Episode-07 - Sync, Async and SetTimeOutZero - Code

Refer to Code for Most of the insights. into this episode.

Highlights :

① const fs = require("node:fs")
- require can also be written like above
- This way of writing is just to represent that the module we are using is a CORE module.
- It is optional to write

② fs.readFileSync ("./gossip.txt", "utf8")
- This readFileSync is used to synchronously read file data. it does this by blocking the code, it does not let the below code to run unless this is completed.

③ Crypto.pbkdf2Sync ("password","salt", 5000000, 50, "sha512") → another core module "crypto"
- synchronous version of pbkdf2, here 2 means 2nd version.
- pbkdf → password based key derivation function
- Arguments → "password" → signifies password
  → "salt" → It is for encryption
  → 500000 → defines no. of iteration → proportional to how strong password will be
  → 50 → No. of key length
  → "sha512" → means which algorithm we want to use

Note: Synchronous functions doesn't have a callback

④ SetTimeOutZero → setTimeout(()=>{}, 0)
→ Trust issues with setTimeOut
→ It will only be called when call stack of main Thread is empty.

→ Explanation ⇒

Since the code for setTimeout zero is offloaded to libuv, then the callback function can only be executed by js engine only when the callstack is empty that is all synchronous code (that is global execution context is popped out of callstack i.e completed its execution) has been executed, hence therefore the callback function of this setTimeout is pushed to JS engine callstack only after callstack runs the whole code. (that is synchronous code)