1. Usage (in PowerShell):
2. install pre-requisitories

npm install sha256 node-xlsx officegen

* 1. ~~On the first run, create the blockchain with the data built in the code, save the chain in the excel:~~

~~node ./main.js reset~~

* 1. count the number of blocks:

node ./main.js count

* 1. check the latest block:

node ./main.js latest

* 1. check the whole blockchain:

node ./main.js check

* 1. add a new block with a timestamp and a github link:

node ./main.js add 10.11.2022 <https://github.com/tibonto/dr/commit/50d0834deba2ce791772be7932055cf1a7bb9545>

working process:

* + 1. load the blockchain from the excel sheet (blockchain\_hist.xlsx)
    2. create a new block
    3. save the new block in Word (DR\_Blockchain\_Status/DR\_xxx\_yyyy\_mm\_dd.docx)
    4. generate an email with a template (email\_template.txt)

1. Test:
   1. Open PowerShell in the folder and Install dependencies:

npm install sha256 node-xlsx officegen

* 1. Reset the chain:

node ./main.js reset

* 1. Add the 39th block:

node ./main.js add 10.11.2022 <https://github.com/tibonto/dr/commit/50d0834deba2ce791772be7932055cf1a7bb9545>

* 1. Check the Word file in the DR\_Blockchain\_Status folder

1. File structure:
   1. main.js: deal with different command input
   2. blockchain.js: block and chain definition and functions regarding the blockchain
   3. word.js: code for Word generation
   4. email.js: code for Email generation
   5. blockchain\_hist.xlsx: saves the data of the blockchain (index, timestamp, github link)
   6. email\_template.txt: template for Email. The information of the latest block is inserted between the symbols “++++++” and “------”
   7. DR\_Blockchain\_Status: folder with the Word files containing the newly-added blocks. The index is generated based on the number of files starting with “DR” in this folder