



UNIVERSIDAD DE
COSTA RICA

IF7102 Multimedios

Sede del Sur, I-2024

JavaScript: Lenguaje de programación interpretado

Para crear servidores web, apps móviles, apps de escritorio...

JavaScript: Variables

nombre

var edad = **22**;

valor

declaración

JavaScript: Variables

var: +bloque de código

const: se utiliza cuando el valor no va cambiar a lo largo del script.

let: -bloque de código

Tipo de datos	Posibles Valores	Descripción
Boolean	Verdadero FALSO	Los booleanos están destinados a operaciones lógicas. Si le preguntas a una computadora algo como: "¿X es igual a 3?" Responderá con un booleano (verdadero o falso).
String	Cualquier serie de personajes.	Las cadenas son la única forma que tenemos de almacenar palabras (series de caracteres). Nota: las cadenas deben estar entre comillas.
Number	Sólo números	Números enteros, números negativos, números decimales, flotantes, etc. Todos los tipos posibles de números. Nota: si escribes un número entre comillas, JavaScript lo tratará como una cadena.
Undefined	El vacío	Cuando una variable no tiene un valor asignado, no está definida.
Array	Una lista de cualquier tipo de valor.	Una sucesión de cualquier tipo de valor. Pueden ser tipos mixtos de valores; por ejemplo: [2, 3, 'Palabra', 2, 1, nulo, 232, 5, 3, 23, 234, 5, 'hola'].
Objects	cualquier objeto	Puede crear sus propios tipos de datos con operaciones más complejas. Hablaremos más sobre esto más adelante.
Null	solo nulo	Se utiliza para especificar cuando la base de datos o cualquier otra función no devuelve nada.

JavaScript: Lógica de operaciones

Operation	Syntaxis	Examples
Equal to	==	Is 5 == 5? True! Is 5 == 4? False! Is 5 == '5'? True!
Not Equal to	!=	Is 5 != 5? False! Is 5 != '5'? False! Is 1 != 'Hello' True!
Greater than	>	Is 5 > 5? False! Is 6 > 3? True!
Less than	<	Is 6 < 12? True
Greater or equal	>=	Is 6 <= 6? True Is 3 <= 6? True
Less or equal	<=	You get the idea 😊

JavaScript: Lógica de operaciones

Operation	Syntaxis	Examples
AND	&&	With AND, both sides HAVE TO BE TRUE in order for everything to become true. Is (5 == 5 && 3 > 1) ? True! Is ('Ramon' == 'Pedro' && 2 == 2) ? False!
OR		Is ('Oscar' != 'Maria' OR 2 != 2)? True! Is (5 == '5' AND 'Ramon' != 'Pedro') OR (2 == 2)? True!
NOT	!	NOT will be the exact opposite of the result of the logical operator: Is !(5 > 5)? True! Is !(True)? False!

JavaScript: Control de flujo

```
1  if (number < 18) {  
2      document.write("Hello");  
3  } else {  
4      document.write("Good bye!");  
5  }
```


JavaScript: Control de flujo

```
1  for (var i = 0; i < 10; i++) {  
2      document.write("This is number" + " " + i);  
3  }
```

JavaScript: Control de flujo

```
1  var sum = 0;
2  var number = 1;
3  while (number <= 50) {
4      sum += number;
5      number++;
6  }
7  console.log("Sum = " + sum);
```

JavaScript: Control de flujo

```
1  var dog = {  
2    species: "Great Dane",  
3    size: "Extra Large",  
4    age: 3 ,  
5    name: "Rocky"  
6  
7  }  
8  
9  for(items in dog){  
10    console.log(dog[items]);  
11  }
```

Plugins para VS Code



JavaScript (ES6) code snippets

charalampos karypidis | 11,870,921 | ★★★★★

Code snippets for JavaScript in ES6 syntax

Installing



Prettier - Code formatter

Prettier prettier.io | 31,664,693 | ★★

Code formatter using prettier

Disable



Uninstall



This extension is enabled globally.



Quokka v1.4.0

Centuplex | 40,191 | ★★★★★(2)

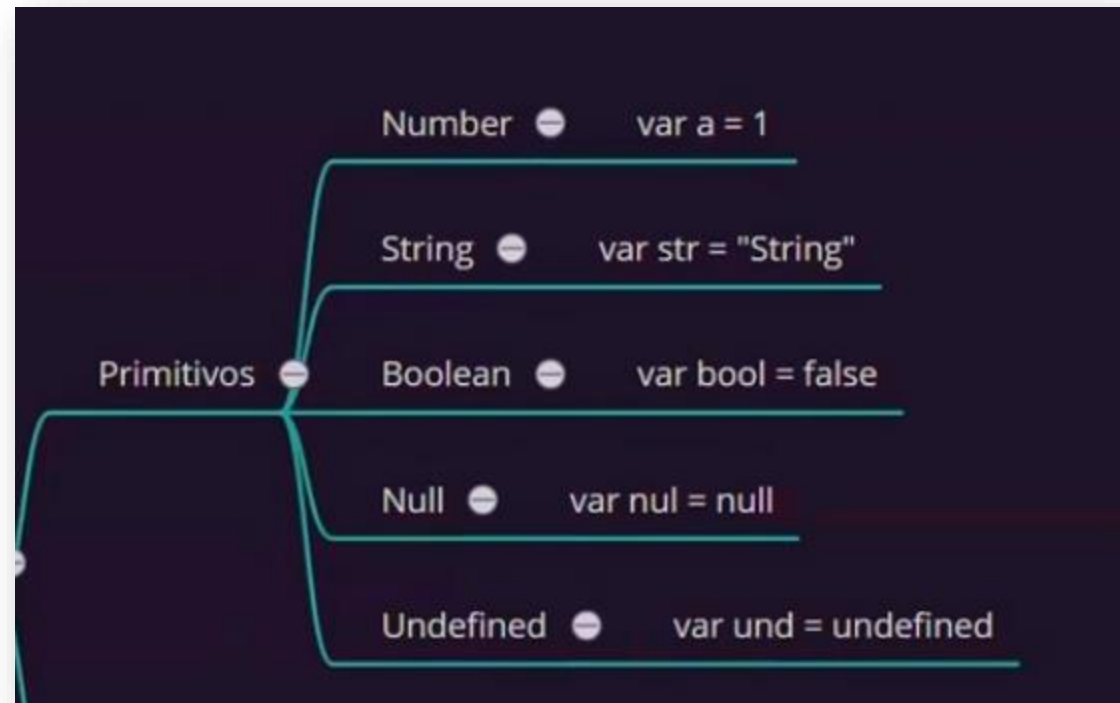
Support for Quokka

Install

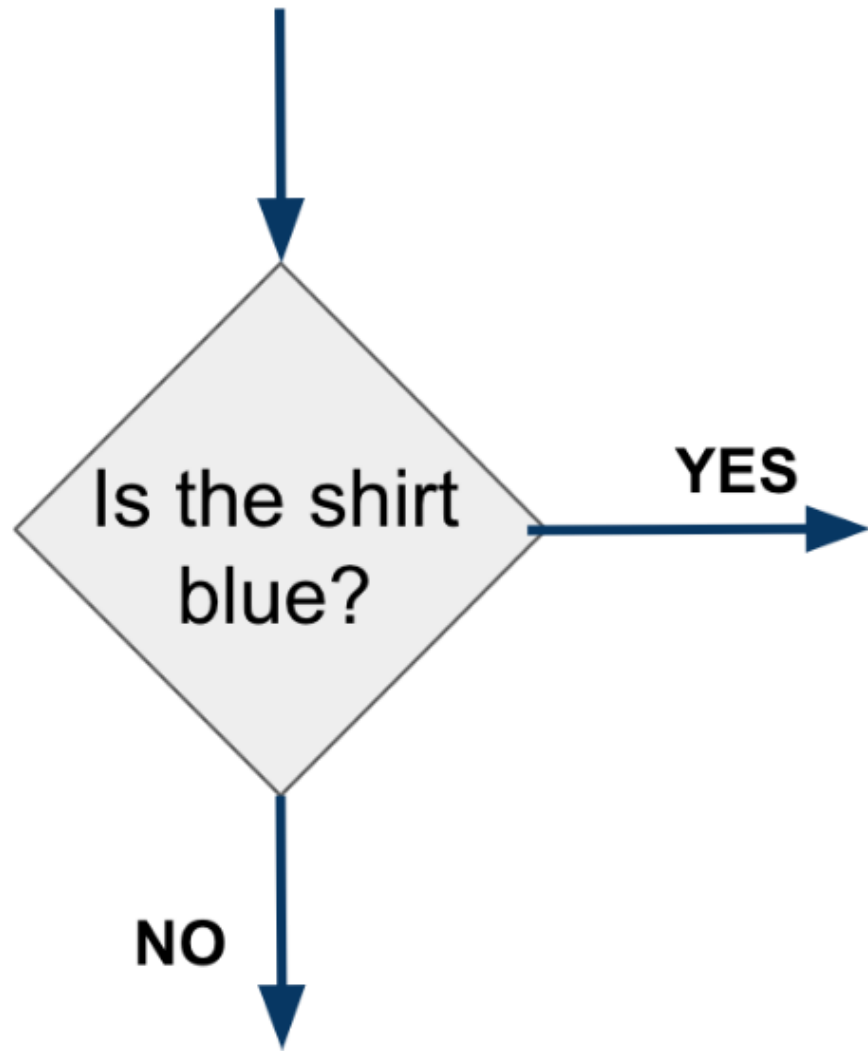


¡Importante dominar el uso de las condiciones en la construcción de algoritmos!

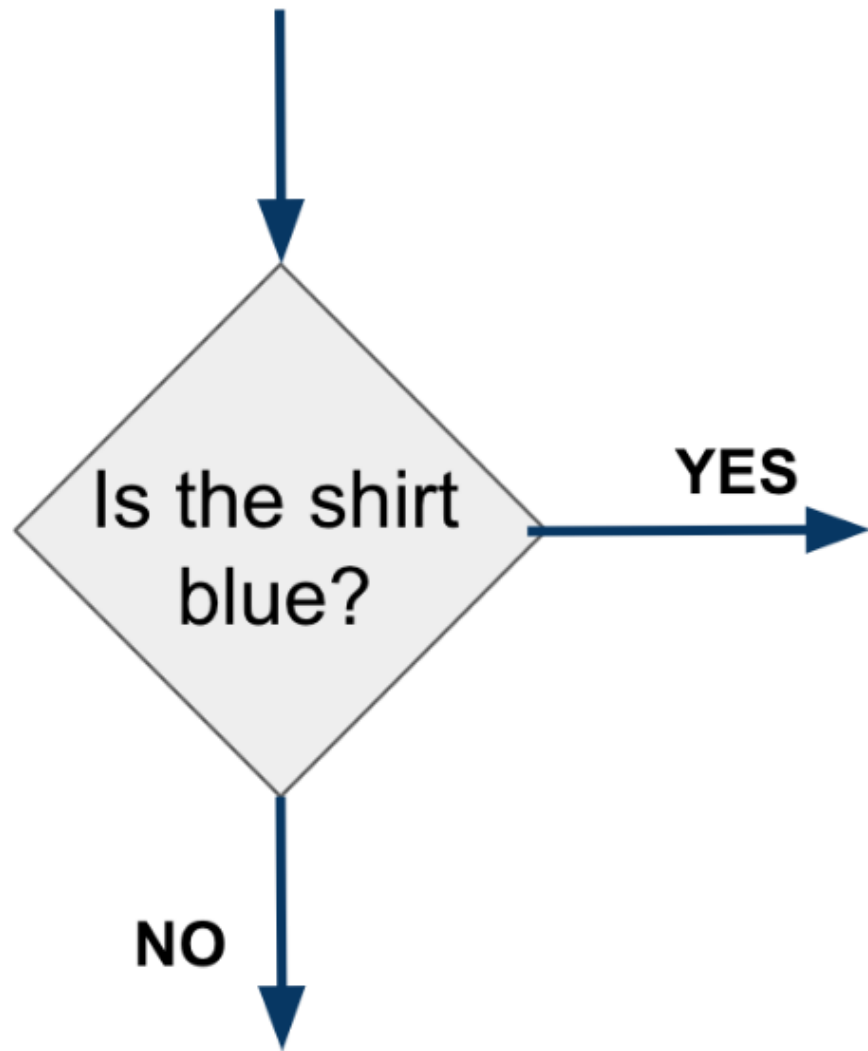
1. Variables
2. Condiciones
3. Objetos
4. Bucles
5. Funciones



IF
ELSE

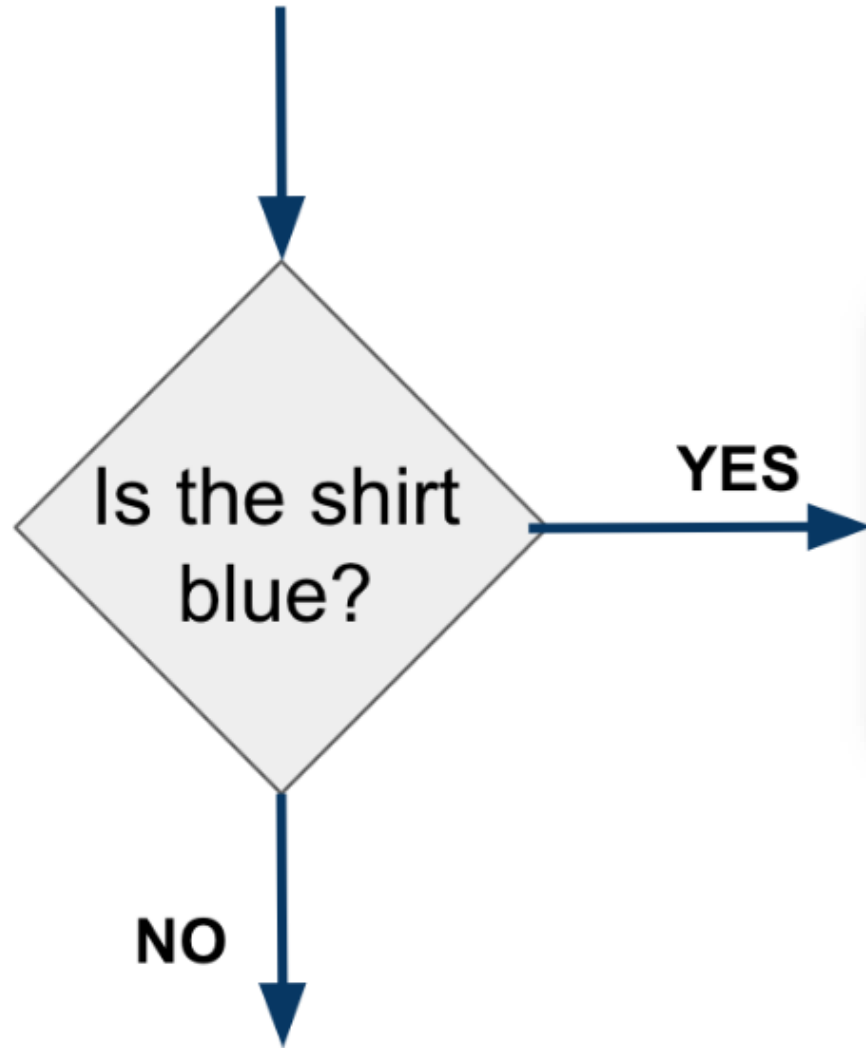


```
1 if(color != 'blue'){  
2     //any code here will run when the color is different than blue  
3 }  
4 else{  
5     //it will only run this code if the color is blue.  
6 }
```



```
1 let feelsLikeHitByTrain = true;
2 let youWereHitByTrain = false;
3
4 if(feelsLikeHitByTrain == true){
5   if(youWereHitByTrain == true){
6     console.log("You don't have a flu");
7   }
8   else{
9     console.log("You have a flu");
10  }
11 }
12 else{
13   console.log("You don't have a flu");
14 }
```

Operadores AND y OR



```
1 if(feelsLikeHitByTrain == false || youWereHitByTrain == true){  
2   console.log("You don't have a flu");  
3 }  
4 else if(feelsLikeHitByTrain == true && youWereHitByTrain == false){  
5   console.log("You have a flu")  
6 }  
7 else{  
8   console.log("I have no idea");  
9 }
```



```

1 if(feelsLikeHitByTrain == false || youWereHitByTrain == true){
2   console.log("You don't have a flu");
3 }
4 else if(feelsLikeHitByTrain == true && youWereHitByTrain == false){
5   console.log("You have a flu")
6 }
7 else{
8   console.log("I have no idea");
9 }

```

Original	Equivalente
en lugar de <code>if(feelsLikeHitByTrain == true)</code>	usted escribe <code>if(feelsLikeHitByTrain)</code>
en lugar de <code>if(feelsLikeHitByTrain == false)</code>	usted escribe <code>if(!feelsLikeHitByTrain)</code>

Operadores mayor que o menor que

>
<

```
1  if(age < 16){  
2      console.log("You cannot drive");  
3  }  
4  else if(age >= 16){  
5      console.log("You can drive");  
6  }
```

Ternario

```
1 const flu = (feelsLikeHitByTrain && !youWereHitByTrain) ? true : false;
```

Funciones

Es código que se agrega entre una estructura “paréntesis”, el cual puede ejecutar en cualquier momento.

Funciones

1. Cada función debe tener un objetivo.
2. Cada función debe tener un nombre propio.
3. Cada función debe devolver algo.
4. Las funciones pueden tener parámetros.

```
1 //the function name is "multiply" and receives 2 param: a & b
2 function multiply(a, b) {
3     //the function returns the multiplication
4     return a * b;
5 }
```

Funciones, ¿para qué?

1. **Divide y conquista.**
2. **Reutiliza tu código.**
3. **Organiza su código.**

```
1
2 //this part of the code is OUTSIDE the 'multiply' function
3
4 function multiply(a, b) {
5
6     //this part of the code is INSIDE the 'multiply' function
7
8     return a * b;
9
10    //this part of the code is INSIDE my function but it will never be executed because it is located AFTER the return statement
11 }
12
13 //this part of the code is OUTSIDE the 'multiply' function
```

```
1 function multiply(a, b) {  
2  
3   let myVariable = 'hello';  
4  
5   return a * b;  
6 }  
7  
8 console.log(myVariable);  
9 // this console.log won't work it will trigger an error because myVariable was declared inside the function multiply,  
10
```


¿Qué es NodeJS?



Creado por **Ryan** Dahl, un profesional en programación en el año **2009**.

Es un **framework** basado en JS capaz de ejecutarse a través de cualquier plataforma.

Facilita la ejecución JS en nuestras máquinas.

¿Qué es NPM?

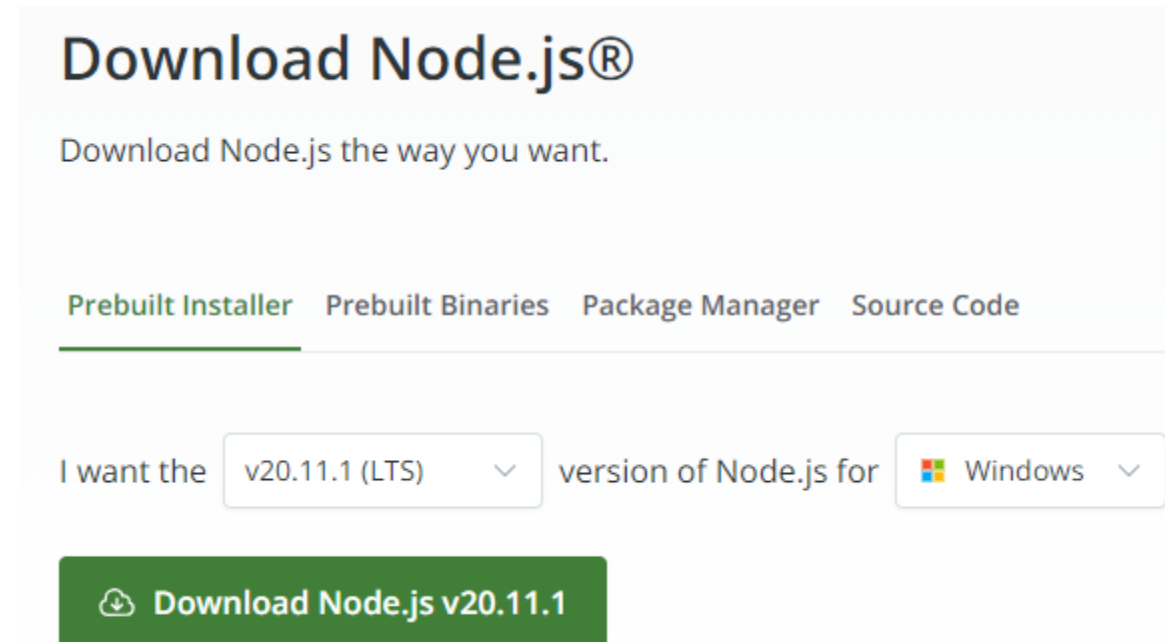
NPM: Node Package Management.

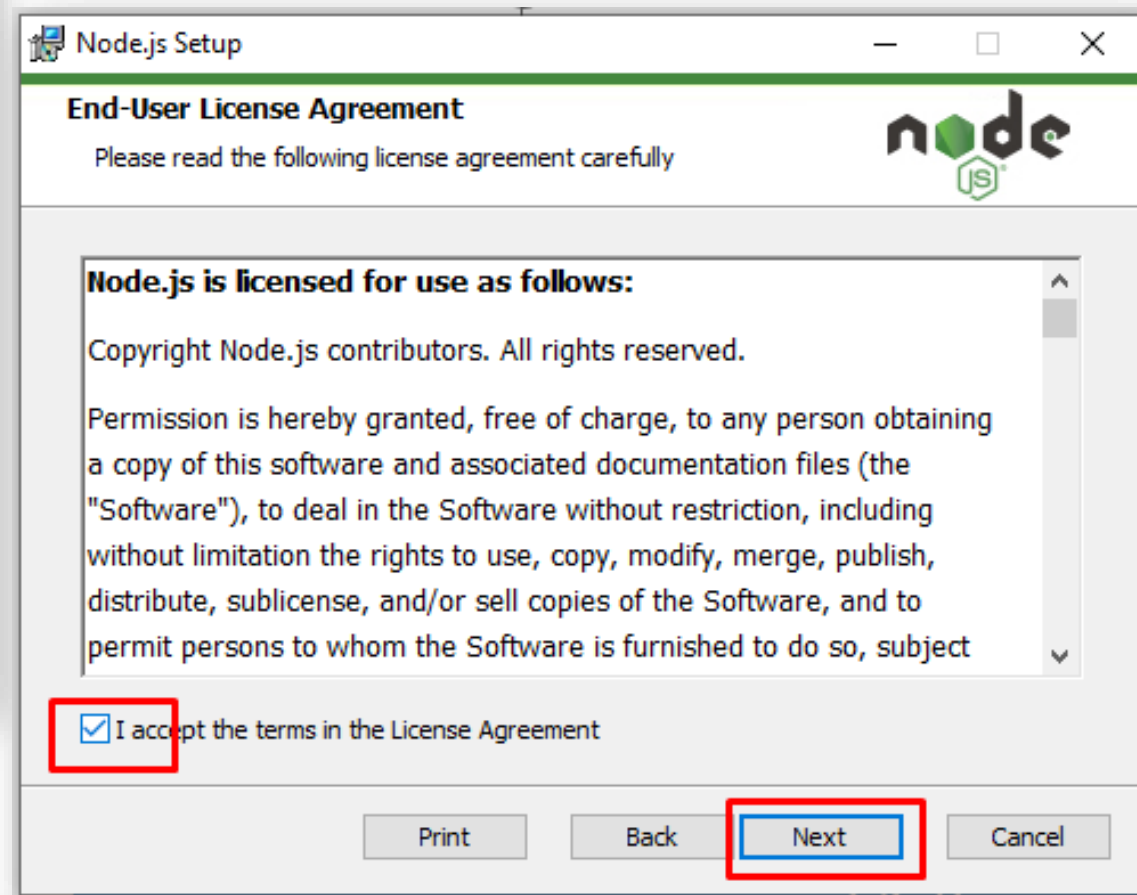
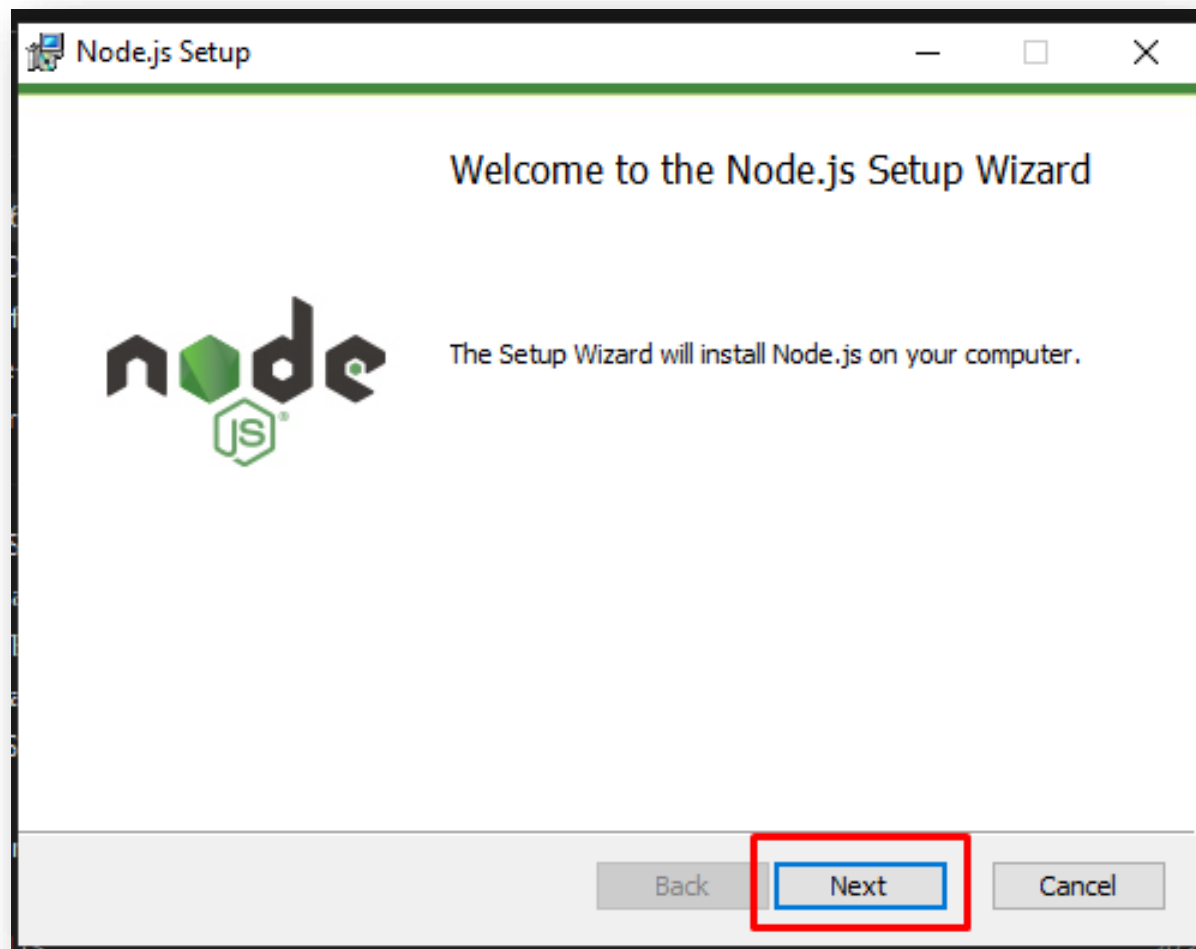
Se encarga de gestionar todos los módulos, paquetes que tenga que ver con el entorno de NodeJs.

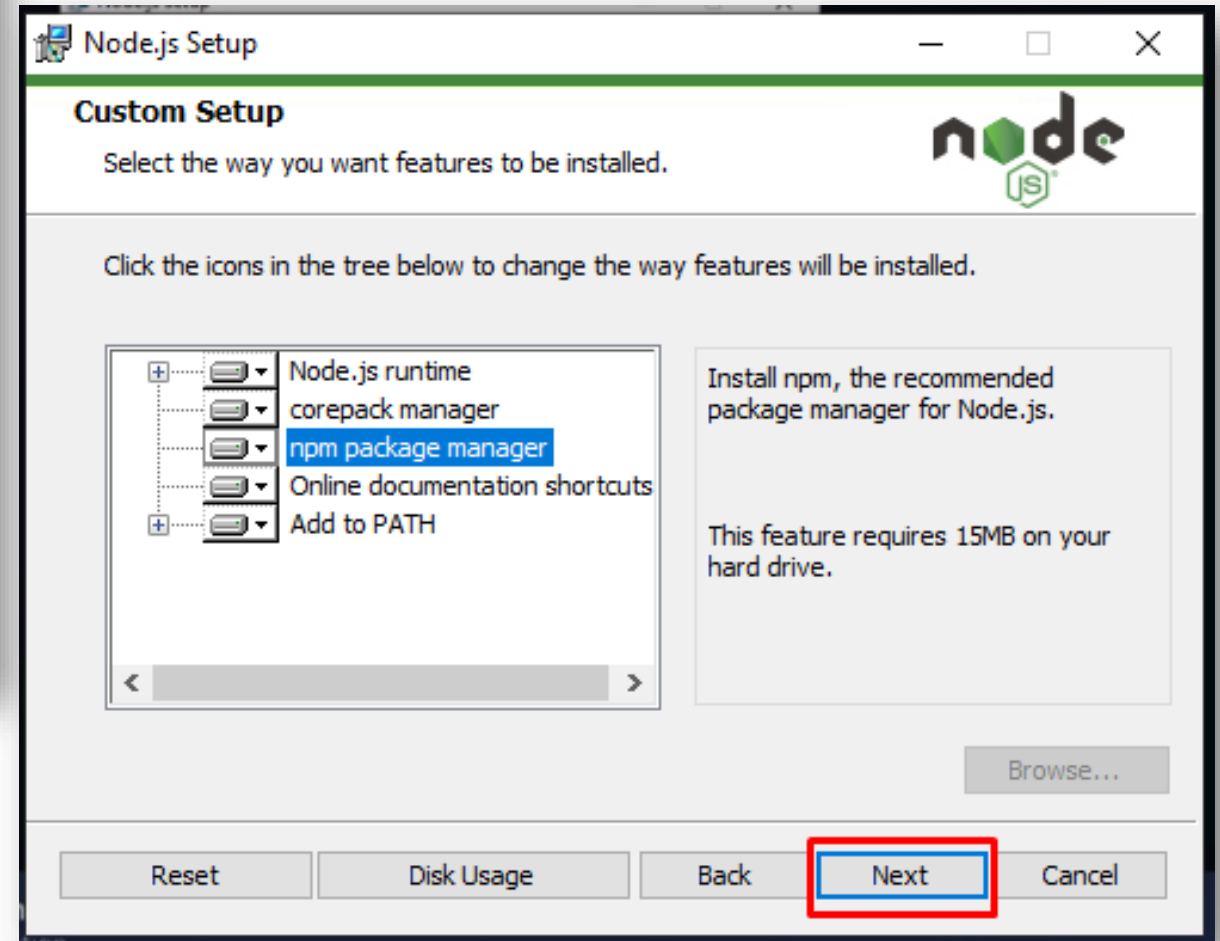
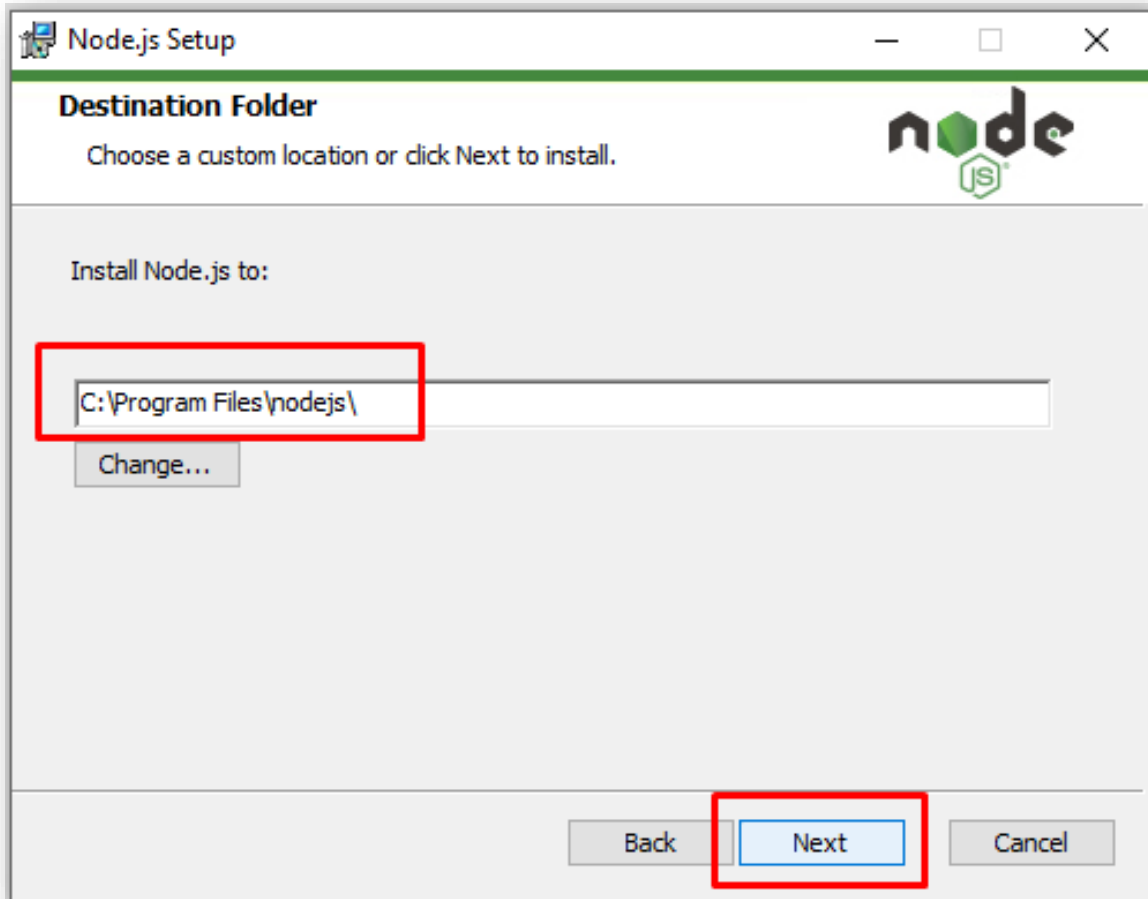
Instalar NodeJS & NPM

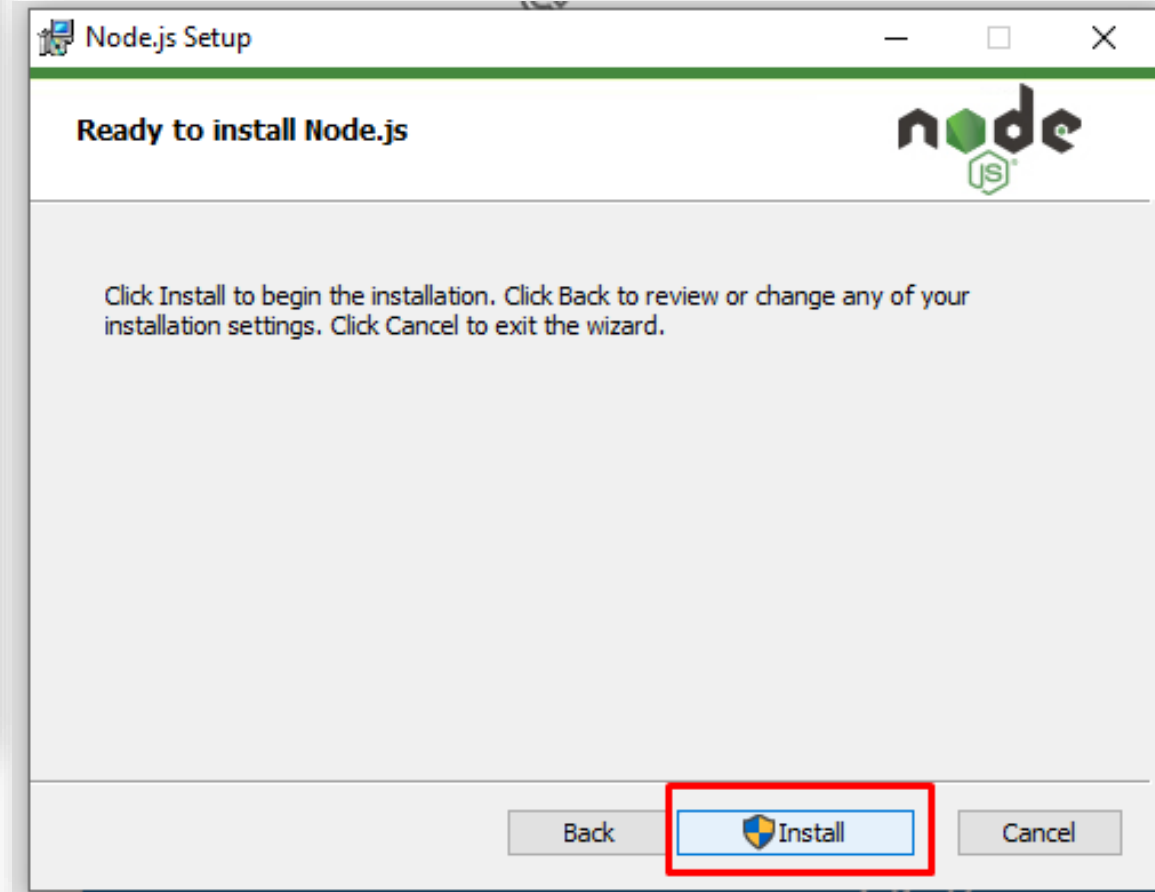
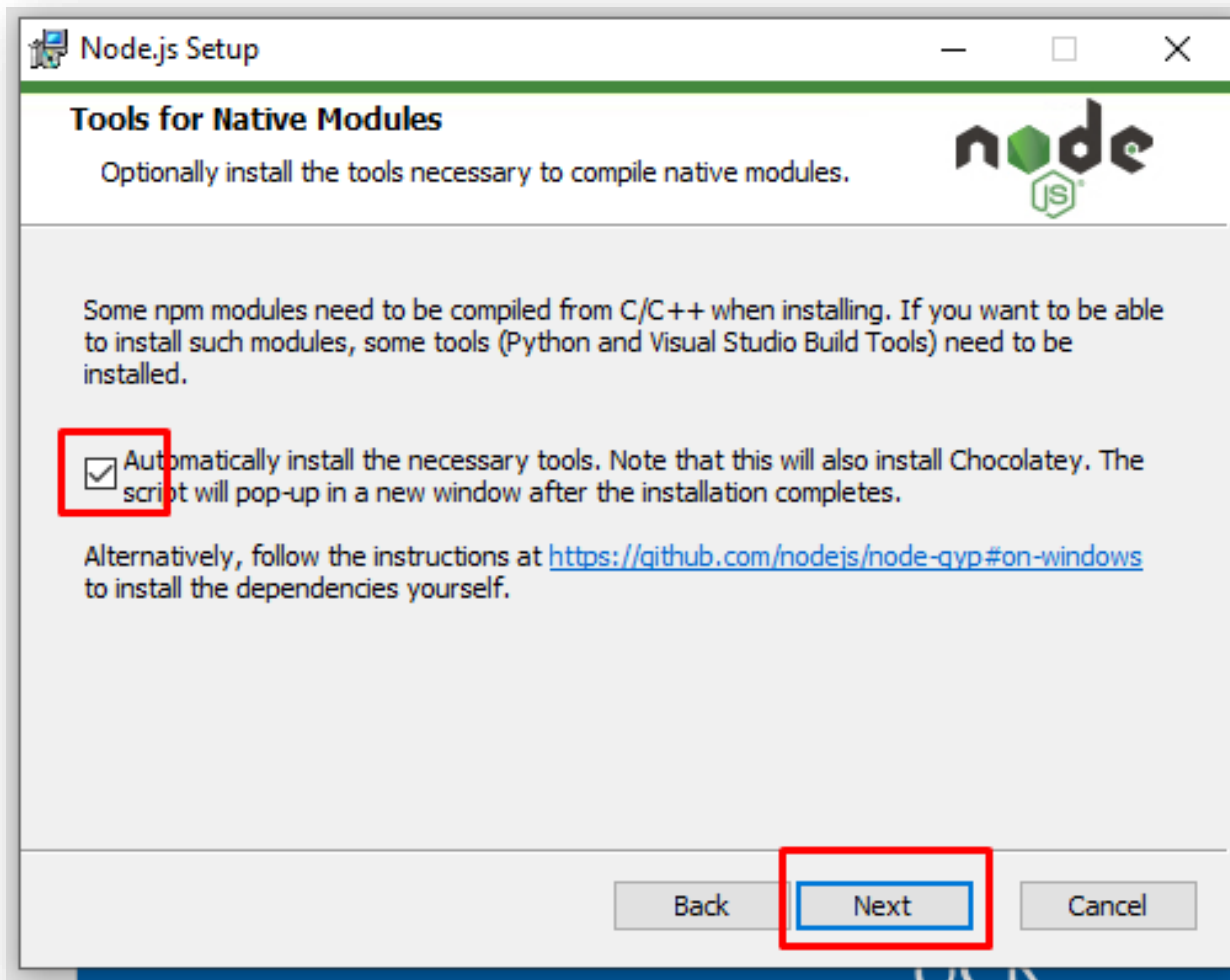


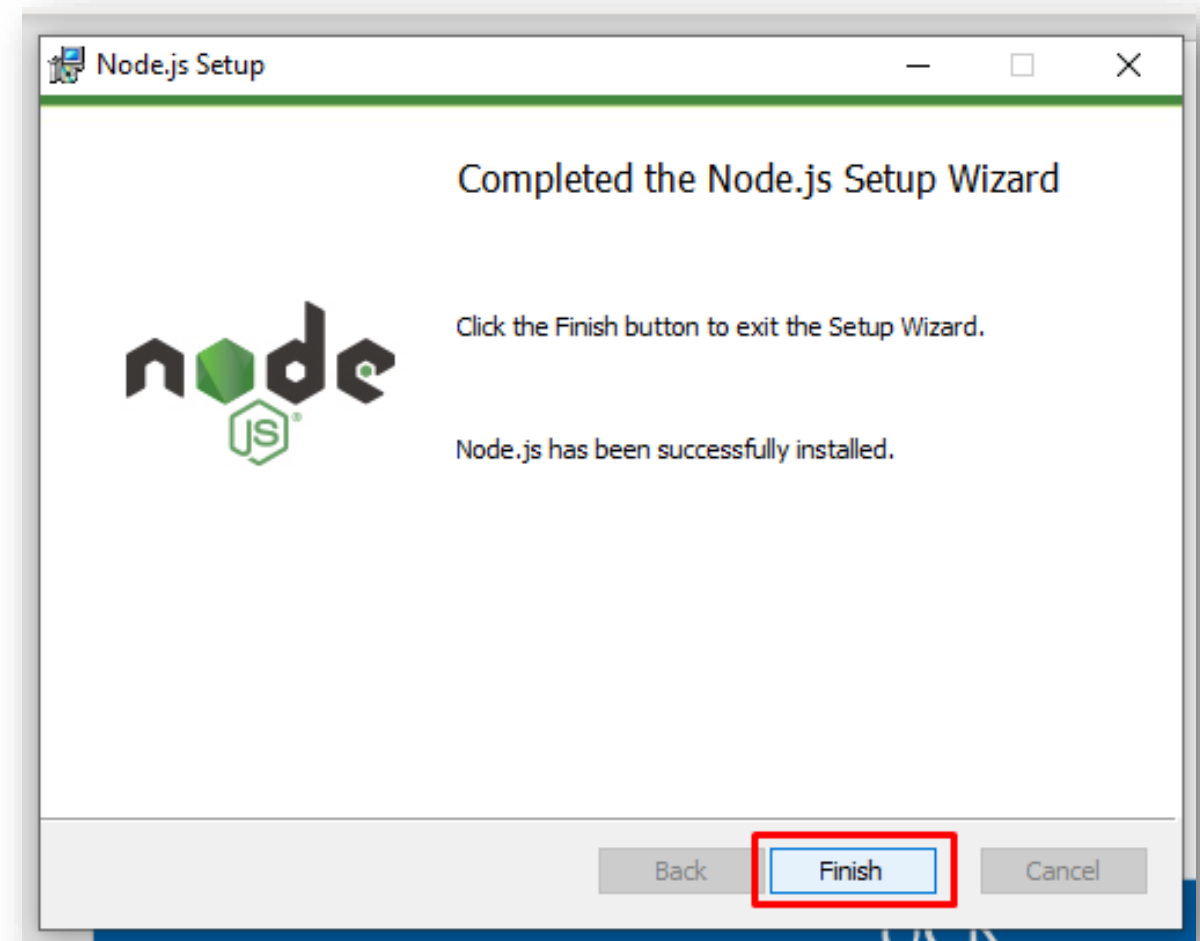
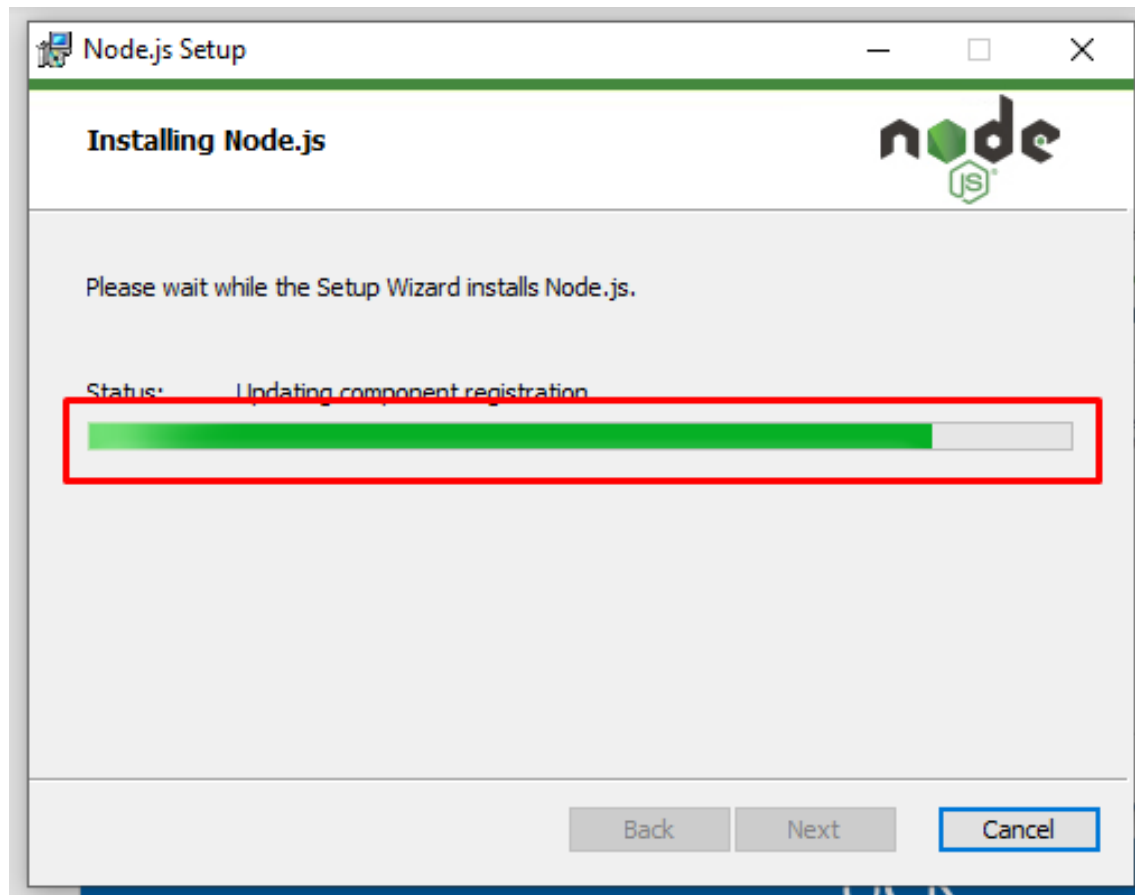
1. Ingresar al sitio web de **Node**.
2. El sitio detecta el **OS** en uso.
3. Descargar la opción estable de Node **recomendada**.











```
C:\Users\Karla>node -v  
v18.16.0
```

```
C:\Users\Karla>npm -v  
9.5.1
```

```
C:\Users\Karla>node --version  
v18.16.0
```

```
C:\Users\Karla>npm --version  
9.5.1
```

```
C:\Users\Karla>
```



1. node -v
2. npm -v
3. node --version
4. npm --version



1. Crear directorio en la máquina, CLI se puede usar **mkdir nombre_Directorio**.
2. Para abrir VS desde cmd digitar CLI **code** .

```
C:\Users\Karla>
```



1. Para abrir línea de comando en VS Code:

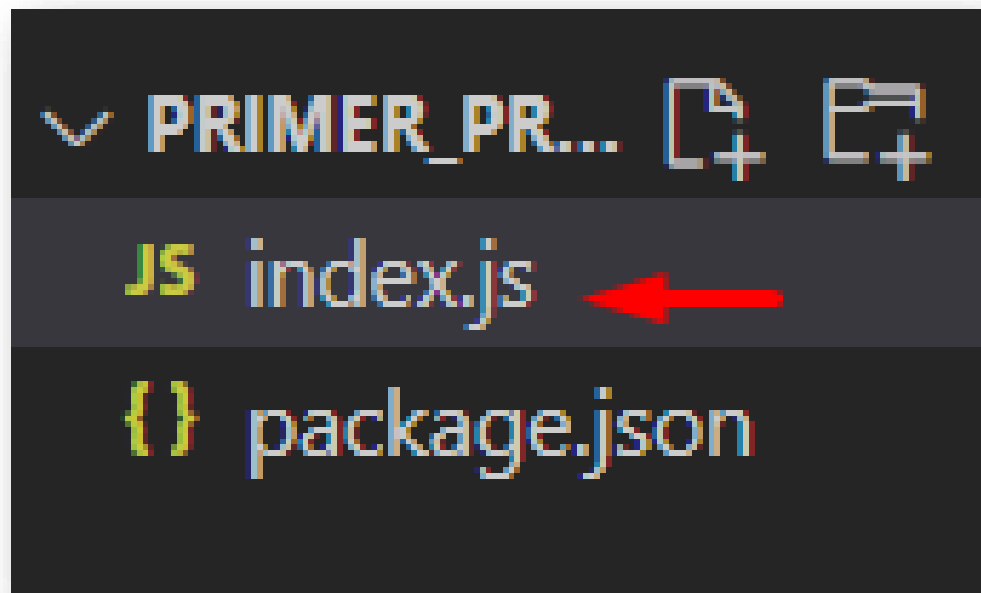
Ctrl + Shift + ñ



Para inicializar: **npm init**

Permitiendo crear un archivo JSON, mismo va tener toda la información del proyecto

```
Press ^C at any time to quit.  
package name: (primer_proyectojs) hola-mundo  
version: (1.0.0)  
description: Primer proyecto usando Javascript con NodeJS en el curso de Multimedios  
entry point: (index.js)  
test command:  
git repository:  
keywords: holamundo  
author: Karla Mayorga  
license: (ISC)  
About to write to C:\Users\Karla\Downloads\kmb\UCR\IF7102_Multimedios\JS\Primer_ProyectoJS\package.json:
```



Crear archivo index.js

A screenshot of the Visual Studio Code interface. The Explorer sidebar on the left shows a project named 'PRIMER_PROYECTOJS' with three files: 'holaMundo.js' (highlighted with a red box), 'index.js', and 'package.json'. The Editor view on the right shows the content of 'holaMundo.js', which is a single line of code: 'console.log('Hola Mundo');' (also highlighted with a red box).

```
EXPLORER  ...  JS holaMundo.js

  PRIMER_PROYECTOJS
    JS holaMundo.js
    JS index.js
    {} package.json

JS holaMundo.js
1 console.log('Hola Mundo');
```

Crear archivo holaMundo.js



```
EXPLORER  ...  JS holaMundo.js

✓ PRIMER_PROYECTOJS
  JS holaMundo.js
  JS index.js
  {} package.json

JS holaMundo.js
1 console.log('Hola Mundo');
```

Agregar **console.log** para
imprimir “**Hola Mundo**”



```
Primer_ProyectoJS> node holaMundo.js
```

Comando de la terminal para
ejecutar el archivo js **node**
holaMundo.js



Resultado de la ejecución:

```
PS C:\Users\Karla\Downloads\kmb\UCR\IF7102_Multimedios\JS\Primer_ProyectoJS> node holaMundo.js
Hola Mundo
PS C:\Users\Karla\Downloads\kmb\UCR\IF7102_Multimedios\JS\Primer_ProyectoJS> |
```




Package.json:

```
▶ Debug  
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1"  
},
```

Package.json:



npm run test

```
5   "main": "index.js",
    ▶ Debug
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
  ...

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  powershell + - [ ] [X]

PS C:\Users\Karla\Downloads\kmb\UCR\IF7102_Multimedios\JS\Primer_ProyectoJS> node holaMundo.js
Hola Mundo
PS C:\Users\Karla\Downloads\kmb\UCR\IF7102_Multimedios\JS\Primer_ProyectoJS> npm run test

> hola-mundo@1.0.0 test
> echo "Error: no test specified" && exit 1

>Error: no test specified"
PS C:\Users\Karla\Downloads\kmb\UCR\IF7102_Multimedios\JS\Primer_ProyectoJS> |
```

Package.json:



```
"scripts": {  
  "start": "node index.js",  
}
```

```
PS C:\Users\Karla\Downloads\kmb\UCR\IF7102_Multimedios\JS\Primer_ProyectoJS> npm start
```

```
> hola-mundo@1.0.0 start
```

```
> node index.js
```

```
Bienvenid@s al curso de Mulmedios. Iniciando la aventura.....
```

```
PS C:\Users\Karla\Downloads\kmb\UCR\IF7102_Multimedios\JS\Primer_ProyectoJS> █
```

npm start

Comentarios en JS:

```
//Esta es la forma de comentar una línea
```

```
/*Esta  
es la forma  
de realizar  
comentarios  
en varias líneas*/
```

```
console.log('Bienvenid@s al curso de Mulmedios. Iniciando la aventura.....');
```

Para comentar un bloque de código **CTRL + K + C**



Actividad:

Crea un nuevo proyecto de Node, y dentro del package.json crea un script que muestre por pantalla:

"Hola, te muestro mi primera actividad con Node realizado durante el curso de Multimedios".

Práctica 1 :



Generador nombres de dominio

Pronombres

Adjetivos

Sustantivos

.extensiones

Práctica 1:



Hacer una función haga un generador de dominio

```
let pronoun = ["El", "Nuestro"];  
  
let adj = ["alto", "grande"];  
  
let noun = ["equipo", "luz"];  
  
let ext = [".com", ".net", ".cr", ".org"];
```



Práctica 2 :

Hacer una función que imprima un número
`random()`;



Salida de ejemplo:

```
/ 1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
.... .... 98
Fizz
Buzz
/
```

- Escribe el código necesario para imprimir en la consola todos los números desde 1 hasta 100.
- Para múltiplos de 3, en lugar del número, imprima Fizz.
- Para múltiplos de 5, imprima Buzz.
- Para números que son múltiplos de 3 y 5, imprima FizzBuzz.

Práctica 4 :

Escribe una función `getPrice` que reciba el número de invitados que asistirán a la boda y devuelva el precio correspondiente a la consola.

_Por ejemplo, si el usuario dice que asistirán 20 personas a la boda, el precio devuelto debe ser 4000

Pista:

Escribe una función `getPrice` que reciba el número de invitados que asistirán a la boda y devuelva el precio correspondiente a la consola.

_Por ejemplo, si el usuario dice que asistirán 20 personas a la boda, el precio devuelto debe ser 4000

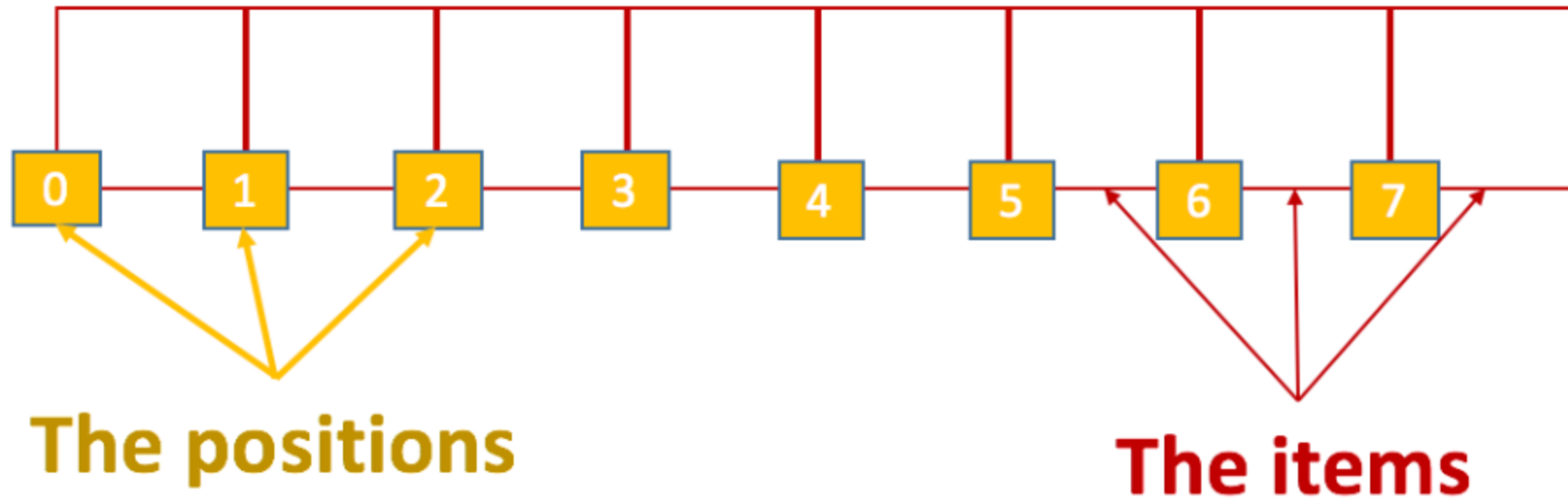


No. de personas	Precio
Hasta 50 personas	\$4,000
Hasta 100 personas	\$10,000
Hasta 200 personas	\$15,000
Más de 200 personas	\$20,000

JavaScript: Array

```
let myArray = ['sunday', 'monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday'];
```

This is an Array with Length=8



Práctica 1 :



```
let myArray = ['sunday','monday','tuesday','wednesday','thursday','friday','saturday'];
```

1. Imprimir el tercer item aquí.
2. Cambiar el valor 'friday' por null.