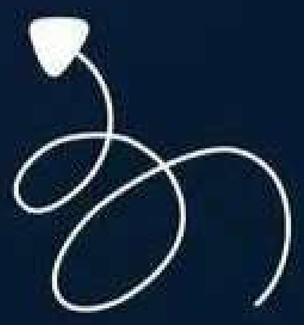
Intro to Solidity Programming Language

Code For Real
@codeforreal

Save

What is Solidity and why is it being so famous?

Give a follow



Code For Real
@codeforreal

Dont’ forget to save it

Save

Solidity is a relatively new programming language developed by **Ethereum**, the second-largest **cryptocurrency market** by capitalization.

It is an object-oriented programming language created specifically for constructing and designing **smart contracts** on **Blockchain** platforms.

The language has lot of similarities with **JavaScript** and **Python** but it is a a **statically typed** language unlike JavaScript and Python though which are **dynamically typed.**

Code For Real
@codeforreal

Save

Contract

Contracts are the fundamental building block of Ethereum applications – they encapsulate all the variables and functions of a smart contract. They're analogous to **classes** in object-oriented languages.

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.14;

contract NFTMarketplace{

  function sayHello() public pure returns (string memory) {
      return "Hello, world!";
  }

}
```

A contract file needs to define **License Identifier** at the top.

A contract file also needs **solidity version** to be predefined with keyword **pragma**.

Code For Real
@codeforreal

Save

Variables

Solidity is statically typed, which means that you must specify the **variable's type during declaration**. There is no concept of "undefined" or "null" in Solidity, which means you must also **initialize** all declared variables with a value based on its type.

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.14;

contract NFTMarketplace{

  uint256 variableUnsignedInt; // default value 0
  string variableString; // default ""
  bool variableBoolean; // default value false
  address variableAddress; //default address(0)

}
```

The **scoping** of variable works exactly like working of variabled in JavaScript.

Code For Real
@codeforreal

Save

Struct, Array & Mapping

Struct allow you to combine multiple pieces of data into a single data type

Mapping stores a single key-value pair datatype.

```
struct Person {
    string name;
    uint age;
    uint gender;
}
```

```
mapping(uint256 => string) varibleMapping;
```

```
uint256[2] fixedArray; // FIxed Array
uint256 dynamicArray; // Dynamic Array
```

If you want **collection of data**, use an array.

Array can be either **fixed** length or **dynamic**.

Code For Real
@codeforreal

Save

Function

A **function** is a group of reusable code which can be called anywhere in your program.

To define a function we should use the **function keyword**, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces.

```
function getResult() public view returns(uint256){
        uint a = 1;
        uint b = 2;
        return a * b;
}
```

A Solidity function can have an optional **return statement.** This is required if you want to return a value from a function.

Code For Real
@codeforreal

Save

Accessibilty

You might be wondering what are those **public**, **private** etc symbols in function and variables?
Well those are called **accessibilty** identifiers.

Public: Is accessible everywhere within single and multiple contracts.

Private: Is accessible only within that contract.

External: Is accessible only from outside like from wallets like Metamask.

Internal: Is accessible only within that contract as well as contracts inherting it.

If none of these is provided, it will be considered **public** by **default**.

Code For Real
@codeforreal

Events

You may have thought to yourself: “But where are the print statements?”

Because smart contracts run on the blockchain, outputting information is a little more complicated than printing like console.log. Instead, we log information using Events.

```
event HelloWorldCalled(address calledBy);

function sayHelloWorld() public payable returns(string){
  // Do some task

  emit HelloWorldCalled(msg.sender);

  return "Hello World";
}
```

You declare an event by using **event** keyword and dispatch an event by using **emit** keyword.

Code For Real
@codeforreal

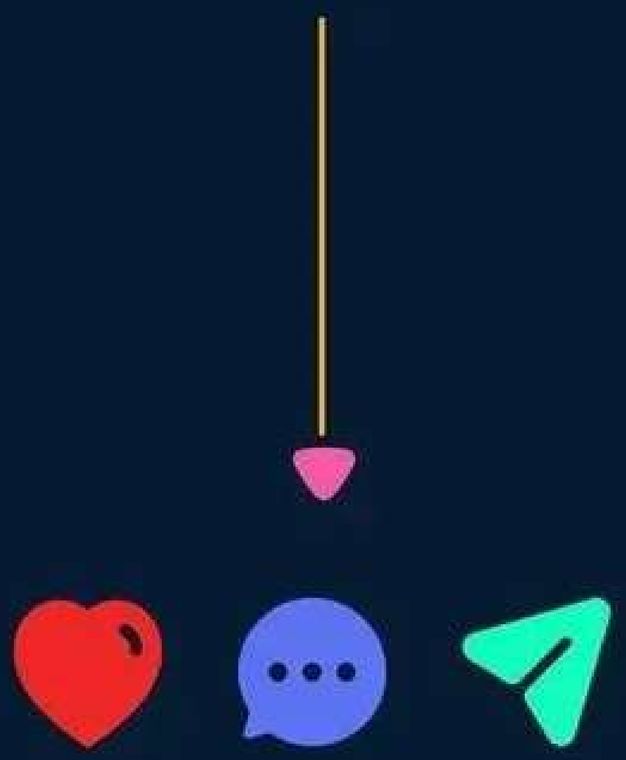
Save

Code For Real

@codeforreal

Did You Like This Post?

Let me know in the comments 👇



Save it