

Packages and Crates

What is a Crate?

A crate is a subset of a package that is used in the actual Cairo compilation. This includes:

- The package source code, identified by the package name and the crate root, which is the main entry point of the package.
- A subset of the package metadata that identifies crate-level settings of the Cairo compiler, for example, the `edition` field in the `_Scarb.toml_` file.

Crates can contain modules, and the modules may be defined in other files that get compiled with the crate, as will be discussed in the subsequent sections.

What is the Crate Root?

The crate root is the `_lib.cairo_` source file that the Cairo compiler starts from and makes up the root module of your crate. We'll explain modules in depth in the ["Defining Modules to Control Scope"]([modules]) chapter.

[modules]: ./ch07-02-defining-modules-to-control-scope.md

What is a Package?

A Cairo package is a directory (or equivalent) containing:

- A `_Scarb.toml_` manifest file with a `[package]` section.
- Associated source code.

This definition implies that a package might contain other packages, with a corresponding `_Scarb.toml_` file for each package.

Creating a Package with Scarb

You can create a new Cairo package using the Scarb command-line tool. To create a new package, run the following command:

```
```bash
scarb new my_package
```
```

This command will generate a new package directory named `_my_package_` with the following structure:

```
```
my_package/
% % % Scarb.toml
% % % src
 % % % lib.cairo
```
```

- `_src/_` is the main directory where all the Cairo source files for the package will be stored.
- `_lib.cairo_` is the default root module of the crate, which is also the main entry point of the package.
- `_Scarb.toml_` is the package manifest file, which contains metadata and configuration options for the package, such as dependencies, package name, version, and authors. You can find documentation about it on the [Scarb reference]([manifest]).

```
```toml
[package]
name = "my_package"
version = "0.1.0"
```

```
edition = "2024_07"
[dependencies]
foo = { path = "vendor/foo" }
``
```

As you develop your package, you may want to organize your code into multiple Cairo source files. You can do this by creating additional `_.cairo_` files within the `_src_` directory or its subdirectories.

```
{{#quiz ../quizzes/ch07-01-packages-crates.toml}}
```

```
[manifest]: https://docs.swmansion.com/scarb/docs/reference/manifest.html
```