Solidity by Example

version 0.8.3

Constructor

A `constructor` is an optional function that is executed upon contract creation.

Here are examples of how to pass arguments to `constructors`.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

// Base contract X
contract X {
    string public name;

    constructor(string memory _name) {
        name = _name;
    }
}

// Base contract Y
contract Y {
    string public text;

    constructor(string memory _text) {
        text = _text;
    }
}

// There are 2 ways to initialize parent contract with parameters.

// Pass the parameters here in the inheritance list.
contract B is X("Input to X"), Y("Input to Y") {

}
```

```
contract C is X, Y {
    // Pass the parameters here in the constructor,
    // similar to function modifiers.

    constructor(string memory _name, string memory _text) X(_name) Y(_t
}

// Parent constructors are always called in the order of inheritance
// regardless of the order of parent contracts listed in the
// constructor of the child contract.

// Order of constructors called:
// 1. Y
// 2. X
// 3. D
contract D is X, Y {
    constructor() X("X was called") Y("Y was called") {}
}

// Order of constructors called:
// 1. Y
// 2. X
// 3. E
contract E is X, Y {
    constructor() Y("Y was called") X("X was called") {}
}
```

Try on Remix

Take a course at Smart Contract Engineer

Smart Contract Programmer

Donate☺

0x95a6...fa243

your ETH will directly go to Khan Academy

contact@solidity-by-example.org

source | license