Type Conversion | From/Into

From/Into

- Rust has a robust type system
  - More reliable & maintainable code
  - Cumbersome to work with similar & wrapper types
    - Usually requires extra repeated code
- Traits can be used to easily convert between types:
  - ***From***
    - Convert **from** one type to another
  - ***Into***
    - Convert one type **into** another type

Traits: *From/Into*

- ***From:***
  - Associated method on a type
    - `TypeName::from()`
  - Implementing ***From*** automatically implements ***Into***
- ***Into:***
  - ***self*** method on a type
    - `variable.into()`

From/Into Example

```rust
let owned = String::from("slice");

let owned: String = "slice".into();

fn to_owned(slice: &str) -> String {
    slice.into()
}
```

Implementing *From*

```rust
enum Status {
    Broken(u8),
    Working,
}

impl From<u8> for Status {
    fn from(code: u8) -> Self {
        match code {
            0 => Status::Working,
            c => Status::Broken(code),
        }
    }
}
```

Using *From/Into* Implementation

```rust
// Returns a status code
fn legacy_interface() -> u8 {
    5
}

let status: Status = legacy_interface().into();
let status = Status::from(legacy_interface());
```

Pro Tips

- ***From/Into*** cannot fail
- Almost always want to implement ***From*** for errors
- Prefer implementing ***From*** instead of ***Into***
    - ***Into*** is automatically implemented with ***From***
- Use ***.into()*** when:
    - Obvious what resulting type will be
- Use ***Type::from()*** when:
    - Important to know the resulting type

Question Mark Operator

```rust
struct Job;

enum JobError {
    Expired,
    Missing,
    Other(u8),
}
```

```rust
impl From<u8> for JobError {
    fn from(code: u8) -> Self {
        match code {
            1 => Self::Expired,
            2 => Self::Missing,
            c => Self::Other(c),
        }
    }
}
```

```rust
fn execute_job(job: Job) -> Result<(), JobError> {
    Err(2)?
}
```

Recap

- ***From/Into*** allow conversion between types
  - The conversion cannot fail
- Prefer implementing ***From*** over ***Into***
  - ***Into*** gets implemented automatically when ***From*** is implemented
- The Question Mark operator will automatically use a ***From*** implementation to convert errors