Ryan Dahl created Node

* node js - A runtime environment for executing javascript code outside of browser
* often use node to build back end services (API - Application Programming Interfacse)
* build highly scalable , data intensive and real time apps

browser is js engine

browser provide runtime environment for javascript code

js code -> js engine -> Machine code

microsoft use Chakra

firefox use SpiderMonkey

chrome use v8

* node.exe = v8+cpp
* Non-Blocking (Asynchronous ) nature of node - by default
* asp.net uses synchronous architechture as one req has single thread
* Node is ideal for i/o intensive apps but not for cpu intensive apps

console.log(window);

in node there is not window or document objects these are part of runtime envinronment we get from browsers

* Node Module System-

os, ts, events, Http

// window

console.log() ----- window.console.log()

setTimeout() --------- -||-

clearTimeout() --------- -||-

setInterval() - call function in intervals --------- -||-

clearInterval() --------- -||-

in node there is no window object instead there is global

//global

console.log() ----- global.console.log()

setTimeout() --------- -||-

clearTimeout() --------- -||-

setInterval() - call function in intervals --------- -||-

clearInterval() --------- -||-

var message = "";

console.log(global.message);

=> undefined

var and functions are not added in global object . they are only scope to the file . they are not available out side of the file app.js

Need of Module

two variables or two functions with the same name dont overrride in another var or func defined somewhere else. they are encapsulated in modules

In node every file is module and var/ func defined in that module is private nd can use only in that module

Creating a module

extra: $jshint app.js - to get details about error

* Module wrapper function

(function (exports, require, module, \_\_filename, \_\_dirname)

{

})

//os Module

// const { clear } = require("console");

// const path = require("path");

// var pathObj = path.parse(\_\_filename);

// console.log(pathObj);

const os = require("os");

var totalmemmory = os.totalmem();

var freememmory = os.freemem();

console.log("Total memory: " + totalmemmory);

//Template string - ES6/ES2015 - ECMAScript 6

console.log(`Total memory: ${totalmemmory}`);

console.log(`Free memory: ${freememmory}`);

npm init

it will create json file

https://www.npmjs.com/package/superheroes