ackee blockchain

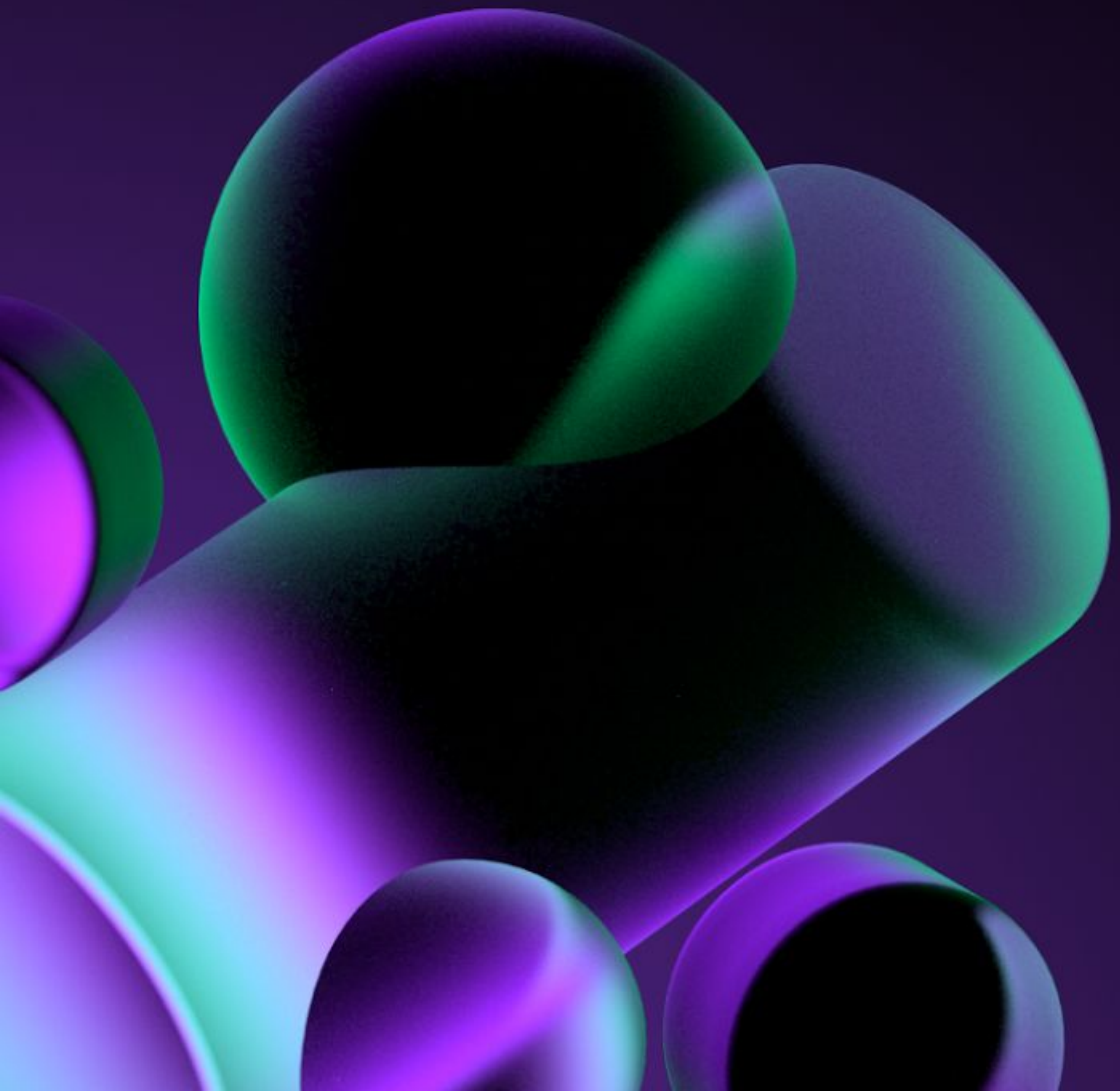
Summer School of Solana

LECTURE 2

Rust for Solana

Tibor Tribus

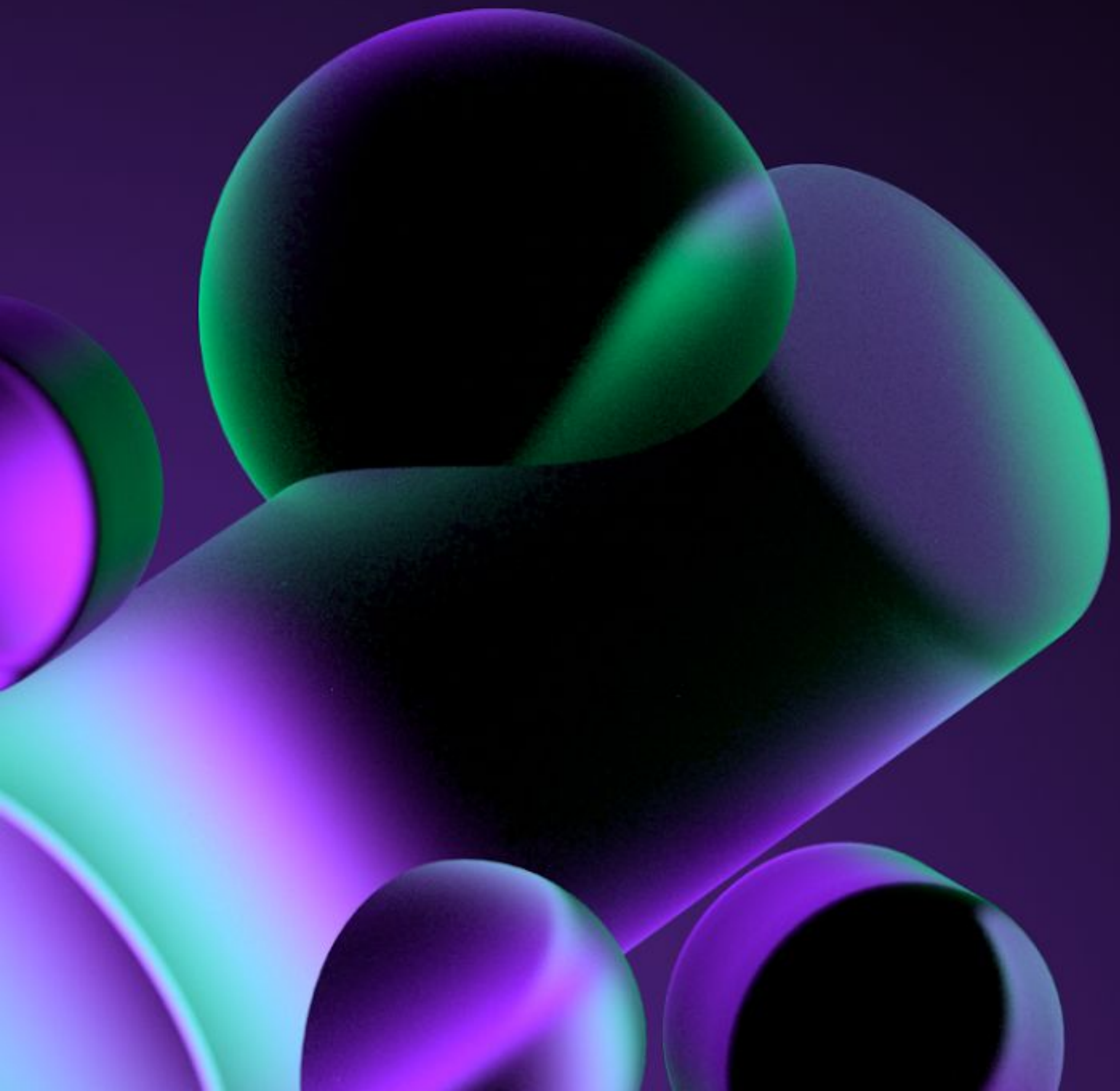
About this lecture

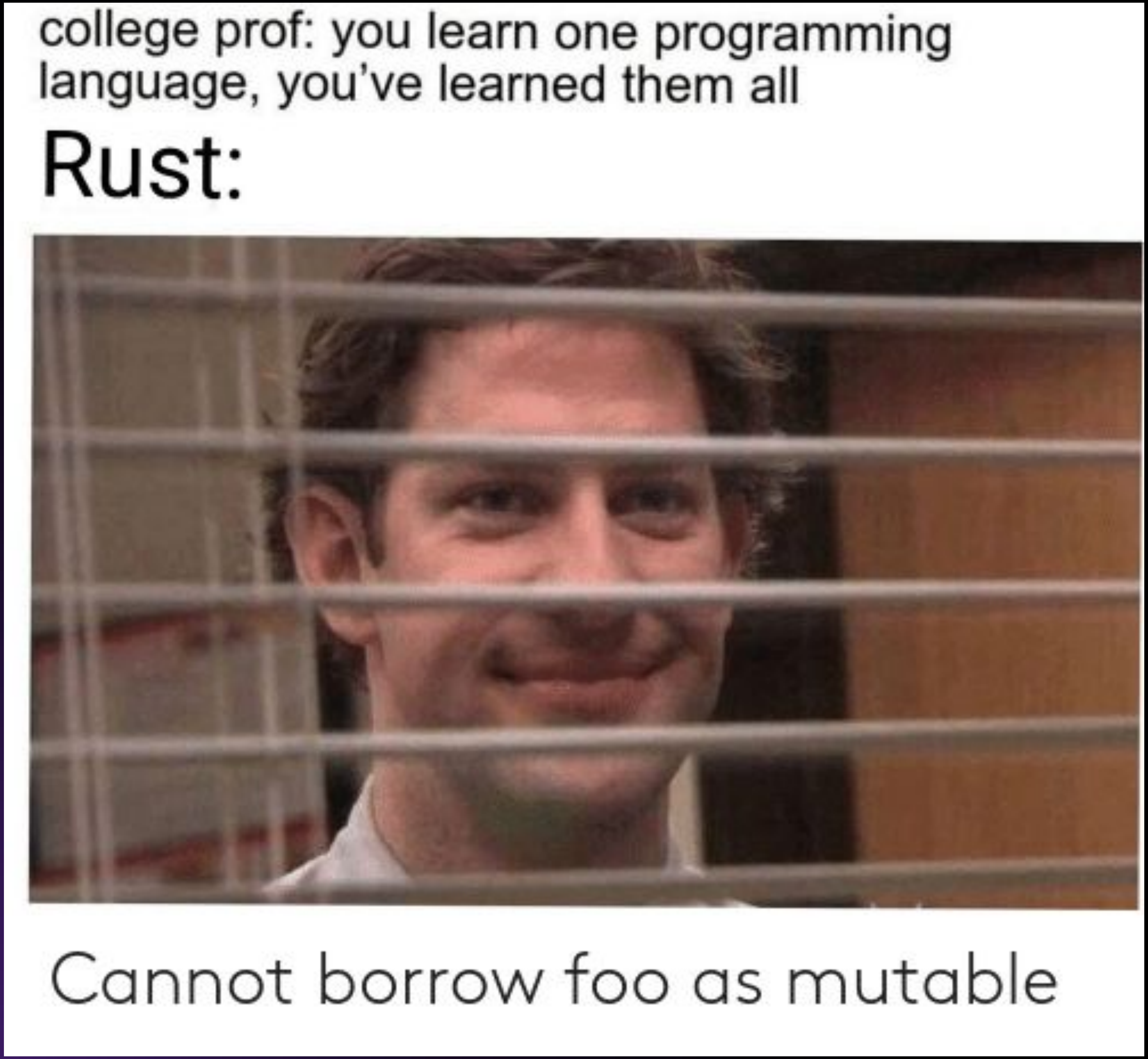


About this lecture

- Lecture organization
- Introduction to Rust
- Hands on example

Rust




college prof: you learn one programming language, you've learned them all
Rust:
Cannot borrow foo as mutable

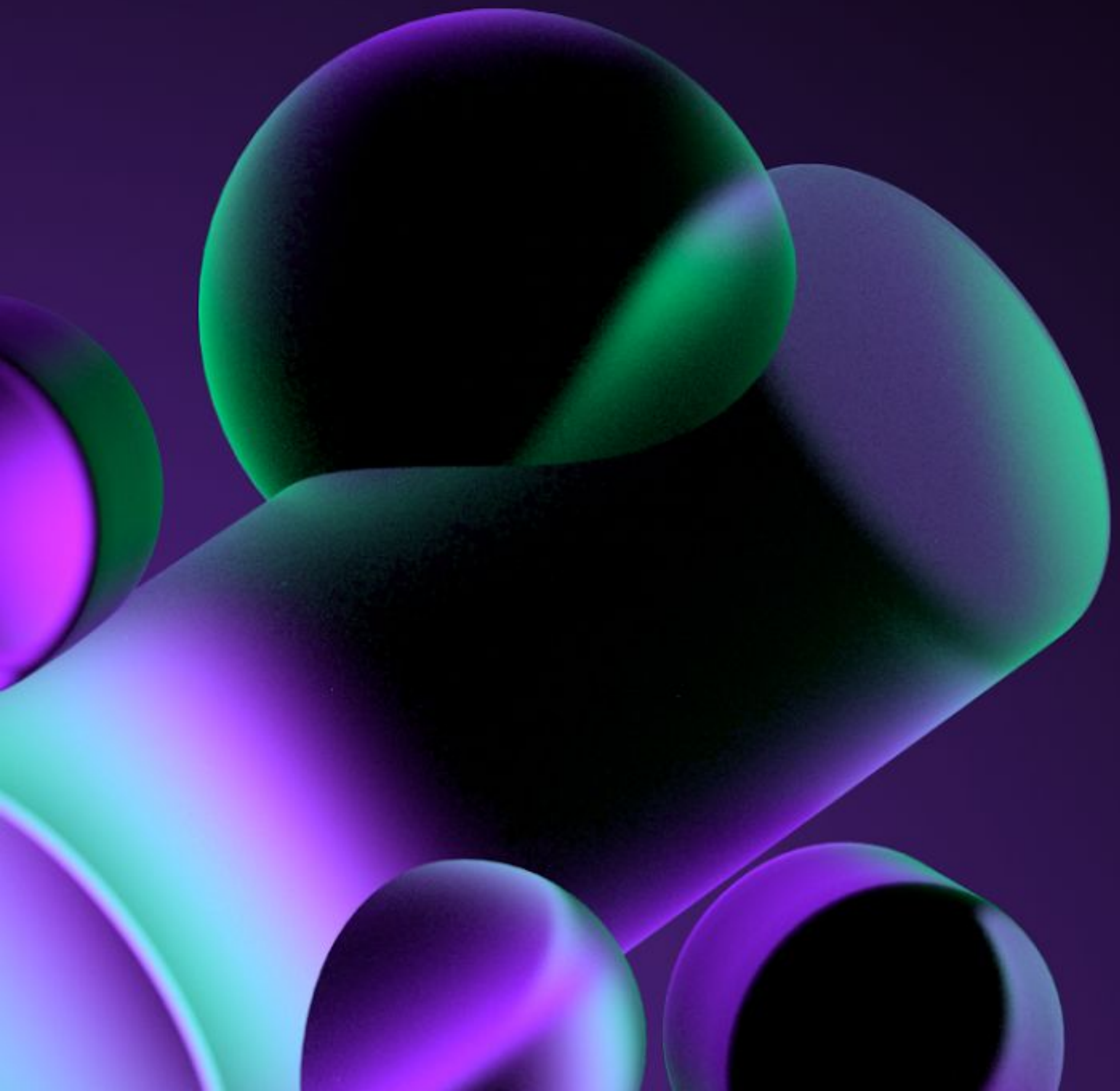
Rust

- Modern systems programming language
  - Safety
  - Speed
  - Concurrency
- Statically and strongly typed
- Graydon Hoare personal project (2006) / @graydon_pub
  - Later acquired by Mozilla

Rust

- Very first release in 2015
- Its own foundation in 2021
- Most loved language since 2016 ❤️
- Companies using Rust
  - Solana
  - CloudFlare
  - 1Password

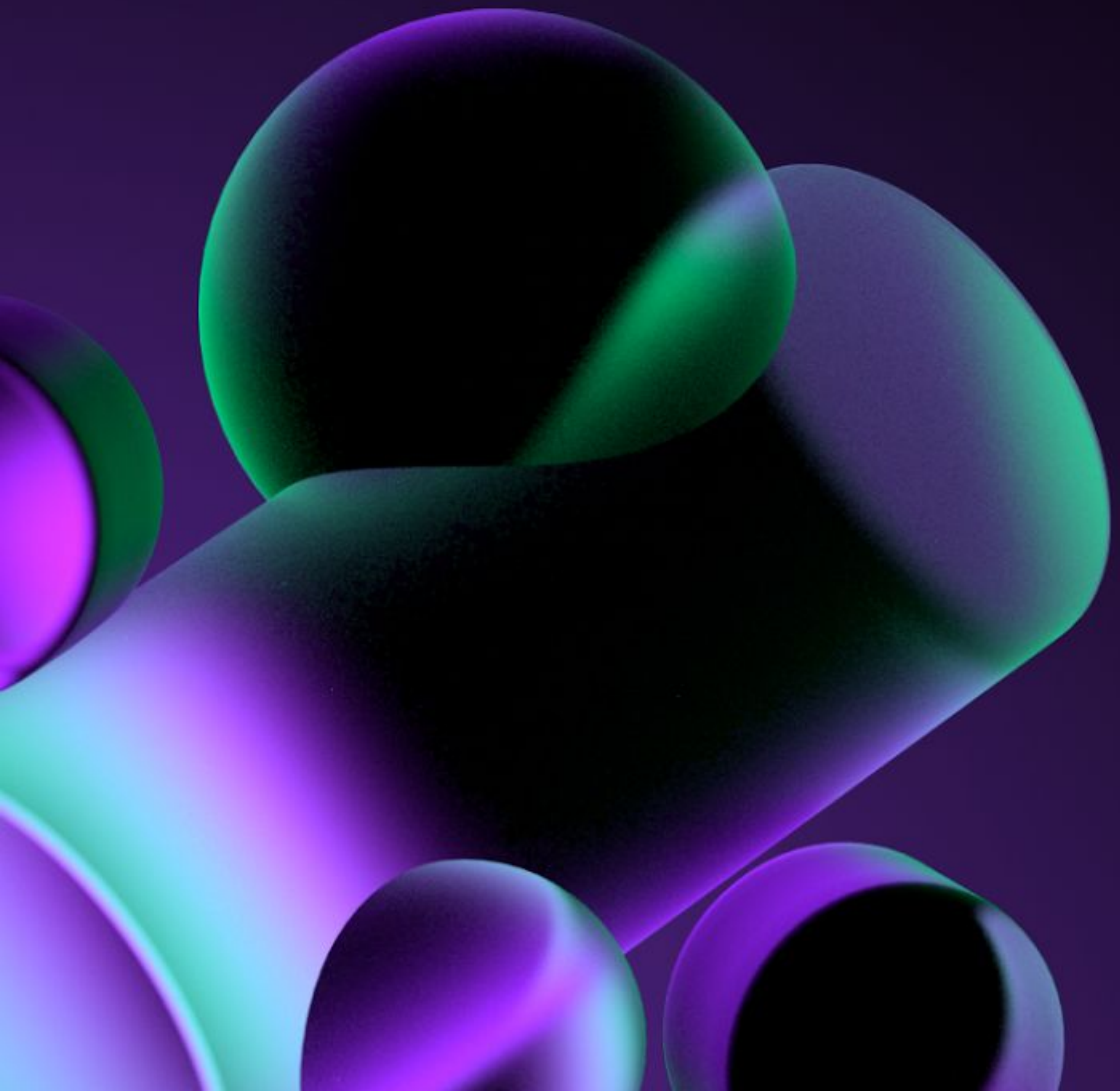
Hello world - again



```
fn main() {
    println!(”Hello, world!”);
}
```

Click here

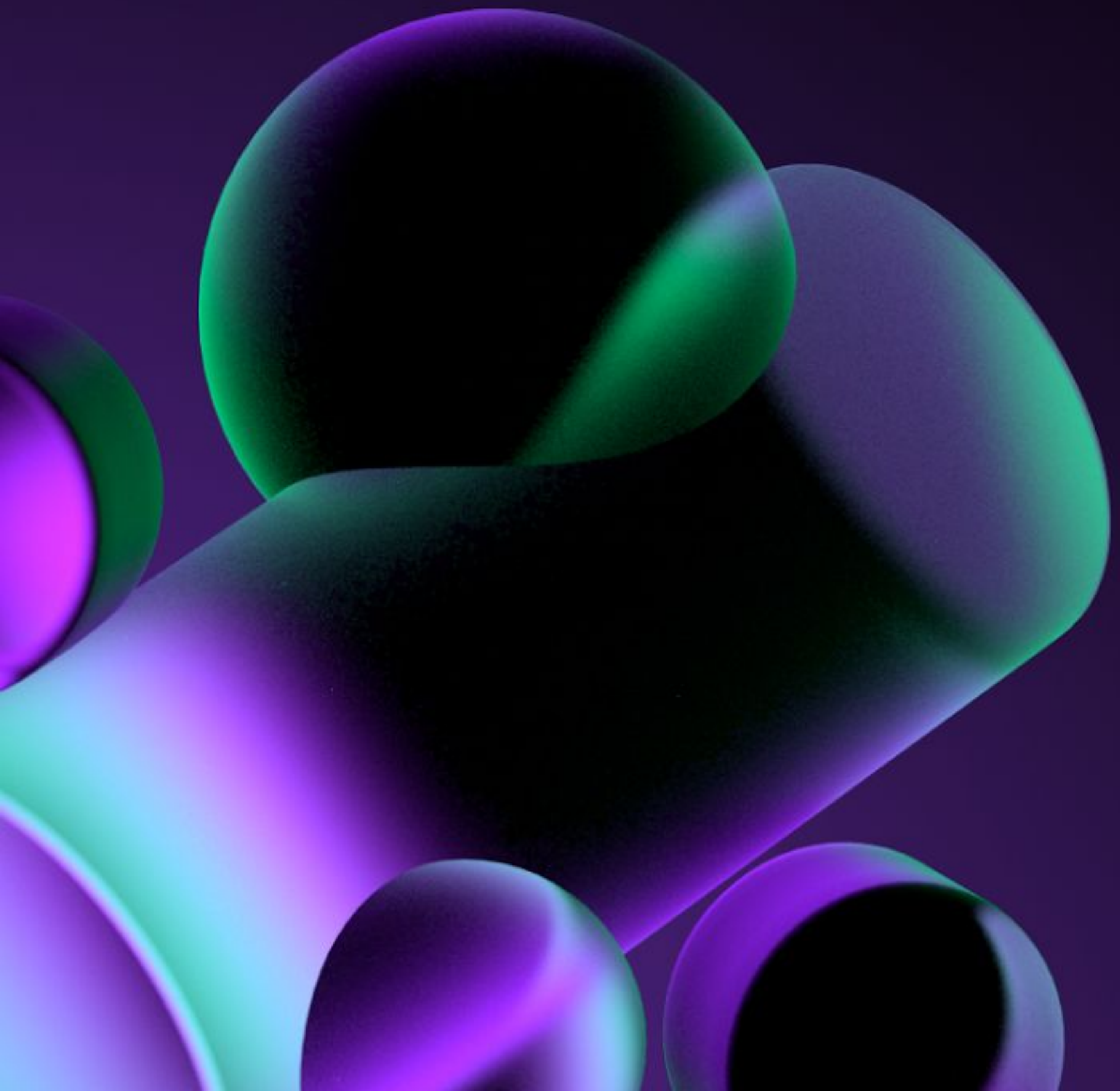
Rust - Variables



Variables

- Letters, digits, and the underscore characters
- Must begin with either a letter or an underscore
- Upper and lowercase letters are distinct
- By default, variables are immutable

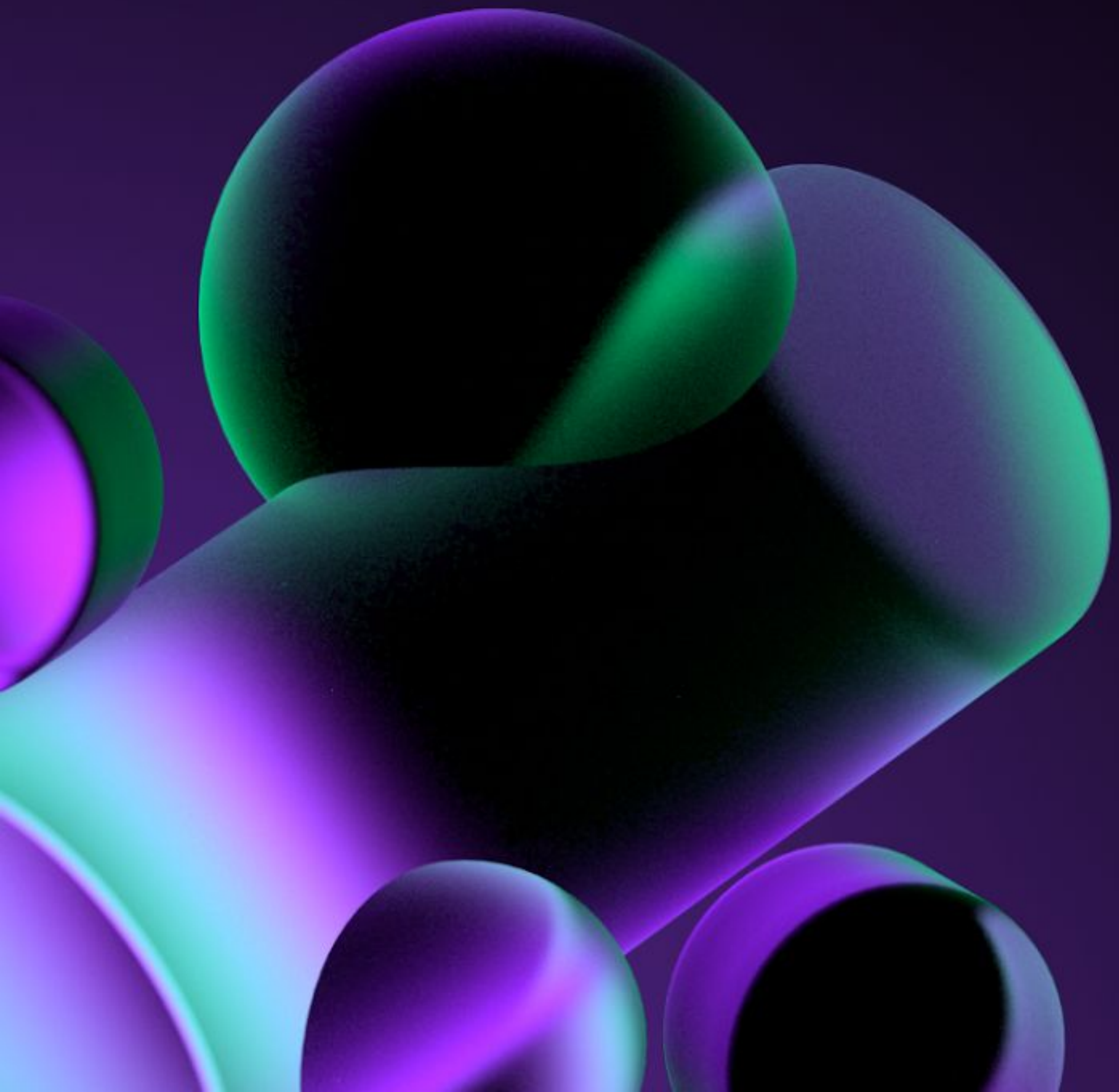
Rust - Ownership

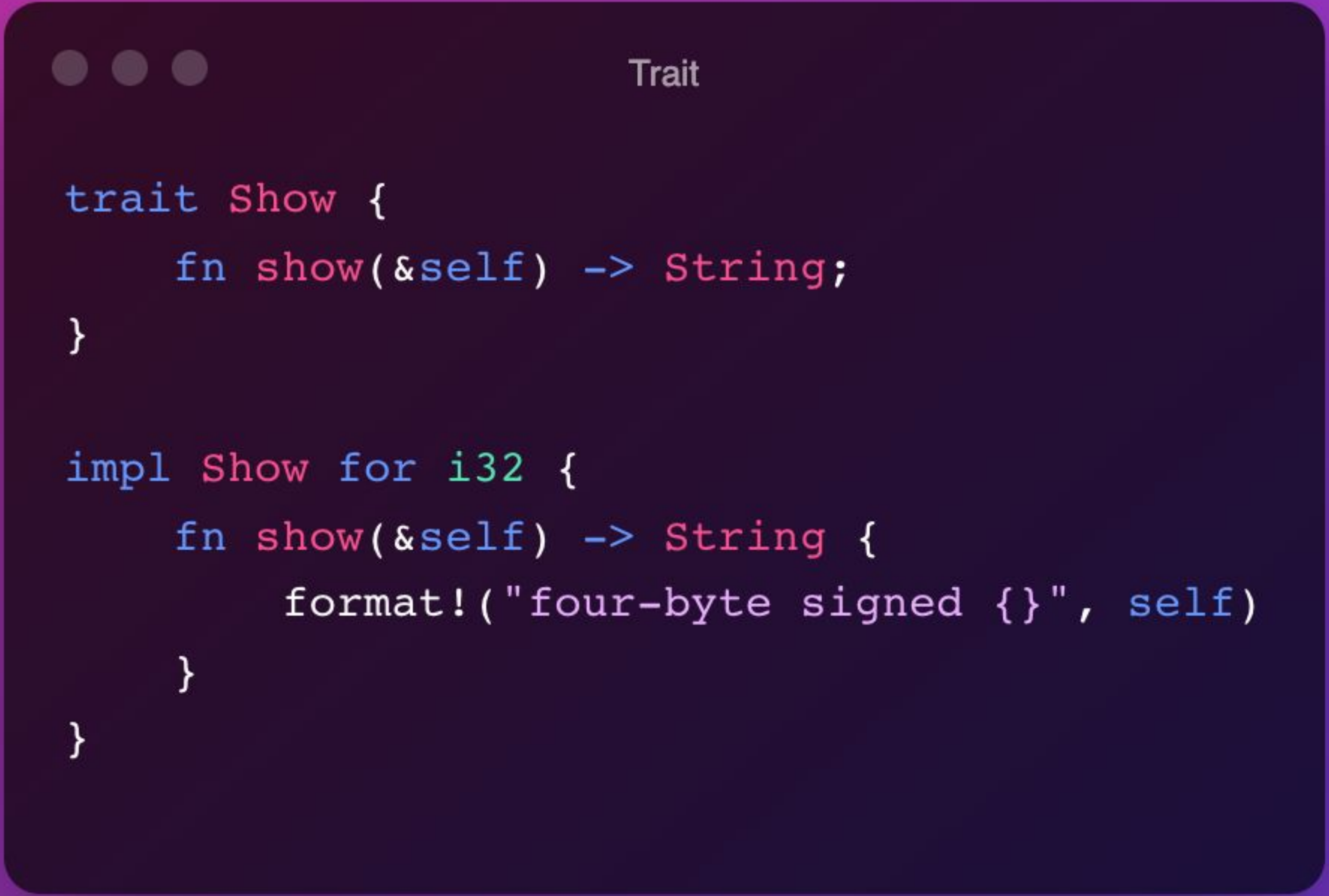


Ownership rules

- Each value in Rust has a variable that’s called its owner
- There can only be one owner at a time
- When the owner goes out of scope, the value will be dropped

Rust - Traits

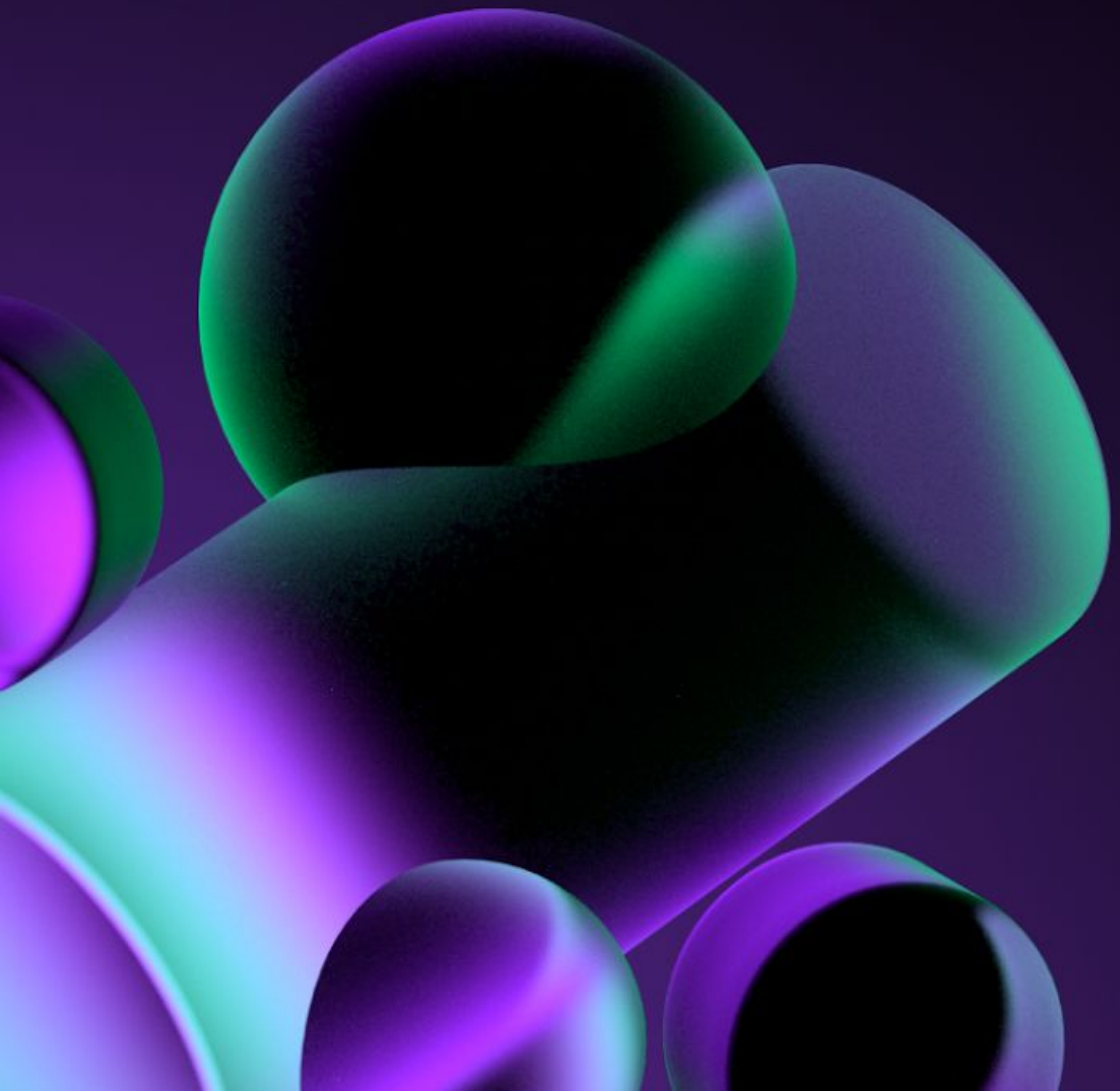


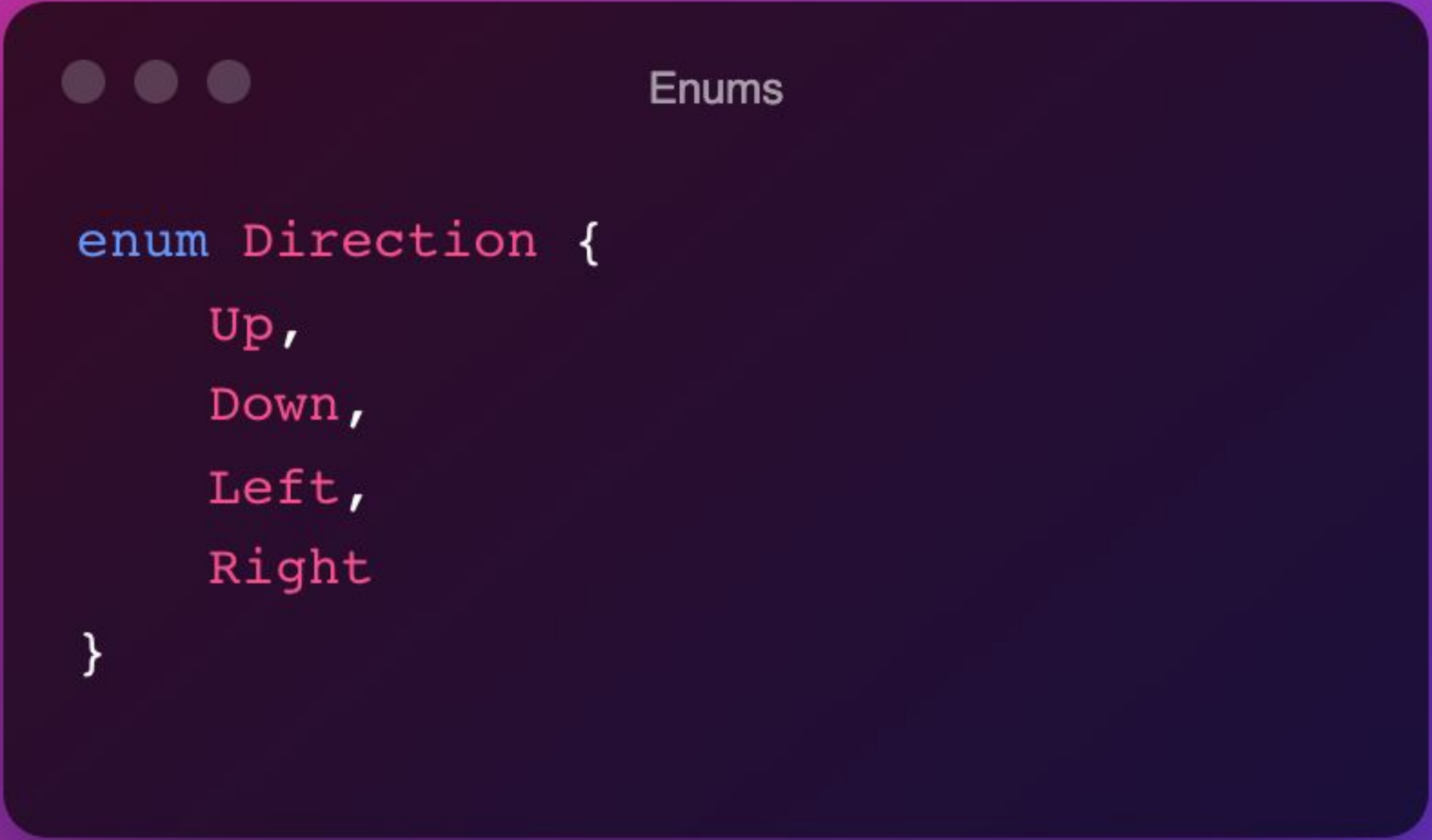


```rust
trait Show {
    fn show(&self) -> String;
}

impl Show for i32 {
    fn show(&self) -> String {
        format!("four-byte signed {}", self)
    }
}
```

Rust - Enums

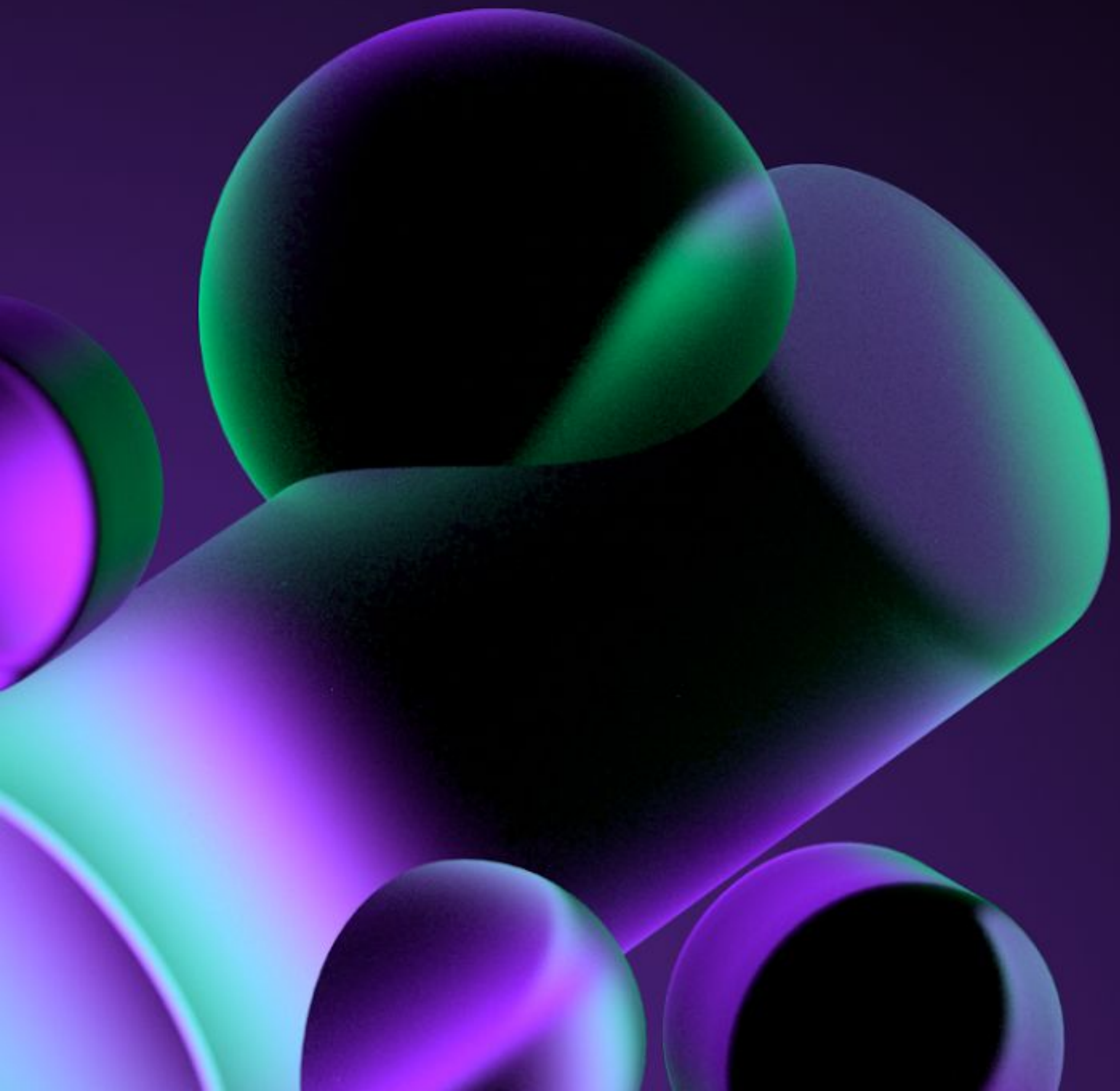


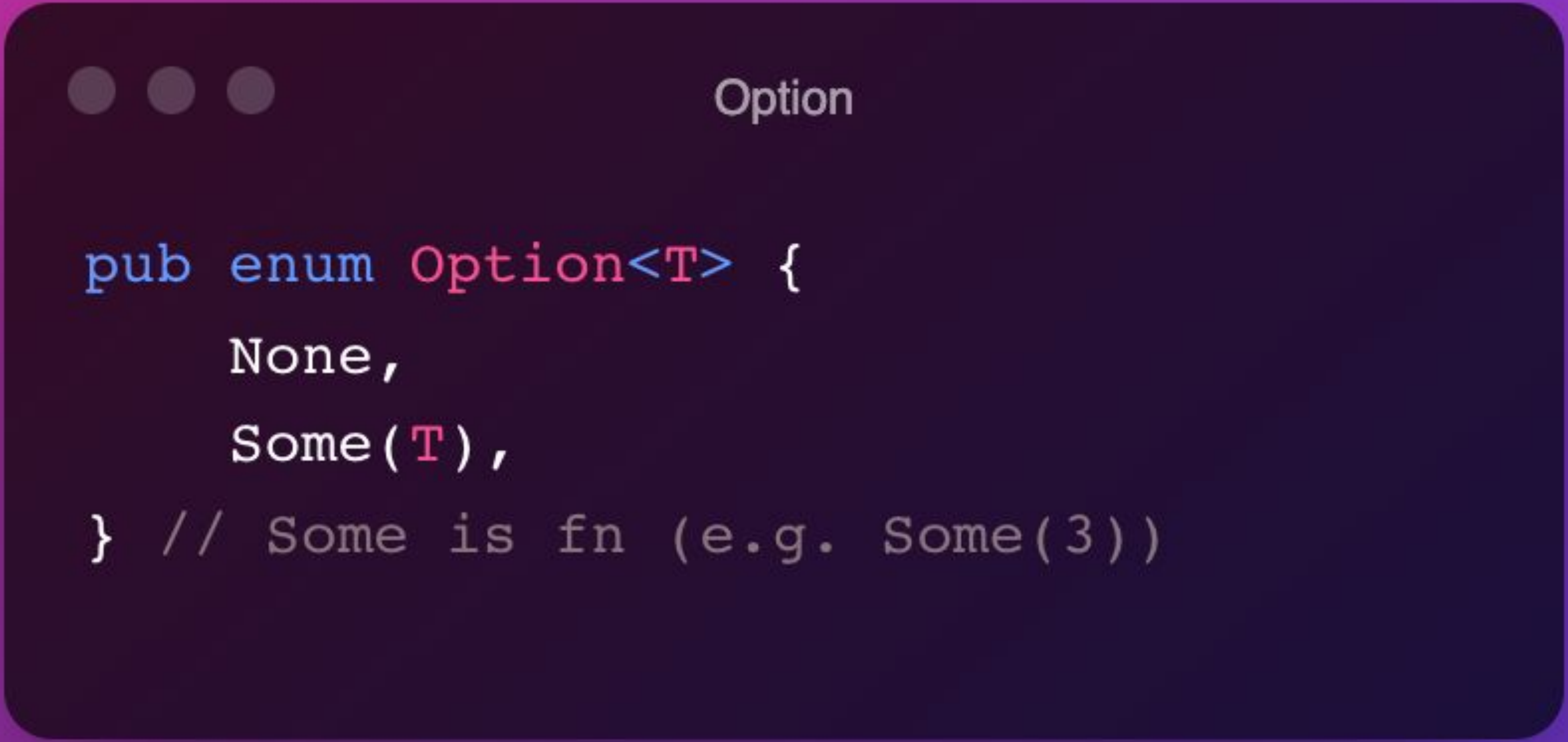


Enums

```
enum Direction {
    Up,
    Down,
    Left,
    Right
}
```

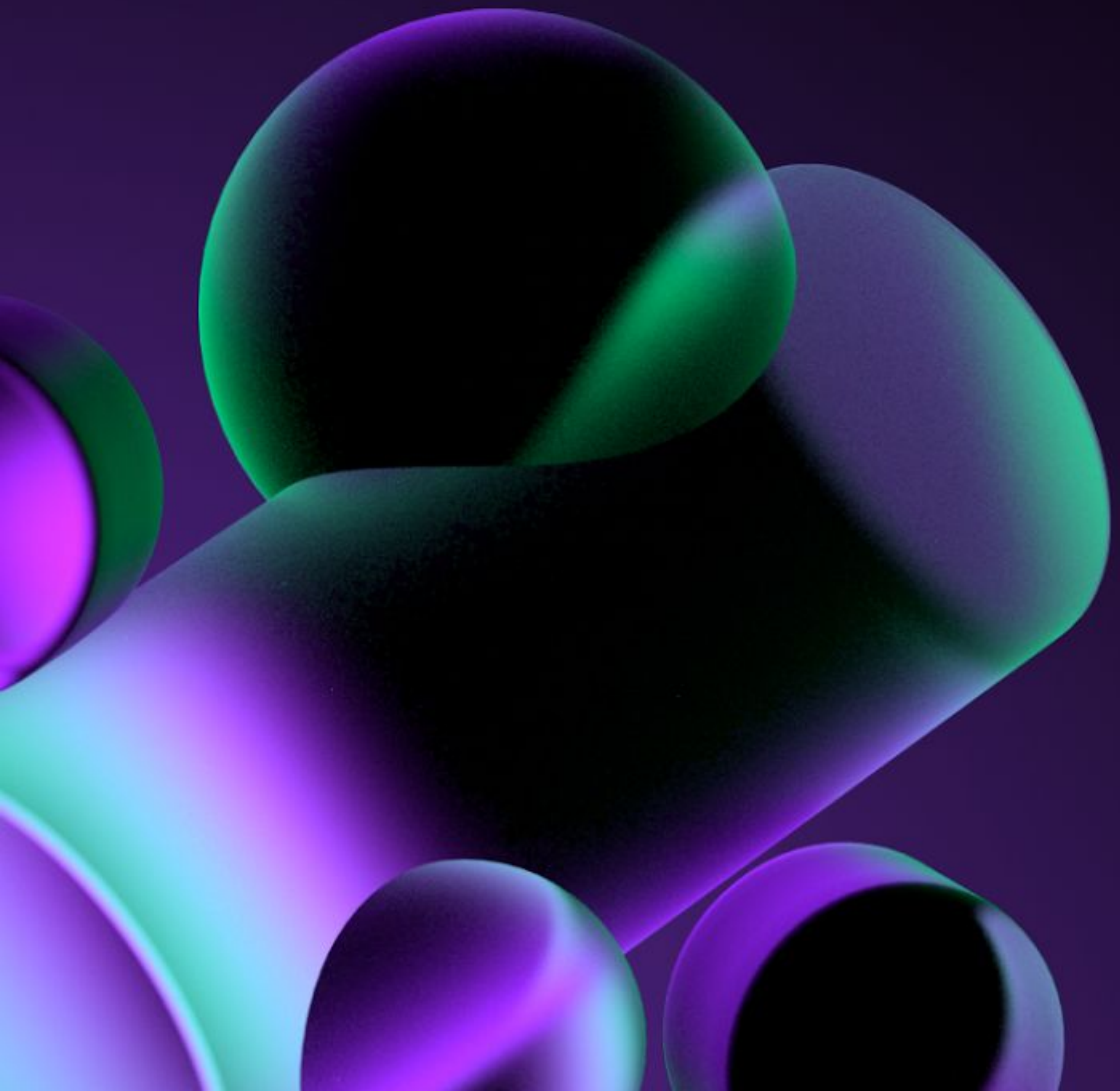
Rust - Option<T>

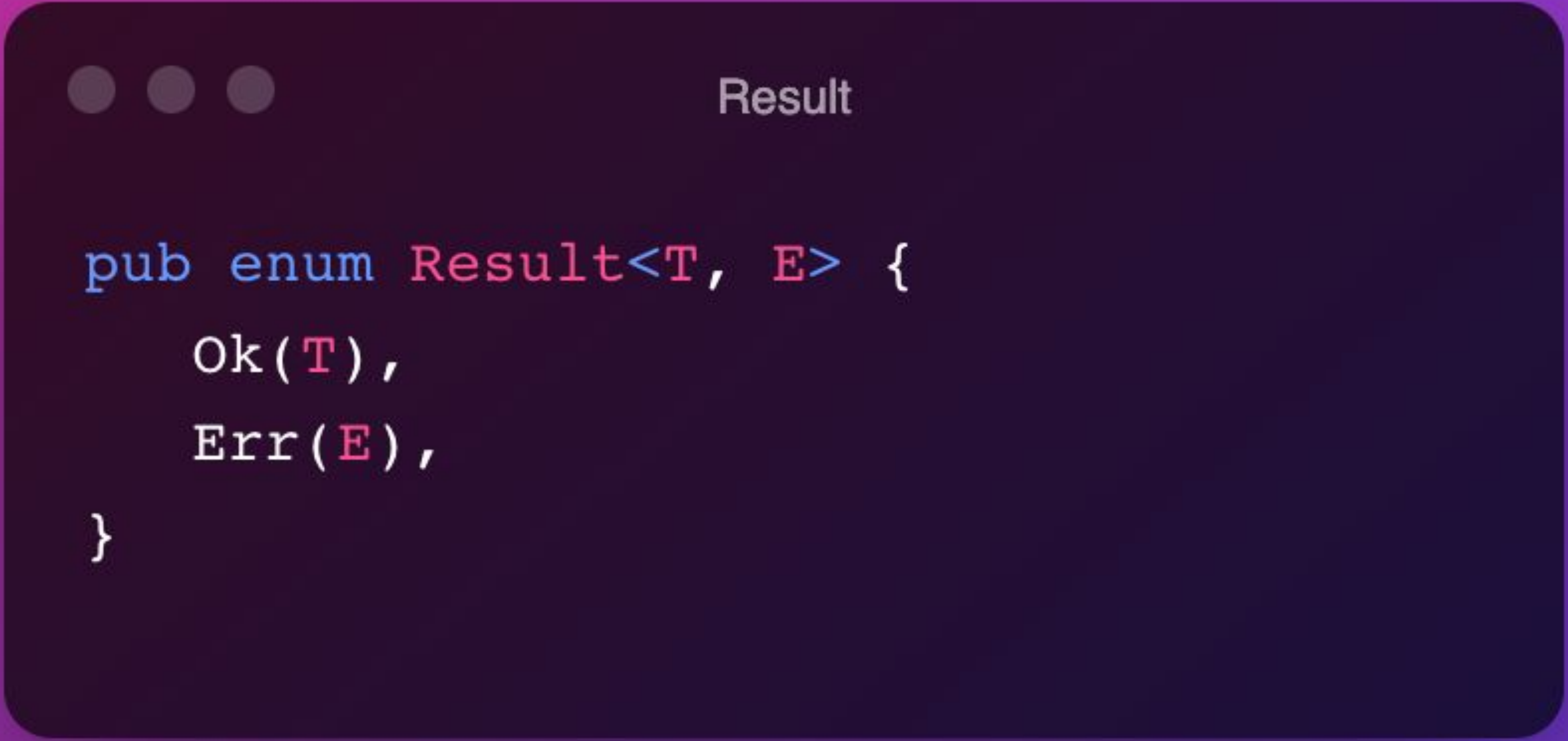




```rust
pub enum Option<T> {
    None,
    Some(T),
} // Some is fn (e.g. Some(3))
```

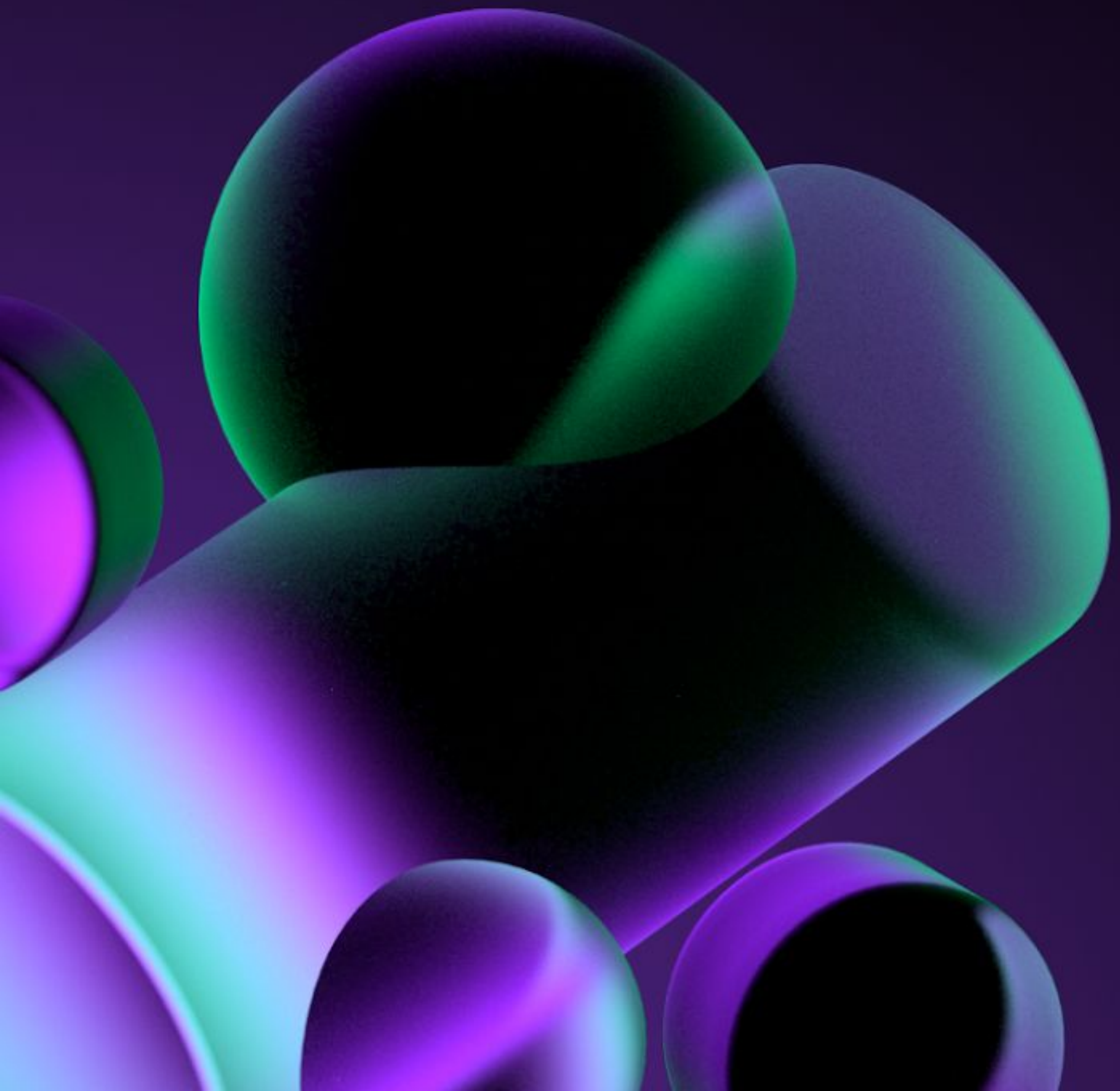
Rust - Result<T, E>

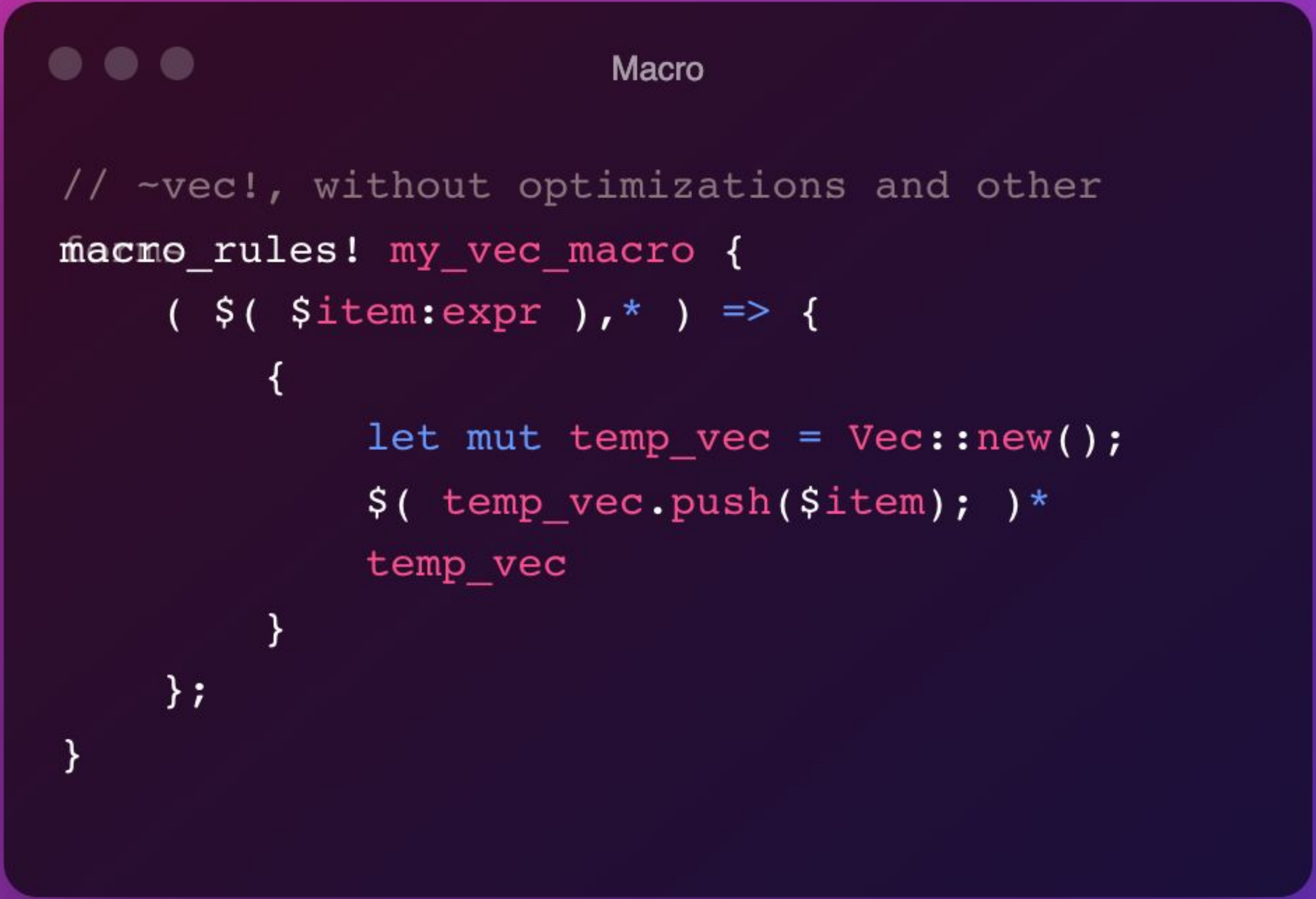




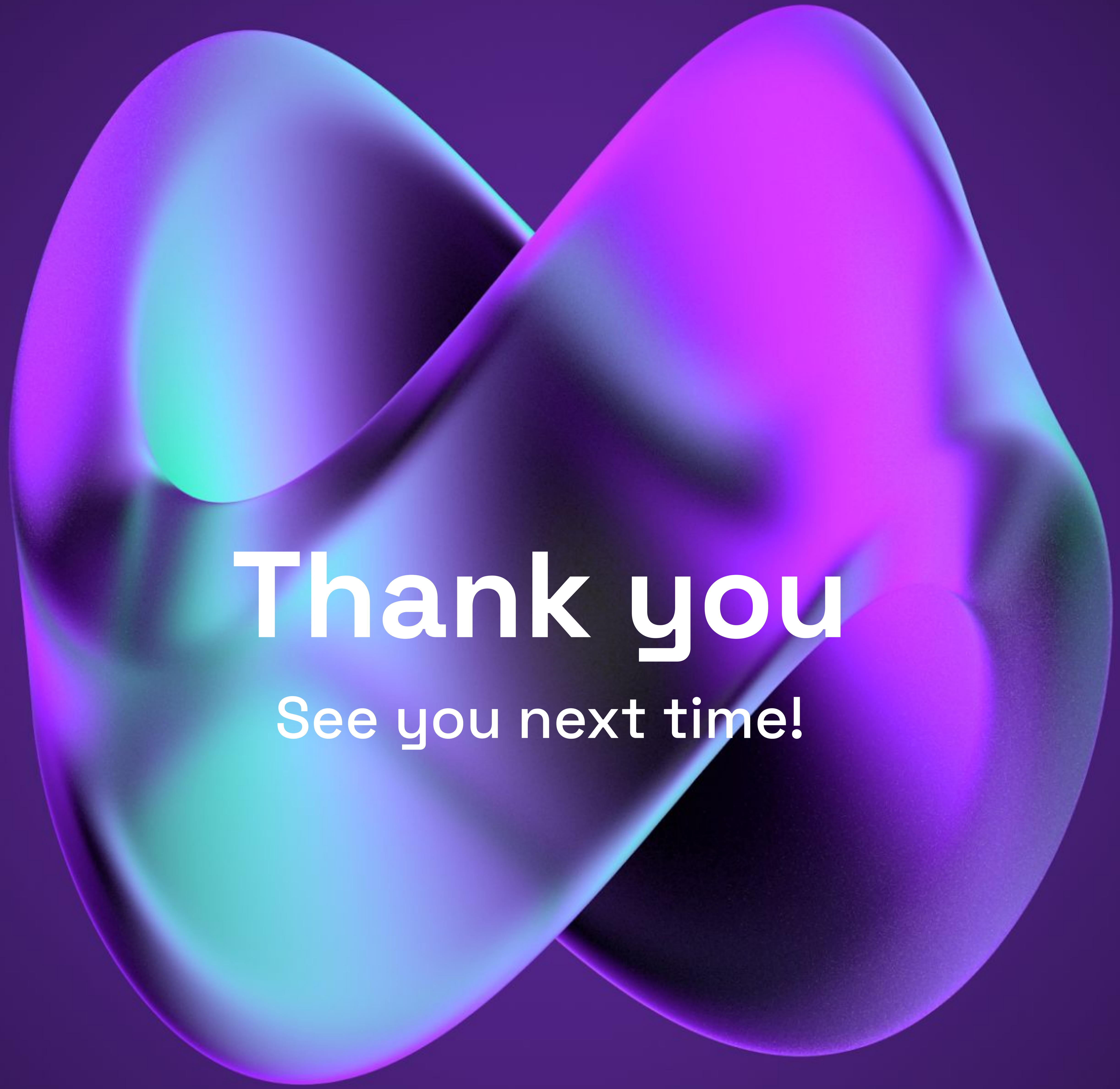
```
pub enum Result<T, E> {
   Ok(T),
   Err(E),
}
```

Rust - Macro





```
// ~vec!, without optimizations and other
macro_rules! my_vec_macro {
    ( $( $item:expr ),* ) => {
        {
            let mut temp_vec = Vec::new();
            $( temp_vec.push($item); )*
            temp_vec
        }
    };
}
```



Thank you

See you next time!