

Rust Error Handling

Jim Fawcett

23 October 2020

Rust Error Handling Types

- `Enum Result<T,E> { Ok(T), Err(E) }`
 - `#[must_use]`
 - `std module import`
- `pub fn is_ok(&self) -> bool`
- `pub fn is_Err(&self) -> bool`
- `pub fn unwrap() -> T`
 - panics if not Ok

Error Types

- `type Result<T> = Result<T,io::Error>;`
 - `std::io module import`
- Custom Error
 - Use `std::io::{Error, ErrorKind};`
 - Let `custom_error =`
`Error::new(ErrorKind::SomeType, custom_error);`

Rust Option

- `pub enum Option<T> { None, Some(T), }`
 - `std::option` module
 - `pub fn is_some(&self) -> bool`
 - `pub fn is_none(&self) -> bool`
 - `pub fn unwrap(self) -> T`
 - panics if `None`