#starkcode

Block By Block.

# BLOCK NONCE #7: Rust enums are great!

## What?

Rust enums let you to name and limit the valid states a variable can take in your domain. If your type can only take on a few known values- you should enum it!

## Why?

Enums make your code self-documenting, more readable, and give the compiler a chance to catch your bugs before your teammate does.

## How?

```
struct LoggingConfig {
    enabled: bool,
    level: Option<LogLevel>,
}
impl LoggingConfig {
    fn log(&self, message: &str) {
        if self.enabled {
            let level = self.level.unwrap(); // Can panic here!
            log(level, message);
        }
    }
}
```

This struct enables invalid configurations like LoggingConfig { enabled: true, level: None } or { enabled: false, level: Some}. Let's **Enum** it:

```
enum LoggingConfig {
    Disabled,
    Enabled(LogLevel),
}
impl LoggingConfig {
    fn log(&self, message: &str) {
          if let Self::Enabled(level) = self { // No risk of panicking.
              log(level, message);
        }
    }
}
```

**Call For Action!**📢💨 Use enums rather then strings in the config, and you'l never has to spel corectly agan.

```
error: invalid value 'executble' for '--program_type <PROGRAM_TYPE>'
[possible values: executable, json]
```