Lesson 3 - Programming tools

Programming setup

- Reference repository
- NPM and Yarn
- VS Code
- VS Code plugins
- source control
- unit tests
- scripts
- Testing passing
- Breaking tests

References

https://github.com/OpenZeppelin/openzeppelin-contracts

https://git-scm.com/book/en/v2/Getting-Started-The-Command-Line

https://classic.yarnpkg.com/en/docs/getting-started

https://classic.yarnpkg.com/en/docs/cli/

https://code.visualstudio.com/docs

Instructions

```
git clone https://github.com/OpenZeppelin/openzeppelin-contracts.git
cd .\openzeppelin-contracts\
yarn install
```

Output example

```
yarn compile
yarn run v1.22.18
$ hardhat compile
...
Compiled 267 Solidity files successfully
Done in 18.79s.
```

Package.json scripts

```
  "scripts": {
    "compile": "hardhat compile",
    "coverage": "env COVERAGE=true hardhat coverage",
    "docs": "oz-docs",
    "docs:watch": "npm run docs watch contracts 'docs/*.hbs'
docs/helpers.js",
    "prepare-docs": "scripts/prepare-docs.sh",
    "lint": "npm run lint:js && npm run lint:sol",
    "lint:fix": "npm run lint:js:fix && npm run lint:sol:fix",
    "lint:js": "eslint --ignore-path .gitignore .",
    "lint:js:fix": "eslint --ignore-path .gitignore . --fix",
    "lint:sol": "solhint 'contracts/**/*.sol' && prettier -c
'contracts/**/*.sol'",
    "lint:sol:fix": "prettier --write \"contracts/**/*.sol\"",
    "clean": "hardhat clean && rimraf build contracts/build",
    "prepare": "npm run clean && env COMPILE_MODE=production npm run
compile",
    "prepack": "scripts/prepack.sh",
    "release": "scripts/release/release.sh",
    "version": "scripts/release/version.sh",
    "test": "hardhat test",
    "test:inheritance": "node scripts/inheritanceOrdering artifacts/build-
info/*",
    "gas-report": "env ENABLE_GAS_REPORT=true npm run test"
  },
```

Test example

```
yarn test .\test\token\ERC20\ERC20.test.js
yarn run v1.22.18
$ hardhat test .\test\token\ERC20\ERC20.test.js
...
  76 passing (5s)
Done in 8.39s.
```

Breaking change

```
code .\contracts\token\ERC20\ERC20.sol
```

```
    function decimals() public view virtual override returns (uint8) {
        return 42;
    }
```

```
yarn test .\test\token\ERC20\ERC20.test.js
yarn run v1.22.18
$ hardhat test .\test\token\ERC20\ERC20.test.js
...
Compiled 28 Solidity files successfully
...
 75 passing (7s)
  1 failing
  1) Contract: ERC20
       has 18 decimals:
      AssertionError: expected '42' to equal '18'
      + expected - actual
      -42
      +18
```

Quality of code

- Composability
- Upgradeability
- Maintainability and readability
- Managing work flow and progress
- Reaching peace of mind

Hardhat setup

- Creating a base repository
- Setup hardhat with typescript
- Configure VS Code
- Hardhat scripts and tasks
- VS Code extensions recommended

References

https://hardhat.org/getting-started/

https://hardhat.org/guides/typescript.html

https://hardhat.org/guides/vscode-tests.html

Steps

```
yarn init
    ...
git init
yarn add hardhat --dev
yarn hardhat init
    "Create an advanced sample project that uses TypeScript"
    ...
    .gitignore? Y
```

```
yarn add --dev [list of suggested dev dependencies above]
Yarn add chai
code .
```

Recommended extensions

Git Graph

Mocha Test Explorer

Solidity

Environment setup

.mocharc.json file:

```
{
  "require": "ts-node/register/files",
  "timeout": 20000
}
```

hardhat.config.ts file:

```
{
const config: HardhatUserConfig = {
...
  paths: { tests: "tests" },
...
  }
}
```

tsconfig.json file:

```
{
const config: HardhatUserConfig = {
...
  "include": ["./scripts", "./tests", "./typechain"],
...
  }
}
```

Run command in root project folder:

```
cp .\.env.example .env
```

Edit the environment with your keys:

```
code .env
```

Test it out:

```
yarn hardhat accounts
yarn hardhat test
```

Coding in VS Code

- Syntax for typescript scripts
- How the project operates
- Writing a test file
- Using Waffle
- Using Ethers.js library
- Testing syntax
- Running a test file

References

https://ethereum-waffle.readthedocs.io/en/latest/

https://docs.ethers.io/v5/

https://mochajs.org/

https://www.chaijs.com/guide/

Clearing template files

```
rm .\contracts\*
rm .\scripts\*
rm .\tests\*
yarn hardhat clean
```

Introduction for Ballot.sol

- Scenario strategy
- The Ballot contract
- Giving voting rights
- Voting
- Delegating
- Proposals

Reference

https://docs.soliditylang.org/en/latest/solidity-by-example.html#voting

TDD methodology

- Writing unit tests that fail first
- Filling code that completes the test
- Refactoring
- Measuring code smell
- Measuring coverage
- Iteration process
- How Ballot.sol could be tested

References

https://www.agilealliance.org/glossary/tdd/

https://www.browserstack.com/guide/what-is-test-driven-development

Homework

- Create Github Issues with your questions about this lesson
- Read the references
- Get to know Hardhat and Ethers.js documentation
- (Optional) Study the sample content provide in hardhat setup
    - Greeter.sol contract
    - Test scripts
    - Deployment scripts
- (Optional) Complete the test scenarios for HelloWorld.sol marked as “TODO”