INTRODUCTION TO SMART CONTRACTS

Last edit: @kziechmann ↗ , March 30, 2021

See contributors

Edit page

On this page >

WHAT IS A SMART CONTRACT?

A "smart contract" is simply a program that runs on the Ethereum blockchain. It's a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain.

Smart contracts are a type of Ethereum account. This means they have a balance and they can send transactions over the network. However they're not controlled by a user, instead they are deployed to the network and run as programmed. User accounts can then interact with a smart contract by submitting transactions that execute a function defined on the smart contract. Smart contracts can define rules, like a regular contract, and automatically enforce them via the code.

PREREQUISITES

Make sure you've read up on accounts, transactions and the Ethereum virtual machine before jumping into the world of smart contracts.

A DIGITAL VENDING MACHINE

Perhaps the best metaphor for a smart contract is a vending machine, as described by Nick Szabo ↗ . With the right inputs, a certain output is guaranteed.

To get a snack from a vending machine:

```
1    money + snack selection = snack dispensed
2
```

This logic is programmed into the vending machine.

A smart contract, like a vending machine, has logic programmed into it. Here's a simple example of how this vending machine might look like as a smart contract:

Show less Copy

```
pragma solidity 0.6.11;

contract VendingMachine {

    // Declare state variables of the contract
    address public owner;
    mapping (address => uint) public cupcakeBalances;

    // When 'VendingMachine' contract is deployed:
    // 1. set the deploying address as the owner of the contract
    // 2. set the deployed smart contract's cupcake balance to 100
    constructor() public {
        owner = msg.sender;
        cupcakeBalances[address(this)] = 100;
    }

    // Allow the owner to increase the smart contract's cupcake balance
    function refill(uint amount) public {
        require(msg.sender == owner, "Only the owner can refill.");
        cupcakeBalances[address(this)] += amount;
    }

    // Allow anyone to purchase cupcakes
    function purchase(uint amount) public payable {
        require(msg.value >= amount * 1 ether, "You must pay at least 1 ETH per cupcake");
        require(cupcakeBalances[address(this)] >= amount, "Not enough cupcakes in stock to
complete this purchase");
        cupcakeBalances[address(this)] -= amount;
        cupcakeBalances[msg.sender] += amount;
    }
}

```

Like how a vending machine removes the need for a vendor employee, smart contracts can replace intermediaries in many industries.

PERMISSIONLESS

Anyone can write a smart contract and deploy it to the network. You just need to learn how to code in a smart contract language, and have enough ETH to deploy your contract. Deploying a smart contract is technically a transaction, so you need to pay your Gas in the same way that you need to pay gas for a simple ETH transfer. Gas costs for contract deployment are far higher, however.

Ethereum has developer-friendly languages for writing smart contracts:

- Solidity
- Vyper

More on languages

However, they must be compiled before they can be deployed so that Ethereum's virtual machine can interpret and store the contract. More on compilation

COMPOSABILITY

Smart contracts are public on Ethereum and can be thought of as open APIs. That means you can call other smart contracts in your own smart contract to greatly extend what's possible. Contracts can even deploy other contracts.

Learn more about smart contract composability.

LIMITATIONS

Smart contracts alone cannot get information about "real-world" events because they can't send HTTP requests. This is by design. Relying on external information could jeopardise consensus, which is important for security and decentralization.

There are ways to get around this using oracles.

SMART CONTRACT RESOURCES

OpenZeppelin Contracts - *Library for secure smart contract development.*

- openzeppelin.com/contracts/ ↗
- GitHub ↗
- Community Forum ↗

DappSys - *Safe, simple, flexible building-blocks for smart-contracts.*

- dapp.tools/dappsys ↗
- GitHub ↗

FURTHER READING

- Smart Contracts: The Blockchain Technology That Will Replace Lawyers ↗ *– Blockgeeks*
- Best Practices for Smart Contract Development ↗ *– Nov 10, 2019 - Yos Riady*

Back to top ↑

Did this page help answer your question?

Yes No

PREVIOUS
Intro to the stack

NEXT
Smart contract languages

Website last updated: June 2, 2021

Use Ethereum

Ethereum Wallets

Get ETH

Decentralized applications (dapps)

Stablecoins

Stake ETH

Learn

What is Ethereum?

What is ether (ETH)?

Community guides and resources

History of Ethereum

Ethereum Whitepaper

Ethereum 2.0

Ethereum Glossary

Ethereum Improvement Proposals

Developers

Get started

Documentation

Tutorials

Learn by coding

Set up local environment

Developer Resources

Ecosystem

Ethereum Community

Ethereum Foundation

Ethereum Foundation Blog ↗

Ecosystem Support Program ↗

Ecosystem Grant Programs

Ethereum Brand Assets

Devcon ↗

Enterprise

Mainnet Ethereum

Private Ethereum

Enterprise

About ethereum.org

About us

Jobs

Contributing

Language Support

Privacy policy