

Ejercicios prácticos

Node.js: Introducción a la programación back-end.

Actividades

1.

Ingresa a la consola de Node con el comando `node` y realiza las siguientes acciones:

- a) Realiza una suma, por ejemplo $10 + 10$.
- b) Imprime el mensaje "Estoy aprendiendo Node.js" por pantalla utilizando `console.log`
- c) Declara una variable `X` y asignale el valor 10. Imprime por consola el valor de esa variable utilizando `console.log`.
- d) Declara una variable `Y` y asignale el valor 4. Sumale 15 a esa variable. Imprime el nuevo valor de `Y` utilizando `console.log`.
- e) Declara una variable que contenga el siguiente array `[23,45,76]`. Imprime por pantalla la primera posición del array.

2.

Creá un programa que imprima un saludo por pantalla. Guardalo como `saludo.js` y ejecutalo desde la consola de node.

3.

Creá un programa que declare un array con números del 1 al 20. Luego realiza un ciclo para recorrer ese array y mostrar todos los números por pantalla. Guarda este programa como `numeros.js` y ejecutalo desde la consola de node.

4.

- a) Escribí el comando para crear el `package.json`.
- b) ¿Recordás para que sirve y la importancia que tiene este archivo?

5.

- a) Escribí el comando para instalar una dependencia.
- b) Si quisieras almacenar esa dependencia en el `package.json` ¿Cómo sería el comando?

6.

¿De qué manera importarías en tu programa una dependencia llamada “dependencia-loca”?

7.

- a) Instalá la dependencia shuffle-array utilizando npm.
- b) Lee la documentación de esta dependencia:
<https://www.npmjs.com/package/shuffle-array>. Lo que hace es mezclar los elementos de un array al azar.
- c) Creá un programa y llamalo mezclador.js.
- d) Importá la dependencia que instalaste.
- e) Creá un array con los números del 1 al 15, llamalo como quieras.
- f) Utilizá la dependencia shuffle-array para mezclar ese array. Luego mostrarlo por pantalla por console.log
- g) Guardá ese programa y ejecutalo en la consola de node.

8.

Identificá la ruta de estas URL:

- a) <https://home.mercadolibre.com.ar/computacion>
- b) <https://computacion.mercadolibre.com.ar/monitores>
- c) <https://www.despegar.com.ar/vuelos>
- d) <https://www.apple.com/iphone>

9.

Escribí el código necesario para importar la dependencia express y crear una nueva instancia de una aplicación Express.

10.

Dado este código:

```
var express = require('express');  
var app = express();
```

¿Qué hace falta agregarle para que tu aplicación comience a escuchar pedidos?

11.

Completa la siguiente función para que cuando se realice un pedido GET a la ruta “/despedida” se envíe como respuesta el mensaje “Hasta luego”.

```
app.get(_____, function(req, res) {  
});
```

12.

Completá la siguiente función para que cuando se realice un pedido GET a la ruta “/sumar” se realice una suma (con los números que quieras) y se envíe como respuesta el resultado de la suma a través del mensaje: “El resultado de la suma es ..”.

```
app.get(_____, function(req, res) {  
});
```

13.

Completá la siguiente función para que cuando se realice un pedido GET a la ruta “/promedio” se calcule el promedio (de 3 números que quieras) y se envíe como respuesta el resultado a través del mensaje: “El promedio es ..”.

```
app.get(_____, function(req, res) {  
});
```

14.

Identificá los parámetros que se envía a través de las siguientes URL. Determiná si son parámetros de ruta (path params) o parámetros de consulta (query params)

- a) <http://www.imdb.com/title/tt0120338/>
- b) <https://www.acamica.com/cursos?tags=arduino>
- c) <http://www.villagecines.com/peliculas/177>
- d) <https://www.acamica.com/cursos?difficulty=intermediate>

15.

Se tiene la siguiente URL:

www.venta-productos.com

¿Cómo la completaría para que contenga un query param que envíe el parámetro “categoría” con el valor “hogar”?

16.

Se tiene la siguiente URL:

www.mis-series.com

Se quieren filtrar las series de esta página por dos criterios:

- Género: Drama
- Temporada: 2

Cómo la completaría esa URL para que envíe a través de query params esos parámetros y valores.

17.

Completá el código para que se puedan obtener los parámetros enviados por la URL.

- a) Se hace un pedido a la ruta /estudiantes?año=5&curso=C

```
app.get("_____", function(req, res) {  
    var año = _____  
    var curso = _____  
});
```

- b) Se hace un pedido a la ruta /socio/18

```
app.get("_____", function(req, res) {  
    var id = _____  
});
```

c) Se hace un pedido a la ruta /restaurant?tipo=italiano&zona=centro

```
app.get("_____", function(req, res) {  
    var tipo = _____  
    var zona = _____  
});
```

d) Se hace un pedido a la ruta /paciente/3570202

```
app.get("_____", function(req, res) {  
    var documento = _____  
});
```

18.

Se tiene el siguiente código:

```
app.get('/libros', function(req, res) {  
  
});
```

Si se realiza un pedido a la ruta: "/libro?autor=dan_brown&editorial=bolsillo, completá el código para almacenar en una variable editorial y en otra libro los parámetros enviados.

19.

Se tiene el siguiente código:

```
app.get('/descuento', function(req, res) {  
  
});
```

Cuando se le realiza un pedido a la ruta 'descuento' calcula el 25% del valor enviado.
Si se realiza un pedido a la ruta: "/descuento?valor=1000, completá el código para obtener el valor enviado, calcular el descuento y enviarlo como respuesta.

20.

Se tiene el siguiente código:

```
var dias_semana = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"];
```

```
app.get('_____', function(req, res) {
```

```
});
```

Completá el código para que la ruta /semana reciba un parámetro de ruta llamado "dia". Si se realiza un pedido a esa ruta, debe enviar como respuesta el correspondiente día de la semana. Por ejemplo, un pedido a la ruta "/semana/2" debería enviar el segundo día de la semana, "Martes".

21.

Escribí el código necesario para determinar que cuando se realice un pedido GET a la ruta "/resta" realice las siguientes acciones:

- Obtenga los parámetros enviados a través de la URL. Los pedidos tienen la siguiente forma "/resta?numero1=32&numero2=44"
- Solo realice la resta "numero1 - numero2" si el resultado no va a ser un número negativo. En el caso de que lo sea, es decir, que numero2 sea mayor a numero1, no debe realizar la resta y enviar el siguiente mensaje: "No se pudo realizar la resta". En el caso de que pueda realizarse, debe enviar el siguiente mensaje: "El resultado de la resta es..."

22.

Escribí el código necesario para determinar que cuando se realice un pedido GET a la ruta "/ciclo" realice las siguientes acciones:

- Obtenga el parámetro enviado a través de la URL. Los pedidos pueden ser de la siguiente forma "/ciclo" o "/ciclo?numero=4".
- Si se envía el parámetro "numero", se debe realizar un ciclo for que cree un array con los números del 0 al número enviado. Si no se envía ningún parámetro, se debe crear por defecto, un array con los números del 0 al 3. En ambos casos, se debe enviar como respuesta el array obtenido.

Soluciones

1.

- a) `10 + 10`
- b) `console.log("Estoy aprendiendo Node.js");`
- c) `var x = 10`
`console.log(x)`
- d) `var y = 4`
`y = y + 4`
`console.log(y)`
- e) `var array = [23,45,76]`
`console.log(array[0])`

2.

En el programa, escribí `console.log("Hola a todos")`. Guardá el archivo como `saludo.js`. Luego, ejecutalo desde la consola haciendo `node saludo.js`.

3.

En el programa, vas a tener el siguiente código que declara un array y lo recorre.

```
var array = [1,2,3,4,5];

for (var i = 0; i < array.length; i++) {
    console.log(array[i]);
}
```

Guardá el archivo como `numeros.js`. Luego ejecutalo desde la consola haciendo `node numeros.js`.

4.

- a) El comando es npm init.
- b) El package.json sirve para almacenar toda la información relevante de un proyecto. Entre otras cosas, contiene el nombre del proyecto, la versión, el repositorio en el cuál está subido, su descripción y las dependencias que utiliza. Es muy importante utilizarlo y mantenerlo actualizado porque lleva el registro de las dependencias que utiliza nuestro proyecto. Si en algún momento queremos compartir ese proyecto, la persona que lo quiera ejecutar, va a necesitar instalar todas las dependencias que utilizar. Para saber cuáles son, no va a leer todo el proyecto y encontrarlas en el código, sino que va a instalar las que estén declaradas en el package.json.

5.

- a) npm install [dependencia]
- b) npm install --save [dependencia]

6.

```
var dependencia-locas = require("dependencia-locas");
```

7.

Para instalar la dependencia utiliza npm install shuffle-array o npm install --save shuffle-array para almacenarla en el package.json.

Tu código debería quedar así:

```
var shuffle = require("shuffle-array");  
var array = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15];  
shuffle(array);  
console.log(array);
```

8.

Identificá la ruta de estas páginas web:

- a) computacion
- b) monitores

- c) vuelos
- d) iphone

9.

```
var express = require('express');  
var app = express();
```

10.

Para que la aplicación comience a escuchar pedidos, necesitamos utilizar la función `listen`. Esta función debe recibir por parámetro el puerto en el que se van a escuchar los pedidos.

```
app.listen(8080);
```

El parámetro puerto es obligatorio, pero también se le puede pasar como parámetro una función `callback`.

```
app.listen(8080, function(){  
  console.log("Escuchando pedidos en el puerto 8080");  
});
```

11.

```
app.get('/despedida', function(req, res) {  
  res.send('Hasta luego');  
});
```

12.

```
app.get('/sumar', function(req, res) {  
  var suma = 10 + 10;  
  res.send('El valor de la suma es '+ suma);  
});
```

13.

```
app.get('/promedio', function(req, res) {  
  var promedio = (10 + 5 + 6)/3;  
  res.send('El promedio es ' + promedio);  
});
```

14.

- a) Se envía el parámetro tt0120338, correspondiente con el id de la película que se está mostrando. Es un path param.
- b) Se envía el parámetro "tags" con el valor "arduino". Es un query param.
- c) Se envía el parámetro 177, correspondiente con el id de la película que se muestra. Es un path param.
- d) Se envía el parámetro "difficulty" con el valor "intermediate". Es un query param.

15.

www.venta-productos.com?categoria=hogar

16.

www.mis-series.com?genero=dama&temporada=2

17.

- a)

```
app.get("/estudiantes", function(req, res) {  
  var año = req.query.año;  
  var curso = req.query.curso;  
});
```
- b)

```
app.get("/socio/:id", function(req, res) {  
  var id = req.params.id;  
});
```
- c)

```
app.get("/restaurant", function(req, res) {  
  var tipo = req.query.tipo;  
  var zona = req.query.zona;
```

```
});
```

```
d) app.get("/paciente/:documento", function(req, res) {  
    var documento = req.params.documento;  
});
```

18.

```
app.get('/libros', function(req, res) {  
    var autor = req.query.autor;  
    var editorial = req.query.editorial;  
  
});
```

19.

```
app.get('/descuento', function(req, res) {  
    var valor = req.query.valor;  
    var descuento = valor * 0,25;  
    res.send(descuento);  
});
```

20.

```
var dias_semana = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"];
```

```
app.get('/semana/:dia', function(req, res) {  
    var dia = req.params.dia;  
    var respuesta = dias_semana[dia - 1];  
    res.send('El día es ' + respuesta);  
});
```

21.

```
app.get('/resta', function(req, res) {  
    var numero1 = req.query.numero1;  
    var numero2 = req.query.numero2;  
    if (numero2 > numero1){  
        res.send('No se puede realizar la resta');  
    }else{  
        var resta = numero1 - numero2;  
        res.send('El resultado de la resta es '+ resta);  
    }  
});
```

22.

```
app.get('/ciclo', function(req, res) {  
    var numero = req.query.numero;  
    if (numero){  
        var array = [];  
        for(var i = 0; i <= numero; i++){  
            array[i] = i;  
        }  
    }else{  
        var array = [0,1,2,3];  
    }  
    res.send(array);  
});
```