Packages and Crates

Learn to Code with Rust

Packages

- Our previous Rust programs all lived in a single **main.rs** file.
- When we run the **cargo new** command, we create a Rust package.
- A **package** is a folder with a **Cargo.toml** file. The **Cargo.toml** file holds metadata about the package like its name and version.

Crates

- A **package** is a collection of one or more **crates.**
- A **crate** is a collection of Rust code that produces an executable or a library.
- A **crate** is the smallest amount of code that the Rust compiler considers at a time.

Types of Crates

- A **binary crate** is a crate that compiles to an executable.
  - A **binary crate** has a **main** function that is the entrypoint for the executable.
- A **library crate** exports functionality for other Rust programs to share and use.
  - A **library crate** does not have a **main** function and does not compile to be an executable program.

Our Project

- The **cargo new** command will default to creating a package with a binary crate.
- The **Cargo.toml** file's **name** field sets the name of the package.
  - Our package name is **warehouse**.
- Cargo will look for a **src/main.rs** file. If it exists, Rust infers that we have a binary crate named **warehouse**.
- Cargo will look for a **src/lib.rs** file. If it exists, Rust infers that we have a library crate named **warehouse**.