

# Lab 04-07-2023

## 1.1 What is Solidity?

Solidity is an object-oriented programming language specifically designed for creating smart contracts on the Ethereum Virtual Machine (EVM).

It is a vital skill for individuals interested in working on blockchain projects and engaging with the Ethereum ecosystem.



Understanding Solidity allows developers to participate in decentralized applications (DApps), build smart contracts, and interact with the Ethereum blockchain.

## 1.2 Features of Solidity:

- **Object-oriented:** Enables modular and reusable code structure.
- **Smart Contract Development:** Designed for Ethereum smart contracts.
- **Industry Relevance:** Vital skill in the blockchain industry, particularly in Ethereum.
- **Open-Source Analysis:** Helps identify scams and ensure project integrity.
- **Professional Opportunities:** Offers lucrative ventures and Ethereum ecosystem integration.
- **Decentralized Application Development:** Supports building decentralized applications.

## 1.3 How to Run Solidity Script

To run a Solidity script, you can use the following methods:

- 1) **Remix IDE:** Use the online Remix IDE to write, compile, and deploy Solidity contracts.
- 2) **Ethereum Development Frameworks:** Utilize frameworks like Truffle or Hardhat for developing, testing, and deploying Solidity contracts.
- 3) **Command-Line Tools:** Compile and execute Solidity scripts using the solc Solidity compiler and tools like Ganache or web3.js.
- 4) **Online Ethereum Sandboxes:** Explore online platforms such as Ethereum Remix IDE or Ethereum Studio for writing and running Solidity scripts.
- 5) **Ethereum Development Networks:** Connect to Ethereum test networks like Ropsten or Rinkeby to deploy and test Solidity contracts.

## 1.4 Additional:

Latest Solidity Version: **v0.8.20**

### 2.1 Remix IDE:

Remix is a recommended smart contract development IDE (Integrated Development Environment) endorsed by Ethereum. It is particularly suitable for beginners as it enables swift deployment and testing of smart contracts directly in the browser, eliminating the need for local program installations.

### 2.2 Characteristics:

Remix IDE is preferred for Solidity development due to its:

- ✓ **Accessibility:** Web-based IDE accessible from any browser.
- ✓ **User-Friendly Interface:** Intuitive interface with code editor and error checking.

- ✓ **Integrated Environment:** Comprehensive tools for writing, compiling, and deploying contracts.
- ✓ **Real-Time Compilation and Debugging:** Instant feedback and debugging features.
- ✓ **Robust Toolset:** Linter, code analyzer, and test framework integration.
- ✓ **Ethereum Network Integration:** Seamless connection to Ethereum networks for deployment and testing.

Remix IDE combines convenience, user-friendliness, and a range of features, making it a popular choice for Solidity development.

## 2.3 Website Link:

Open your web browser and navigate to the Remix IDE website at <https://remix.ethereum.org/>.

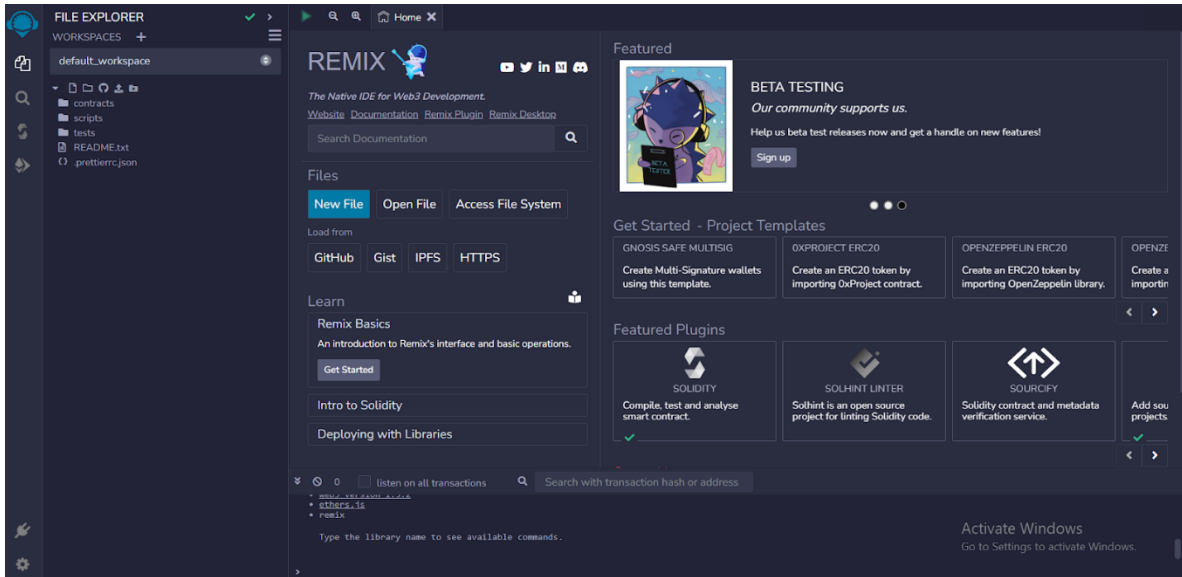
## 2.4 Additional:

- Current version : **v0.34.0**
- You can install it in your own system using this link <https://github.com/ethereum/remix-ide>
- For complete documentation, visit <https://remix-ide.readthedocs.io/en/latest/>

## 3.1 Getting Start:

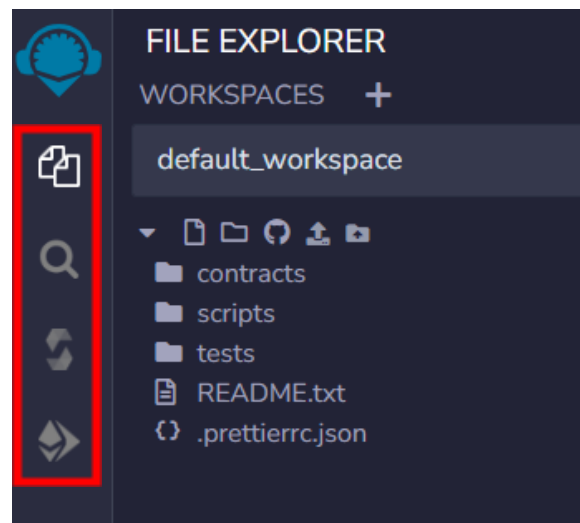
- 1) **Open Remix:** Using your web browser, navigate to the Remix IDE website at <https://remix.ethereum.org/>

Website looks something like this,

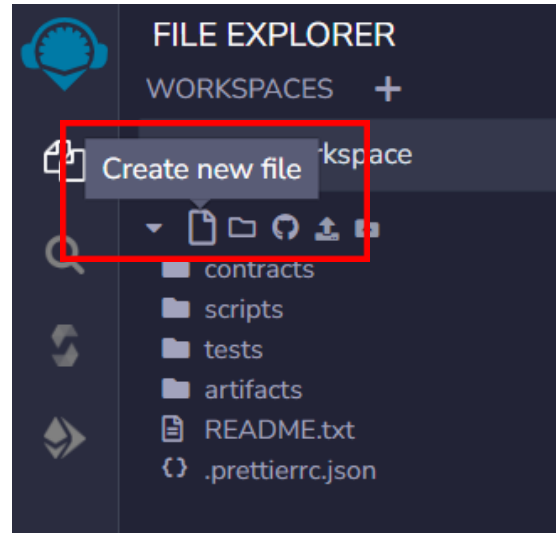


Upon accessing Remix, you'll notice a vertical menu on the left side with four buttons:

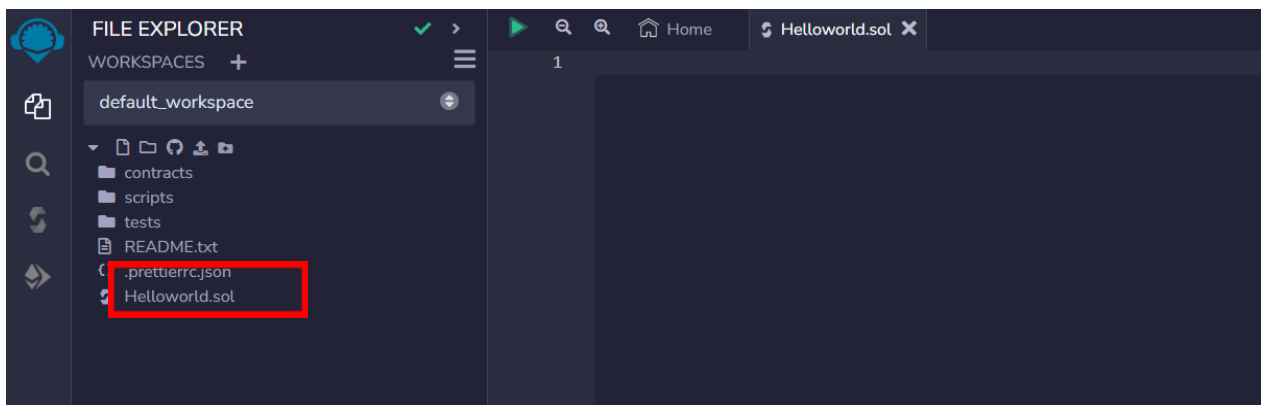
- i. *FILE EXPLORER* (for code writing)
- ii. *SEARCH IN FILES* (to find and replace files),
- iii. *SOLIDITY COMPILER* (for code execution)
- iv. *DEPLOY & RUN TRANSACTIONS* (for on-chain deployment).



- 2) **Create a new file:** Once you're on the Remix IDE website, create a blank Solidity contract by clicking the Create New File button.



On clicking, name the file as **'HelloWorld.sol'**. New file will be created with empty panel as



- 3) **Write the Solidity script:** In the newly created file, write the following "Hello, World!" Solidity script:

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

// compiler version must be greater than or equal to 0.8.17 and less than
0.9.0

pragma solidity ^0.8.17;

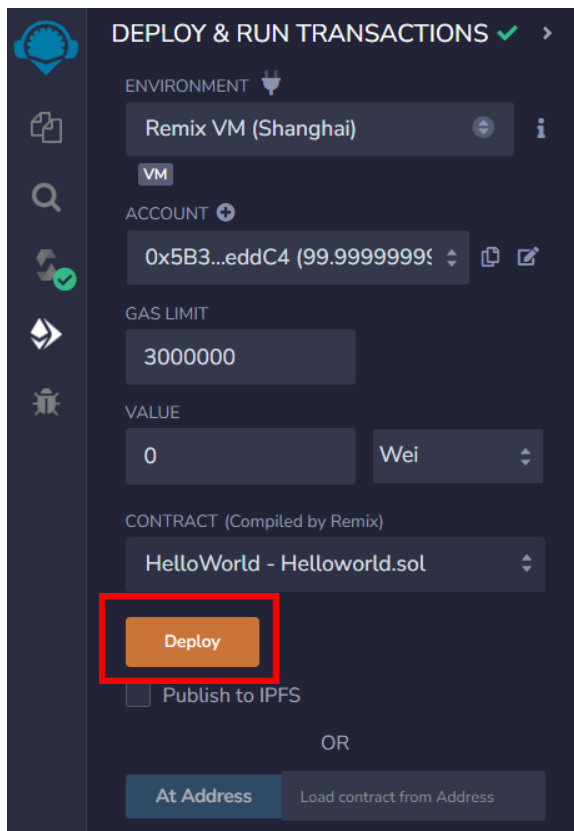
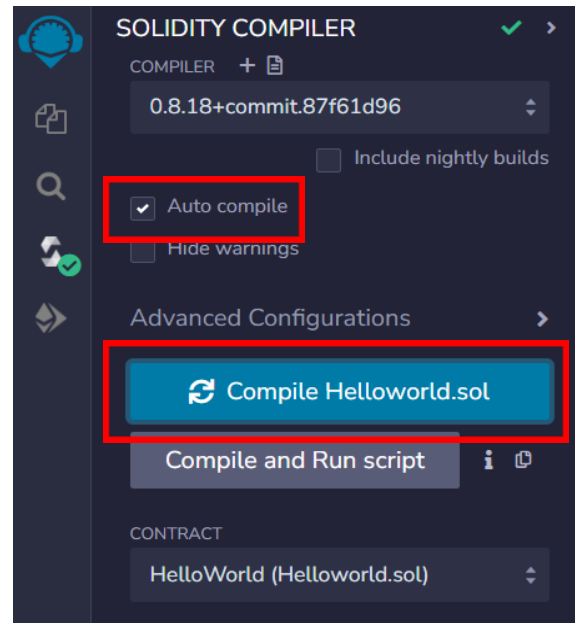
contract HelloWorld {

    string public greet = "Hello World!";

}
```

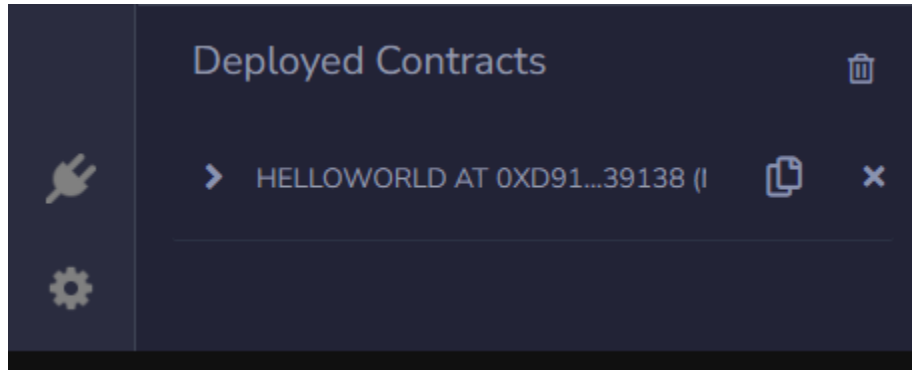
4) **Compile the contract:**

- Click on the *"Solidity Compiler"* tab in the Remix IDE. Ensure that the compiler version matches the pragma statement in your Solidity script (in this case, ^0.8.17).
- Click on the *"Compile HelloWorld.sol"* button to compile the contract.
- **Make sure to Enable Auto Compile.**



5) **Deploy the contract:** Click on the "Deploy & Run Transactions" tab in the Remix IDE. Without changing any of the values as shown above just click "Deploy" button to deploy your smart contract. Once deployed you will find your smart contract just below in "Deployed Contracts" heading.

***Congratulations!*** You have successfully created and deployed a "Hello World!" Solidity script using Remix on getting the following .



### Code Explanation:

- ***SPDX-License-Identifier:*** MIT: Indicates that the contract is licensed under the MIT license, which specifies the terms of use and redistribution.
- ***pragma solidity ^0.8.17:*** Sets the required compiler version for the contract to be equal to or greater than 0.8.17 and less than 0.9.0.
- ***contract HelloWorld:*** Defines a contract named "HelloWorld".
- ***string public greet = "Hello World!";*** Declares a public string variable named "greet" and assigns it the value "Hello World!". The public visibility allows other contracts and external entities to access this variable.