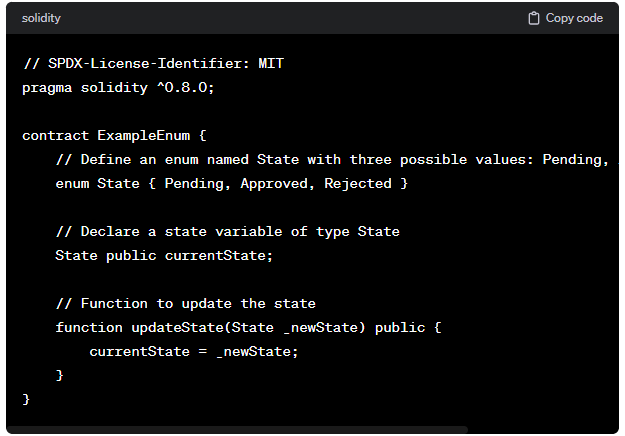
**Enum:**

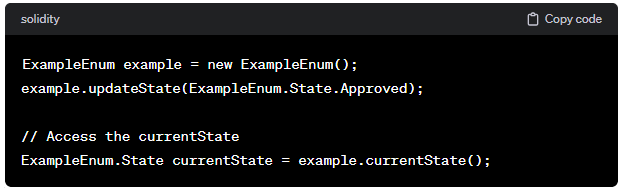
Enum, or enumerations, in Solidity are a way to create a user-defined data type with a finite set of possible values. Enum are useful when you want to represent a set of discrete, distinct values, typically used to define states or categories.

In Solidity, you can declare an enum using the **enum** keyword.

It's a user define data types.



You can access enum values using the dot notation.



Enum are commonly used in smart contracts to represent different states of a contract or to define options for certain parameters. They provide a clear and readable way to work with a predefined set of values.

**Code:**

//SPDX-License-Identifier: GPL-3.0

pragma solidity ^0.8.0;

contract Enum{

    enum Status{

        Pending,

        Shipped,

        Accepted,

        Rejected,

        Cancel

    }

    Status status;

    function getStatus() public view returns(Status){

        return status;

    }

    function setstatus(Status \_status) public {

        status = \_status;

    }

    function rej() public{

        status = Status.Rejected;

    }

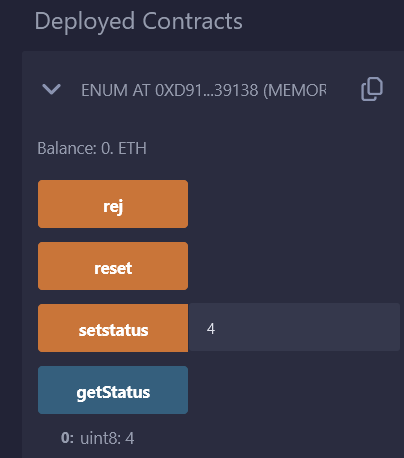
    function reset() public{

        delete status;

    }

}

**Output:**



//SPDX-License-Identifier: GPL-3.0

pragma solidity ^0.8.0;

contract Enum{

    enum user{allowed, not\_allowed,wait}

    user public u1= user.allowed;

    uint public lottery=1000;

    function owner() public {

        if(u1==user.allowed)

        {

            lottery=0;

        }

    }

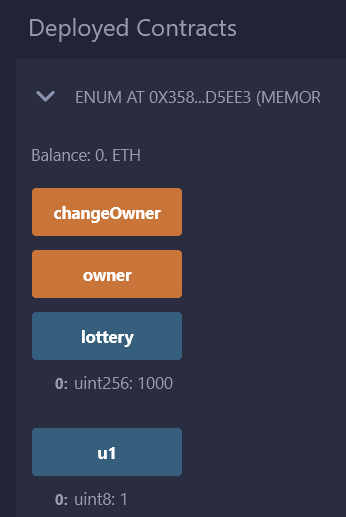
    function changeOwner() public{

        u1=user.not\_allowed;

    }

}

**Output:**



Enum can be used inside the contract and Outside the contract.