

Apply Fundamentals of Blockchain



Lecturer 4

By: Japhet Moise H.

Apply Solidity Basics

- Solidity is a brand-new programming language created by Ethereum which is the second-largest market of cryptocurrency by capitalization, released in the year 2015 and led by Christian Reitwiessner.



SOLIDITY

Features of solidity

- Solidity is a high-level programming language designed for implementing smart contracts.
- It is a statically typed object-oriented(contract-oriented) language.
- Solidity is highly influenced by Python, c++, and JavaScript which run on the Ethereum Virtual Machine(EVM).
- Solidity supports complex user-defined programming, libraries, and inheritance.
- Solidity is the primary language for blockchains running platforms.
- Solidity can be used to create contracts like voting, blind auctions, crowdfunding, multi-signature wallets, etc.

Key definition

1. **Identifier In Solidity:** identifiers are names given to various elements within a contract, such as variables, functions, structs, and events. They serve as labels that allow you to reference these elements within your code.
2. **Structs in Solidity:** are user-defined data types that allow you to group related data elements together under a single name.
3. **State Variables in Solidity:** State variables are variables that store data persistently within a smart contract.

- **Smart Contracts in Solidity:** Smart contracts are self-executing contracts with terms directly written into lines of code. They run on a blockchain network and automatically execute when predefined conditions are met.
- **Visibility and Access Control in Solidity:** Visibility modifiers in Solidity control the accessibility of contract elements (functions, variables, and modifiers) from other contracts or external accounts. Types of modifiers: public, private, internal, external.
- **Ethereum is a decentralized platform that runs smart contracts:** applications that run exactly as programmed without any possibility of censorship, downtime, or third-party interference.
- **Ethereum Virtual Machine (EVM):** The Ethereum Virtual Machine (EVM) is the runtime environment that executes smart contracts on the Ethereum blockchain. It's a virtual computer that processes transactions and executes the code contained within smart contracts.

Ethereum development tools

1. Integrated Development Environments (IDEs)

- **Remix:** A web-based IDE that allows you to write, compile, deploy, and test smart contracts directly in your browser. It's a great starting point for beginners.
- **Truffle Suite:** A comprehensive suite of tools for developing, testing, and deploying Ethereum applications. It includes a development environment, testing framework, and deployment scripts.
- **Ganache:** A personal blockchain for developers, providing a local environment to test smart contracts without interacting with the main Ethereum network.
- **Visual Studio Code:** A popular code editor with a variety of Ethereum extensions available, including Solidity language support, debugging, and contract deployment.

2. Programming Languages and Compilers:

- Solidity: The primary programming language for writing smart contracts on Ethereum. It's a high-level language with syntax similar to JavaScript.
- Solc: The Solidity compiler, which converts Solidity code into bytecode that can be executed on the Ethereum Virtual Machine (EVM).

3. Testing Frameworks:

- Truffle Test: A testing framework included in the Truffle Suite, allowing you to write unit tests for your smart contracts.
- Hardhat: A development environment and testing framework for Solidity, providing a flexible and modular approach to testing.

4. Deployment Tools

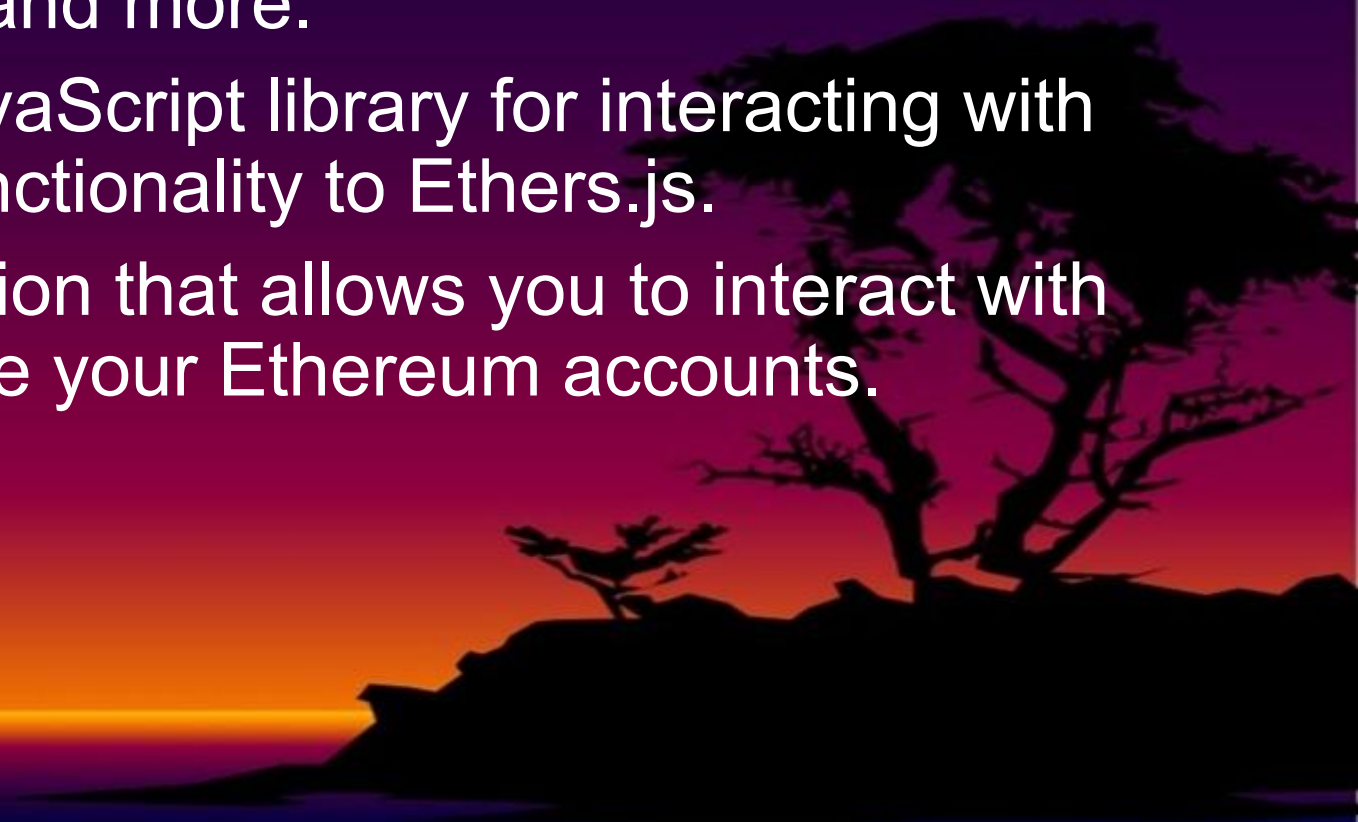
- Truffle Deploy: A tool included in the Truffle Suite for deploying smart contracts to various Ethereum networks.
- Infura: A service that provides access to the Ethereum network and other blockchains, making it easy to deploy and interact with smart contracts.

5. Debugging Tools

- Remix Debugger: A built-in debugger in Remix that allows you to step through your code, inspect variables, and identify errors.
- Truffle Debugger: A standalone debugger for Truffle projects, providing advanced debugging features.

6. Other Tools

- Ethers.js: A JavaScript library for interacting with the Ethereum blockchain, providing a variety of functions for sending transactions, querying data, and more.
- Web3.js: Another popular JavaScript library for interacting with Ethereum, offering similar functionality to Ethers.js.
- MetaMask: A browser extension that allows you to interact with Ethereum dApps and manage your Ethereum accounts.



Thank you!!!

