

What is Nested Mapping?

A **nested mapping** means a mapping *inside another mapping*

Simple Example: Student Marks

We will store **each student's marks for each subject** using nested mapping.

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract NestedMappingExample {

    // Nested mapping
    // student => (subject => marks)
    mapping(address => mapping(string => uint)) public marks;

    // Set marks for a student in a subject
    function setMarks(string memory subject, uint score) public {
        marks[msg.sender][subject] = score;
    }

    // Get marks of a student in a subject
    function getMarks(address student, string memory subject)
        public
        view
        returns (uint)
    {
        return marks[student][subject];
    }
}
```

The screenshot shows the REMIX IDE interface version 1.3.0. On the left, there's a sidebar with various icons for deployment, transactions, and contracts. The main area is divided into two sections: "DEPLOY & RUN TRANSACTIONS" and the "Contract Editor".

Contract Editor:

- Compile dropdown:** Set to "Compile".
- Search bar:** Contains "tpp.sol".
- Code:** Displays the Solidity code for "NestedMappingExample".

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract NestedMappingExample {
    // Nested mapping
    // student => (subject => marks)
    mapping(address => mapping(string => uint)) public marks;

    // Set marks for a student in a subject
    function setMarks(string memory subject, uint score) public {
        marks[msg.sender][subject] = score;
    }

    // Get marks of a student in a subject
    function getMarks(address student, string memory subject) public view returns (uint) {
        return marks[student][subject];
    }
}
```

Deployed Contracts:

- A deployed contract named "NESTEDMAPPINGEXAMPLE AT" is listed.
- Balance:** 0 ETH.
- setMarks:** A transaction is shown with "math95" as the subject and a value of 95.
- GETMARKS:** A function call is shown with "student: 0x5B38Da6a701c568545dCfcB03F" and "subject: math".
- Call data:** Shows the ABI encoded data for the call.
- Parameters:** Shows the parameters for the call.
- call button:** Allows the user to execute the function call.
- Output:** Shows the result of the call: "0: uint256: 95" and "marks: 0x5B38Da6a701c568545dC".