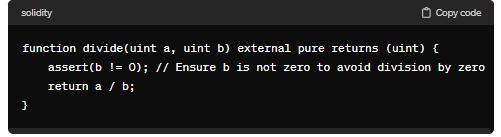
**Revert and Assert:**

Mainly used for error handling.

In Solidity, **revert** and **assert** are two mechanisms used for handling errors and unexpected conditions in smart contracts. While they might seem similar, they serve slightly different purposes:

**Revert**: The **revert** keyword is used to revert the current transaction due to an error condition or to revert the state changes made within a contract function. It allows you to provide a custom error message that explains why the transaction is reverting. Reverting is typically used for expected conditions where it's acceptable for the transaction to fail without causing any harm to the blockchain state.

**Assert**: The **assert** keyword is used to ensure that a condition is always true. If the condition evaluates to **false**, the transaction is reverted, and any state changes are rolled back. **Assert** is used to check for internal errors or invariant violations that should never occur if the contract is functioning correctly.



In summary, **revert** is used to handle expected conditions and provide custom error messages, while **assert** is used to enforce internal consistency and invariant conditions within the contract. Both are important for writing secure and robust smart contracts.

The key difference between **require** and **revert** lies in their intended use cases and semantics. **require** is typically used for input validation and to enforce preconditions, while **revert** is more versatile and can be used for any situation where you need to revert the transaction with a custom error message or handle exceptional conditions. Additionally, **require** is often used for conditions that are expected to hold true in normal circumstances, while **revert** is used for exceptional or unexpected conditions.

Use of assert -> is used for 2 thing: Bug identification and Security

**Advantages:**

1) This do the same thing while returning the unused gas.

2) Undo the value of state variable if it change.

Revert having power of custom error.

**Code:**

//SPDX-License-Identifier: GPL-3.0

pragma solidity ^0.8.0;

contract Revert\_assert{

    address public holder = msg.sender;

    uint public age=25;

    error gettingerror(string,address);

    function Revert(uint \_x) public {

        age = age+5;

        if(\_x<2){

            revert gettingerror("value of x is less than 2",msg.sender);

        }

    }

    function onlyholder() public {

        if(holder !=msg.sender){

            revert("you are note the owner");

        }

        age = age-2;

    }

    //Assert

    function checkOwner() public view{

        assert(holder==0x5B38Da6a701c568545dCfcB03FcB875f56beddC4);

    }

}

**Output:**

