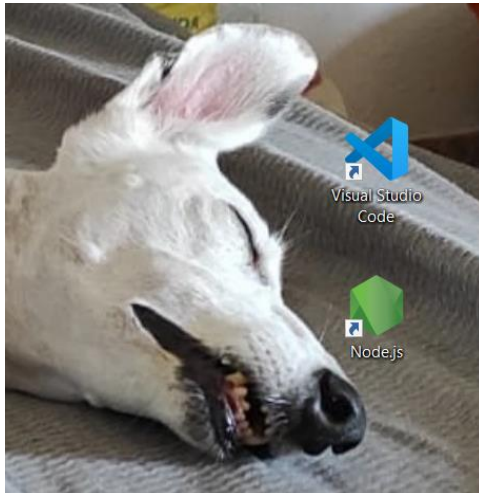


Introducción a JavaScript

Preparación

1. Instalar los requisitos de software indicados



2. Hacer un *fork* del repositorio base https://github.com/pedroprieto/practica_dwec_gestor_presupuesto.git en tu cuenta de GitHub



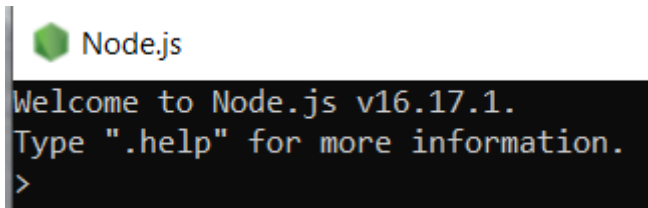
3. Abrir un terminal
4. Clonar **tu repositorio** (el que se ha creado en tu cuenta al hacer el *fork*, no el repositorio base) al equipo local mediante `git clone`

```
PS C:\Users\rabic\OneDrive\Escritorio\Curso-github> git clone https://github.com/CintiaCastilloCareglio/practica_dwec_gestor_presupuesto.git
Cloning into 'practica_dwec_gestor_presupuesto'...
remote: Enumerating objects: 370, done.
remote: Counting objects: 100% (370/370), done.
remote: Compressing objects: 100% (138/138), done.
Receiving objects: 99% (367/370) used 352 (delta 200), pack-reused 0
Receiving objects: 100% (370/370), 836.99 KiB | 1.75 MiB/s, done.
Resolving deltas: 100% (209/209), done.
```

5. Acceder a la carpeta del repositorio

```
PS C:\Users\rabic\OneDrive\Escritorio\Curso-github> cd .\practica_dwec_gestor_presupuesto\
PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto> █
```

Instala el entorno **NodeJS** en tu ordenador. Puedes utilizar la versión actual para tener acceso a las últimas características.



El REPL de JavaScript

JavaScript ofrece un entorno **REPL** (acrónimo que significa *Read Evaluate Print Loop* o Bucle de Lectura, Evaluación e Impresión). Este entorno no es más que una consola que nos permite escribir cualquier código JavaScript como código de entrada y devolver el resultado a la consola después de la ejecución.

El REPL es muy útil a la hora de hacer pruebas rápidas o pequeños prototipos. Podemos acceder al REPL de JavaScript en el **navegador** y en **NodeJS**.

El REPL en el navegador

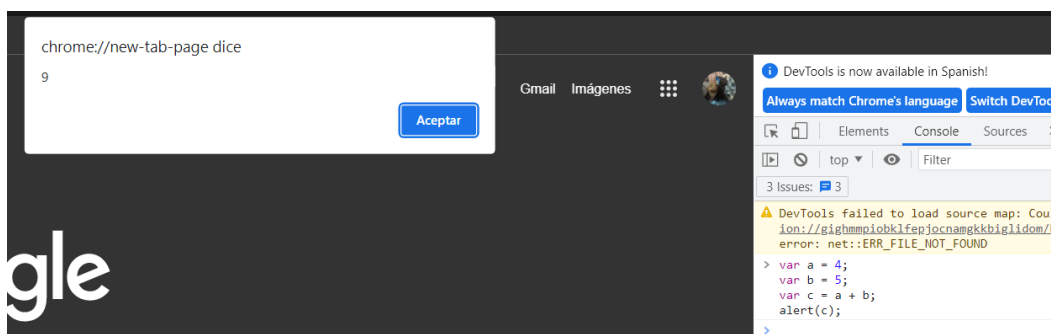
Para acceder al REPL en el navegador, accede a las [herramientas de desarrollo](#) (en la mayoría de los casos se accede pulsando F12). Una vez ahí, selecciona **consola**.

Si estás en una pestaña vacía (sin ninguna web cargada) no habrá ningún código JavaScript cargado; si por el contrario no lo está, seguramente haya algún código JavaScript cargado con el que podrías interactuar.

Para empezar, escribe el siguiente código en la consola y comprueba su ejecución:

```
var a = 4;
var b = 5;
var c = a + b;
alert(c);
```

Verás que el código se ejecuta y que por pantalla se muestra un cuadro de alerta con el resultado.

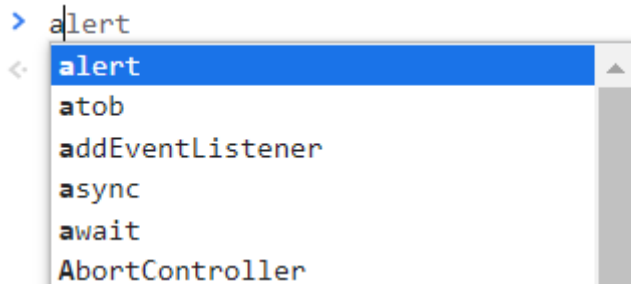


También verás que tras ejecutar cada línea (o tras ejecutar la última, si las has copiado juntas) aparece el resultado `undefined`. Esto se debe a que en entornos REPL todo se considera una función que devuelve un valor. Como todas las líneas de nuestro primer programa no devuelven valores (`var` es una declaración de variables y `alert` no devuelve ningún valor en la consola, sino que solo muestra el texto en el cuadro de alerta), el resultado de cada ejecución es `undefined`.

Puedes hacer alguna prueba más con algún código sencillo y ver cómo funciona.

Por último, **lee el siguiente artículo** sobre la [consola de JavaScript en el navegador](#) y familiarízate con los comandos de **autocompletar**, **historial** y **modo multilínea**.

Autocompletar:



Modo multilínea con *shift+enter* :

```
> var a=9;
   var b=7;
   var c=a-b;
   var d=a-c;
   alert(c)
   alert(d)
< undefined
```

El REPL en NodeJS

Para acceder al REPL en NodeJS abre una consola en el sistema operativo (CMD en Windows o un terminal en Linux o Mac) y ejecuta el comando `node`. Si lo has instalado correctamente, verás que aparece un intérprete de comandos de JavaScript.

```
PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto> node --version
v16.17.1
```

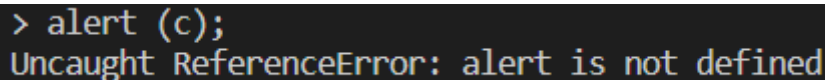
Para salir de la consola de Node puedes ejecutar CTRL + D pulsar CTRL + C dos veces. También puedes escribir `.help` para obtener información.

```
PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto> node
Welcome to Node.js v16.17.1.
Type ".help" for more information.
> 
```

Para empezar, escribe el siguiente código en la consola y comprueba su ejecución:

```
var a = 4;  
var b = 5;  
var c = a + b;  
alert(c);
```

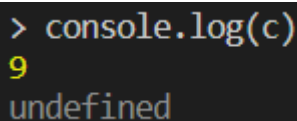
Verás que en este caso las tres primeras líneas se ejecutan sin problemas pero la llamada a la función `alert` produce un error. Esto es debido a lo que hemos comentado antes sobre las APIs web: NodeJS solo es intérprete de JavaScript y no tiene acceso a las APIs de navegador, entre las que se encuentra la función `window.alert`.



```
> alert (c);  
Uncaught ReferenceError: alert is not defined
```

Por el contrario, si ejecutas el siguiente código verás que funciona correctamente:

```
var a = 4;  
var b = 5;  
var c = a + b;  
console.log(c);
```



```
> console.log(c)  
9  
undefined
```

En este último caso, el programa no hace referencia a ninguna API de navegador.

Como en el caso del navegador, puedes hacer alguna prueba más con algún código sencillo y ver cómo funciona. Comprueba también que puedes utilizar teclas de **autocompletar**.

Ejecución de código en archivos externos

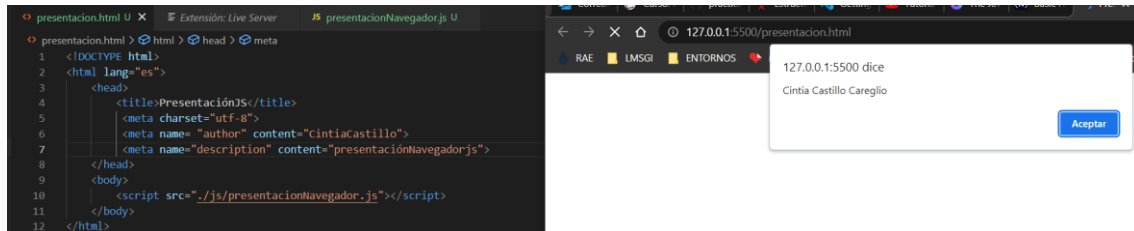
Navegador

Tal como se indica en el artículo [Hola Mundo](#), lo habitual es que los scripts de una mínima complejidad se ejecuten en archivos independientes.

Crea en la raíz del repositorio un fichero denominado `presentacion.html`. Dicho archivo deberá contener un [esqueleto de página HTML](#) que cargue un script llamado `presentacionNavegador.js`, que deberá estar almacenado en la carpeta `js` del repositorio.

El script deberá mostrar una **alerta** por pantalla con tu **nombre y apellidos**.

```
js > JS presentacionNavegador.js
1 alert ("Cintia Castillo Careglio")
```



NodeJS

Por último, Node también permite ejecutar programas JavaScript que estén almacenados en ficheros. Para ello, **crea un archivo** llamado `presentacionNode.js` dentro de la carpeta `js` que **muestre en la consola tu nombre y apellidos**. Una vez guardado, puedes ejecutarlo mediante el comando `node presentacionNode.js` (recuerda acceder a la ruta donde esté almacenado el fichero si en la consola no estás en la misma carpeta que él).

```
js > JS presentacionNode.js
1 console.log ("Cintia Castillo Careglio");
```

```
PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto\js> node presentacionNode.js
Cintia Castillo Careglio
```

Formato de la entrega

- Cada persona trabajará en su **repositorio personal** que habrá creado tras realizar el *fork* del repositorio base.
- Todos los archivos de la práctica se guardarán en el repositorio y se subirán a GitHub periódicamente. Es conveniente ir subiendo los cambios aunque no sean definitivos. **No se admitirán entregas de tareas que tengan un solo commit.**

```
PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto> git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    js/presentacionNavegador.js
    js/presentacionNode.js
    presentacion.html

nothing added to commit but untracked files present (use "git add" to track)
```

- Como mínimo se debe realizar un **commit** por **cada elemento de la lista de tareas** a realizar (si es que estas exigen crear código, claro está).

```
PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto> git add .\presentacion.html
PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto> git commit -m "Presentación HTML"
[master ee7d389] Presentación HTML
1 file changed, 12 insertions(+)
create mode 100644 presentacion.html
```

```
PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto> git add .\js\presentacionNavegador.js
PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto> git commit -m "Presentación navegador"
[master 438308c] Presentación navegador
1 file changed, 1 insertion(+)
create mode 100644 js/presentacionNavegador.js
```

```
PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto> git add .\js\presentacionNode.js
PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto> git commit -m "Presentación Node"
[master 77b0b58] Presentación Node
1 file changed, 1 insertion(+)
create mode 100644 js/presentacionNode.js
```

- Para cualquier tipo de **duda o consulta** se pueden abrir Issues haciendo referencia al profesor mediante el texto @pedroprieto dentro del texto del Issue. Los issues deben crearse en **tu repositorio**: si no se muestra la pestaña de Issues puedes activarla en los Settings de tu repositorio.
- Una vez **finalizada** la tarea se debe realizar una Pull Request al repositorio base indicando tu **nombre y apellidos** en el mensaje.