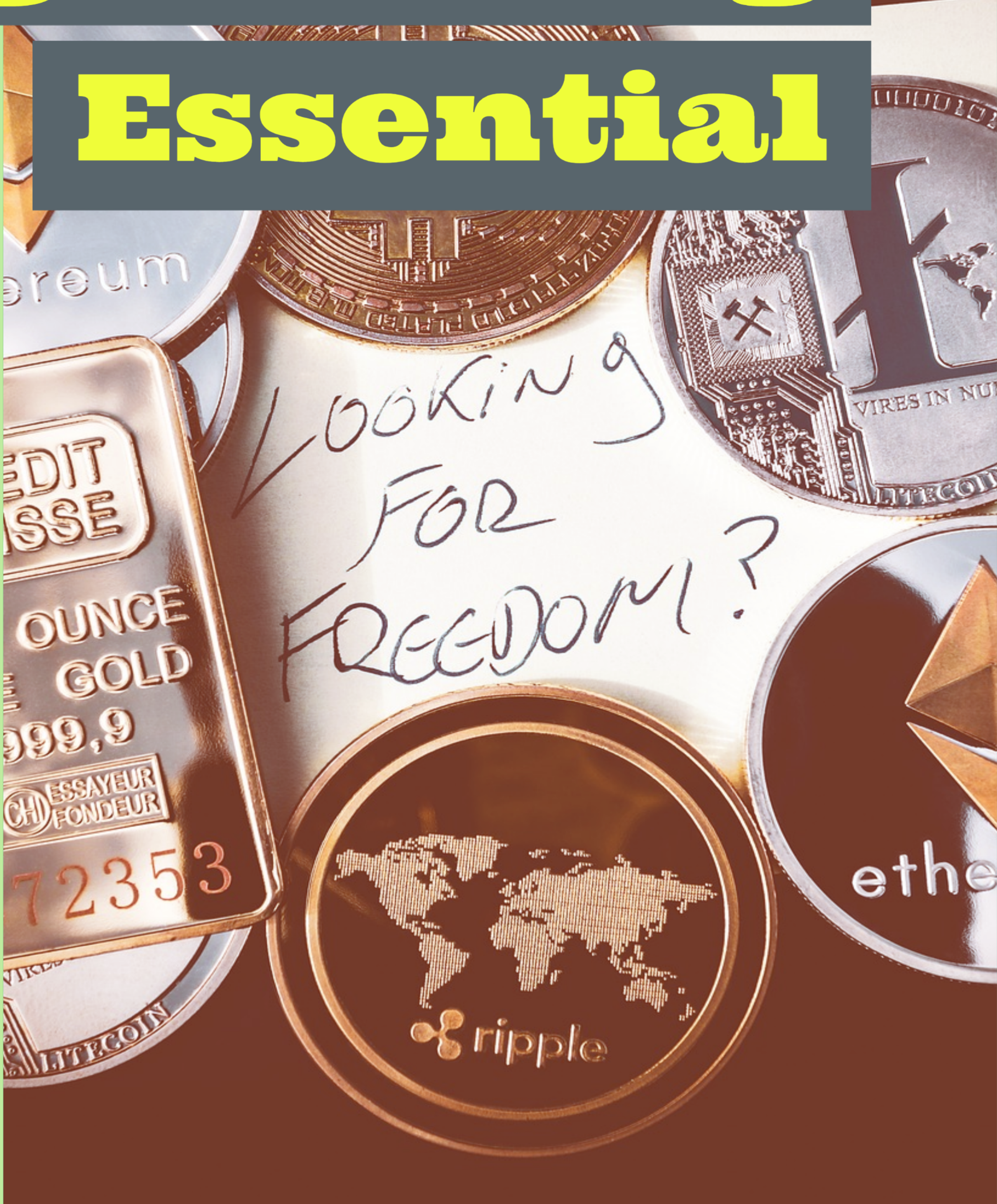
SOLIDITY Programming Essential



Author:
Anish Nath

Solidity Programming Essentials

Anish Nath

This book is for sale at http://leanpub.com/solidity

This version was published on 2021-07-05

Tweet This Book!

Please help Anish Nath by spreading the word about this book on Twitter!

The suggested hashtag for this book is #solidity blockchain ethereum.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

#solidity blockchain ethereum

Contents

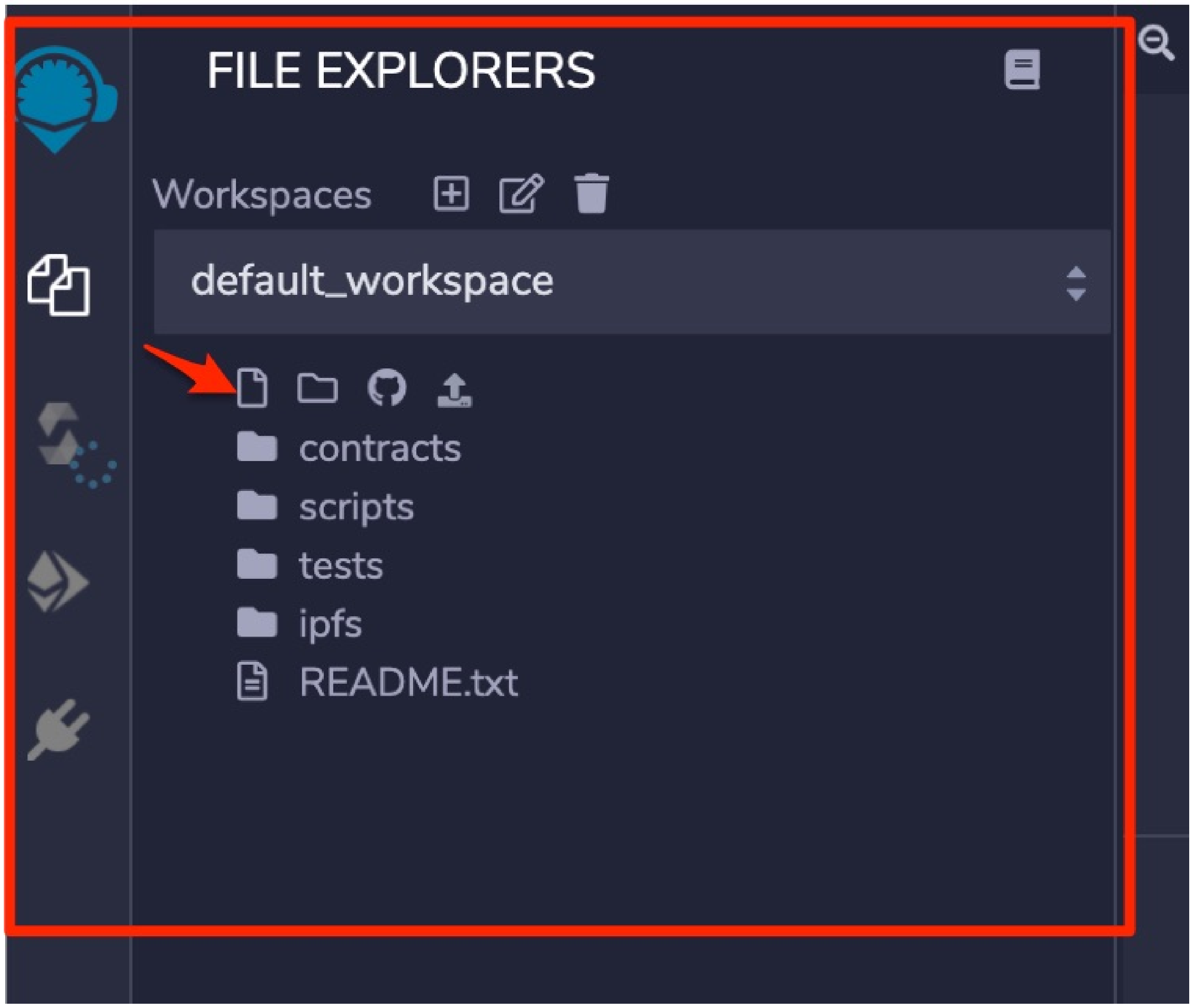
Remix Environment

You can try out code examples in this book directly in your browser with the Remix IDE. The remix is a web browser-based IDE that allows you to write, deploy and administer Solidity smart contracts, without the need to install Solidity locally.

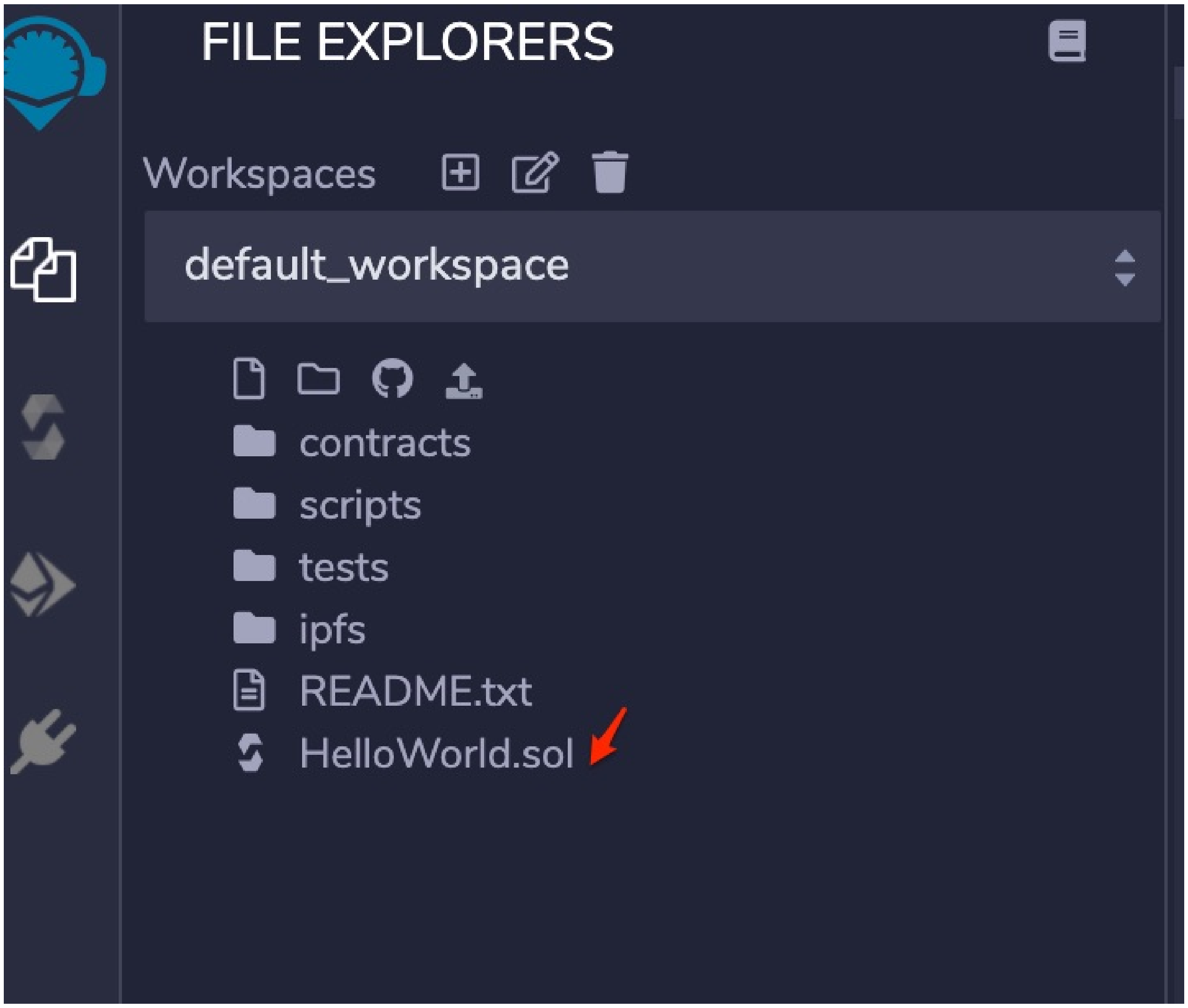
Open the https://remix.ethereum.org/ you will be presented with the entire remix IDE in the web browser and your IDE is ready to use.

HelloWorld with Remix

Leave the default setting as it is, in your chosen workspace go to the file explorer and locate the Icon shown in the below image to create a new file and this will be our first `HelloWorld.sol` smart contract deployment in Remix



Type the file name *“HelloWorld.sol”* and enter the following code into it

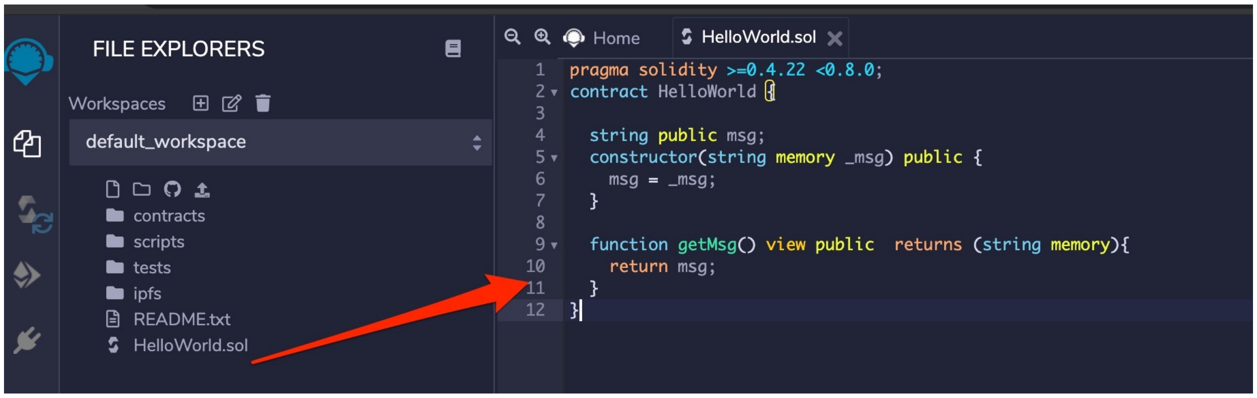


```
pragma solidity >=0.4.22 <0.8.0;
contract HelloWorld {

  string public msg;
  constructor(string memory _msg) public {
    msg = _msg;
  }

  function getMsg() view public  returns (string memory){
    return msg;
  }
}
```

Once done click the icon just below the *"file explorer"* icon as shown below:

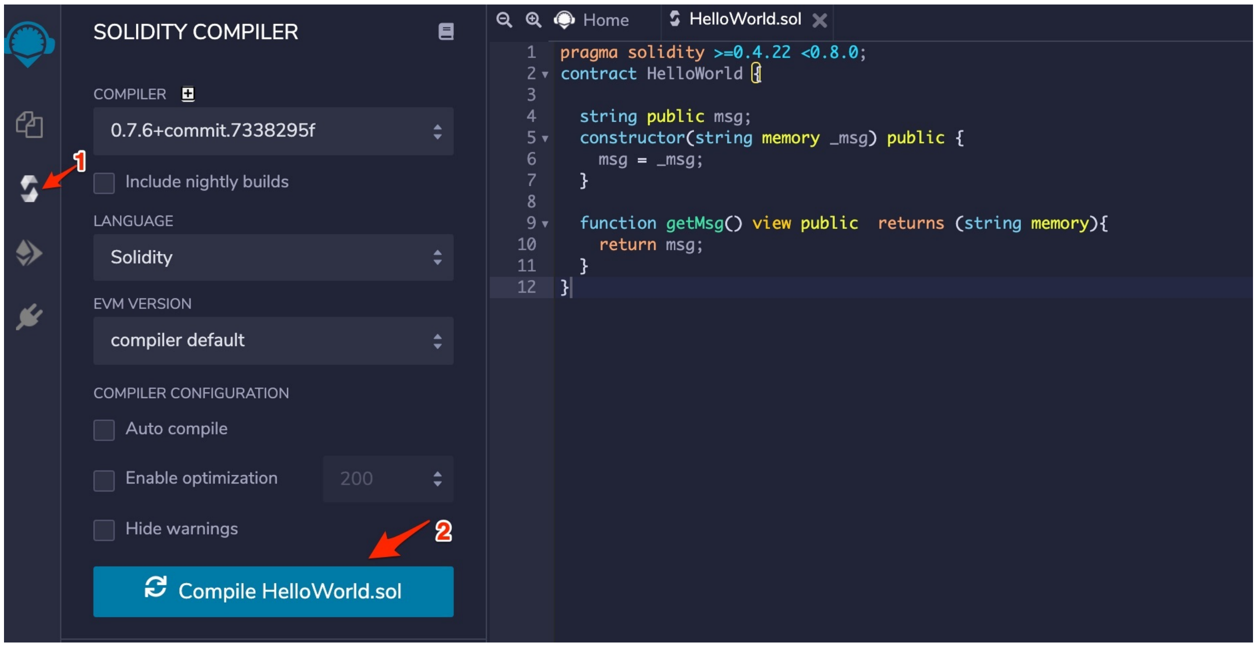


```
pragma solidity >=0.4.22 <0.8.0;
contract HelloWorld {

  string public msg;
  constructor(string memory _msg) public {
    msg = _msg;
  }

  function getMsg() view public  returns (string memory){
    return msg;
  }
}
```

Compile your first `HelloWorld.sol` Contract



COMPILER: 0.7.6+commit.7338295f

Include nightly builds

LANGUAGE: Solidity

EVM VERSION: compiler default

COMPILER CONFIGURATION

Auto compile

Enable optimization 200

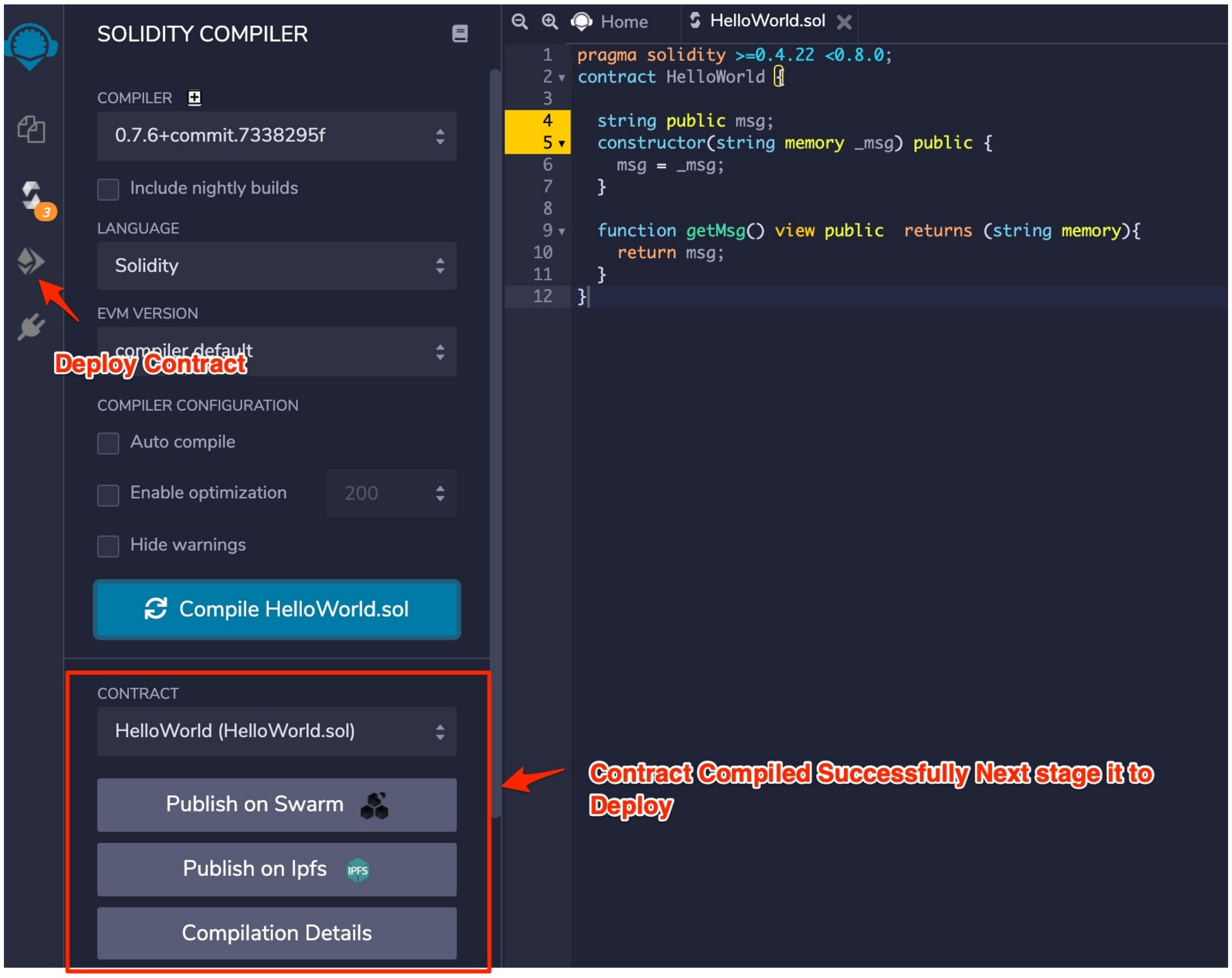
Hide warnings

```
pragma solidity >=0.4.22 <0.8.0;
contract HelloWorld {

  string public msg;
  constructor(string memory _msg) public {
    msg = _msg;
  }

  function getMsg() view public  returns (string memory){
    return msg;
  }
}
```

On successful compilation, the IDE will give you multiple Compiler options which can be explored. Now it's time to Deploy the **HelloWorld** smart contract. On file, explorer click on the Deploy option



COMPILER: 0.7.6+commit.7338295f

LANGUAGE: Solidity

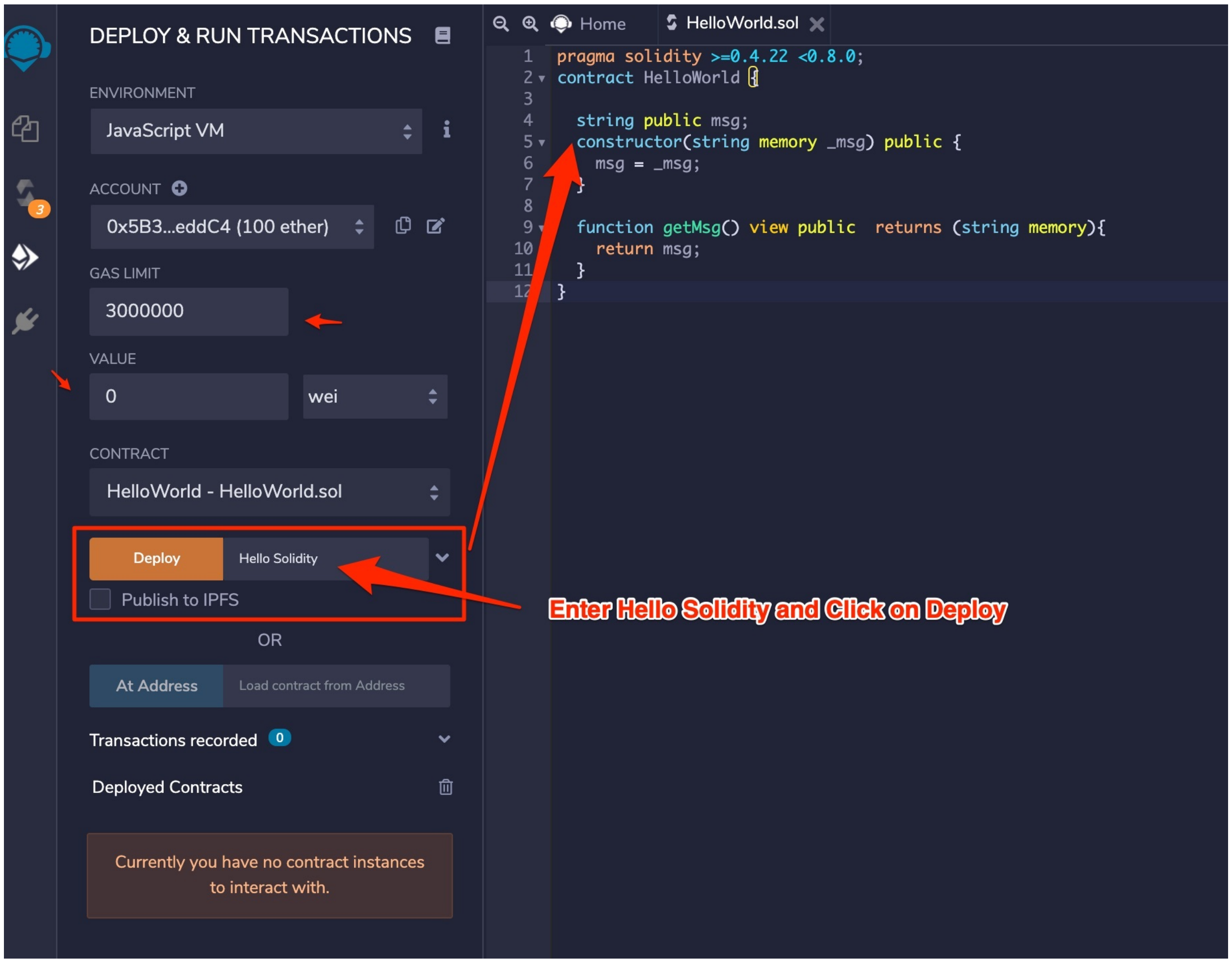
EVM VERSION: compiler default

```
pragma solidity >=0.4.22 <0.8.0;
contract HelloWorld {

  string public msg;
  constructor(string memory _msg) public {
    msg = _msg;
  }

  function getMsg() view public  returns (string memory){
    return msg;
  }
}
```

Fill out the constructor parameter “**Hello Solidity**” required for the HelloWorld contract and click on Deploy

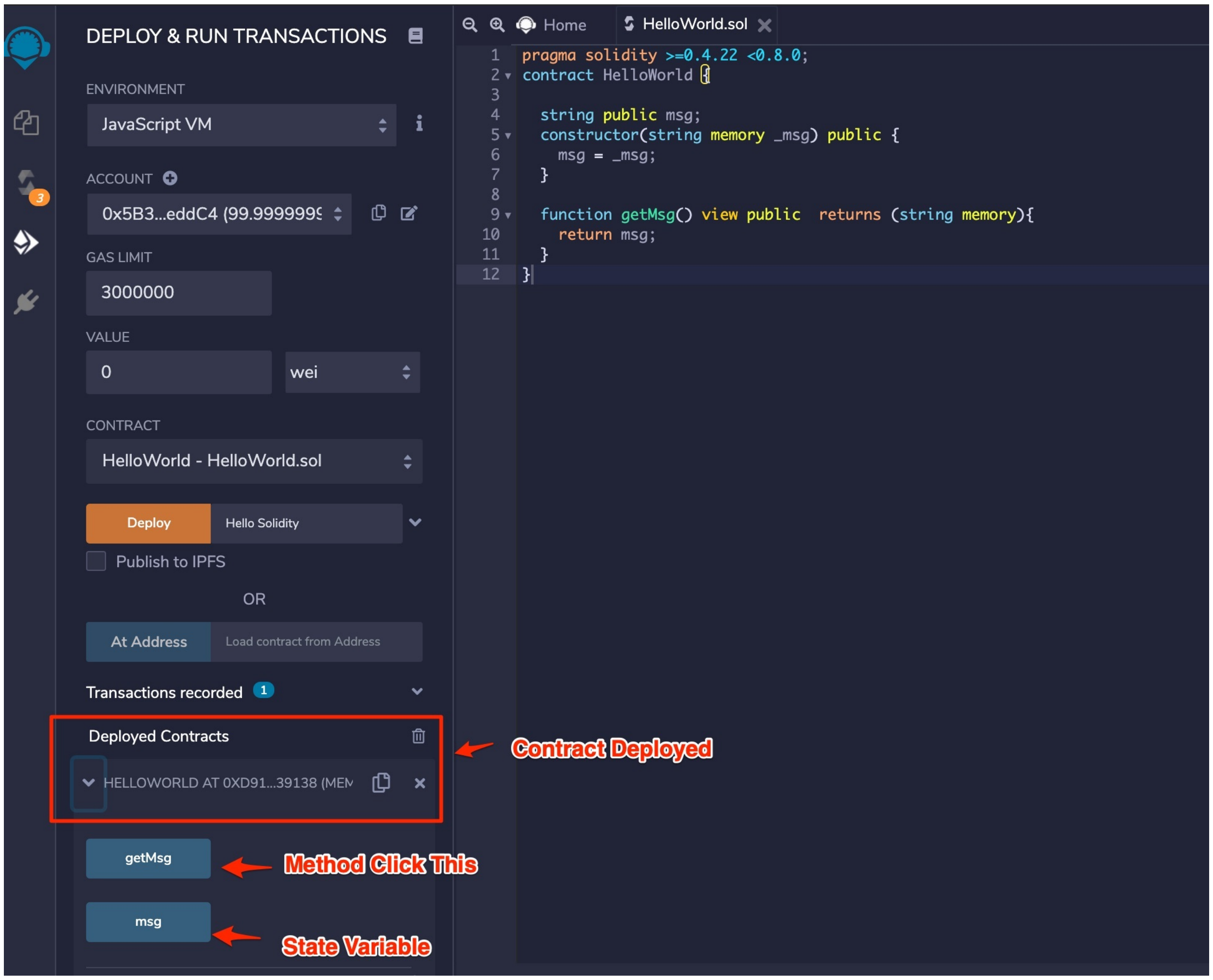

DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
JavaScript VM
ACCOUNT
0x5B3...eddC4 (100 ether)
GAS LIMIT
3000000
VALUE
0
wei
CONTRACT
HelloWorld - HelloWorld.sol
Deploy
Hello Solidity
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 0
Deployed Contracts
Currently you have no contract instances to interact with.
Home
HelloWorld.sol
Enter Hello Solidity and Click on Deploy

```
pragma solidity >=0.4.22 <0.8.0;
contract HelloWorld {

  string public msg;
  constructor(string memory _msg) public {
    msg = _msg;
  }

  function getMsg() view public  returns (string memory){
    return msg;
  }
}
```

You will see the next screen where you can see the Deployed contracts, expand that tab and it will give you the method and state variable available for transactions. Click on **getMsg()** as defined in the contract definition



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT
JavaScript VM

ACCOUNT
0x5B3...eddC4 (99.9999999

GAS LIMIT
3000000

VALUE
0 wei

CONTRACT
HelloWorld - HelloWorld.sol

Deploy Hello Solidity

Publish to IPFS

OR

At Address Load contract from Address

Transactions recorded 1

Deployed Contracts

HELLOWORLD AT 0XD91...39138 (MEM

getMsg

msg

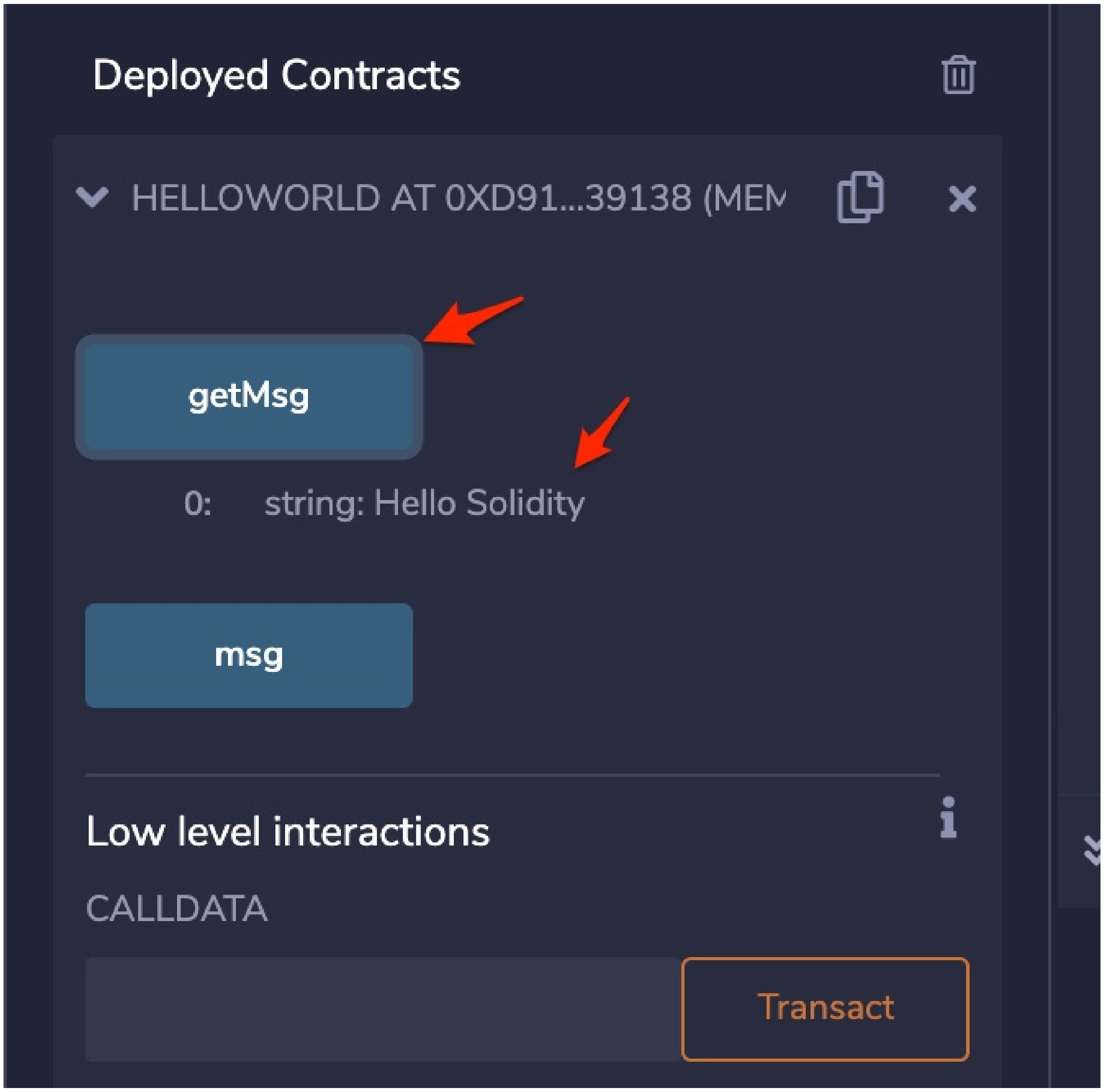
Home HelloWorld.sol

```
pragma solidity >=0.4.22 <0.8.0;
contract HelloWorld {

  string public msg;
  constructor(string memory _msg) public {
    msg = _msg;
  }

  function getMsg() view public  returns (string memory){
    return msg;
  }
}
```

The smart contract will display the output of the program



0: string: Hello Solidity

Use the Remix IDE to examine the transacation of hash, address, execution cost and others. This information will be available on the bottom pane of the remix IDE.

The Remix IDE will also show the entire calldata for e.g for the above transacation the following information is captured.

```
transaction hash        0xf772277ffa74574891ead016341ec9cd81481139c8377f6a99539453bc686207
 from        0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
 to        HelloWorld.getMsg() 0xd9145CCE52D386f254917e481eB44e9943F39138
 execution cost        24201 gas (Cost only applies when called by a contract)
 hash        0xf772277ffa74574891ead016341ec9cd81481139c8377f6a99539453bc686207
 input        0xb5f...deb23
 decoded input        {}
 decoded output        { "0": "string: Hello Solidity" }
 logs        []
```