BLOCKCHAIN | Batch 1 - Day 5 | Overview

GitHub Accounts

String

Memory - Keyword (Which is used for storing data type in the memory block)

Msg Objects

Stop, start, pause and destroy a Smart Contract

Maps

Struct

Array

Constructor

Pure

View

Solidity - Strings:

Solidity supports String literal using both double quote (") and single quote ('). It provides string as a data type to declare a variable of type String.

```
pragma solidity >=0.5.13 <0.7.3;


contract test {
   string data = "LetsUpgrade";
}
```

In the above example, "LetsUpgrade" is a string literal and data is a string variable. The more preferred way is to use byte types instead of String as string operation requires more gas as compared to byte operation. Solidity provides inbuilt conversion between bytes to string and vice versa. In Solidity, we can assign String literal to a byte32 type variable easily. Solidity considers it as a byte32 literal.

```
pragma solidity >=0.5.13 <0.7.3;


contract test {
   bytes32 data = "LetsUpgrade";
}
```

Msg Objects:

Msg objects are like global inbuilt variables or functions.

msg.gas (uint): To get the remaining gas.

msg.sig (bytes4): first four bytes of the call data (i.e. function identifier)

msg.value (uint): number of Wei sent with the message.

msg.data (bytes): complete call data.

msg.sender: function indicated the sender of the current message or (current call).

Structs:

Defining a Struct,

To define a Struct, you must use the struct keyword. The struct keyword defines a new data type, with more than one member. The format of the struct statement is as follows –

```
struct struct_name {
   type1 type_name_1;
   type2 type_name_2;
   type3 type_name_3;
}struct_name membershipVariables;
```

constructor :

A constructor is a special function declared using constructor keyword. It is an optional function and is used to initialize state variables of a contract. The following are the key characteristics of a constructor.

A contract can have only one constructor.

A constructor code is executed once when a contract is created and it is used to initialize the contract state.

After a constructor code executed, the final code is deployed to the blockchain. This code includes public functions and code reachable through public functions. Constructor code or any internal method used only by constructor is not included in the final code.

A constructor can be either public or internal.

An internal constructor marks the contract as an abstract.

In case, no constructor is defined, a default constructor is present in the contract.

```
pragma solidity >=0.5.13 <0.7.3;

contract Test {
  constructor() public {}
}
contract Base {
  uint data;
  constructor(uint _data) public {
    data = _data;
// We are constructing the data by using _data in public
  }
```

Maps or Mapping :

Mapping is a reference type as arrays and structs. Following is the syntax to declare a mapping type.

```
mapping(_KeyType => _ValueType)
```

Where,

```
}
```

_KeyType − can be any built-in types plus bytes and string. No reference type or complex objects are allowed.

_ValueType − can be any type.

```
pragma solidity >=0.5.13 <0.7.3;


contract charitySmartContract{
    address public owner;


    mapping(address => uint) public addressToEtherMap; // Here we are pointing out the address of the owner to the unit variable(valuetype).
```

Pure & view:

Pure:

Functions set to pure should not modify or read from the state. It prevents pure functions from:

Calling functions that are not marked.

Using inline assembly, containing specific opcodes.

Reading from state variables.

Accessing address(this).balance or <address>.balance.

Accessing members of the block, tx, msg (except msg.sig and msg.data).

```
pragma solidity >=0.5.0 <0.7.0;


contract C {
    function f(uint a, uint b) public pure returns (uint) {
        return a * (b + 42);
    }
}
```

View:

Functions set to view do not change the state. These actions indicate a modification of the state:

Creating other contracts

Making Solidity transfer Ether through calls

Writing to state variables

Removing events

```
pragma solidity >=0.5.0 <0.7.0;


contract C {
    function f(uint a, uint b) public view returns (uint) {
        return a * (b + 42) + now;
    }
}
```

Github:

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

Step 1: Create your own account by your Gmail and unique password

Step 2: Create a GitHub repository.

A repository is nothing but a directory or storage space where your projects and code can get stored and collaborate with your co-mates.