



## BLOCK NONCE #7: Rust enums are great!

### What?

Rust enums let you name and limit the valid states a variable can take.  
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 InputSource {  
    is_stdin: bool, // Either a file or stdin.  
    path: Option<String>,  
}  
  
...  
let input = if input.is_stdin {  
    read_from_stdin();  
} else if let Some(path) = &input.path {  
    read_from_file(path);  
} else {  
    panic!("Invalid InputSource configuration!");  
};
```

This struct enables invalid configurations like: `InputSource{is_stdin:false, path:None}`.  
Let's **Enum** it:

```
enum InputSource {  
    Stdin,  
    File(String),  
}  
  
...  
let input = match input {  
    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'll never have to spell correctly again.

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