

Introduction

What is CosmWasm?

CosmWasm is a smart contracting platform built for the Cosmos ecosystem. Simply put, it's the Cosmos (Cosm) way of using WebAssembly (Wasm) hence the name.

CosmWasm is written as a module that can plug into the Cosmos SDK. This means that anyone currently building a blockchain using the Cosmos SDK can quickly and easily add CosmWasm smart contracting support to their chain, without adjusting existing logic.

[Rust](#) is currently the most used programming language for CosmWasm, in the future, it is possible to have different programming languages like [AssemblyScript](#)

The purpose of this documentation is to give a deep dive into the technology for developers who wish to try it out or integrate it into their products. Particularly, it is aimed at Go developers with experience with the Cosmos SDK, as well as Rust developers looking for a blockchain platform.

info [Here](#) you can read the launch article of CosmWasm.

How to use CosmWasm

As CosmWasm is another Cosmos SDK module, a binary is enough to start integrating it into your blockchain.

A sample binary of CosmWasm integrated into the `gaiad` binary, called `wasmd` is provided and can be found [here](#). Using `wasmd` it is possible to launch a new smart-contract enabled blockchain out of the box, using documented and tested tooling and the same security model as the Cosmos Hub.

A running blockchain is needed to host and interact with the contracts. It can be either localhost, testnet, or a mainnet blockchain.

The details on how to [connect to a testnet](#) or [set up a local devnet](#) will be explained in the later sections.

Sections

- [Getting Started](#)
 - dives you into hands-on training. It gently leads you through
 - modifying, deploying, and executing a smart contract on a local blockchain. It is the ideal place to go through and
 - get acquainted with all the aspects of the system, without too much hard work coding.
- [Architecture](#)
 - explains much of the high-level design and architecture of
 - CosmWasm. It is crucial to understand the mental model and capabilities of the
 - system before designing products using it. However, if you prefer to learn by coding then you can skip this section and visit as you need it.
- Learn
 - demonstrates developing smart contracts from zero to production with step
 - by step explanations, code snippets, scripts and more.
- - [Dev Academy](#)
- - provides structured learning content starting from basics of blockchains and smart contracts to Cosmos SDK, CosmWasm smart contracts and clients.
- - [Tutorials](#)
- - demonstrates developing smart contracts from zero to production with step by step explanations, code snippets, scripts, and more.
- [Workshops](#)
 - has a great collection of demonstrations and verbal explanations of CosmWasm
 - tech stack recorded in various events and organizations.
- [Plus](#)
 - is for state-of-the-art, production ready CosmWasm smart contracts.

Additional Resources

Lots of valuable information that will help you in your CosmWasm journey are also available outside of this documentation.

Here, a few of them are listed:

- [A set of example smart contracts](#)
- for experimenting.
- Rustdoc for the [core contract libs](#)
- .
- Rustdoc for the [storage helpers](#)
- .

There are quite a few [high-level articles on medium](#) that explain the various components of our stack and where we are going.

Many thanks to the [Interchain Foundation](#) for funding most of the development work to bring CosmWasm to production. [Next Your First Contract](#)