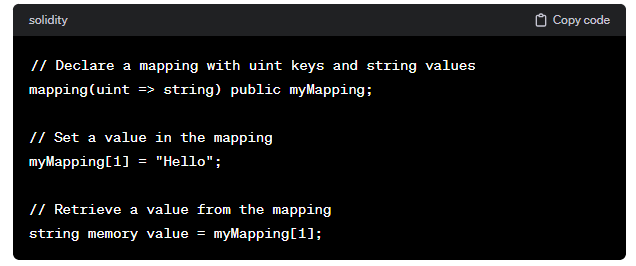
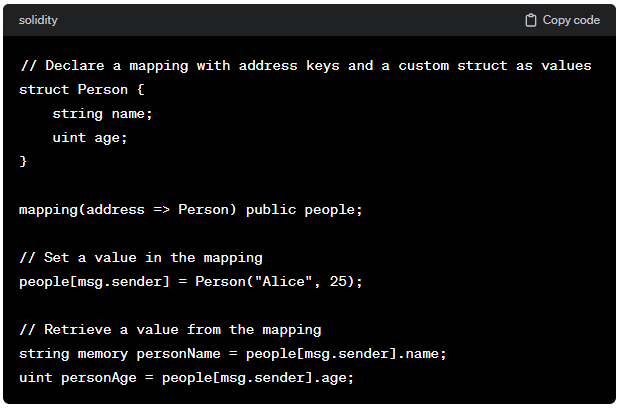
**(Mapping in Solidity | Solidity Latest Version):**

In Solidity, mapping is a data structure used to store key-value pairs, somewhat similar to dictionaries or hash maps in other programming languages.



Mappings can be used with various data types as keys and values, including other mappings or custom data structures. It's important to note that mappings in Solidity are not iterable, meaning you cannot directly loop through all the keys or values in a mapping. If you need to iterate over the keys or values, you might need to maintain an additional data structure like an array or use a different approach depending on your specific use case.



* Concept of keys and values.
* Mapping (key=> value).

**Code:**

//SPDX-License-Identifier: GPL-3.0

pragma solidity 0.8.0;

contract demo{

    mapping(uint=>string) public roll\_no;

    function setter(uint keys,string memory value) public

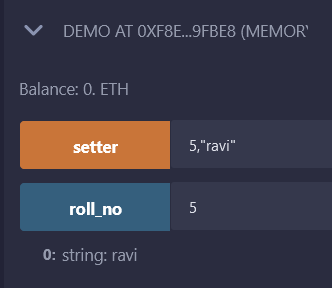
    {

        roll\_no[keys]=value;

    }

}

**Output:**



//SPDX-License-Identifier: GPL-3.0

pragma solidity 0.8.0;

contract Mapping{

    mapping(uint=>string) public emp\_id;

    function setIds() public {

        emp\_id[31]="gaurav";

        emp\_id[27]="aman";

        emp\_id[33]="sadqwd";

        emp\_id[20]="amdrewe";

        emp\_id[14]="dqsasas";

    }

    function getId(uint \_id) public view returns(string memory){

        return emp\_id[\_id];

    }

}



**Now the code for advance mapping**

Here we are trying to add older donation + new donation so finally we get the updated amount.

//SPDX-License-Identifier: GPL-3.0

pragma solidity 0.8.0;

struct donor\_dts{

    string name;

    uint age;

    string add;

    uint don;

}

contract advMapping{

    mapping(address=>donor\_dts) public acc\_info;

    function set(string memory \_name, uint \_age, string memory \_add, uint \_don) public{

        acc\_info[msg.sender] = donor\_dts(\_name,\_age,\_add,acc\_info[msg.sender].don+\_don);

    }

    function delete\_Info() public{

        delete acc\_info[msg.sender];

    }

}

**Output:**

