

File Edit Selection ... ← → ⚡ Node.js

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

powershell + ⌂ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌊ ⌋

```
PS D:\My Workspace1\Web Development(Delta)\WEB CODE\Node js\Day 28\Figlet> node ./index.js
  _\ \ / \ \ \ \ / / | _ / / _ \
 | | / - \ \ V / | _ \ | [ ] |
 [ ] / / \ \ \ | [ ] / \ \ /
```

```
PS D:\My Workspace1\Web Development(Delta)\WEB CODE\Node js\Day 28\Figlet>
```

main ⌁ ⌂ 0 ▲ 0 ⌂ 0 ⌂

Ln 10, Col 23 Spaces: 4 UTF-8 CRLF {} JavaScript ⌂ Go Live ⌂

## \* Library v/s Framework

- 1) Library is a collection of pre-written code that can be used to perform specific tasks.
- 2) e.g. Axios
- 3) e.g. express

## \* Express :

- A Node.js web application framework that helps us to make web application.
- It is used for server side programming.

## \* Usage : 1] Listen for incoming request

- 2] Parse
- 3] match response with routes
- 4] response

## \* For installation of express :

```
$ npm install express
```

## \* Getting Started with Express

```
const express = require("express");
const app = express();
let port = 3000;

* app.listen(port, () => {
    console.log(`app listen on port ${port}`);
})
```

// For request

Port are the logical endpoints of a network connection that is used to exchange information bet<sup>n</sup> a web server & web client.

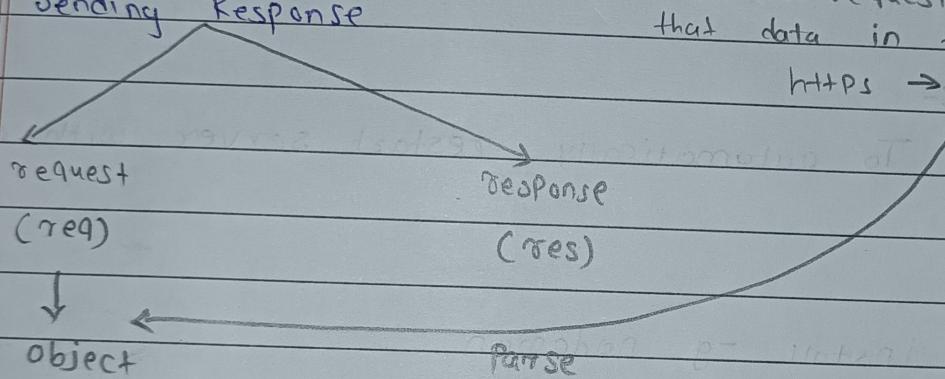
`app.listen()` : listen fn create new web server for accepting request.

\* ~~for~~ `use` line of code is used to mount middleware F<sup>n</sup> in the app's request-response cycle

```
app.use(req, res) => {
  console.log("Request received");
}
```

(Listen, request) (Handle Req)

\* Sending Response  
When we request to other API, website that data in text format  
https → text



```
res.send("This is basic response");
```

The body parameter can be buffer object, string, object, boolean, or an array.

\* Routing :

it is a process of selecting a path for traffic in a network or bet<sup>n</sup> or access multiple network.

```
app.get("/", () => {
    res.send("this is root path");
});
```

```
app.post("/", () => {
    res.send("this is root path");
});
```

if you enter path that not exist that resopnse

using

```
app.get("*", () => {
    res.send("this path doesn't exist");
});
```

#### \* Nodemon :-

To automatically restart server with code changes.

```
$ npm install -g nodemon
```

```
$ nodemon -v
```

for run :

```
nodemon filename.js
```

#### \* Path- Parameter :

req. Param

```
app.get("/ig/:username", (req, res) => {
```

```
let $username = req.param;
```

```
res.send(`This account belongs to @${username}`);
```

```
});
```

\* query string:

req. query

app.get("/search", (req, res) =>

{

let { q } = req.query;

if (!q)

res.send("No search query");

~~res.send(`These are the result for: \${q}`);~~

});