Workshop: Your First Smart Contract

Deploy to a Real Blockchain in 15 Minutes

BSc Blockchain, Crypto Economy & NFTs

FS2026

Workshop Overview

What You'll Do:

1. Set up MetaMask wallet
2. Write a simple Solidity contract
3. Deploy to Sepolia testnet
4. Interact with your contract

Prerequisites:

- Browser (Chrome recommended)
- 15 minutes
- No coding experience required!

Related Lessons: W01, L13, L15

By the end, you'll have a contract running on a real blockchain

Step 1: Install MetaMask

MetaMask is your blockchain wallet

1. Go to **metamask.io**
2. Click **Download**
3. Add to your browser
4. Create new wallet
5. **SAVE YOUR SEED PHRASE!**

Important:

- Never share your seed phrase
- Write it down on paper
- This is your master key to all funds

MetaMask manages your keys and signs transactions

Step 2: Get Test ETH

Switch to Sepolia Testnet:

1. Open MetaMask
2. Click network dropdown (top)
3. Settings → Show test networks
4. Select **Sepolia test network**

Get Free Test ETH:

1. Copy your wallet address from MetaMask
2. Go to **sepoliafaucet.com**
3. Paste address, click request
4. Wait 1-2 minutes

Alternative Faucets:

- Alchemy: alchemy.com/faucets/ethereum-sepolia
- Chainlink: faucets.chain.link/sepolia

Testnets let you experiment without real money

Step 3: Open Remix IDE

Remix is a browser-based Solidity IDE

1. Go to **remix.ethereum.org**
2. Close welcome tabs
3. In file explorer, click **contracts** folder
4. Create new file: `HelloWorld.sol`

Remix Features:

- No installation required
- Built-in compiler
- Direct deployment to networks
- Contract interaction UI

Remix is perfect for learning and prototyping

Step 4: Write Your Contract

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

contract HelloWorld {
    string public greeting;

    event GreetingChanged(string oldGreeting, string newGreeting);

    constructor(string memory _greeting) {
        greeting = _greeting;
    }

    function getGreeting() public view returns (string memory) {
        return greeting;
    }

    function setGreeting(string memory _newGreeting) public {
        string memory old = greeting;
        greeting = _newGreeting;
        emit GreetingChanged(old, _newGreeting);
    }
}
```

This contract stores and updates a greeting message

Code Explanation

Line	Meaning
`SPDX-License`	License declaration (required)
`pragma solidity`	Compiler version requirement
`contract HelloWorld`	Defines your contract
`string public greeting`	State variable stored on-chain
`event GreetingChanged`	Log for tracking changes
`constructor`	Runs once at deployment
`view`	Function doesn't modify state (free to call)
`memory`	Variable stored temporarily in RAM

State variables persist between calls; memory doesn't

Step 5: Compile

In Remix:

1. Click **Solidity Compiler** icon (left sidebar)
2. Select compiler: `0.8.20` or higher
3. Click **Compile HelloWorld.sol**
4. Green checkmark = success!

Common Errors:

Error	Solution
Parser error	Check semicolons and brackets
Type error	Verify variable types match
Version mismatch	Select correct compiler version

Compilation converts Solidity to bytecode

Step 6: Deploy

Connect MetaMask:

1. Click **Deploy & Run** icon (left sidebar)
2. Environment: **Injected Provider - MetaMask**
3. MetaMask popup: Click **Connect**
4. Verify: Network = Sepolia, Account = Your address

Deploy Contract:

1. Under Deploy, enter greeting: `"Hello, Blockchain!"`
2. Click **Deploy**
3. MetaMask popup: Click **Confirm**
4. Wait 10-30 seconds for confirmation

Save Your Contract Address!

Deployment is a transaction that costs gas

Step 7: Interact

Read Functions (Free):

- Click `greeting` → Shows your message
- Click `getGreeting` → Same result
- These are **view** functions - no gas cost

Write Functions (Cost Gas):

1. In `setGreeting`, enter new message
2. Click `setGreeting`
3. MetaMask: Click **Confirm**
4. Wait for confirmation
5. Check `greeting` again - it changed!

Read = free, Write = costs gas

Step 8: View on Etherscan

Verify Your Contract Exists:

1. Copy your contract address
2. Go to **sepolia.etherscan.io**
3. Paste address in search
4. See your contract and transactions!

What You'll See:

- Contract creation transaction
- All subsequent interactions
- Event logs
- Contract bytecode

Optional: Verify Source

- Makes source code public
- Enables Etherscan interactions
- Builds trust

Etherscan is the blockchain explorer for Ethereum

Congratulations!

You deployed your first smart contract!

What You Accomplished:

- ✓ Set up blockchain wallet
- ✓ Wrote Solidity code
- ✓ Deployed to real network
- ✓ Interacted with your contract

Next Steps:

- Try the **Create Cryptocurrency** workshop
- Explore more Solidity features
- Build something useful!

Your contract will exist on Sepolia forever

Troubleshooting

Problem	Solution
"Insufficient funds"	Get more test ETH from faucet
"Transaction failed"	Check gas limit, retry
MetaMask not connecting	Refresh page, unlock MetaMask
Wrong network	Switch to Sepolia in MetaMask
Compilation error	Check syntax, semicolons, brackets

Resources:

- Notebook: `projects/notebooks/05_first_contract.ipynb`
- Web Guide: `.../projects/first-contract/`
- Solidity Docs: `docs.soliditylang.org`