Basics

* To check current node version and successful installation type **node -v** in CMD/Power shell/Terminal.
* Node.JS is a Javascript runtime built on Google’s open-source V8 Javascript Engine.
* It is a single-threaded event driven and non-blocking I/O model.
* It is perfect for building fast and scalable data-intensive apps.
* Basic CLI operations
  + In CLI enter **node** to start the virtual environment just like console in chrome known as REPL (Read-Eval-Print-Loop).
  + On pressing **tab key** on keyboard will provide all the global variables present in node.
  + To view all the variable and functions related to any Global variables, first right the variable and the press tab key.
    - **Example**

String. //Press tab key will provide all the string variable and functions.

* + **\_** is a variable which store previous results.
    - **Example**

3\*8

24

\_+6

30

* + To run a js file in the selected folder open cmd and type

**node** Filename.js

* + To exit use **ctrl + c**, or type **.exit**
* **Using NPM**
  + To initialize npm in any project just type **npm init** in CLI.
  + It will ask a bunch of information and at the end will create a **package.json** file.
  + There are two types of packages-
    - * Simple dependencies -> contains code on which we will write our code. Example = express.js.
        + **npm install** dependencyName
        + To install a specific version, use **npm install** dependencyName**@version Number**
      * Development dependencies-> used for building our code may be not required in prod.
        + **npm install** dependencyName**--save-dev**
        + To install anything globally (so that it is present throughout the system locally)

**npm install** dependencyName **--global**

* **Using Nodemon**
  + Nodemon is a tool used to make a node.js file run again on any change found in the src code.
  + It automatically checks for any change after save and re initializes the cli making life easier.
  + Behind the scenes it calls node filename.js again on any src code change.
  + If we install it globally then we can use it directly by using **nodemon filename.js**
  + But if installed locally we need to add same command in **npm start script** of package json.
* **NPM package versions**
  + Consider this line from package.json dependencies

"nodemon": "^2.0.7"

* + - ^ symbol means the project **will receive minor or patch releases**.
    - ~ symbol means the project will receive only patch releases.
    - \* symbol means the project will receive all update.
    - 2 is **major version** (huge new release that can break old versions).
    - 0 is **minor version** (minor enhancements/features which are backward compatible)
    - 7 is **patch version** (bug fixes).
    - To check for outdated packages according to what symbol(^/~/\*) we have provided, use **npm outdated.**
    - To update a package to latest version according to symbol (^/~/\*) used , use **npm update** dependencyName.
    - To uninstall a package use **npm uninstall** dependencyName.
* **How web works**
  + Timeline

    Description automatically generated
  + Diagram

    Description automatically generated