Functions

Functions serve as the building blocks of smart contracts. Functions can be eitherpublic , ie they are publicly available for anyone to see and can directly interact with public state, orprivate , meaning they are executed completely client-side in thePXE . Read more about how private functions workhere .

For a more practical guide of using multiple types of functions, follow thetoken tutorial .

Currently, any function is "mutable" in the sense that it might alter state. However, we also support support static calls, similarly to EVM. A static call is essentially a call that does not alter state (it keeps state static).

Initializer functions

Smart contracts may have one, or many, initializer functions which are called when the contract is deployed.

Initializers are regular functions that set an "initialized" flag (a nullifier) for the contract. A contract can only be initialized once, and contract functions can only be called after the contract has been initialized, much like a constructor. However, if a contract defines no initializers, it can be called at any time. Additionally, you can define as many initializer functions in a contract as you want, both private and public.

Oracles

There are also special oracle functions, which can get data from outside of the smart contract. In the context of Aztec, oracles are often used to get user-provided inputs.

Explore this section to learn:

- How function visibility works in Aztec
- Public, private, and unconstrained functions
- , and how to write them
- How to write aninitializer function
- Calling functions from within the same smart contract and from different contracts
- , including calling private functions from private functions, public from public, and even private from public
- Oracles
- and how Aztec smart contracts might use them
- How functions work under the hood Edit this page

Previous What a contract looks like Next Function Context