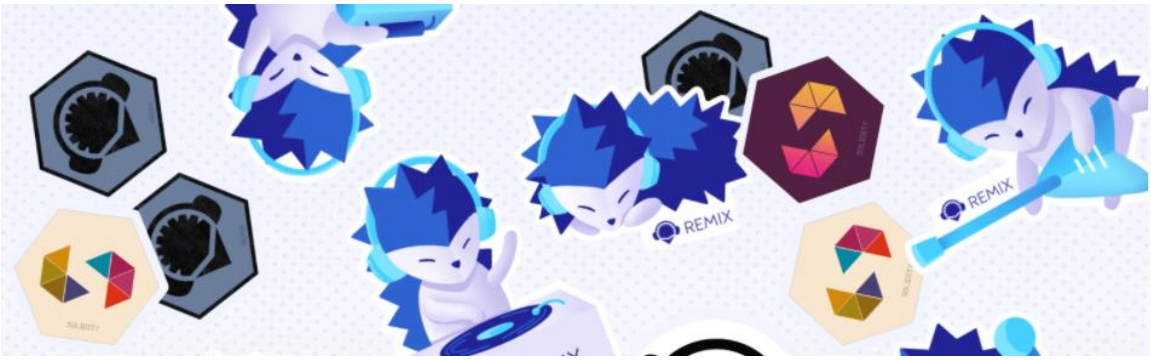
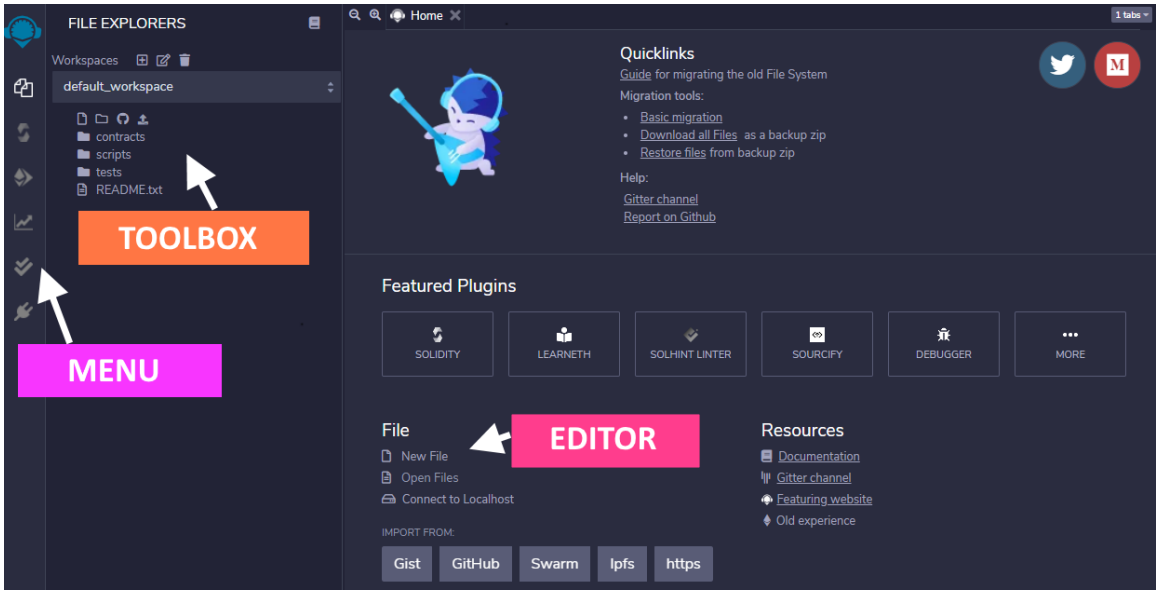
Deploying Smart Contracts



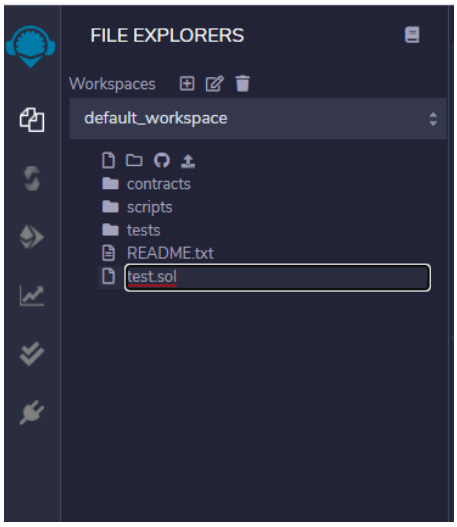
Ganache and Metamask is required to run this activity.

You write code at the Remix editor hosted at: https://remix.ethereum.org/
Full documentation: https://remix-ide.readthedocs.io/en/latest/index.html

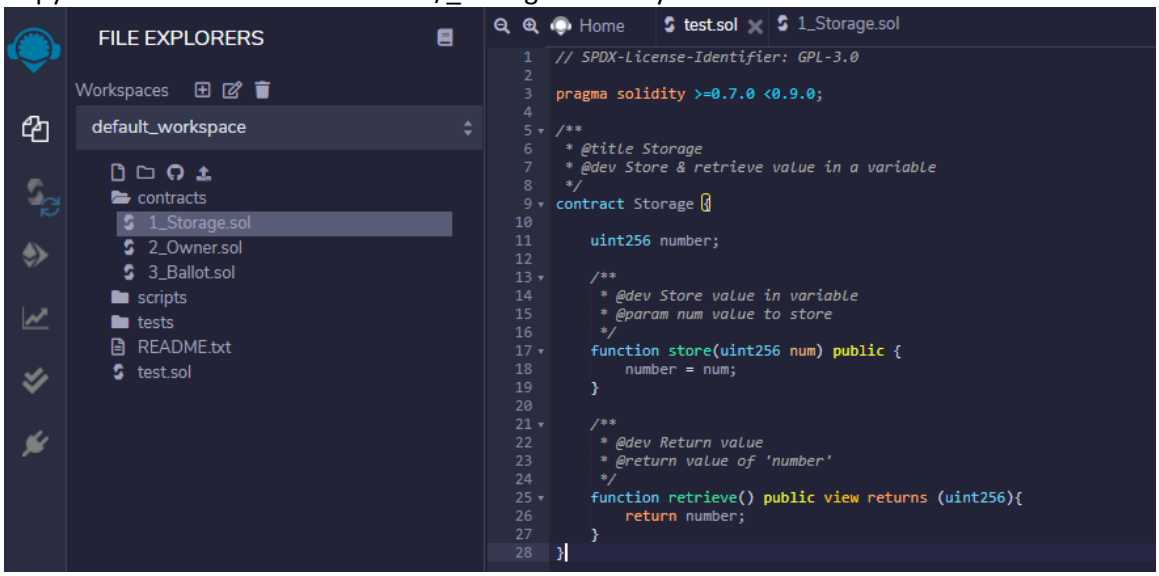


Click on New File to open the File Editor, an empty file will appear in the default_workspace.

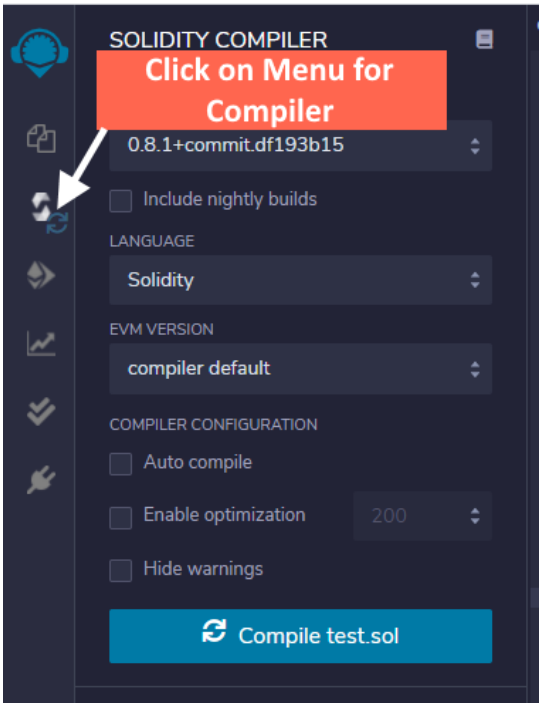
Key in a contract file name.



Copy the code from folder contracts/_Storage.sol into your new file.



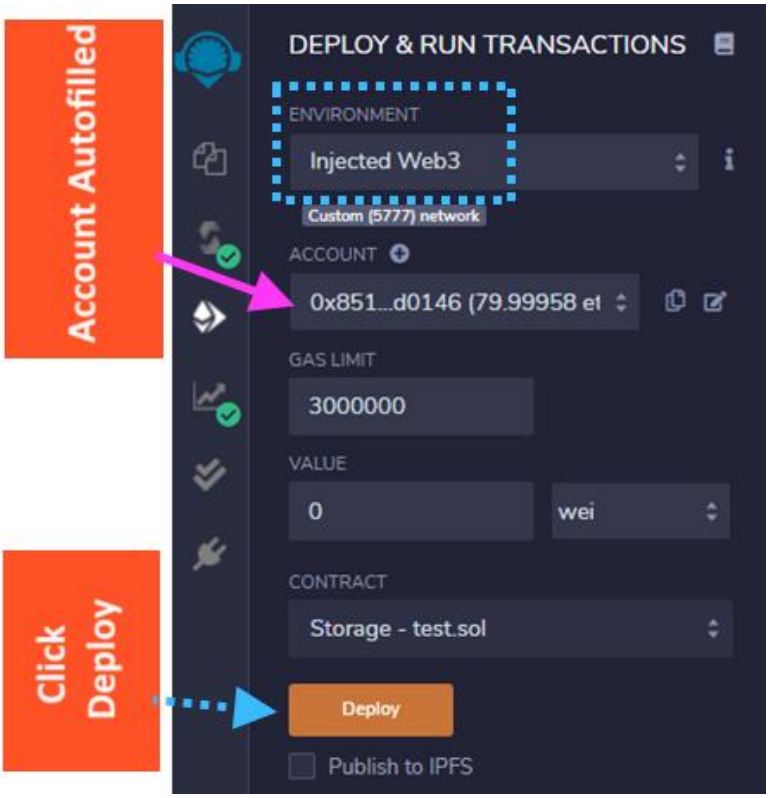
Click on the Compiler and compile the code.

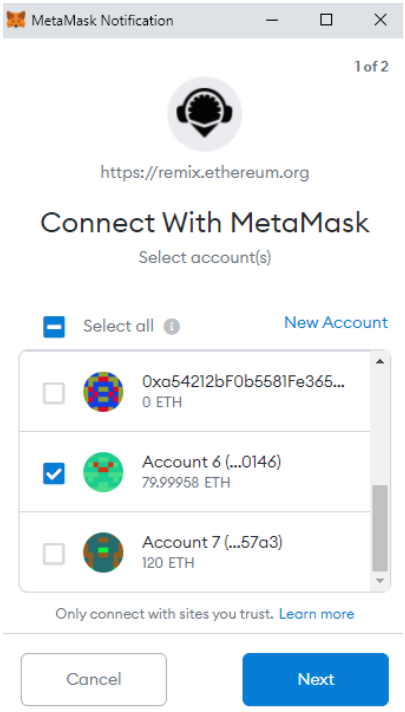


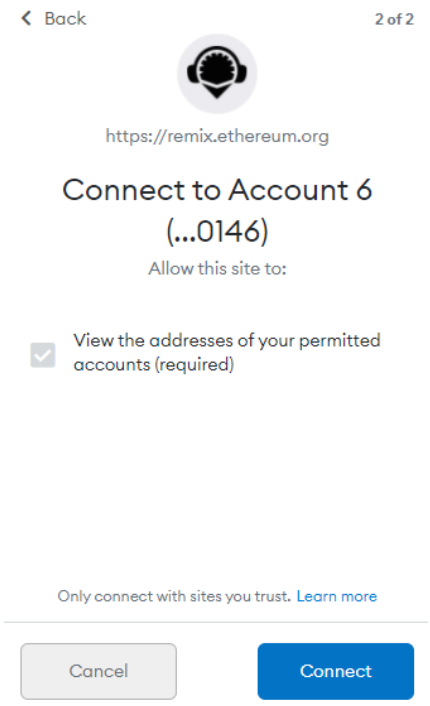
After compiling successfully, click on the next menu item**: Deploy and Run.**
Select:

Injected Web3 for Environment.

You will be asked to connect to Metamask. Select the account that is available at Ganache. You will have the account number filled automatically in Remix.


Account Autofilled
DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Injected Web3
Custom (5777) network
ACCOUNT
0x851...d0146 (79.99958 et
GAS LIMIT
3000000
VALUE
0
wei
CONTRACT
Storage - test.sol
Click Deploy
Deploy
Publish to IPFS

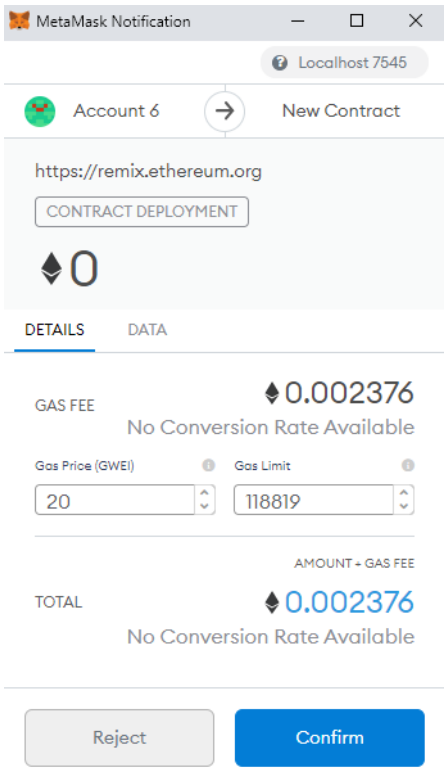

MetaMask Notification
1 of 2
https://remix.ethereum.org
Connect With MetaMask
Select account(s)
Select all
New Account
0xa54212bF0b5581Fe365...
0 ETH
Account 6 (...0146)
79.99958 ETH
Account 7 (...57a3)
120 ETH
Only connect with sites you trust. Learn more
Cancel
Next



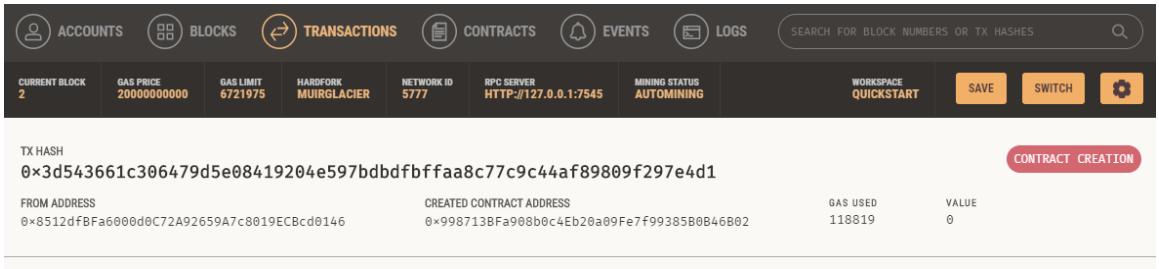
After connecting to Metamask, click at Remix:

Deploy

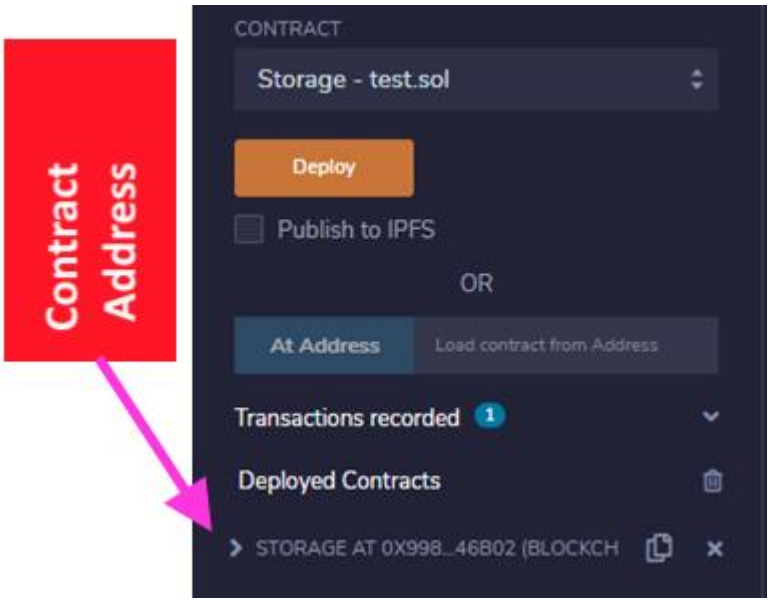
The Metamask window will appear asking to confirm the deployment of contract.



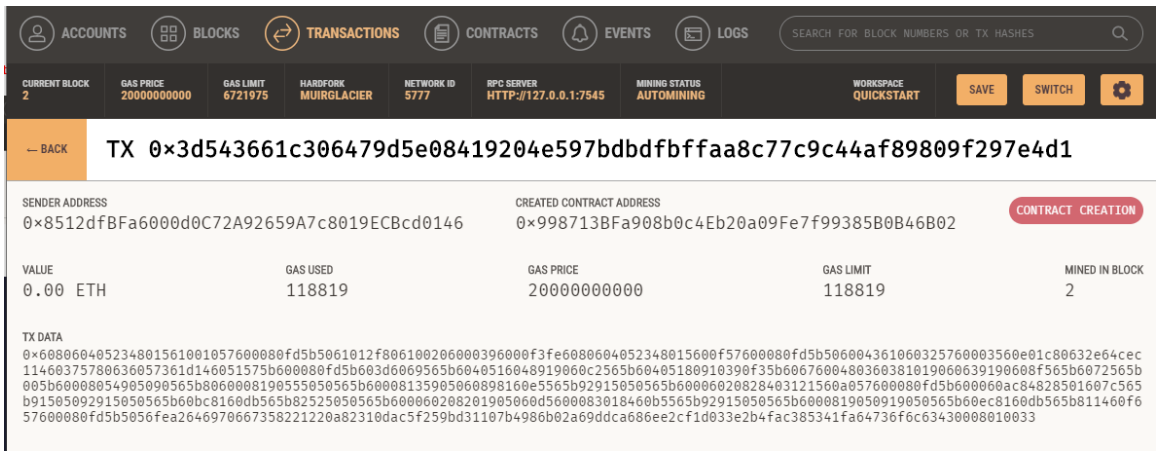
Go back to Ganache, at the Transactions will you will see **contract creation** at the transaction.



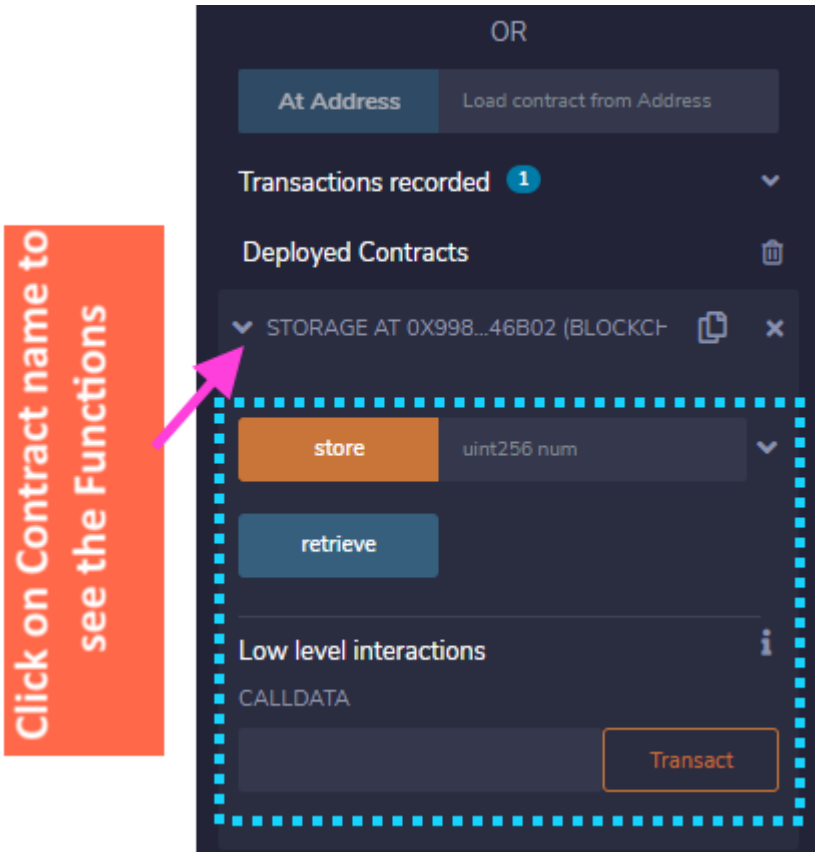
Back at Remix, you can see the address that the Contract (named Storage) is deployed.



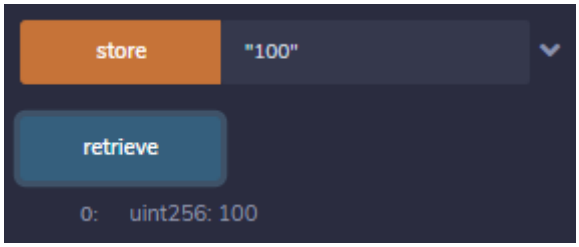
You can also confirm the address is actually created at Ganache.



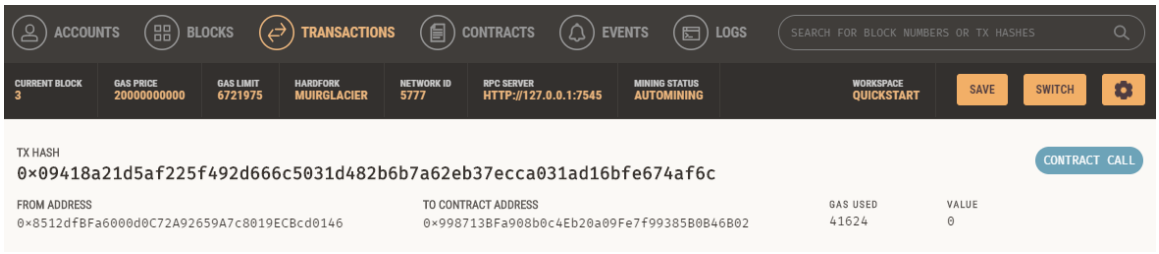
Go back to Remix to call the contract from an account. Click on the Contract name to expand to its functions (methods).



Key in a number at **store**: 100. Metamask will ask for confirmation. Click **retrieve** to see the value that was previously stored.



Transactions at Ganache registered a contract call.



What the Code does

The code simply stores a number for the associated account. Here are some questions that you might ask yourself:

1. What is the sender's address?
2. What is the contract address?
3. Can the contract address be changed?
4. Who is the contract owner?
5. Where is the number stored?
6. Can the contract store a number for another account?
7. Does the store method create a transaction in a block?
8. What happens when the store is called more than once?
9. Does the retrieve method create a transaction block?
10. Is creating a contract easy?

Exercise

Modify the code so that you have an **add** method that adds to the existing store value instead of overwriting it.

Compile it and check that it is running correctly.

Conclusion

You have successfully executed a smart contract that stored a number on a private Ethereum blockchain. This number could well be a token or a crypto currency. This is how tokens are created.

Make sure you understand all the steps, as we will be using them a fair bit. We will discuss further details on Contract writing in the following weeks.

Answer: 3:N;6:N;7:Y;9:N;10:Y