

The Iterator and Intoliterator Traits

Learn to Code with Rust / Section Review

Vectors

- A **vector** is an collection type for storing homogenous elements in order.
- A **vector** is similar to an array but it can grow and shrink in size.
- Rust's vector type is **Vec**. It has one generic which represents the type of the stored elements.

Create a Vector

- The **Vec::new** constructor function returns an empty vector.
- The **Vec::new** function requires a manual type annotation.
- As soon as code inserts an element into the vector, the compiler can infer its generic type.
- The **vec![]** macro creates a vector with pre-populated elements.

Vector Methods

- The **push** method appends an element to the end of the vector.
- The **insert** method adds an element at a specific index position.
- The **pop** method attempts to remove the last element from the vector. It returns an **Option** enum.
- The **remove** method removes an element by index position.

Reading Vector Elements

- Use square brackets to extract a vector element by its index position
- Rules of ownership apply. If a type does not implement the copy trait, use the borrow operator (&) to borrow without moving ownership.
- The **get** method accepts an index position and returns an **Option** enum.
- Borrow a slice with the borrow operator, the value, square brackets, and a range.

Writing Vector Elements

- Use square brackets to target an index position, then overwrite its value with an equal sign.
- Rust permits one mutable reference to a value at a time.
- Rust permits any number of immutable references to a value at a time.