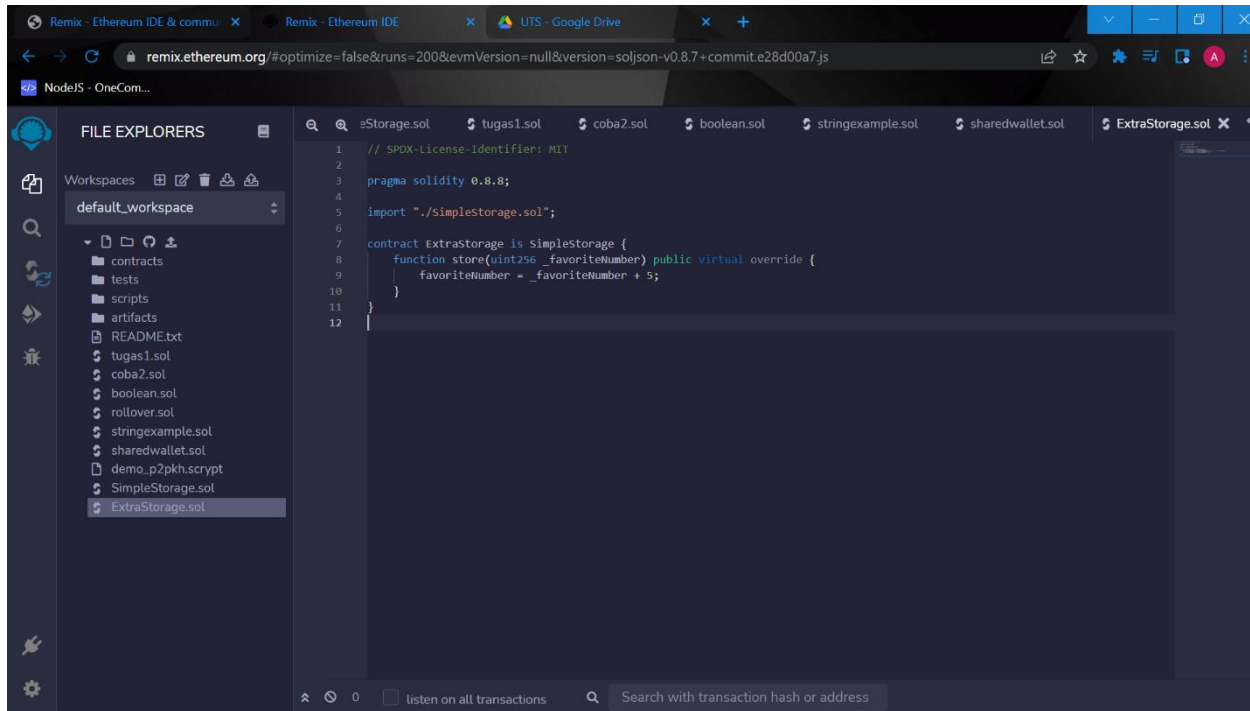
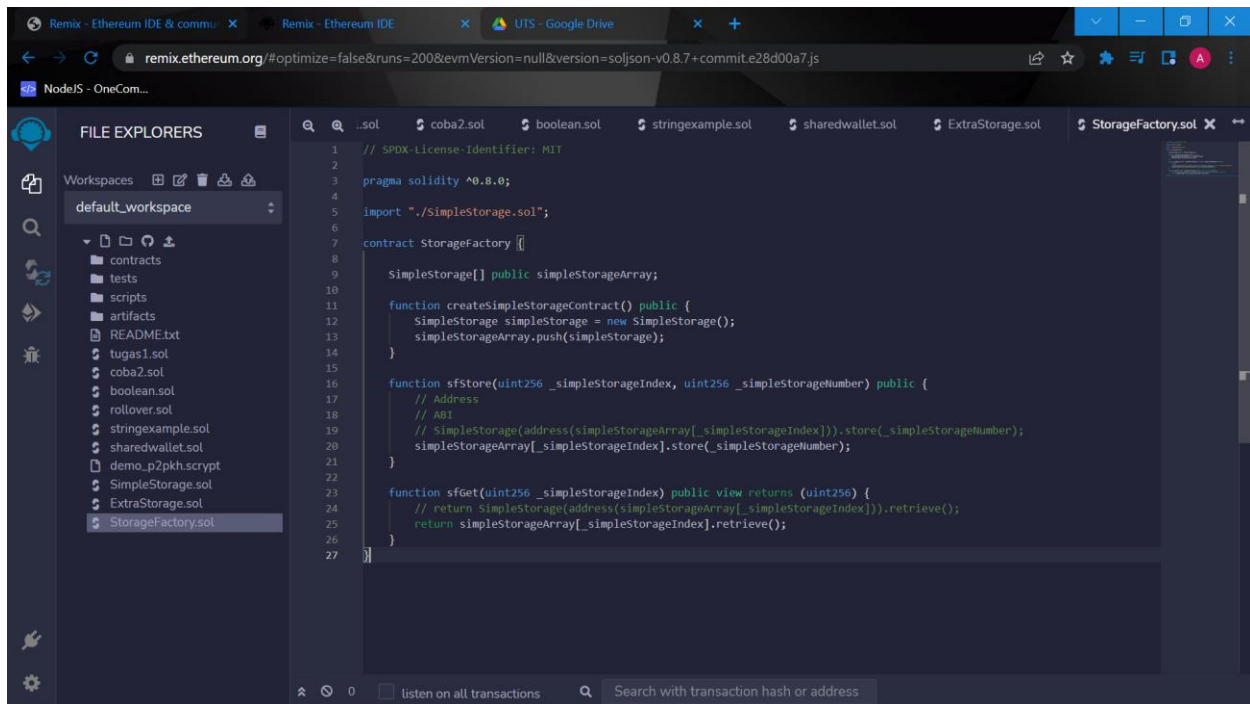


Storage Factory



The screenshot shows the Remix IDE interface with the 'ExtraStorage.sol' file open. The left sidebar displays the 'FILE EXPLORERS' panel with a tree view of the project files, including 'contracts', 'tests', 'scripts', 'artifacts', and various Solidity files. The main editor area shows the code for 'ExtraStorage.sol'.

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity 0.8.8;
4
5 import "./SimpleStorage.sol";
6
7 contract ExtraStorage is SimpleStorage {
8     function store(uint256 _favoriteNumber) public virtual override {
9         _favoriteNumber = _favoriteNumber + 5;
10    }
11 }
12
```



The screenshot shows the Remix IDE interface with the 'StorageFactory.sol' file open. The left sidebar displays the 'FILE EXPLORERS' panel with a tree view of the project files, including 'contracts', 'tests', 'scripts', 'artifacts', and various Solidity files. The main editor area shows the code for 'StorageFactory.sol'.

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.0;
4
5 import "./SimpleStorage.sol";
6
7 contract StorageFactory {
8
9     SimpleStorage[] public simpleStorageArray;
10
11     function createSimpleStorageContract() public {
12         SimpleStorage simpleStorage = new SimpleStorage();
13         simpleStorageArray.push(simpleStorage);
14     }
15
16     function sfStore(uint256 _simpleStorageIndex, uint256 _simpleStorageNumber) public {
17         // Address
18         // ABI
19         SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).store(_simpleStorageNumber);
20         simpleStorageArray[_simpleStorageIndex].store(_simpleStorageNumber);
21     }
22
23     function sfGet(uint256 _simpleStorageIndex) public view returns (uint256) {
24         // return SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).retrieve();
25         return simpleStorageArray[_simpleStorageIndex].retrieve();
26     }
27 }
```