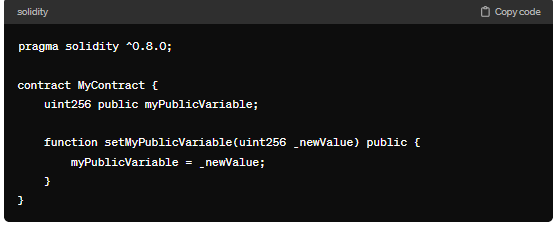
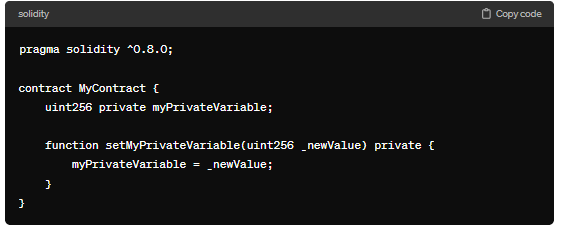
**(Public, Private, Internal, External):**

In Solidity, visibility modifiers control how functions and state variables can be accessed by other contracts or accounts. There are four visibility levels:

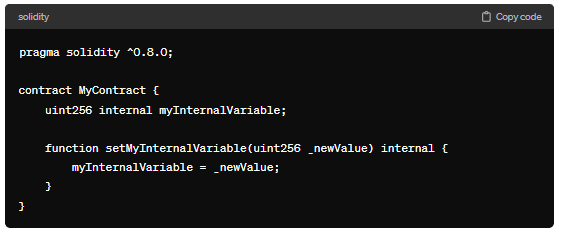
**Public**: Functions and state variables declared as public can be accessed from outside the contract as well as from within the contract and its derived contracts. When you declare a state variable as public, Solidity automatically creates a getter function with the same name as the variable.



**Private**: Functions and state variables declared as private can only be accessed from within the contract that defines them. They are not visible to external contracts or accounts.

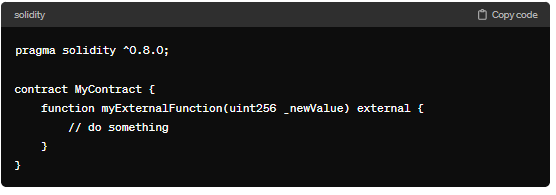


**Internal**: Functions and state variables declared as internal can only be accessed from within the contract and its derived contracts. They are not visible to external contracts or accounts.

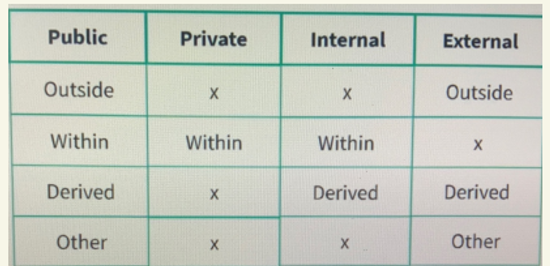


**External**: Functions declared as external can only be accessed from outside the contract. They cannot be called internally, meaning they cannot be called by the contract itself or its derived contracts.

Example:



These visibility modifiers provide flexibility in managing access to functions and state variables within Solidity contracts, ensuring proper encapsulation and security.



**Visibility:**

It is used for:

1) State variable

2) Function

Visibility tell us for above one who is performing the call / access.

4 visibility are -> (Public, Private, Internal, External)

Potential Caller are of 4 type:

Gas increased -> (Private (least) -> Internal(less) -> External (more) -> Public (high)

**Private:**

1) MyContract -> contract call itself.

2) DerivedContract is MyContract -> in this derived contract call function. Or state variable from parent contract.

3) Another Contract -> 2 contract in one file can use the state variable and functions of another contract. 4) Outside World: Remix id (where button created).

**Internal:**

1) MyContract-> within the contract

2) DerivedContract is Mycontract -> as child contract -> can access state variable or function of parent variable.

3) AnotherContract -> not applicable

4) Outside world -> not applicable

**External:**

(State variable we cannot use for external it’s only for function and it cannot be use for My contract and Derived contract.)

1) MyContract -> no

2) DerivedContract is MyContract-> no

3) Another Contract -> yes

4) Outside World -> yes

**Public:**

1) MyContract -> yes

2) DerivedContract is MyContract-> yes

3) Another Contract -> yes

4) Outside World -> yes

By Default => State variable (Internal) And Function (Public). If we are not defining anything in case of State variable then it consider as Internal. But in case of Function. If we are not defining anything then it give us an error.

**Code**

//SPDX-License-Identifier: GPL-3.0

pragma solidity ^0.8.0;

contract A {

    function f1() public pure returns(uint)

    {

        return 1;

    }

    function f2() private pure returns(uint)

    {

        return 2;

    }

    function f3() internal pure returns(uint)

    {

        //uint x=f1();

        return 3;

    }

    function f4() external pure returns(uint)

    {

        return 4;

    }

}

contract B is A

{

    uint public bx=f3();

}

contract C

{

    A obj= new A();

    uint public cx=obj.f4();

}

**Output:**

