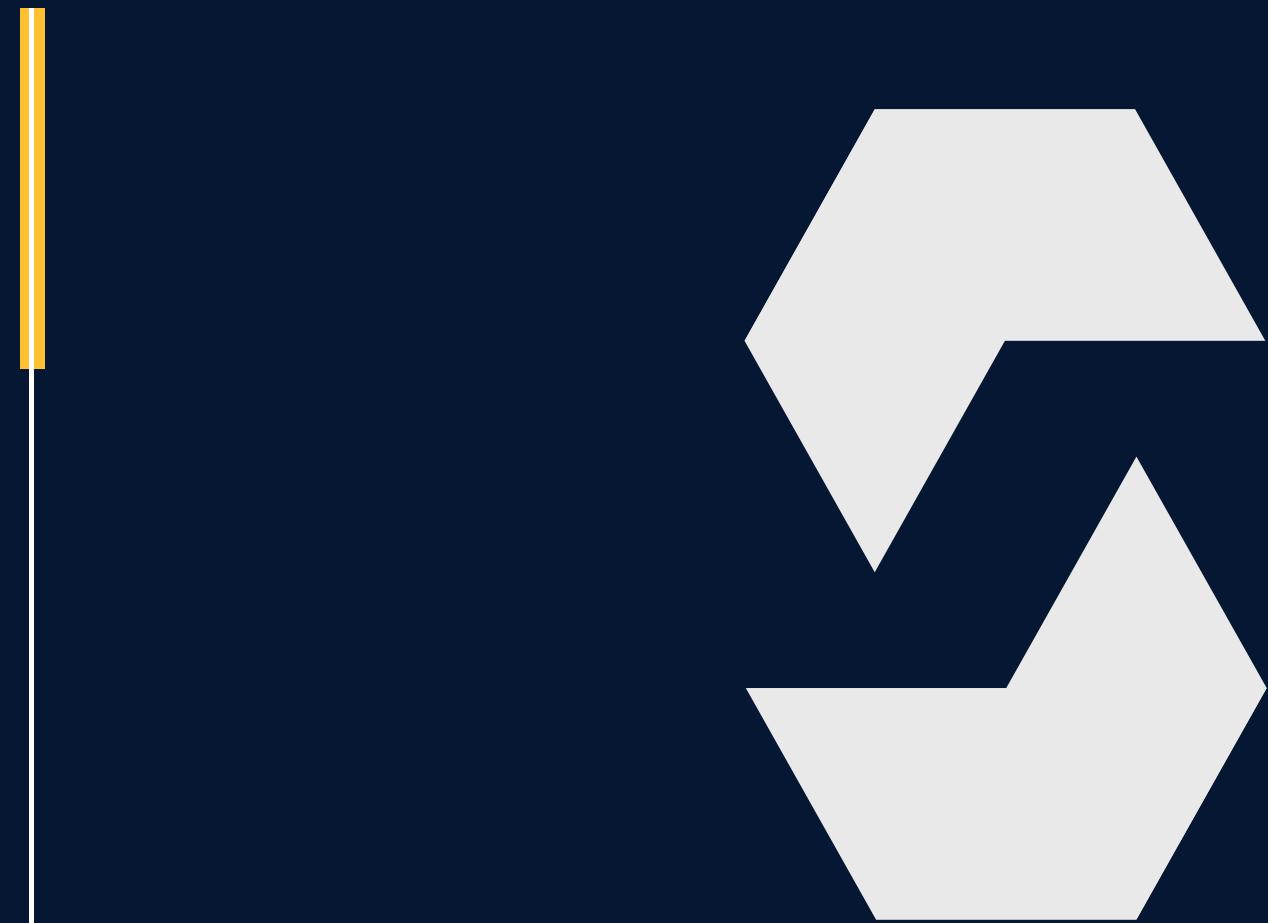


INTRODUCTION TO SOLIDITY

What Is Solidity?

Solidity is a type of object-oriented programming language. This programming language is developed specifically for smart contracts.

This type of language is widely used in creating smart contracts features in blockchain platforms. It's influenced by C++, JavaScript and Python. Solidity also uses Ethereum Virtual Machine to function properly.



...

Concepts You Should Know to Understand Solidity

...

Ethereum

Ethereum is an open-source blockchain platform that offers smart contract facilities. Solidity was first introduced as a new type of programming language for the Ethereum platform.

Ether is the primary token for the platform. This platform is dedicated to developers for helping them develop and deploy decentralized applications.

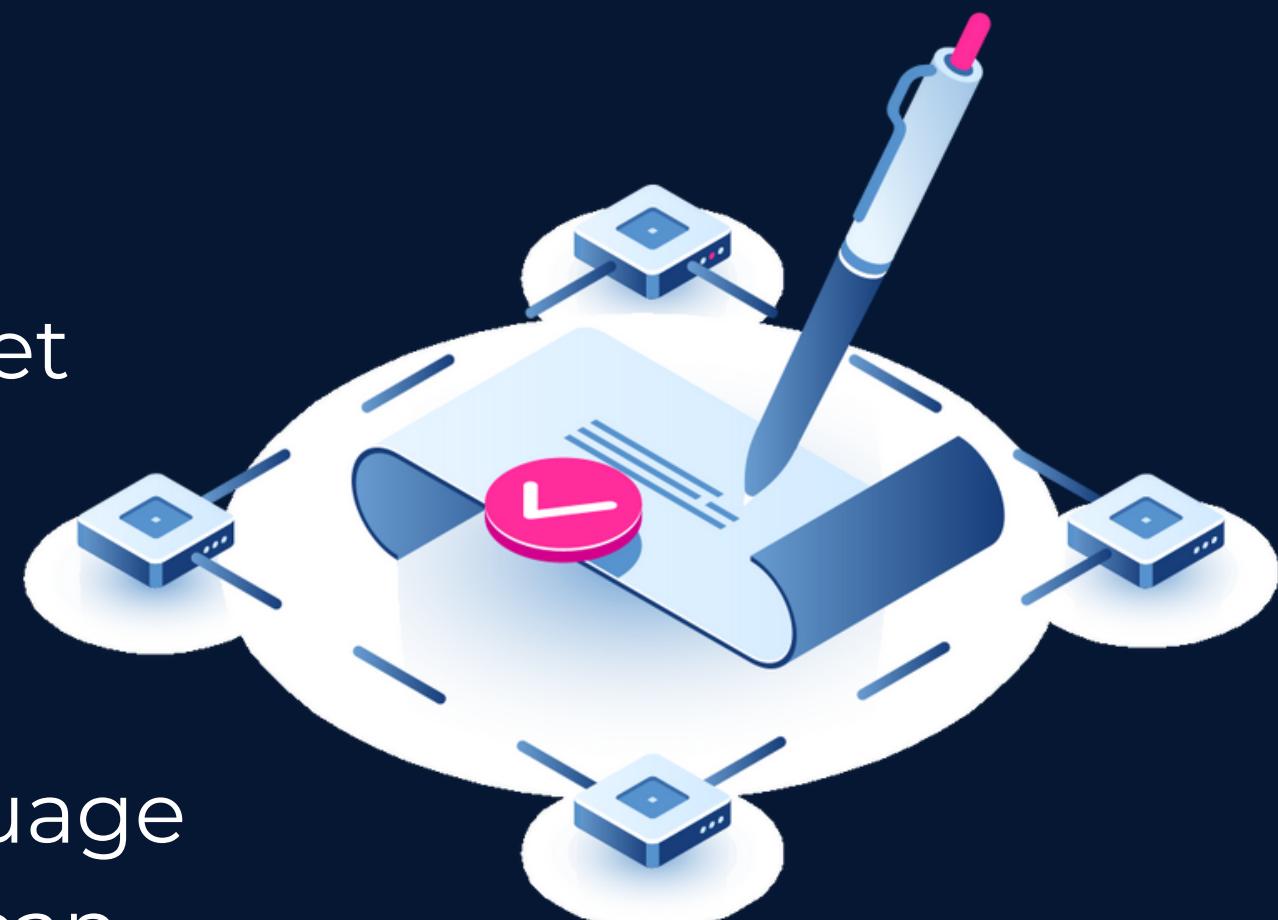


Concepts You Should Know to Understand Solidity

...

Smart Contracts

Smart contracts are a type of automated digital contract. Here, parties within the contract will set up specific rules and based on that rules, the contracts will get executed.



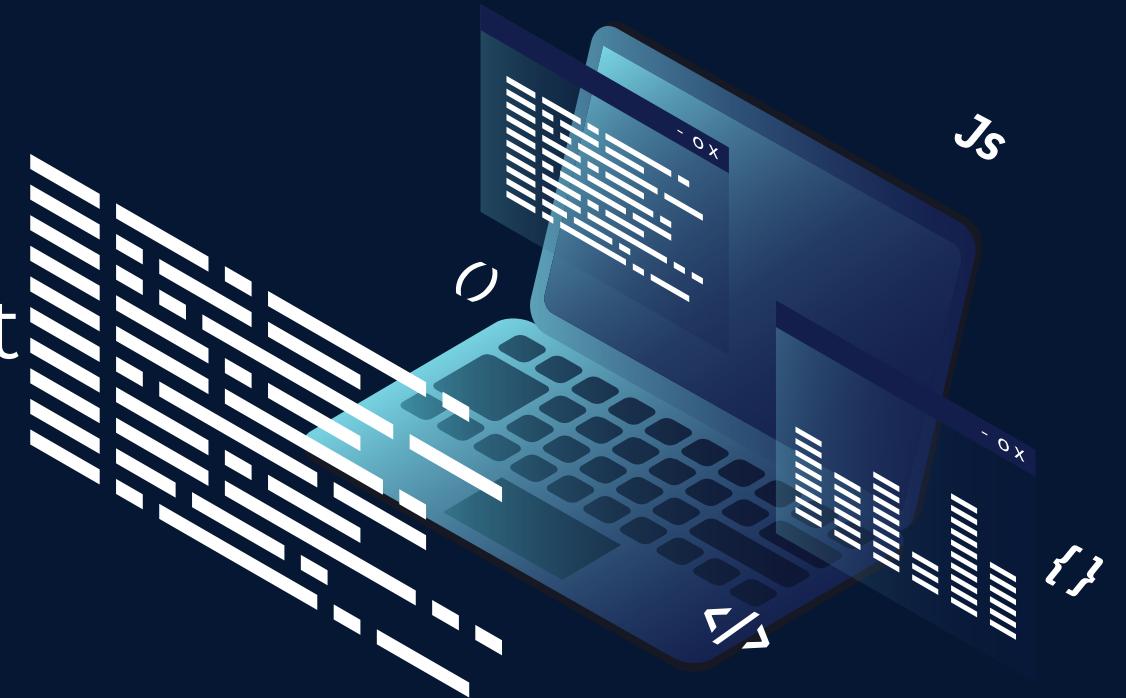
Developers use the Solidity programming language to develop smart contracts. Using Solidity, you can program the contracts to do any type of task.

Concepts You Should Know to Understand Solidity

...

Ethereum Virtual Machine

Ethereum Virtual Machine is a crucial part of understanding Solidity. EVM offers a runtime environment for smart contracts. Using EVM, smart contracts can execute their core objectives with higher security.



It's also very effective in preventing DOS or Denial-of-Service attacks. EVM comes with a specific set of codes or instructions to run specific tasks within the execution environment.

Different Methods of Setting Up Solidity Compiler Environment





-  **Remix:** Remix IDE is an application that provides plugins and a development environment for smart contracts. Users can use this application online without installing any software for the environment.
-  **Node.js/npm:** This is the easiest way to set up the environment. Here, you will install a solidity compiler called solc-js. However, solc-js offer limited functionalities for accessing the compiler.
-  **Docker Image:** Docker image offers simple steps in setting up the environment. Docker images offer a template to build a container for running the Solidity compiler.
-  **Binary Packages:** Binary packages are archive files that will have all directories and files for installing the Solidity compiler on your device.

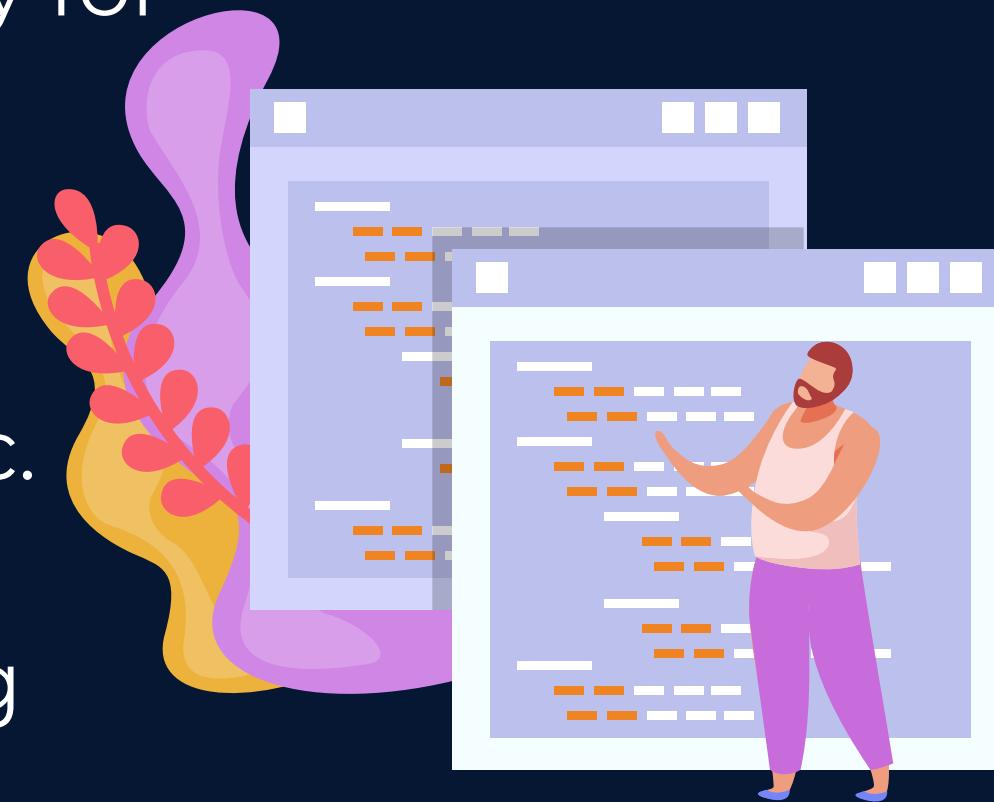
Understanding the Solidity Syntax

- **Pragma:** In Solidity, a pragma language will specify how the compiler will process any type of input. Typically, the first line of code in Solidity based smart contracts contains the pragma.
- **Contract:** uintstoredData component within the code will denote the Solidity contract. This part will contain all the data and code needed for locating a particular address within the blockchain.
- **File Importing:** Solidity offers similar support for file import systems like JavaScript.



Value Types in Solidity

- Variables are only locations that are reserved for storing values. Thus, users can reserve any amount of memory for a certain variable.
- Solidity comes with a vast list of value types, such as Boolean, double floating-point, integers, character, etc.
- The operating system will be responsible for allocating the memory and selecting the data to store for a particular variable.



Data Types

- Unsigned and signed fixed-point numbers varying in sizes
- 8 bits to 256 bits Unsigned integer
- 8 bits to 256 bits Signed integer
- Boolean
- Signed fixed-point number
- Unsigned fixed-point number



...

Variables in Solidity

- **State Variable:** Users can locate the state variables within the contract storage where values are permanently stored.
- **Local Variable:** Users can find the value of any local variables within the defining function. This value is not permanently stored.
- **Global Variables:** Global variables help in fetching data or information from the blockchain platform and any associated transaction processes.



Financial Services



Using Solidity, it's possible to offer smart contracts facilities to financial service providers. It can reduce errors within contracts and help to make the process transparent.

Solidity based smart contracts can also streamline cross-border payments, insurance claims, national bonds, mortgages, etc.

Trade Finance



Solidity smart contracts can help to streamline the issuing of trade finance documentation that takes up a lot of resources and time.

It can help to reduce the risks of error while maintaining a fast execution time.

Data Storage

Smart contracts based on Solidity can act as data storage for valuable information or documents.

Here, Solidity can define the file type and separately store those data. When needed, this type of contract can be called by other contracts for using the stored data.



Digital Identity

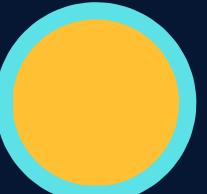
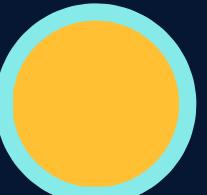
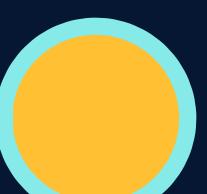
Solidity based smart contracts can help to solve issues related to identity thefts and data monopoly.

These contracts are a good option for storing confidential information, as it allows the user to have full control over the data.





Top Solidity Issues

- | | |
|---|--|
|  Unchecked External Call |  Reentrancy |
|  Costly Loops and Gas Limit |  Clearing Mappings |
|  Arithmetic Precision |  Unexpected Ether |
|  Relying On tx.origin |  Default Visibilities |
|  Overflow and Underflow |  Timestamp Manipulation |

Thank You