

Introduction to Starknet Smart Contracts

All through the previous sections, you've mostly written programs with a ``main`` entrypoint. In the coming sections, you will learn to write and deploy Starknet contracts. One of the key applications of the Cairo language is writing smart contracts for the Starknet network. Starknet is a permissionless decentralized network that leverages zk-STARKs technology for scalability. As a Layer 2 (L2) scalability solution for Ethereum, Starknet aims to provide fast, secure, and low-cost transactions. It operates as a validity rollup, commonly known as a zero-knowledge rollup, and is built on top of the Cairo VM. Starknet contracts are programs specifically designed to run within the Starknet OS. The Starknet OS is a Cairo program itself, which means that any operation executed by the Starknet OS can be proven and succinctly verified. Smart contracts can access Starknet's persistent state through the OS, enabling them to read or modify variables in Starknet's state, communicate with other contracts, and interact seamlessly with the underlying Layer 1 (L1) network.

If you want to learn more about the Starknet network itself, its architecture and the tooling available, you should read the [Starknet Book][starknet book]. In this book, we will only focus on writing smart contracts in Cairo.

[starknet book]: <https://book.starknet.io/>

Scarb

Scarb facilitates smart contract development for Starknet. To enable this feature, you'll need to make some configurations in your `_Scarb.toml_` file (see [Installation][scarb installation] for how to install Scarb).

First, add the ``starknet`` dependency to your `_Scarb.toml_` file. Next, enable the Starknet contract compilation of the package by adding a `[[target.starknet-contract]]` section. By default, specifying this target will build a Sierra Contract Class file, which can be deployed on Starknet. If you omit to specify the target, your package will compile but will not produce an output that you can use with Starknet.

Below is the minimal `_Scarb.toml_` file required to compile a crate containing Starknet contracts:

```
``toml
[package]
name = "package_name"
version = "0.1.0"
[dependencies]
starknet = ">=2.8.0"
[[target.starknet-contract]]
``
```

To compile contracts defined in your package's dependencies, please refer to the [Scarb documentation][compile dep contract].

[scarb installation]: [./ch01-01-installation.md](#)

[compile dep contract]: <https://docs.swmansion.com/scarb/docs/extensions/starknet/contract-target.html#compiling-external-contracts>

Starknet Foundry

Starknet Foundry is a tool chain for Starknet smart contract development. It supports many features, including writing and running tests with advanced features, deploying contracts, interacting with the Starknet network, and more.

We'll describe Starknet Foundry in more detail in [Chapter 17][testing with snfoundry] when discussing Starknet smart contract testing and security.
[testing with snfoundry]: ./ch17-02-testing-smart-contracts.md#testing-smart-contracts-with-starknet-foundry