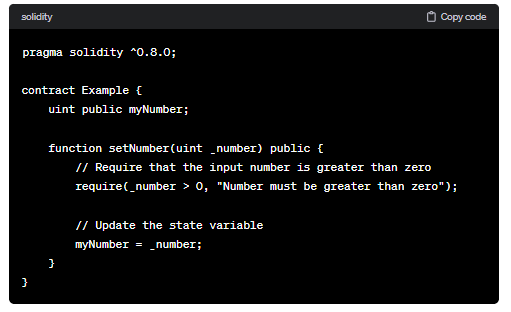
**Require:**

In Solidity, the **require** statement is a powerful tool used to validate conditions and ensure that certain requirements are met before proceeding with the execution of a function. It's commonly used as a guard clause to check inputs, state conditions, or other requirements that must be satisfied for the contract's logic to proceed.





Using **require** statements helps to enforce constraints and make your contracts more robust by preventing invalid state transitions and inputs. It's an essential tool for writing secure and reliable smart contracts.

We can use the same in case of Error Handling.

1) Use for input validation

2) for access control

**Advantages:**

1) Return the gas (remaining gas refund)

2) Run a function in it if we have a “require” statement. If we are changing doing in state variables. In the case of the required statement, this state variable is false. So whatever the changes did in the state variable of the function is reverted.

**Code:**

//SPDX-License-Identifier: GPL-3.0

pragma solidity ^0.8.0;

contract Require{

    address public newowner = msg.sender;

    uint public age =25;

    function testrequire(uint \_x) public {

        age = age+5;

        require(\_x>5,"value of x is lesser than 5");

    }

    function testOwner() public {

        require(newowner==msg.sender,"welcome you are the owner");

        age = age-2;

    }

}

**Output:**

