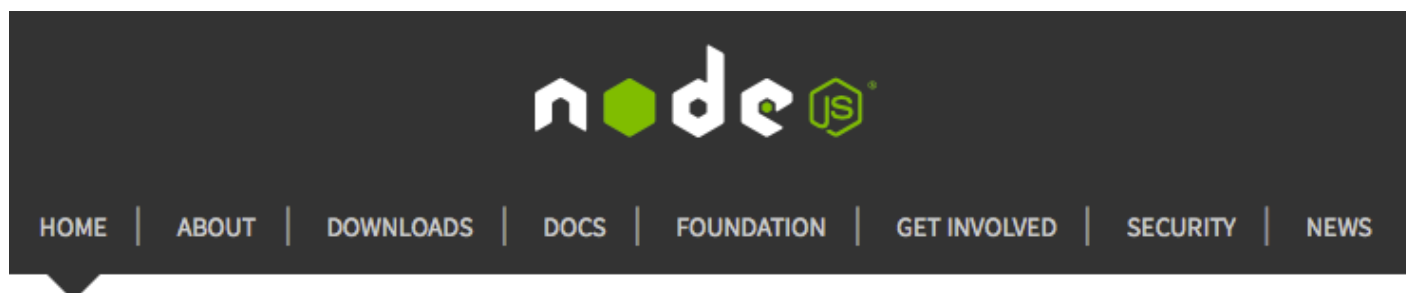


Ejercicios Tema 1

Ejercicio 1

En primer lugar vamos a ver como instalar Node.js. Para ello accedemos a la página web: <http://nodejs.org/>

Nos aparecerá una ventana como la siguiente:



Node.js® is a JavaScript runtime built on **Chrome's V8 JavaScript engine**. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, **npm**, is the largest ecosystem of open source libraries in the world.

Download for OS X (x64)



[Other Downloads](#) | [Changelog](#) | [API Docs](#) [Other Downloads](#) | [Changelog](#) | [API Docs](#)

Pulsando el botón *download* se descargará un ejecutable que siguiendo los pasos nos debería instalar correctamente Node.js en nuestro ordenador.

Para comprobar que ha sido instalado podemos poner el siguiente comando en nuestro terminal o ventana de comandos en windows: `node --version`

Este comando nos mostrará la versión de Node.js instalada. Si es así, ya podemos introducir comandos JavaScript en nuestra consola de Node.js. En primer lugar ponemos `node` para que se abra la terminal de Node.js dentro de nuestra terminal.

Después podemos escribir comandos como: `console.log("Prueba en Node.js")`

Ejercicio 2

En este ejercicio vamos a hacer una calculadora a través del navegador, de forma que introduciendo diferentes rutas y parámetros nos devuelva el resultado, por ejemplo:

<http://localhost:3000/suma?numero1=4&numero2=1> devolverá $4+1 = 5$

<http://localhost:3000/resta?numero1=20&numero2=15> devolverá $20-15 = 5$

<http://localhost:3000/mult?numero1=3&numero2=2> devolverá $3*2 = 6$

<http://localhost:3000/div?numero1=10&numero2=2> devolverá $10/2 = 5$

Para ello, en primer lugar vamos a crear el módulo que realiza las operaciones al igual que vimos en teoría. Este fichero se llamará **operaciones.js**:

```
function suma(a,b){
    return parseInt(a) + parseInt(b);
}

function resta(a,b){
    return parseInt(a) - parseInt(b);
}

function mult(a,b){
    return parseInt(a) * parseInt(b);
}

function div(a,b){
    return parseInt(a) / parseInt(b);
}
```

Por último sólo nos quedará añadir la función export para poder acceder a las funciones creadas del módulo

```
module.exports = {
    suma: suma,
    resta: resta,
    mult: mult,
    div: div
}
```

Ha llegado el momento de crear nuestro servidor. Lo haremos en otro fichero que llamaremos `server.js` y en primer lugar contendrá lo siguiente:

```
var http = require("http");
var url = require("url");
var operaciones = require('./operaciones');
```

Importamos los módulos `http` para crear el servidor como vimos en la teoría. También importaremos `url` para poder recortar la url que se ha introducido a nuestro antojo. Y por último importaremos nuestro modulo recién creado `operaciones`.

Una vez hecho esto, crearemos en primer lugar el esqueleto de nuestro servidor:

```
var http = require("http");
var url = require("url");
var operaciones = require('./operaciones');

var server = http.createServer(function (peticion, respuesta){

    //TODO Comprobar la ruta introducida para saber de qué función se trata.
    //TODO Obtener primer argumento.
    //TODO Obtener segundo argumento.

    //TODO Escribir la cabecera de la respuesta.

    //TODO Comprobar de qué función se trata para realizar la operación correspon-
    diente.
    //TODO Escribir la respuesta que devolverá cada operación.

    respuesta.end();
});

server.listen(3000, function(){
    console.log("tu servidor está listo en " + this.address().port);
});
```

En primer lugar tenemos que obtener la url que ha sido llamada, para ello tenemos la el método `parse` del módulo `url`. Esto crea un objeto del cual podemos obtener los parámetros que necesitamos:

```
var pathname = url.parse(peticion.url, true).pathname;
var num1 = url.parse(peticion.url, true).query.numero1;
var num2 = url.parse(peticion.url, true).query.numero2;
```

Después escribimos la cabecera:

```
respuesta.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'});
```

Con todo preparado ya sólo nos quedaría añadir las funciones pertinentes:

```

if (pathname == "/suma") {
    respuesta.write(num1 + "+" + num2 + " = " + operaciones.suma(num1, num2));
} else if (pathname == "/resta") {
    respuesta.write(num1 + "-" + num2 + " = " + operaciones.resta(num1, num2));
} else if (pathname == "/mult") {
    respuesta.write(num1 + "*" + num2 + " = " + operaciones.mult(num1, num2));
} else if (pathname == "/div") {
    respuesta.write(num1 + "/" + num2 + " = " + operaciones.div(num1, num2));
}

```

El código completo de nuestro servidor quedaría así:

```

var http = require("http");
var url = require("url");
var operaciones = require('./operaciones');

var server = http.createServer(function (peticion, respuesta) {

    var pathname = url.parse(peticion.url, true).pathname;
    var num1 = url.parse(peticion.url, true).query.numero1;
    var num2 = url.parse(peticion.url, true).query.numero2;

    respuesta.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'});

    if (pathname == "/suma") {
        respuesta.write(num1 + "+" + num2 + " = " + operaciones.suma(num1, num2));
    } else if (pathname == "/resta") {
        respuesta.write(num1 + "-" + num2 + " = " + operaciones.resta(num1, num2));
    } else if (pathname == "/mult") {
        respuesta.write(num1 + "*" + num2 + " = " + operaciones.mult(num1, num2));
    } else if (pathname == "/div") {
        respuesta.write(num1 + "/" + num2 + " = " + operaciones.div(num1, num2));
    }
    respuesta.end();

});

server.listen(3000, function() {
    console.log("tu servidor está listo en " + this.address().port);
});

```

Para ejecutar el servidor nos desplazamos en el terminal hasta la carpeta en la que hemos creado nuestros ficheros. Y ejecutamos el comando:

Si funciona adecuadamente nos mostrará el mensaje: “tu servidor está listo en 3000”

Y para probarlo sólo nos quedará realizar llamadas al servidor como las mostradas de ejemplo al principio del ejercicio 2.