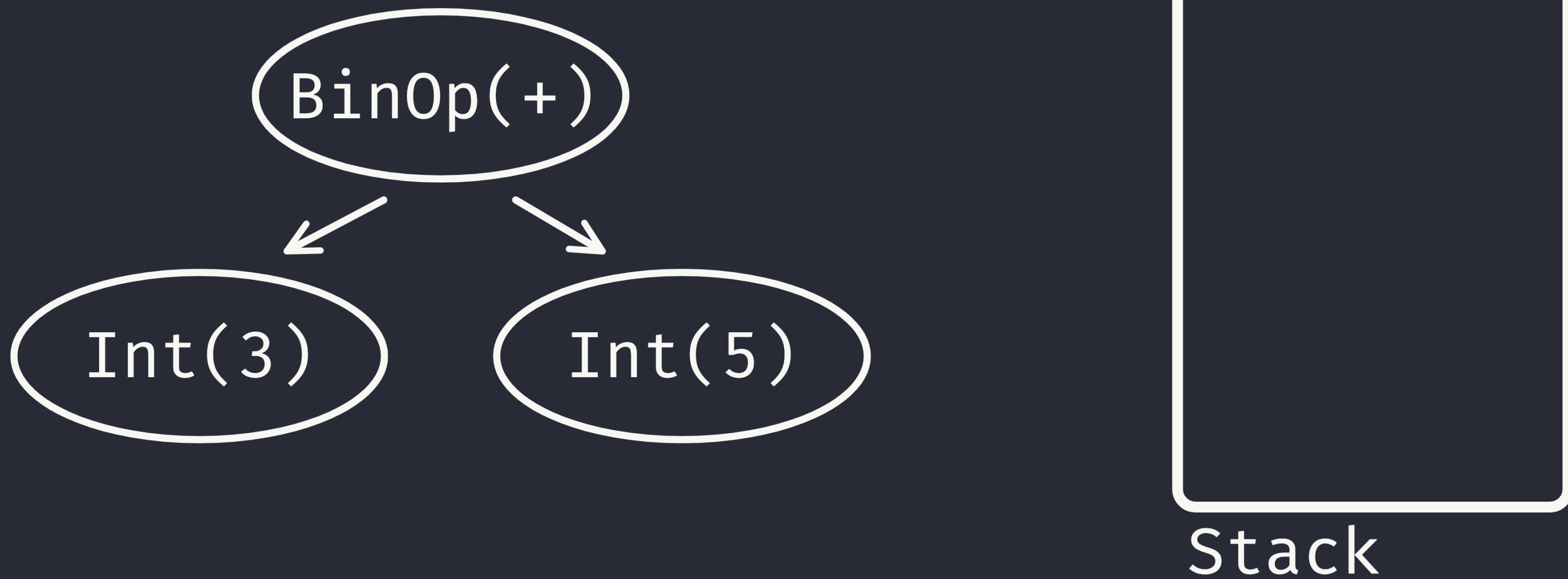
Token(INT, 3)

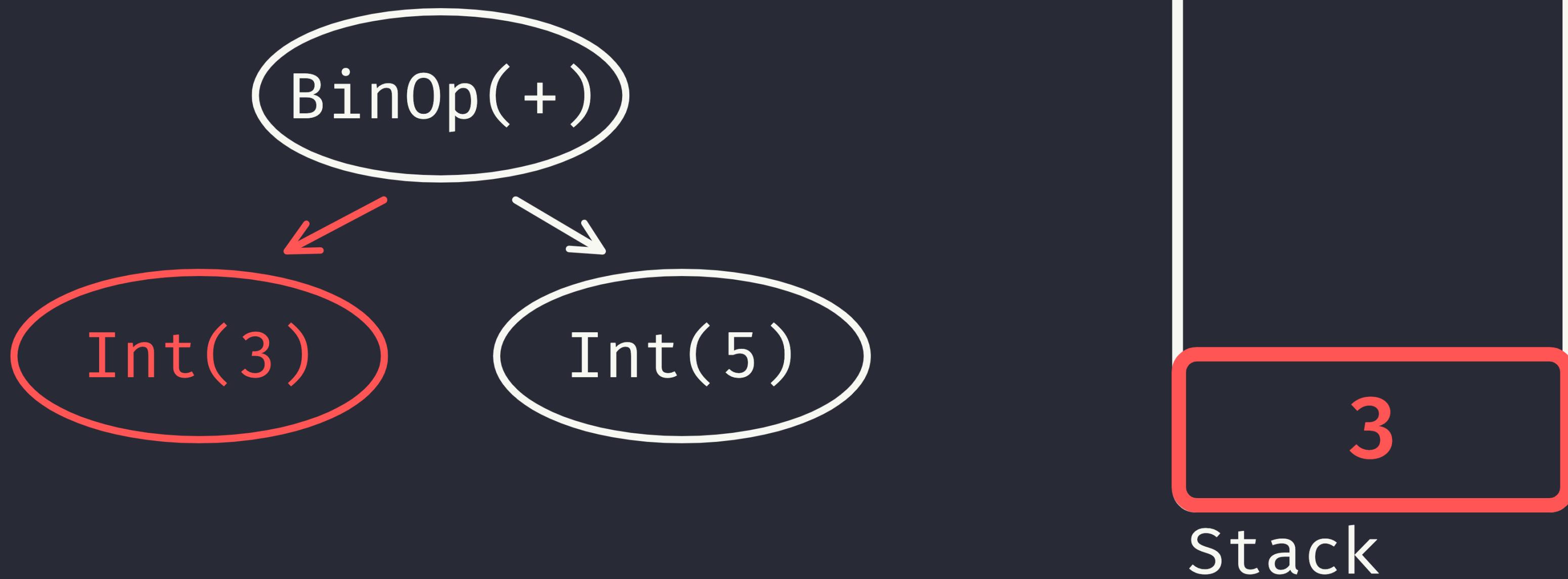
Token(PLUS) ----->

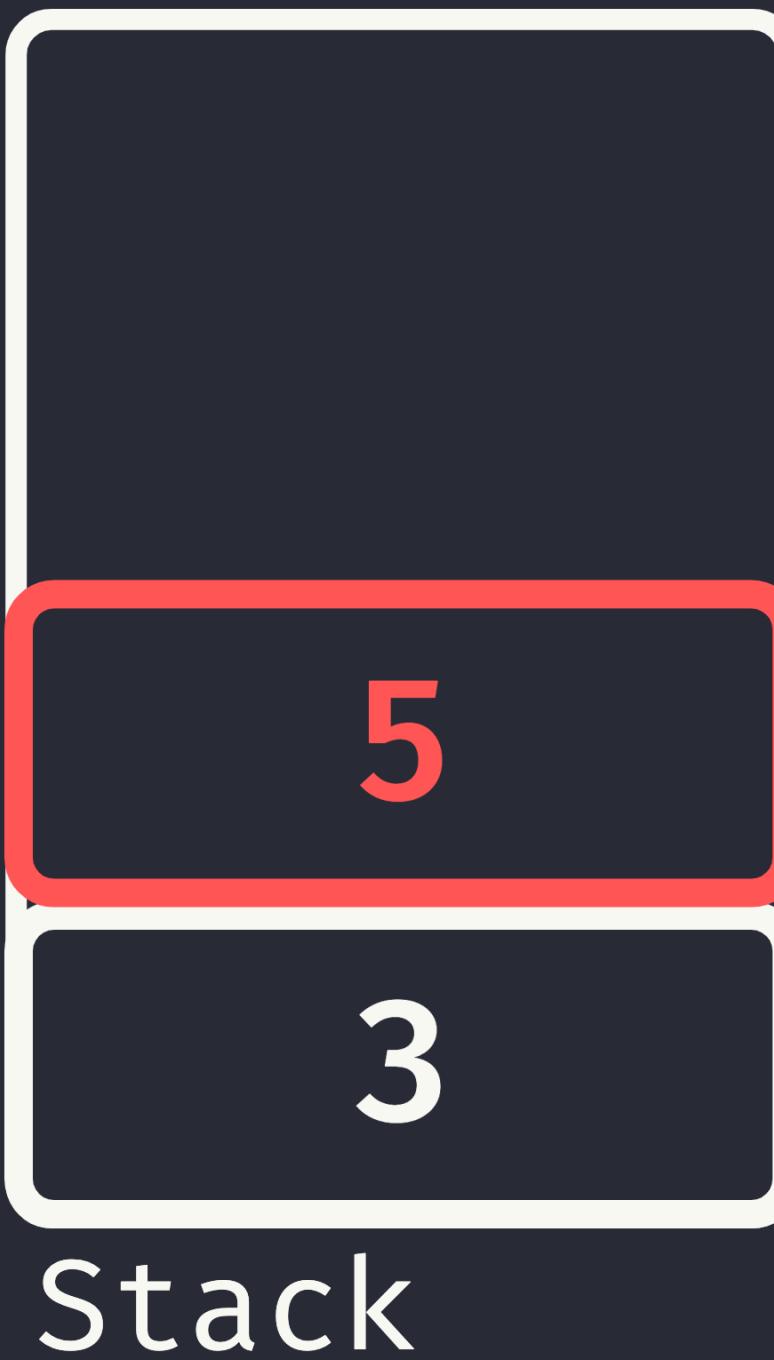
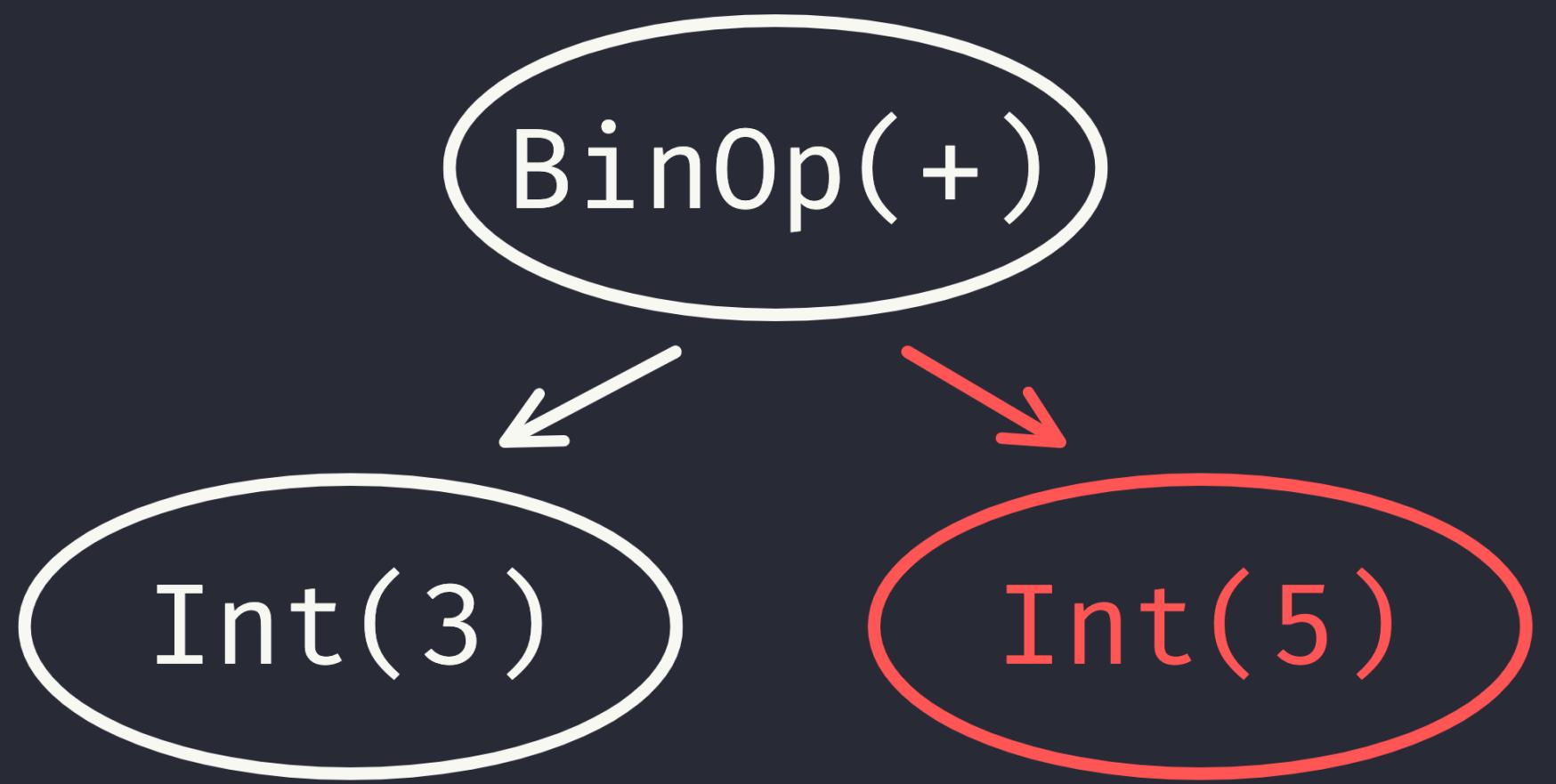
Parser

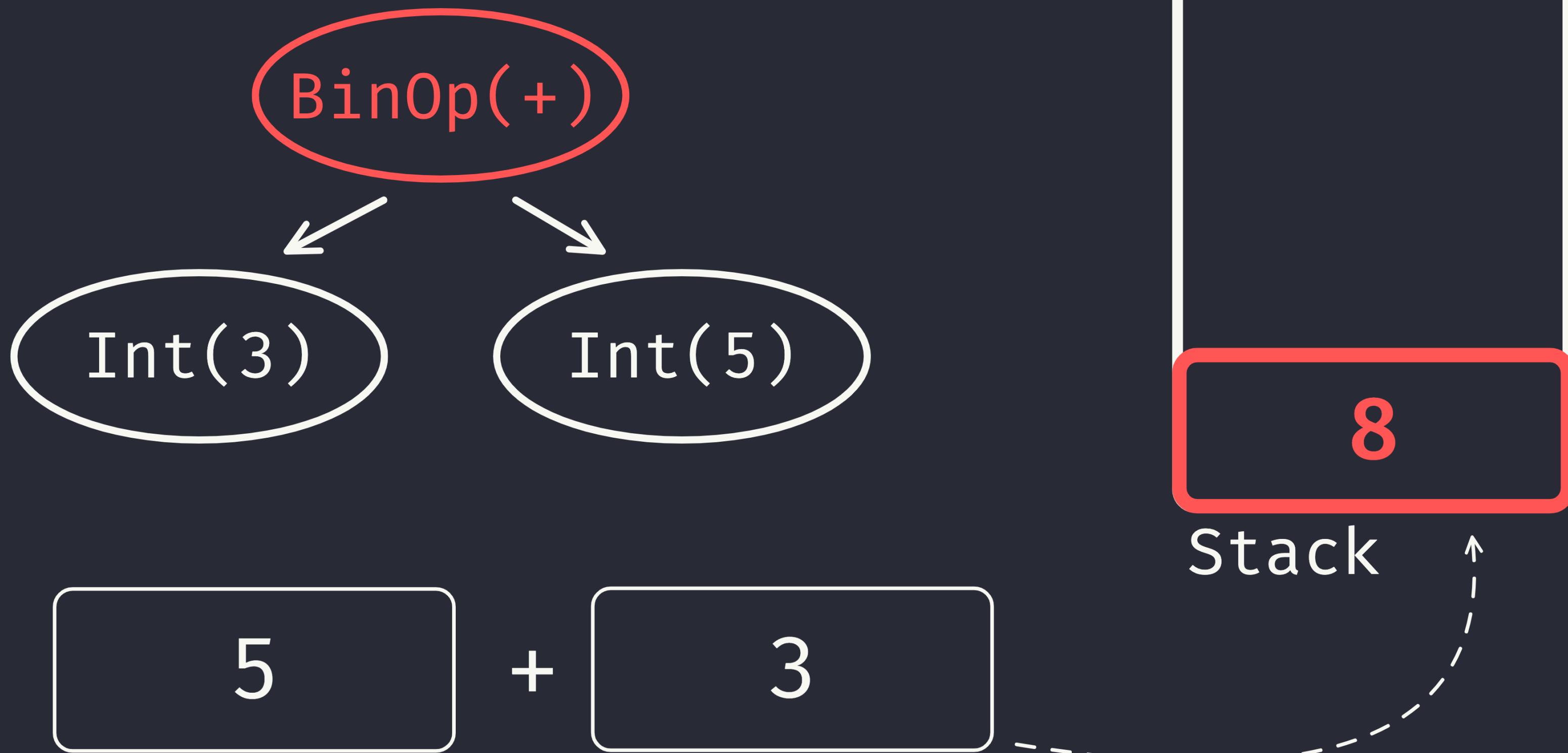
Token(INT, 5)













BC\_PUSH, 3

BC\_PUSH, 5

BC\_BINOP, +

----->

**Interpreter**

-----> 8

# Code

- Create tokens for \* /
- Create them inside next\_token

Can you tokenize 2+ digit integers?

- The level of the rule influences precedence
- Rule at the bottom = higher precedence

Can you tokenize 2+ digit integers?

Can you add parenthesised expressions?

- BinOp's children must be compiled first

Can you tokenize 2+ digit integers?

Can you add parenthesised expressions?

Can you add multiple statements?

- Operands must come *out* of the stack
- Careful with the left/right order

Can you tokenize 2+ digit integers?

Can you add parenthesised expressions?

Can you add multiple statements?

I hope you had fun

(& learned something!)

