



The `.unwrap()` Method in Rust

Core Purpose

`.unwrap()` is a method that extracts the value from `Option<T>` or `Result<T, E>` types, but **panics** if the value isn't there. It's essentially saying: "I'm certain this has a value, and if I'm wrong, crash the program."

The Two Implementations

`Option::unwrap()`

```
// Simplified from std library
impl<T> Option<T> {
    pub fn unwrap(self) -> T {
        match self {
            Some(val) => val,
            None => panic!("called `Option::unwrap()` on a `None` value"),
        }
    }
}
```

What happens:

1. Takes ownership of the `Option<T>` (consumes it)
2. Pattern matches on the enum variant
3. If `Some(val)`, returns the inner `T` value by moving it out
4. If `None`, calls `panic!()` macro

`Result::unwrap()`

