

Bootcamp Java Developer

Fase 2 - Java Web Developer
Módulo 17

Consola

Consola de JavaScript

La consola de JavaScript es una herramienta **disponible en todos los navegadores que sirve para la depuración del código.**

Cuando hablamos de depuración (*debug* en inglés) nos referimos al **mantenimiento y corrección de un código fuente.**

La consola ofrece dos herramientas principales:

- La vista del navegador (mediante la tecla **F12**, **Ctrl + Shift + I** ó **Click Derecho > Inspeccionar**). Esta vista nos permite **leer** las salidas por consola.
- El objeto **console** (mediante la palabra '*console*' en JavaScript). Este objeto nos permite **escribir** en la consola.

console.assert

Registra un mensaje y envía una traza de error a la consola si el primer argumento es falso:

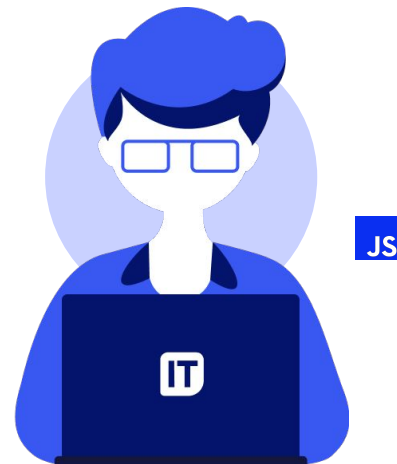
```
1  var edad = 14;  
2  
3  console.assert(edad > 18, 'No es mayor de edad');  
4
```

Este método sirve para evaluar y probar un código mediante los *Assertion Test*, que son pruebas en las que se evalúa el estado de una variable. A diferencia de un `if/else`, **console.assert** es útil solo para el desarrollador.

console.clear

Limpia la consola, **borra todos los mensajes previos:**

```
1 console.clear();  
2
```



console.error

Muestra un mensaje de error.

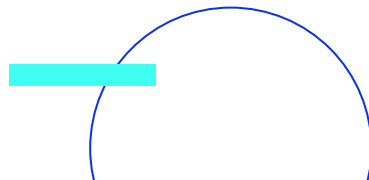
```
1 console.error('Error, usuario no autorizado')  
2
```



console.count

Registra el **número de veces** que esta línea ha **sido llamada** con la etiqueta dada:

```
1 console.count('Mi contador');  
2 console.count('Mi contador');  
3 console.count('Mi contador');  
4 console.count('Mi contador');  
5
```



console.countReset

Reinicia el valor del contador para la etiqueta dada:

```
1 console.count('Mi contador');  
2 console.count('Mi contador');  
3  
4 console.countReset('Mi contador');  
5  
6 console.count('Mi contador');  
7 console.count('Mi contador');  
8
```



console.debug

Registra un mensaje con el nivel de debug. El nivel debug es un nivel menos grave que el error.

```
1 console.debug('Este mensaje es grave, pero no tan grave como un error')  
2
```

console.info

Muestra un **mensaje de información** en la consola:

```
1 console.info('Este mensaje es una información para el desarrollador')  
2
```



console.info

console.log

Muestra un mensaje en la consola:

```
1 console.log('Este mensaje es una salida general')  
2
```



Cronómetros

Una herramienta útil para **medir el rendimiento** de un programa es el temporizador.

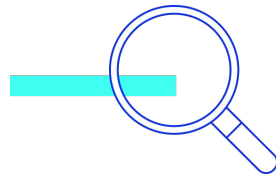
Con el temporizador es posible evaluar **cuánto tarda el navegador en ejecutar un código determinado**.

Con esta información se puede verificar si es necesario cambiar el código para que funcione más rápido.

```
1 // Inicia un cronómetro de nombre 'MedidaPerformance'
2 console.time('MedidaPerformance');
3
4 // Obtiene el valor actual de ese cronómetro
5 var valor = console.timeLog('MedidaPerformance');
6 console.log(valor);
7
8 // Finaliza el cronómetro
9 console.timeEnd('MedidaPerformance');
10
```

Revisión

- Examinar la **interfaz de la consola**.
- Comprender que la consola te sirve como desarrollador para el **mantenimiento de tu código**.
- Hacer ***Assertion Test*** con la consola.
- **Mostrar diferentes tipos de mensajes** por la consola.
- **Medir el tiempo de ejecución** de un programa.



**¡Sigamos
trabajando!**