Brownie Tutorial Note # remember to check sidechain testing later.

* How to initialize the Brownie contract
* Make a new folder named newproject or something else [mkdir newproject]
* Go to the new folder [cd newproject]
* Check the automatically created folders [dir]
* Know the meaning of each folder 🡪 brownie structure
  + /contracts: where you are going to have contracts, can be solidity or vyper
  + /interfaces: keep the basic resources for interfaces, not essential part of your project
  + /scripts: store python scripts used to deploy contracts or automate any common tasks or interactions
  + /tests: where you write your unit tests, if you run the brownie test, it will assume you tests are in the test folder and it is going to use the pi test framework for unit testing
  + You will see build and report folders, brownie will use those folders to generate files automatically, produce test results and reports for some json files. You do not need to put anything in these.
* For example, you can go to <https://github.com/brownie-mix> to check existing files. Try to run token-mix.
* Make a new working directory
* Run [brownie bake token]
* Check [cd token] then [dir], then everything we need
* In the scripts folder, there is a token test python file: [scripts\token.py], it is going to have a basic deployment scripts; by default it will have 18 decimal places
* It will automatically create brownie config yaml file: excluding safemath when evaluating test coverage
* Interact with basic brownie token 🡪 brownie comes with the full console [brownie console], it will realize that TokenProject is the active project.
* [accounts]: it will automatic ally open brownie local block and create 10 accounts that you can work with
* [accounts[0]]: it is the first account, most important one
* Get the basic console action [my\_token = Token.deploy('Test', 'TST', 18, 1e21, {'from':accounts[0]})]
* Then you will get the my\_token and interact with it using full power of web3 such as [my\_token.name()]
* You can transfer token from one to another: [my\_token.transfer(accounts[1], 1e10, {‘from’: accounts[0]})]; you can check the hash code of this transaction