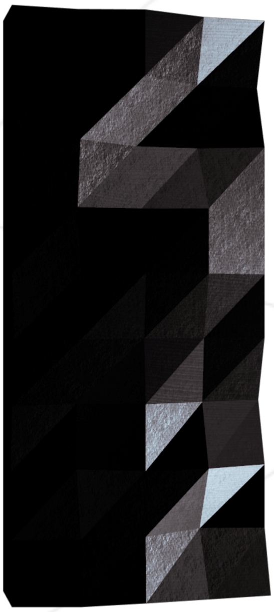


Tokenizer

Build your own token

Summary: This document is a Web3 related exercise.

Version: 1.1

Contents

Chapter I

Preamble

This subject is the production of a partnership between 42 and BNB Chain.

Build N Build (BNB) Chain is a distributed blockchain network upon which developers and innovators can build decentralized applications (DApps) as part of the move to Web3.

As of October 2022, BNB Chain is the world's largest smart-contract blockchain in terms of transaction volume and daily active users. At the time of writing, it has processed 3 billion transactions from 232 million unique addresses, and has an ecosystem of more than 1,500 active DApps. The decentralized nature of the network means anyone can build a product on BNB Chain without having to ask for permission, and potentially reach a massive audience.

You can retrieve Tbnb for free and with no minimum coins on your wallet via this faucet: BNB Chain Faucet

Figure I.1: https://www.bnbchain.org/

Chapter II

Introduction

Welcome to the exciting world of blockchain technology!

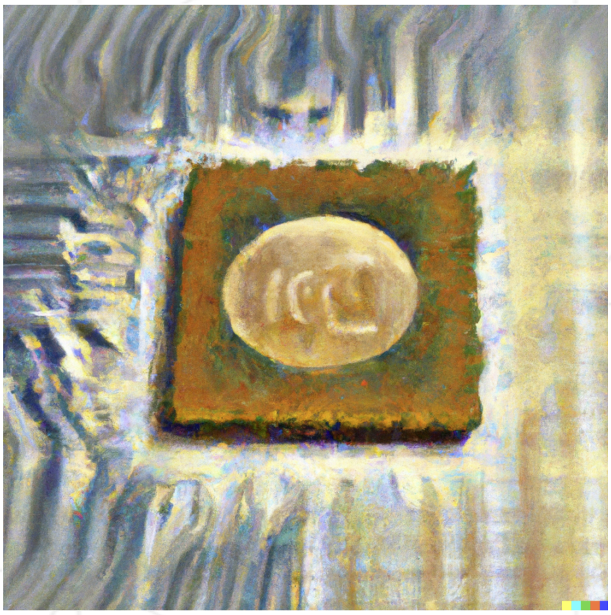
Have you ever dreamed of creating your own digital token?

Now is your chance to make that dream a reality.

Blockchain technology allows for the creation and distribution of unique digital assets, known as tokens. These tokens can represent a wide range of things, from a simple representation of currency to more complex assets like artwork or even a real-world asset. The process of creating your own token is not without its challenges, but with the right knowledge and resources, it can be a rewarding and fulfilling experience.

So, why wait?

Start your journey towards creating your very own token on the blockchain today!



Chapter III

Objectives

As a participant in this project, you will have the opportunity to contribute to the creation of a digital asset on the blockchain. This project is designed to challenge you in several areas, including your ability to master multiple programming languages and your familiarity with public blockchain technology.

While a strong background in cryptography is not required for this project, you should be prepared to learn and adapt as you work towards creating your own digital asset. This project will require you to think critically and creatively, as well as to push yourself out of your comfort zone as you navigate the complexities of blockchain technology.

Ultimately, your participation in this project will not only help you develop valuable skills and knowledge, but it will also allow you to be part of something truly innovative and exciting. Are you ready to take on the challenge?

Let's get started!

Chapter IV

Mandatory part

In order to create a token, there are several technical requirements that must be met.

> You are free to choose the name of your token. Your only constraint is to have 42 in it. It is of course forbidden to use insulting terms under penalty of punishment.

You must therefore create a `README.md` file at the root of your repository explaining the choices you had to make and the reasons why you made these choices.

> The language used is of course free, but you must respect the standards of the block chain you are going to use (for example ERC20 for ETH or BEP-20 for BSC).

First and foremost, you will need to choose a blockchain platform that supports the creation of tokens. There are many different options to choose from, each with its own unique features and capabilities.

Once you have selected a platform, you will need to become proficient in the programming language used by that platform in order to develop your token. Different platforms use different programming languages, so you will need to ensure that you have the necessary skills to work with the language of your chosen platform such as IDE, Truffle, Remix or Hardhat.

> Make sure you understand what you are doing. You will never be asked to use real money to do this project. There are test chains to avoid this problem.

You must submit the code used to create your token in a `code` folder located at the root of your repository. You should be careful to comment out your code and to use readable and explicit variable/function names.

During your evaluation there will be a code review.

You must be very careful about how you demonstrate the operation of your token. You must be able to perform minimalist actions to demonstrate its operation. You need to think about all aspects of security such as ownership or privileges.

You should also put all the things you need for the deployment part of your token in a second folder with the name you want.

After deploying your token on a public blockchain. You will define its ticker and publish it on a blockchain explorer (ex: blockscan or bscscan). Please mention the smart contract address and the network used, in your `Git` repository.

Finally, you should have a folder containing the documentation for this project. This folder is called `documentation` and should be at the root of your repository. It should be possible to understand how it works and what is needed to use your token.

You will need to have a clear understanding of how your token will be used and what it will represent. This may require the development of a whitepaper or other documentation outlining the features and functionality of your token.

You must take the time to make a clear and explicit documentation.
This will be reviewed during your evaluation.

Consider also creating a demo video to showcase your token and its features to potential users and investors.

If you want to make a video demo you don't have to push the video on your repository but a simple link will do!

Creating a demo video is not required. You will not get a better grade by creating this video.

Below is an example of the expected directory structure:

```
$> ls -al
total XX
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 .
drwxrwxrwt 17 wil wil 4096 avril 42 20:42 ..
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 README.md
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 code
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 deployment
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 documentation
```

Chapter V

Bonus part

To ensure the security of your token and prevent fraudulent activity, you may want to consider implementing a multisignature system, also known as a `multisig`.

This feature requires multiple parties to sign off on a transaction before it can be executed, providing an extra layer of protection for high-value assets or sensitive financial transactions.

Setting up a multisig system is easy using your preferred programming language by creating a smart contract that mandates multiple signatures for every transaction. Determine the number of signatures required and who is authorized to sign to enhance security and gain the trust of your token's users.

You must adapt this bonus to the mandatory part of this project.

⚠ The bonus part will only be assessed if the mandatory part is PERFECT. Perfect means the mandatory part has been integrally done and works without malfunctioning. If you have not passed ALL the mandatory requirements, your bonus part will not be evaluated at all.

Chapter VI

Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your folders and files to ensure they are correct.

Exceptionally for this project, we recommend that you share your project via your personal git account when your project is valid. Feel free to use different hashtags depending on the programming language used, but also web3 etc...