

Type Conversion | TryFrom/TryInto

■ *TryFrom/TryInto*

- ◆ *Fallible* type conversion
 - Use when there is the possibility of failure
- ◆ Just like *From/Into*, except it returns a ***Result***
 - *TryFrom* will auto-implement *TryInto*

■ Implementing *TryFrom*

```
use std::convert::TryFrom;
enum NonZeroError {
    IsZero,
}
struct NonZero(i32);

impl TryFrom<i32> for NonZero {
    type Error = NonZeroError;

    fn try_from(value: i32) -> Result<Self, Self::Error> {
        if value == 0 {
            Err(NonZeroError::IsZero)
        } else {
            Ok(NonZero(value))
        }
    }
}
```

Usage

```
struct NonZero(i32);
```

```
use std::convert::{TryFrom, TryInto};
```

```
match NonZero::try_from(9) {  
    Ok(nonzero) => println!("not zero"),  
    Err(e) => println!("is zero!"),  
}
```

```
let whoops: Result<NonZero, _> = 0_i32.try_into();  
match whoops {  
    Ok(nonzero) => println!("not zero"),  
    Err(e) => println!("is zero!"),  
}
```

■ Recap

- ◆ *TryFrom/TryInto* allow conversion between types
 - Conversion can fail
- ◆ Prefer implementing *TryFrom* over *TryInto*
 - *TryInto* gets implemented automatically when *TryFrom* is implemented