Node.JS

Node modules

SmartPhone → Apps ← Shopping
                      Games
                      Financial
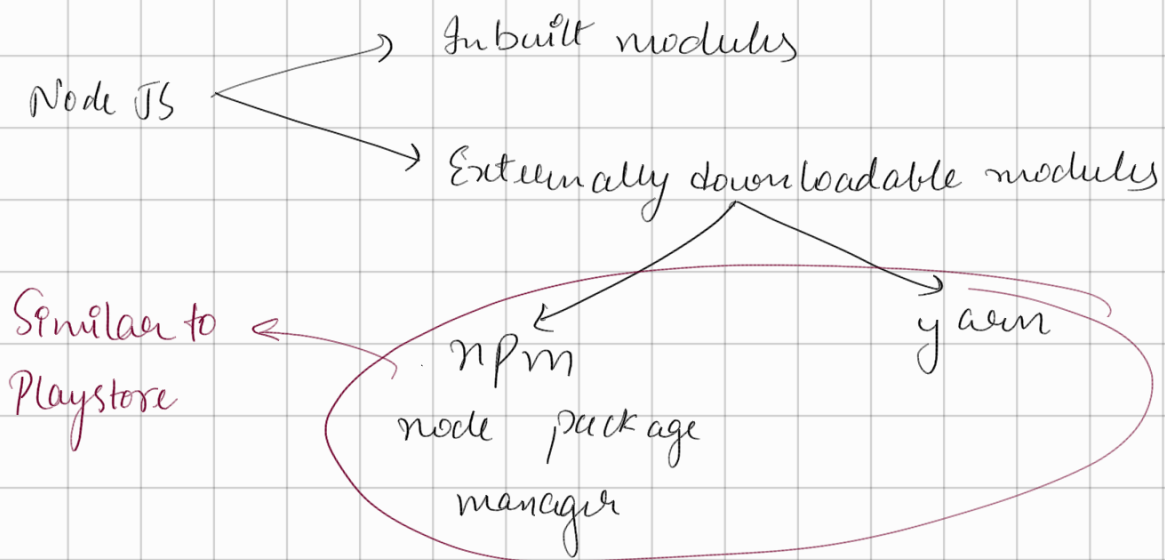   :

We cannot make one universal app for all functions.
Similarly, node modules are separate modules in Node JS

Node JS → Inbuilt modules
        → Externally downloadable modules

Similar to → npm   yarn
Playstore   node package
              manager

Inbuilt modules in Node JS

① Child process

to pass, keyword → `require (<___>)`
                              module
                              name

for child process module

        `require (child_process)`

without the quotes. & store it in any variable

Similar to opening calc or VS code in cmd

Eg: const cp = require('child-process')

    cp.execSync('calc')   // opens calculator
    cp.execSync('Code')   // opens VS code editor

Can also be used to handle websites & browsers

When running another file using cp.execSync.
the output comes in Buffer form.

→ ≈ Binary form.

→ To convert into string, use concatenation

② OS module ⟹ operating system interaction.

Eg: os.arch() ⟶ returns ×32 or ×64
    os.platform() ⟶ gives platform
    os.networkInterfaces() ⟶ network details
    os.cpus() ⟶ cpu core

✰ Larger modules.

① FS : file system module

    - reading ⟶ readFileSync (<filename/path>)
    - writing ⟶ writeFileSync (<filename/path>, data)
    - append ⟶ appendFileSync (<filename/path>, data)
    - delete ⟶ unlinkSync (<filename/path>)

for directories

   mkdirSync()

   rmdirSync()   } → argument = directory Name

Path

   existsSync() ⟶ true if exists or false

   lstatsSync() ⟶ gives statistics of a file or folder

   stats. isFile() ⟶ checks if passed arg is file

   -- , isDirectory() ⟶ checks if passed arg is directory

   fs.readDirSync(folderPath) ⟶ gives details of content in
                                                a directory

   path.basename(Path) ⟶ gives base name in path

   path.join(Path1, Path2) ⟶ gives joined path from given 2

   fs.copyfileSync(srcFile, destPath) ⟶ copies file from src to
                                                   destination.