

06 NodeJs: El REPL

Tras haber instalado Node.js, nuestra primera interacción directa con el entorno será a través de su **REPL**. El término REPL es un acrónimo en inglés que significa **Read-Eval-Print Loop** (Leer-Evaluar-Imprimir Bucle). Describe un entorno interactivo simple que realiza los siguientes pasos de forma continua:

1. **Read (Leer):** Lee la entrada proporcionada por el usuario.
2. **Eval (Evaluar):** Evalúa (ejecuta) la entrada como código JavaScript.
3. **Print (Imprimir):** Imprime el resultado de la evaluación o cualquier salida generada (como por `console.log`).
4. **Loop (Bucle):** Vuelve al paso 1, esperando la siguiente entrada.

Si han utilizado la consola de JavaScript del navegador, el concepto de REPL les resultará familiar, ya que opera de manera muy similar. Es una herramienta excelente para probar pequeños fragmentos de código, realizar cálculos rápidos o explorar las características de JavaScript y los objetos específicos de Node.js.

Accediendo al REPL:

Para iniciar el REPL de Node.js, sigan estos pasos:

1. Abran su terminal o símbolo del sistema (Command Prompt en Windows, Terminal en macOS/Linux).
2. Escriban el comando `node` y presionen `Enter` .

```
node
```

Deberían ver un indicador (`>`) que les señala que Node.js está esperando su entrada. Es posible que también vean un mensaje de bienvenida indicando la versión de Node.js que están utilizando.

```
Welcome to Node.js vXX.Y.Z.  
Type ".help" for more information.  
>
```

Ejecutando Código JavaScript Básico:

Ahora pueden escribir cualquier expresión válida de JavaScript y Node.js la evaluará:

```
> 2 + 3  
5  
> let saludo = "Hola desde Node.js REPL";  
undefined  
> saludo  
'Hola desde Node.js REPL'  
> function sumar(a, b) { return a + b; }  
undefined
```

```
> sumar(10, 5)
15
```

Observen que, al igual que en la consola del navegador, la asignación de variables (`let saludo = ...`) devuelve `undefined` (ya que la asignación en sí misma no produce un valor), pero luego podemos acceder al valor de la variable escribiendo su nombre.

El Clásico "Hola Mundo":

Para imprimir un mensaje en la consola, utilizamos el conocido `console.log()` , exactamente como lo harían en el navegador:

```
> console.log('Hola Mundo');
Hola Mundo
undefined
>
```

Node.js ejecuta la función `console.log` , que imprime la cadena "Hola Mundo" en la siguiente línea. La línea `undefined` que aparece después es el valor de retorno de la propia función `console.log` .

Saliendo del REPL:

Para salir del REPL de Node.js y volver a la línea de comandos de su terminal, pueden hacer una de las siguientes cosas:

- Escribir `.exit` y presionar `Enter` .
- Presionar `Ctrl + C` dos veces seguidas.

Utilidad del REPL:

El REPL es sumamente útil para:

- **Pruebas rápidas:** Verificar sintaxis o cómo funciona una pequeña porción de código sin necesidad de crear un archivo.
- **Exploración:** Investigar los objetos globales y módulos incorporados de Node.js (lo veremos más adelante).
- **Aprendizaje:** Experimentar interactivamente con las características del lenguaje JavaScript.

Aunque el REPL es excelente para estas tareas interactivas, para escribir programas más complejos, normalmente crearemos archivos `.js` y los ejecutaremos con Node.js desde la terminal (por ejemplo, `node mi_script.js`), lo cual cubriremos más adelante.