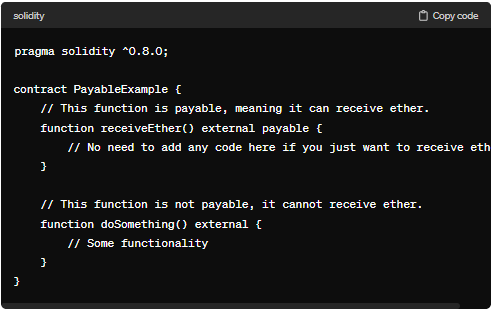
**Payable:**

In Solidity, the **payable** modifier is used to indicate that a function can receive ether (the cryptocurrency used on the Ethereum blockchain) along with a transaction.



When a payable function is called, the sender can include ether with the transaction, and this ether will be sent to the contract and will be accessible within the function through the **msg.value** variable.

**Keypoints:**

1) This is used to make address payable.

2) This is used to make function payable.

If we make any function payable in any contract then we can transfer ether in that contract.

As its payable we cannot use pure or view in case of function as we are doing changes in the Blockchain.

In case of function we can write payable after public also but in case of address we need to use payable before public.

Difference between why we are making function payable not only constructor, as constructor run’s once at the time of deployment but we can function many time.

**Red: payable**

**Blue: view, pure, state**

**Yellow: Simple transactional function**

**Code:**

//SPDX-License-Identifier: GPL-3.0

pragma solidity ^0.8.0;

contract Payable{

    address payable public user = payable(msg.sender);

    constructor() payable{

    }

    function newETH() payable public{

    }

    function checkAmount() public view returns(uint){

        return address(this).balance;

    }

}

**Output:**

