

Smart Contract

Smart contracts are pieces of executable code stored in the [account's state](#) that have their own storage, and perform transactions in the account's name.

They are written in Javascript or Rust, and then compiled and deployed to an account so everyone can interact with them through their public methods.

Why Smart Contracts Matter

Smart contracts enable to create fully decentralized applications .

In fact, smart contracts enable to create a multitude of [awesome apps](#) such as:

- Decentralized Autonomous Organizations, where users create and vote proposals.
- Art Marketplaces, where users create and commercialize digital art pieces.
- Decentralized exchanges, where users can trade different currencies.

Developing Contracts in NEAR

Developers can choose between using [Javascript](#) or [Rust](#) to write smart contracts in NEAR.

Indistinctly from the language chosen, the NEAR SDK will help you to compile the contract into WebAssembly, from which point it can be deployed and executed on the NEAR platform.

tip See how simple it is to build a contract in NEAR with our [Quickstart Guide](#) . You will spin-up your first dApp in a matter of seconds.

Advantages of NEAR Contracts

1. Simple to Build

Thanks to our unique [Javascript SDK](#) and our [vast documentation](#) .

2. Simple to Maintain

Since the contract's code is separated from [its storage](#) , contracts of [non-locked](#) account can be updated .

3. Some Methods are Free to Call

Public methods that perform only read operations can be called for free . [Edit this page](#) Last updated on Feb 1, 2023 by Damián Parrino Was this page helpful? Yes No

[Previous](#) [Access Keys](#) [Next](#) [State](#)