Rust Links

Books

- Rust Standard Library
- Rust Book
- Rust Reference
- Cargo Book
- Rust by Example
- Rust Edition Guide 2018
- The Little Book of Rust Macros
- Half an Hour to Learn Rust
- Rust Unstable Book

Articles

- Arrays, References and Slices in Rust
- Where to Put the Turbofish

Cheat Sheet

- Rust Language Cheat Sheet

Associated Types

Use associated type when there is a one-to-one relationship between the type implementing the trait and that type. It's referred to as an *output type* because it's the output of applying a trait to the parent type.

- Associated Types

Tools

- Rust Playground
- GCC/LLVM Code Coverage
- Tarpaulin Code Coverage Tool

Programming

I/O

Standard out, `io::stdout()` and standard in, `io::stdin()`.

Read a line to a string `io::stdin().read_to_string(&mut input)?;`. `std::io::Read` trait must be in scope.

Write a line to a file `writeln!(io::stdio(), "{}", line)`. `std::io::Write` trait must be in scope.

String

Use `for line in input.lines() { }` to iterate all lines of a string.

Use `let value: i32 = line.parse()?;` to parse an integer. Type of `value` implicitly defines `parse` method used.

See if a string is ASCII - `line.is_ascii()`

Get an iterator for the characters of a string - `str1.chars()`

Errors

Declare `type Result<T> = ::std::result::Result<T, Box<dyn ::std::error::Error>>;` to save typing.

Collections

Bring in scope`use std::collections::HashSet;`. Use `insert(T)` to add. Use `contains(&T)` to test.

Arrays

Get array length - `arr.len()`.

Arrays must be indexed by `usize`.

Iteration

To get a mutable iterator - `for f in list.iter_mut()`

Zip entries from two iterators together - `iter1.zip(iter2)`

Filter items from an iterator - `iter1.filter(|&item| item != 0)`

Useful Crates

- lazy_static
- regex