Introducción (práctica)

Tiempo estimado: 20min

El objeto de esta práctica es afianzar, reforzar y consolidar los conocimientos teóricos presentados en la lección.

Al finalizarla, el estudiante:

- Habrá instalado Node.js.
- Habrá ejecutado el motor de Node en modo interactivo.
- Habrá ejecutado código JavaScript desde la línea de comandos.

Objetivos

El objetivo de la práctica es mostrar cómo instalar Node.js y cómo trabajar en modo interactivo con su motor de JavaScript.

Instalación de Node.js

En una organización, los procesos de instalación se deben encontrar redactados en sus propios documentos para que las instalaciones sean siempre similares, independientemente de quién las realice. En nuestro caso, para comenzar la práctica, vamos a instalar Node.js según los documentos que se adjuntan con la lección en la sección de recursos adicionales. Actualmente, hay dos versiones del documento: una para entornos Windows y otra para distribuciones basadas en Debian.

Modo interactivo

Una vez instalado Node, vamos a jugar un poco con el modo interactivo de su motor de JavaScript.

1. Abrir una consola.
2. Ejecutar node en modo interactivo:

   ```
   $ node
   >
   ```

   El símbolo > es el *prompt* de node. Es el indicador de que está esperando que le digamos qué ejecutar.
3. Mostrar el valor de la variable _:

   ```
   > _
   undefined
   >
   ```
4. Ejecutar la expresión 1 + 2 + 3:

   ```
   > 1 + 2 + 3
   6
   >
   ```

   El intérprete analiza la expresión y la evalúa, mostrando por pantalla el resultado.
5. Mostrar el valor de la variable _:

   ```
   > _
   6
   >
   ```

   Recordemos que la última expresión ejecutada por el intérprete, en modo interactivo, se almacena en la variable _.
6. Mostrar los primeros 10 números naturales mediante una única proposición JavaScript:

   ```
   > for (let i = 1; i <= 10; i+=1) console.log(i);
   1
   ```



```
2
3
4
5
6
7
8
9
10
undefined
>
```

El objeto console contiene funciones con las que escribir en la pantalla. console.log() escribe en la salida estándar.

7. Introducir la sentencia anterior, pero con el cuerpo del bucle for en una línea aparte:

```
> for (let i = 1; i <= 10; i+=1)
... console.log(i);
1
2
3
4
5
6
7
8
9
10
undefined
>
```

Cuando el intérprete detecta que la proposición no ha terminado, muestra un segundo *prompt*, indicando así que continuemos escribiéndola. Este nuevo *prompt* es ....

8. Pulsar la tecla ↑ varias veces.

   Observe que la tecla permite movernos a lo largo de la historia de líneas introducidas.

9. Pulsar la tecla ↓ varias veces.

10. Pulsar Ctrl+C una sola vez.

    Esta combinación de teclas le indica al intérprete interactivo que no deseamos ejecutar la línea actual y deseamos nos muestre de nuevo el *prompt*.

11. Escribir con y, a continuación, pulsar el tabulador:

```
> cons
const
console
constructor
> cons
```

Muestra todas las posibles opciones que el intérprete cree que estamos buscando.

12. Escribir una o, para obtener conso y hacer clic en el tabulador.

    El intérprete detecta una única opción que comience por conso y rellena el resto: console.

13. Pulsar Ctrl+C una sola vez para cancelar la línea en curso.

14. Mostrar los comandos del modo interactivo:

```
> .help
.break    Sometimes you get stuck, this gets you out
.clear    Alias for .break
.editor   Enter editor mode
.exit     Exit the repl
.help     Print this help message
.load     Load JS from a file into the REPL session
.save     Save all evaluated commands in this REPL session to a file
>
```

15. Salir del intérprete:

```
> .exit
```



```
$
```
También se puede utilizar Ctrl+D o dos Ctrl+C seguidos.

Modo comando

Ahora, vamos a ejecutar código JavaScript desde la línea de comandos. Para ello, usaremos las opciones -e y -p.

1. Ir a la consola.
2. Solicitar a node que ejecute la expresión 1 + 2 + 3 con las opciones -e y -p:

```
$ node -e "1 + 2 + 3"
$ node -p "1 + 2 + 3"
6
$
```

Observe que la opción -e no muestra el resultado de la proposición ejecutada, mientras que la opción -p sí lo hace. Si necesita que el resultado sea mostrado, puede utilizar la opción -e con console.log() o bien la opción -p como acabamos de ver. Veamos cómo hacerlo con la opción -e:

```
$ node -e "console.log(1 + 2 + 3)"
6
$
```

