BLOCK NONCE #8: Rust enums are great!

What?

Rust enums let you name and limit the valid states a variable can take.

Why?

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

How?

```rust
struct InputSource {
    is_stdin: bool, // Either a file or stdin.
    path: Option<String>,
}
...
let input = if source.is_stdin {
    read_from_stdin();
} else if let Some(path) = &source.path {
    read_from_file(path);
} else {
    panic!("Invalid InputSource configuration.");
};
```

This struct enables invalid configurations such as:

InputSource { is_stdin: false, path: None}. Let's **Enum** it:

```rust
enum InputSource {
    Stdin,
    File(String),
}
...
let input = match source {
    InputSource::Stdin => read_from_stdin(),
    InputSource::File(path) => read_from_file(path),
};
```

> **Call For Action!**📢❄️ Use enums rather than strings in the config, and you'l never hav to spel corectly agan.
>
> ```
> error: invalid value 'executble' for '--program_type <PROGRAM_TYPE>'
> [possible values: executable, json]
> ```

#starkcode