

How to call functions from other functions

A contract is a collection of persistent [state variables](#) , and [functions](#) which may manipulate these variables. Functions and state variables within a contract's scope are said to belong to that contract. A contract can only access and modify its own state. If a contract wishes to access or modify another contract's state, it must make a call to an external function of the other contract. For anything to happen on the Aztec network, an external function of a contract needs to be called.

Contract

A contract may be declared and given a name using the `contract` keyword (see snippet below). By convention, contracts are named in `PascalCase` .

```
contract keyword contract MyContract
```

```
{
```

```
// Imports
```

```
// Storage
```

```
// Functions }
```

A note for vanilla Noir devs There is no [main\(\)](#) function within a Noir contract scope. More than one function can be an entrypoint.

Directory structure

Here's a common layout for a basic Aztec.nr Contract project:

```
—— my_aztec_contract_project |—— src | |—— main.nr <-- your contract |—— Nargo.toml <-- package and  
dependency management * See the vanilla Noir docs for more info on packages * . * You can review the structure of a  
complete contract in the token contract tutorial here * . Edit this page
```

[Previous Visibility](#) [Next How to write an initializer function](#)