
NodeJS: Sintaxis

Gabriel Rodríguez Flores

October 4, 2022

- Repaso de programación y sintaxis
 - Variables
 - Operadores
 - Bucles
 - Funciones
- Ejercicios
- Entrega por Repo+Replit

Contents

1	Teoría	3
1.1	Qué es NodeJS	3
1.2	Ejecución de NodeJS	3
1.2.1	REPL	3
1.2.2	Ficheros	3
1.3	Recursos	3
1.4	Sintaxis	4
1.4.1	Comentarios	4
1.4.2	Variables	4
1.4.3	Operadores	6
1.4.4	Condicionales	6
1.4.5	Bucles	7
2	Ejemplos	8
2.1	Comprobar instalaciones	8
2.2	Ejemplo sintaxis	8
2.3	Ejemplo variables	9
3	Ejercicios	9
4	Entregables	9
4.1	En clase	9
4.2	Tarea	9

1 Teoría

1.1 Qué es NodeJS

Intérprete de Javascript en el servidor.

Según la web oficial:

Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de Node.js, npm, es el ecosistema mas grande de librerías de código abierto en el mundo.

1.2 Ejecución de NodeJS

1.2.1 REPL

Escribe `node` en la consola y ¡ a jugar !

- Puede usarse de calculadora
 - `'_'` recoge el valor inmediatamente anterior
- Orientado a hacer pruebas rápidas y de sintaxis

1.2.2 Ficheros

Se invoca el fichero existente precedido del comando `node`

```
1 echo "console.log('Hola Mundo !')" > hello.js
2 node hello.js
```

1.3 Recursos

- W3School
- Cheat sheet
- Tutorial Completo
- Apuntes y guía de aprendizaje

1.4 Sintaxis

- No se tienen en cuenta los espacios en blanco y las nuevas líneas
- Se distinguen las mayúsculas y minúsculas
- No es un lenguaje tipado: No se define el tipo de las variables
- No es necesario terminar cada sentencia con el carácter de punto y coma ; ¡Pero si es recomendable!

1.4.1 Comentarios

```
1 // Comentario Javascript
2
3 /* Comentario Javascript */
4
5 /**
6  * Comentario estilo documentación
7  */
```

1.4.2 Variables

Referencia

1.4.2.1 Vida de las variables

```
1 variableGlobal = "global";
2 var variableLocal = "local"; // fichero
3 let variableBloque = "bloque";
4 const constante = "constante"; // inmutable
```

• Ejemplo let

```
1 let hello = "say Hi";
2
3 if (true) {
4   let global = "global";
5   let hello = "say Hello instead";
6   console.log(hello);
7 }
8
9 console.log(hello);
10 console.log(global);
```

• Ejemplo var

```
1 var hello = "say Hi";
2
3 if (true) {
4   var global = "global";
5   var hello = "say Hello instead";
6   console.log(hello);
7 }
8
9 console.log(hello);
10 console.log(global);
```

Demostrar error con const de lo anterior

- Ejemplo const

```
1 const hello = { saludo: "hola", eliminar: true};
2
3 hello.saludo = "ey";
4 hello.despedida = "adios";
5 delete hello.eliminar;
6
7 console.log(hello);
```

1.4.2.2 Tipos de variables Referencia

- Ejemplo de cada tipo usando typeof

```
1 let numero = 123;
2 let objeto = {};
3 let booleano = true; // Truthy ?
4
5 let texto = "";
6 let texto2 = '';
7 let texto3 = `literal ${texto}`;
8
9 let array = new Array("", 123, {});
10 let array2 = ["", 123, {}];
11
12 console.log(`
13   numero: ${typeof numero}
14   objeto: ${typeof objeto}
15   booleano: ${typeof booleano}
16   texto: ${typeof texto}
17   array: ${typeof array}
18 `);
19 console.log('hola');
20 console.log(`
21   hola
```

```
22   `);
```

1.4.3 Operadores

Referencia

Ejemplo comparadores

```
1 1 == '1' // true
2 1 === '1' // false
```

- Ejemplos matemáticos

```
1 const a = 2 ** 10;
2 console.log(a);
```

- Ejemplo operador && y ||

```
1 const x = undefined;
2 const y = 23;
3
4 const a = x && 'no';
5 const b = y && 'yes';
6 const c = x || 'yes';
7 const d = y || 'no';
8
9 console.log(a); // undefined
10 console.log(b); // yes
11 console.log(b); // yes
12 console.log(c); // 23
```

Tip: Se puede aplicar el acumulador a los operadores:

```
1 const x &= 'no';
```

1.4.4 Condicionales

- if-else

```
1 if(true){
2   // TODO ...
3 }
4
5 if(false){
6
```

```
7 }else{
8   // TODO ...
9 }
10
11 if(false){
12
13 }else if(true){
14   // TODO ...
15 }else{
16
17 }
18
19 switch(){
20   case 'value':
21     // TODO ...
22     break;
23   default;
24 }
```

- **Ejemplo True/Truthy y False/Falsy**

- Ver aqui
- Resume
- Otros ejemplos

```
1 const a = 7;
2 a === a // true
3 a === !!a // false
```

1.4.5 Bucles

- **Ejemplo de for**

```
1 for(let i; i<10; i++){
2   // TODO 10 times ...
3 }
```

- **Ejemplo de for..in**

```
1 const object = { a: 1, b: 2, c: 3 };
2 for (const property in object) {
3   console.log(`${property}: ${object[property]}`);
4 }
```

- **Ejemplo de for..of**

```
1 let iterable = [10, 20, 30];
2 for (const value of iterable) {
3   console.log(value);
4 }
```

- Ejemplo de while

```
1 let i = 0;
2 while(i < 10){
3   i += 1;
4   // TODO 10 times ...
5 }
```

- Ejemplo de do..while

```
1 let i = 0;
2 do {
3   i += 1;
4   // TODO 10 times ....
5 } while (i < 10);
```

2 Ejemplos

2.1 Comprobar instalaciones

```
1 node --version
2 npm --version
```

2.2 Ejemplo sintaxis

```
1 var name = 'Gabri';
2 var age = 29;
3 var hasHobbies = true;
4
5 function summarizeUser(userName, userAge, userHasHobby) {
6   return (
7     'Name is ' +
8     userName +
9     ', age is ' +
10    userAge +
11    ' and the user has hobbies: ' +
12    userHasHobby
13  );
14 }
```



```
14 }  
15  
16 console.log(summarizeUser(name, age, hasHobbies));
```

2.3 Ejemplo variables

```
1 const name = 'Gabri';  
2 let age = 29;  
3 const isBirthday = true;  
4  
5 if(isBirthday){  
6   age++;  
7 }  
8  
9 console.log(`${name} is ${age} years old`);
```

3 Ejercicios

- Ejercicio fizzbuzz
- Ejercicio que le pasas un texto y devuelve true/false si tiene el mismo numero de 'x' que de 'o'
- Piedra, papel o tijeras
- Adivina el numero
- Pintar formas (cuadrado, triangulo, ...)
- Imprimir la sucesión de fibonacci hasta el número dado
- Programa conversor de tiempo: Indica segundos y saca el desglose más eficiente de tiempo
 - Mod: Primer argumento valor, segundo argumento unidad (segundos, minutos, horas, etc...)

4 Entregables

4.1 En clase

- Realizar el ejercicio fizzbuzz

4.2 Tarea

- Realizar dos ejercicios de la lista