

1. Cree una aplicación NodeJS que muestre los números del 1 al 100.
2. Crear una biblioteca NodeJS que tenga las funciones suma, resta, multiplicación y división. Cargue esta biblioteca desde otro archivo y escriba un programa de ejemplo que use las cuatro funciones dadas.
3. Escriba un programa que guarde en un archivo los números del 1 al 100 en modo síncrono.
4. Escriba un programa que guarde en un archivo los números del 1 al 100 en modo asíncrono.
5. Escriba un programa que lea el archivo generado en el ejercicio anterior y lo muestre en pantalla de forma síncrona.
6. Escriba un programa que lea el archivo generado en el ejercicio anterior y lo muestre en pantalla de forma asíncrona.
7. Haga un solo programa que guarde en un archivo el texto “Hola mundo” y lo lea, mostrando el resultado en pantalla de forma síncrona.
8. Haga un solo programa que guarde en un archivo el texto “Hola mundo” y lo lea, mostrando el resultado en pantalla de forma asíncrona.
9. Cree una biblioteca con una función que sume los números del uno al 100 y devuelva el resultado de la suma usando “return” de forma síncrona. Cargue la biblioteca desde otro archivo y muestre el funcionamiento de la función.
10. Cree una biblioteca con una función que sume los números del uno al 100 y devuelva el resultado de la suma usando “return” de forma asíncrona. Cargue la biblioteca desde otro archivo y muestre el funcionamiento de la función.
11. Sea el siguiente código:

```

    async function saludar() {
        return "Hola mundo"
    }

    saludar().then(
        (saludo) => {
            let cadena = "****"
            saludar().then(
                (saludo) => {
                    console.log(cadena + saludo)
                }
            )
        }
    )

```

Fíjese que la variable “cadena” no está definida dentro de la función marcada en rojo y no es una variable global. ¿Qué se puede deducir las variables que está definidas fuera de la función y se usan dentro aunque no están definidas dentro de ellas?

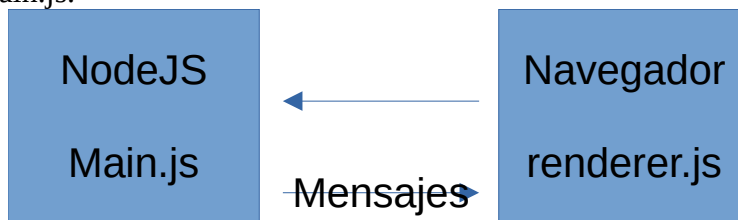
12. Hay situaciones en las que interesa esperar a que una función asíncrona termine, por ello se usa la palabra clave await. Hay muchas bibliotecas que no proporcionan versión síncrona de sus métodos y hay que transformarlos en síncronos usando await. Para demostrar el uso de await, haga un programa que guarde un archivo y lo vuelva a leer. Use la forma asíncrona de leer archivos para hacer este ejemplo.
13. El comando npm permite instalar cientos de bibliotecas. Instale la biblioteca <https://www.npmjs.com/package/md5>, explique para qué sirve y haga un ejemplo.
14. Busque y ejecute un ejemplo de una biblioteca que:
  1. Genere archivos PDF desde NodeJS
  2. Mande un correo electrónico. Haga un ejemplo en que envíe un correo a su cuenta de EducaMadrid.
15. Cree un proyecto usando npm. Instale una biblioteca del ejercicio anterior en dicho proyecto usando npm install y haga un ejemplo. Vea en el archivo que ha generado el proyecto cuál es el

archivo que se va a ejecutar por defecto y ponga allí el código. Ejecute el proyecto usando “npm start”.

16. “Apache Cordova” (<https://cordova.apache.org/>) es una aplicación que permite crear aplicaciones en Javascript que funcionan en diversos sistemas (Linux, Windows, Mac, Android, iPhone,...). Básicamente consiste en un navegador en que se ejecutará la página web que se cree. Numerosas aplicaciones que se usan en los móviles son páginas web encapsuladas con “Apache Cordova” que hacen llamadas Ajax a servidores.

Busque un ejemplo sencillo de cómo crear una aplicación con “Apache Cordova”.

17. Electron es un paquete npm que consiste en un navegador web (de hecho es Chromium) adaptado para crear aplicaciones de escritorio. Una vez creadas se pueden empaquetar para diversos sistemas operativos. Aplicaciones como Visual Studio Code, Teams o Skype son realmente aplicaciones creadas con electron y empaquetadas para diversos sistemas operativos. Con un navegador web normal como Firefox o Chrome no es posible hacer ciertas tareas que realizan las aplicaciones de escritorio, como puede ser escanear el disco duro en busca de archivos o ejecutar comandos. Con electron se pueden ejecutar módulos de "Node.js" sin problemas. Para ello se disponen de dos partes (main.js y renderer.js) que se ejecutan de forma independiente y pasan mensajes entre ellas. Main.js permite ejecutar órdenes javascript como cualquier script de NodeJS. Renderer.js se ejecuta en el navegador y permite enviar y recibir mensajes de main.js.



Leer los apuntes que se indican en <https://selairi.github.io/apuntesjavascript/electron.html> y ejecutar los ejemplos.

Invente una aplicación sencilla que haga uso de electron.