Variables Globales

USANDO DOTENV

npm init

npm i dotenv –D

Verifica el pakage.json que se instaló la librería

Crea un archivo en el mismo directorio .env con las variables globales

PORT=4000*;*

password=123

TOKEN=1233321

Crea un archivo variable.js

**require**('dotenv').**config**();

console.**log**(process.env.PORT);

Ejecuta en la terminal

node variablesGlobales.js

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

USANDO EXPORT

Crea un archivo llamado área.js

const PI = 3.14;

function **areaC**(radio) {

    return PI \* radio \*\* 2;

}

module.exports.**areaC** = **areaC**;

y un Segundo archivo llamado importacion.js

let area = **require**('./variablesglobales2')

console.**log**('area ' + area.**areaC**(10));

Modifica area.js

const PI = 3.14;

function **areaC**(radio) {

    return PI \* radio \*\* 2;

}

function **z1**(a) {

    return "hola";

}

module.exports = { **areaC**, **z1** }

module.exports.PI = PI;

También importancion.js

**require**('dotenv').**config**();

console.**log**(process.env.PORT);

let area = **require**('./variablesglobales2')

console.**log**('area ' + area.**areaC**(10));

console.**log**(area.PI);

console.**log**(area.**z1**());

ejemplo de lectura y escirtura asincrono

const **fs** = **require**('fs'); *//incluimos la libreria para flujo de archivo: lectura y escritura*

var sensor = "Este es un ejemplo de lectura y escritura"

**fs**.**writeFile**("archivo2.txt", sensor, function(err) { *//creamos un archivo no bloqueo*

    if (err) {

        console.**log**(err);

    } else {

        console.**log**("Estoy guardando la información");

    }

});

**fs**.**readFile**('archivo2.txt', 'utf-8', (err, data) => {

    if (err) {

        console.**log**('error: ', err);

    } else {

        console.**log**(data);

    }

});

Typeof

console.**log**(typeof 03 - 03 - 2222);

NaN

var x = 5 / 0;

console.**log**(x);

infinite

var x;

console.**log**(x);

undefine

var **x** = () => {}

console.**log**(**eval**(**x**));

var x = 4.4;

console.**log**(**parseInt**(x));

var x = 4.4;

console.**log**(**parseFloat**(x));

var x = 5 / 0;

if (**isFinite**(x))

    console.**log**(x);

let now = new **Date**();

console.**log**('La fecha actual es', now);

console.**log**('UNIX time:', now.**getTime**());

console.**log**(typeof now)

let arreglo = [1, 2, 3];

console.**log**(typeof arreglo);

let objeto = {

    id: "hola",

    nombre: "ivan"

}

console.**log**(typeof objeto);

npm I express

const express = require('express')

const app = express()

const port = 3000

app.get('/', (req, res) => {

res.send('Hello World!')

})

app.listen(port, () => {

console.log(`Example app listening on port ${port}`)

})

npm install --save bcrypt

const bcrypt = require('bcrypt');

// encriptamos la cadena "chaomundo"

const plaintext = "chaomundo";

const salt = 10;

const hash = bcrypt.hashSync(plaintext, salt);

console.log(plaintext);

console.log(hash);

// validamos cual de las dos coincide con el cifrado anterior

const compA = bcrypt.compareSync("holamundo", hash); // true

const compB = bcrypt.compareSync("chaomundo", hash); // false

console.log(compA);

console.log(compB)