

Lab Session: Smart Contract Interaction

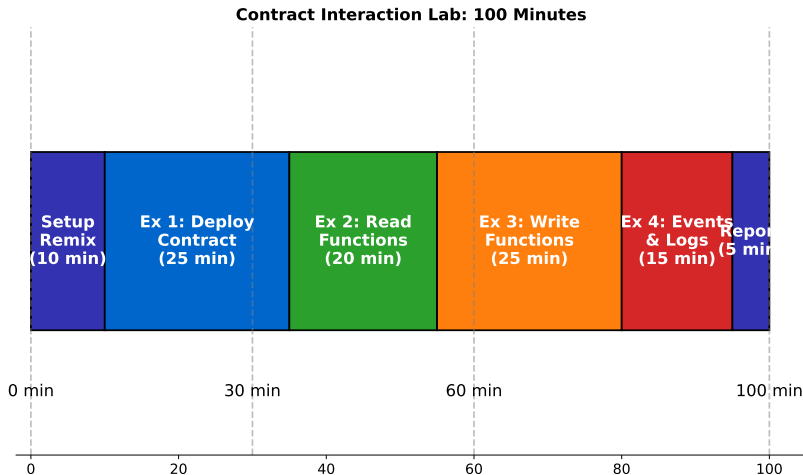
BSc Blockchain, Crypto Economy & NFTs

Course Instructor

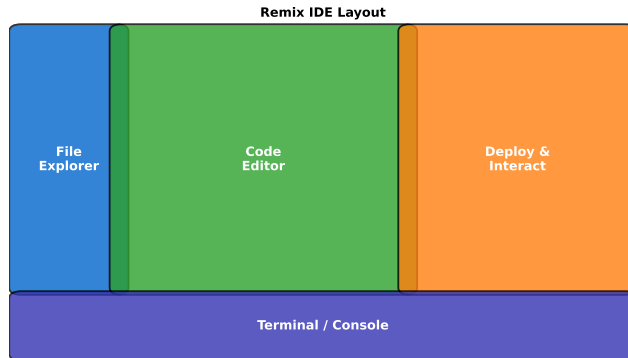
Module B: Ethereum & Smart Contracts

By the end of this lab session, you will be able to:

- Deploy smart contracts using Remix IDE
- Interact with contract read functions (view/pure)
- Execute contract write functions (state-changing)
- Monitor events and transaction logs
- Debug failed transactions



Focus on hands-on interaction with deployed contracts



Deploy & Interact panel is where you'll spend most of your time

Exercise 1: Deploy a Simple Contract

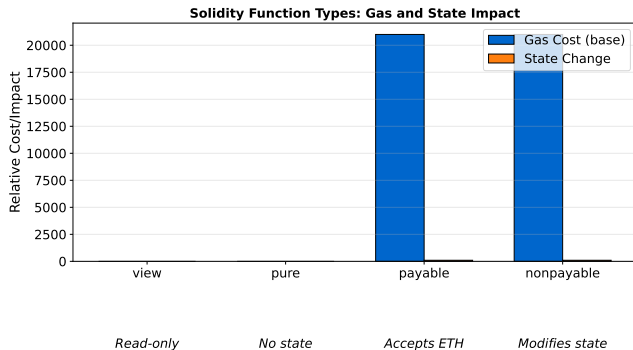
Steps:

- 1 Open Remix IDE: <https://remix.ethereum.org>
- 2 Create new file: `SimpleStorage.sol`
- 3 Write a storage contract with `get/set` functions
- 4 Compile using Solidity compiler
- 5 Deploy to JavaScript VM (local testing)

Key Concepts:

- Contract deployment creates new address
- Constructor runs once at deployment
- State variables persist on blockchain

Function Types in Solidity



View/pure functions are free; write functions cost gas

Exercise 2: Read Functions

Characteristics:

- **view:** Reads state, no modification
- **pure:** No state access, computation only
- No gas cost when called externally
- Instant response (no transaction needed)

Tasks:

- 1 Call `get()` function to read stored value
- 2 Observe: no MetaMask popup, instant result
- 3 Note the return value in Remix console

Exercise 3: Write Functions

Characteristics:

- Modify blockchain state
- Require gas payment
- Create transaction (needs confirmation)
- May emit events

Tasks:

- 1 Call `set(42)` to store new value
- 2 Observe: transaction created, gas consumed
- 3 Verify change by calling `get()`
- 4 Check transaction in Remix terminal

Why Events?

- Notify external applications of state changes
- Cheaper than storage for historical data
- Indexed parameters enable efficient filtering

Tasks:

- 1 Add event to contract: `event ValueChanged(uint newValue)`
- 2 Emit event in set function
- 3 Redeploy and test
- 4 View logs in Remix console

Submit:

1 Solidity Contract:

- SimpleStorage.sol with get, set, and event

2 Screenshots:

- Deployed contract interface
- Read function result
- Write function transaction
- Event log output

Submission Deadline: One week from lab session

Key Takeaways

- Remix IDE is the standard tool for contract development
- View/pure functions are free (no gas)
- Write functions cost gas and create transactions
- Events provide efficient notification mechanism
- Always test on local VM before deploying to testnet