



smerf.net

Ethereum Blockchain Development Week 22

Introduction

The intention of this lab is to look at functions and how variables are stored. All the exercises are completed in [Remix IDE](#) or the VM provided.

Code Completion

Writing code in a new language can be a steep learning curve. The approach here is to provide some code with underscores (_) that you are required to complete. These underscores are there to help you. By completing these exercises you will be improving your skills and knowledge of Solidity.

Each exercise starts on a new page. The **red** numbers in the right-hand margin are estimated minutes you should spend on each exercise.

1 Simple Contract in Truffle

25-30

This exercise has to be completed in the VM provided.

Open a terminal and complete the instructions below:

1. Open a terminal and create a solidity environment:
 - (a) `mkdir simple` and `cd simple`
 - (b) `truffle init`
 - (c) `code .`
2. Edit and modify the `truffle-config` file as shown in Fig. ??
3. Add a new file, `2_simple.js`, in the migrations directory. This file should contain the deployment of the contract and is listed in Fig ??
4. Open Ganache and add a new project, `simple`, as a new environment. Link this project to the `truffle-config.js` file completed above. Ensure that the ports are both matching 7545 and it is up and running with 10 accounts
5. Back in VS Code, add a new file, `simple.sol`, and complete the solidity listing in Fig. ?? in the contracts directory.
6. Compile: In the terminal compile the code, `truffle compile`. Correct any errors, until the compilation is error free.
7. Migrate: In the terminal migrate the code, `truffle migrate`
8. Console: In the terminal go to the truffle console: `truffle console`
9. Complete the following Web3.0 exercises from the truffle console:
 - (a) Retrieve the provider and display the network details
 - (b) Display all the addresses and check they match in Ganache
 - (c) Display the balance of the first and second accounts
 - (d) Display the address of the owner and check this is correct
 - (e) Display the current block number
 - (f) Deploy the smart contract, `simple`

```
Code Completion

1 //SPDX-License-Identifier: MIT
2 _____ solidity ^0.8.0;
3
4 _____ simple{
5     uint public x;
6
7     _____ incrementX() _____ {
8         x++;
9     }
10
11    _____ decrementX() _____ {
12        x--;
```

Figure 1.1: Interface code for Simple contract, available from [simple-incomplete.sol](#)

```
Code Completion

1 development: {
2     host: "127.0.0.1", // Localhost (default: none)
3     port: 7545, // Standard Ethereum port (default: none)
4     network_id: "*", // Any network (default: none)
```

Figure 1.2: Uncomment the development network with the following configurations, code for truffle configuration file

- (g) Instantiate a transaction that calls the `incrementX` function, repeat this three times
- (h) Display the value of `x` from the smart contract, is it 3?
- (i) Display the current block number, has it increased?

```
Code Completion
1 const simple = artifacts.require("simple");
2
3 module.exports = function (deployer) {
4   deployer.deploy(simple);
5 };
```

Figure 1.3: Migrations code to be saved in migrations folder as 2_simple.js