Crafting Interpreters

(in rust)

What is this?

programming book club
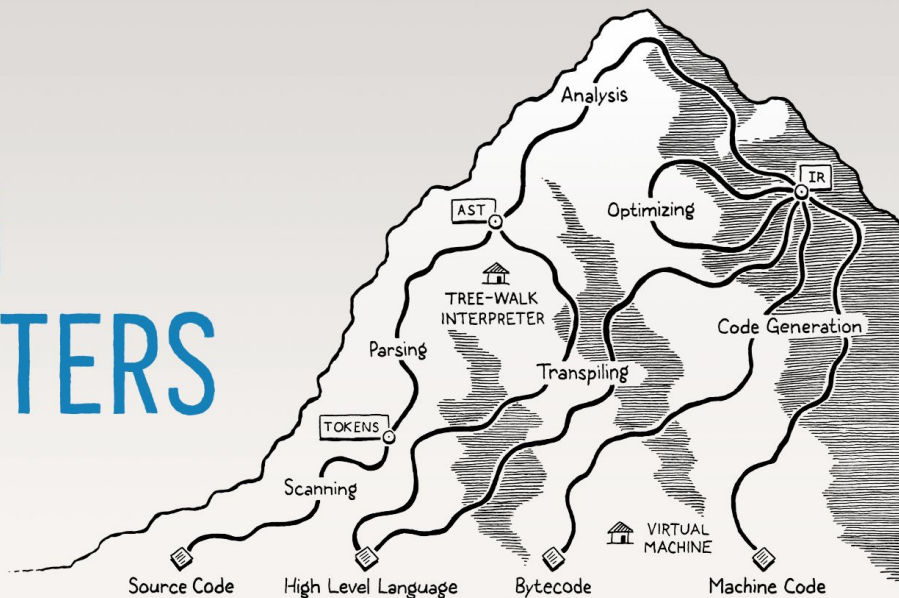
casual Rust programming

have first book selected, but open to suggestions for next books

What's the book?

What will we do?

Language implemented: lox

Like a stripped-down python with { } instead of indentation

```
if (condition) {
  print "yes";
} else {
  print "no";
}
```

```
var a = 1;
while (a < 10) {
  print a;
  a = a + 1;
}
```

```
for (var a = 1; a < 10; a = a + 1) {
  print a;
}
```

```
fun addPair(a, b) {
  return a + b;
}

fun identity(a) {
  return a;
}

print identity(addPair)(1, 2); // Prints "3".
```

```
class Breakfast {
  cook() {
    print "Eggs a-fryin'!";
  }

  serve(who) {
    print "Enjoy your breakfast, " + who + ".";
  }
}
```

What will we do?

Language implemented: lox

Like a stripped-down python with { } instead of indentation

```
if (condition) {
  print "yes";
} else {
  print "no";
}
```

```
var a = 1;
while (a < 10) {
  print a;
  a = a + 1;
}
```

```
for (var a = 1; a < 10; a = a + 1) {
  print a;
}
```

First stage: tree-walk interpreter in Java (make it work as straightforwardly as possible)

Second stage: bytecode interpreter in C (make it performant, similar design to CPython)

We will port both to Rust as we read ([someone else has done this too](#))

What will we do?

Code will go here: https://github.com/DireLines/lox-rs

We'll try VS Code live sessions

Get Rust if you don't have it: https://rustup.rs/
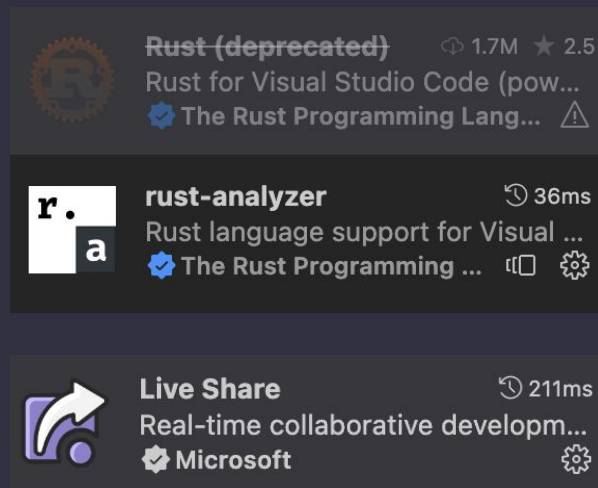
Suggested VS Code extensions:

      rust-analyzer for language support

      Live Share for collaborative sessions

      CodeLLDB for debugging

Who am I?

Nathaniel Saxe (Nate)

UVA CS 2020

2021-2023: Comcast

now: Runpod (GPU rental startup)

hobby Rust programmer, game developer

When will we meet?

Most likely some weeknight 7-8 pm every 2 weeks

but will put out a when2meet to judge availability

Let's get started!

Tentative plan

Now - Chapters 1 & 2

Next time - Chapter 3, lox demo, set up coding environment

Afterward: read a chapter, code a chapter per meeting