**NODE JS**

Node.js is an open-source and cross-platform runtime environment built on Chrome’s V8 JavaScript engine for executing JavaScript code outside of a browser.

It provides an event-driven, non-blocking (asynchronous) I/O and cross-platform runtime environment for building highly scalable server-side applications using JavaScript.

Event – Driven: The event driven system means the control flow of the program is depending

PHP file request

1. Sends the task to the computer's file system.
2. Waits while the file system opens and reads the file.
3. Returns the content to the client.
4. Ready to handle the next request.

Node JS handling the file

1. Sends the task to the computer's file system.
2. Ready to handle the next request.
3. When the file system has opened and read the file, the server returns the content to the client.

What is a Module?

A modules are like JavaScript libraries that consists of set of functions which we can use in our application.

JavaScript files are considered as modules.

To include a module we use require() function with name of module. Require() function is local to each module.

**Module Wrapper Function**

Function(exports, require, model, \_\_filename, \_\_dirname)

Node must not be used for CPU-intensive apps it must be used for data intensive and network intensive apps.

In node we have global object “global”.

Path module

const path = require('path');

basename\_ext = path.basename('D:\\NodeJS\\content\\subfolder\\text.txt');

//output: text.txt

basename\_no\_ext = path.basename('D:\\NodeJS\\content\\subfolder\\text.txt', '.txt');

//output: text

console.log(basename\_ext);

console.log(basename\_no\_ext);

dirname\_ = path.dirname('content\\subfolder\\text.txt');

console.log(dirname\_);

//content/subfolder

console.log(path.extname('app.js')); //.js

/\*

path.format(pathObject)

The path.format() method returns a path string from an object.

priority rules:

1. pathObject.root is ignored if pathObject.dir is provided

2. pathObject.ext and pathObject.name are ignored if pathObject.base exists

\*/

//root is ignored as dir is provided.

let path\_1 = path.format({

    root: '/ignored',

    dir: '/home/user/dir',

    base: 'file.txt'

  });

console.log(path\_1); // /home/user/dir\file.txt

// ext is ignored as base is provided.

let path\_2 = path.format({

    root: '/',

    base: 'file.txt',

    ext: 'ignored'

  });

console.log(path\_2);

// name + ext is used as base not provided.

let path\_3 = path.format({

  root: '/',

  name: 'file',

  ext: '.txt'

});

console.log(path\_3);

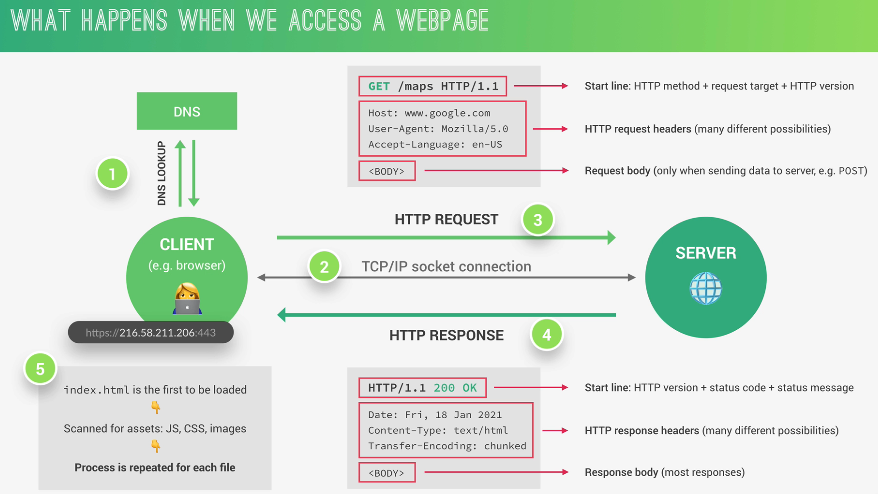
OS module

const os = require('os');

let totalMemory = os.totalmem(); //returns total memory in bytes.

let freeMemory = os.freemem(); //unit bytes

console.log(`Total Memmory is ${totalMemory} and freeMemory is ${freeMemory}`);



**Middleware**

**Middleware** functions are functions that have access to the [request object](https://expressjs.com/en/4x/api.html" \l "req) (req), the [response object](https://expressjs.com/en/4x/api.html" \l "res) (res), and the next function in the application’s request-response cycle. The next function is a function in the Express router which, when invoked, executes the middleware succeeding the current middleware.