

 Hello Solidity – Writing First Smart Contract

**Objective/Aim:**  
  
 The objective is to write and deploy a basic smart contract to a testnet using the Remix IDE and a MetaMask wallet. We will then interact with the live contract and use Etherscan to observe and verify the resulting on-chain transactions.

**Apparatus/Software Used:**

* Laptop/PC
* Remix.ide
* A blockchain explorer (e.g., Etherscan.io)
* Extention Wallet (MetaMask)

**Theory/Concept:**

Code & Compile in Remix IDE

* Writing the Code: Use Remix as your browser-based editor to write the smart contract. This includes the pragma, contract, state variable (string public message), constructor, and an updateMessage function.
* Compilation: Use the "Solidity Compiler" tab in Remix to compile your code. This critical step turns your human-readable Solidity code into machine-readable Bytecode (the contract's logic) and ABI (the menu of functions).

Deploy via MetaMask Wallet

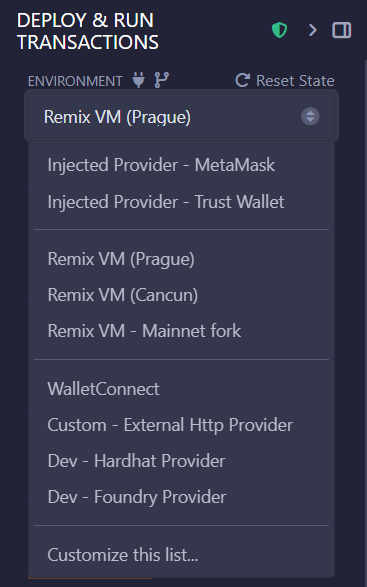
* Connect Your Wallet: In the "Deploy & Run Transactions" tab in Remix, change the environment from "Remix VM" to "Injected Provider - MetaMask". This connects the IDE to your personal wallet.
* Sign & Pay Gas: Click "Deploy". MetaMask will pop up, asking you to confirm the transaction. You are now signing to authorize the deployment and agreeing to pay the necessary gas fee from your wallet to put the contract on the blockchain (e.g., a testnet like Sepolia).

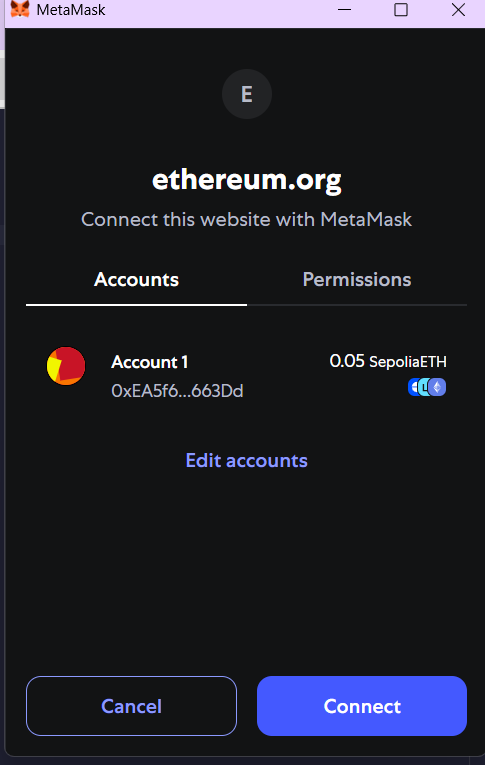
Verify with Etherscan

* Interact in Remix: Once deployed, your contract appears at the bottom of the Remix panel.
* Verify on Etherscan: Copy the deployed contract's address from Remix or MetaMask. Paste it into a blockchain explorer like Etherscan. You will see public proof of your deployment and a complete history of all interactions (like the updateMessage calls) you have made with your contract.



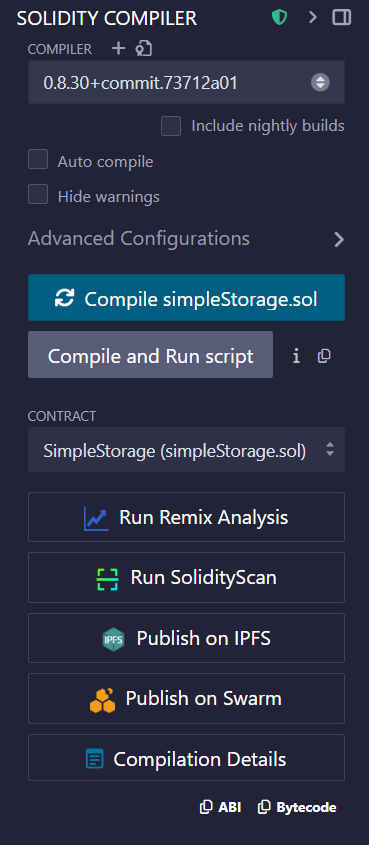
**Procedure:**

Writing the SimpleStorage smart contract with its variables and functions in the Remix code editor 

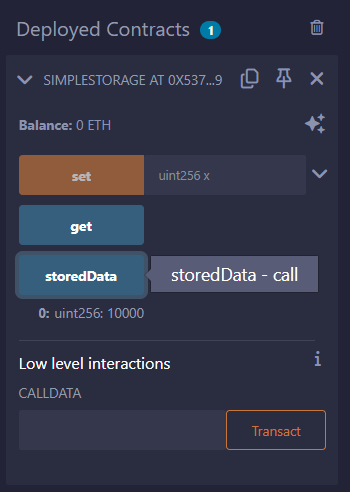
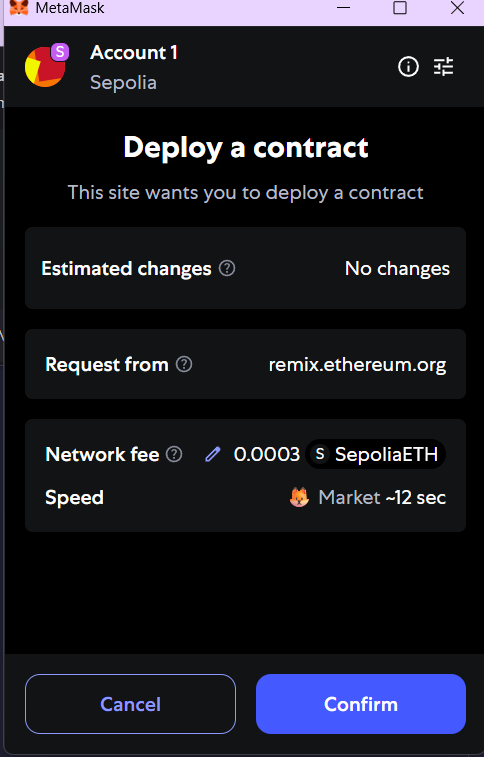
Selecting "Injected Provider - MetaMask" to prepare for deploying the contract using an external wallet

Approving the connection request from the Remix website to your MetaMask wallet.

.



Compiling the SimpleStorage.sol contract using the Solidity Compiler tab in Remix.

Confirming the contract deployment transaction and its gas fee of 0.0003 SepoliaETH in MetaMask. 

Interacting with the live deployed contract's set and get functions through the Remix UI.

**Observation Table:**

we wrote and compiled a basic SimpleStorage contract in the Remix IDE. To get it onto a live testnet, we switched Remix’s environment to connect with our MetaMask wallet.

MetaMask popped up twice—first to get permission to connect, and then again to have us confirm the deployment. This final confirmation showed us exactly what we were paying, a gas fee of 0.0003 SepoliaETH. Once it was live, the contract appeared in Remix with a simple interface: a blue button to read our data for free, and an orange button to write a new value, which would require another gas-paying transaction.