

Bienvenidos al módulo de

Intro a JavaScript



Rol de Sensei y Co-Sensei - Conociendo a tus senseis



FELIPE ALBERTO

Bienvenidos chicos, yo soy Felipe Luna y esta es mi trayectoria:

- Fui alumno en los primeros batch de DEV.F.
- En el 2022 me gradué de la Facultad de Ciencias de la UNAM como Licenciado en Ciencias de la Computación.
- En 2015 inicio mi carrera profesional en Vexi Tarjeta de Crédito.
- En 2019 me certifico como Scrum Master.
- En 2023 comienzo a trabajar para Walmart México.

Yaxche Manrique



- Hace un año tuve mi “Desgraduación” de DEV.F
- Estudié Actuaría en la Facultad de Ciencias
- Soy Frontend Developer para una empresa Canadiense.
- Tengo dos perros y me gusta salir a patinar ..



Objetivos del Módulo

Obtener bases sólidas de programación con JavaScript y emplearlas para aprender a manipular documentos HTML mediante el DOM y el manejo de eventos

Temario



dev.f

Semana 1

	Explicar objetivos del módulo	07-08-2023
	Mostrar el proyecto de clase que se va a desarrollar a lo largo del módulo	
	Presentar la prueba de empleabilidad como reto a desarrollar de forma independiente	
Introducción al Módulo	Qué es JavaScript	
	Consulta y uso de la documentación oficial	
	Comparativas con otros lenguajes de programación	
	Consola del navegador	
JavaScript	Cómo ejecutar JavaScript	
	Sintaxis básica y creación de scripts	
	Entradas y salidas	
	Tipos de datos	
	Variables	
	Operadores	09-08-2023
	Valores truthy / falsy	
Estructuras de Control	Estructuras de control	
	Condiciones If-else	

Semana 2

Funciones	Funciones Scope	14-08-2023
Arreglos	Arreglos y sus métodos	16-08-2023
Objetos	Objetos literales y métodos	
Estructuras de Control	Ciclos For	

Semana 3

Estructuras de Control	Ciclos For	
DOM	Qué es el Document Object Model	21-08-2023
	Manipulación del DOM	
	Manejo de eventos	
Framework	Qué es un Framework	23-08-2023
	Frameworks de CSS	
	Bootstrap	
	Bootstrap Grid	

Semana 4

28-08-2023 - Dudas del Proyecto

30-08-2023 - Presentación de Proyecto

Proyecto de Módulo

Crear una aplicación web de un cajero automático
empleando HTML, CSS y JS

Intro a Javascript



dev.f

Un Poco de Historia

- JavaScript fue creado (en 10 días) originalmente en 1995 por Brendan Eich, quien trabajaba en Netscape.
- JavaScript se diseñó para que pudiera ejecutarse en el navegador del usuario y proporcionar una experiencia interactiva en tiempo real.
- La idea era que los desarrolladores pudieran escribir código que pudiera manipular los elementos de la página web de manera dinámica, en respuesta a la interacción del usuario.



JS

Qué es?

JavaScript consiste en un **lenguaje de programación interpretado**, que habitualmente se utiliza en sitios web para ejecutar acciones en el lado del cliente, estando embebido en el código fuente de la página web

Hablemos de JavaScript



!=



¿Qué es Ecmascript?



dev.f

Ecmascript

- Hay que dejar en claro que Javascript no es lo mismo que Ecmascript .
- Ecmascript es el estándar de lenguaje de programación que cumple Javascript.
- Ecmascript define la sintaxis y semántica del lenguaje de programación.



Ejecutando JavaScript



dev.f

Expressions & Statements

Expression: Elementos que conforman a los statements

Statement: Grupo de palabras y operadores que realizan una acción en específico

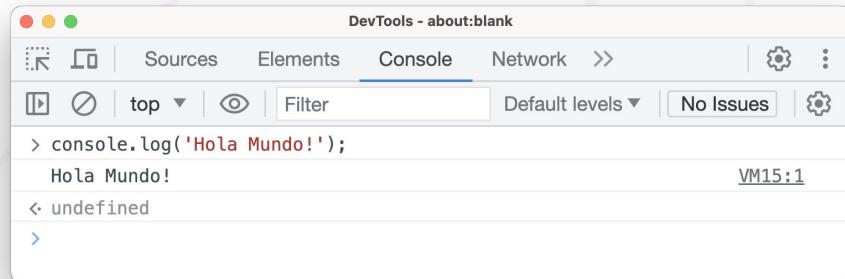
Expression ≈ Palabra

Statement ≈ Oración

console.log() - Salida en Consola / Terminal



```
1 console.log('Hola Mundo!');
```



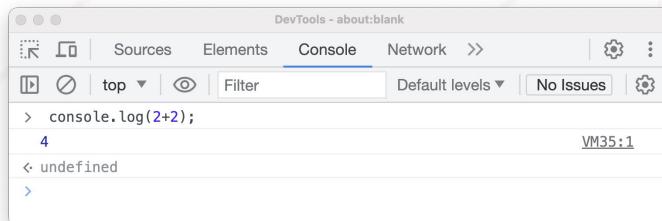
DevTools - about:blank

Sources Elements Console Network > | ⚙️ ⋮

Filter Default levels No Issues

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

VM15:1



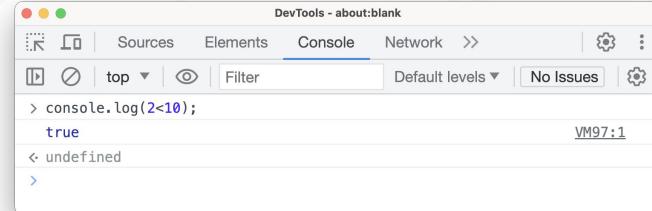
DevTools - about:blank

Sources Elements Console Network > | ⚙️ ⋮

Filter Default levels No Issues

```
> console.log(2+2);
4
< undefined
```

VM35:1



DevTools - about:blank

Sources Elements Console Network > | ⚙️ ⋮

Filter Default levels No Issues

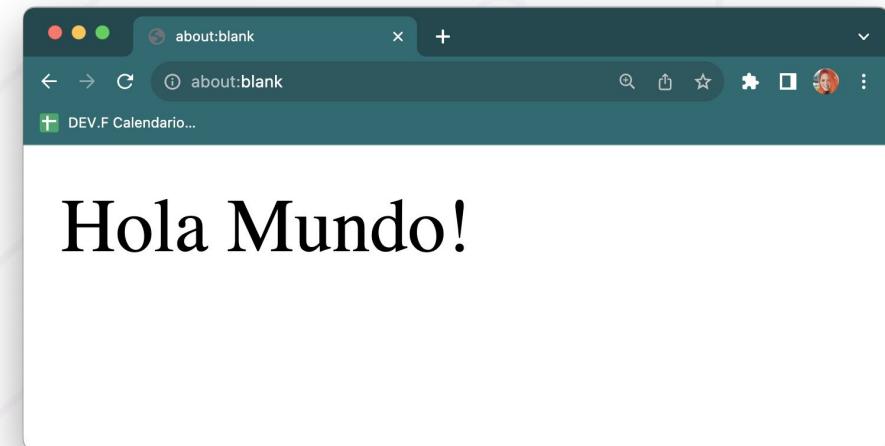
```
> console.log(2<10);
true
< undefined
```

VM97:1

document.write() - Salida en Navegador



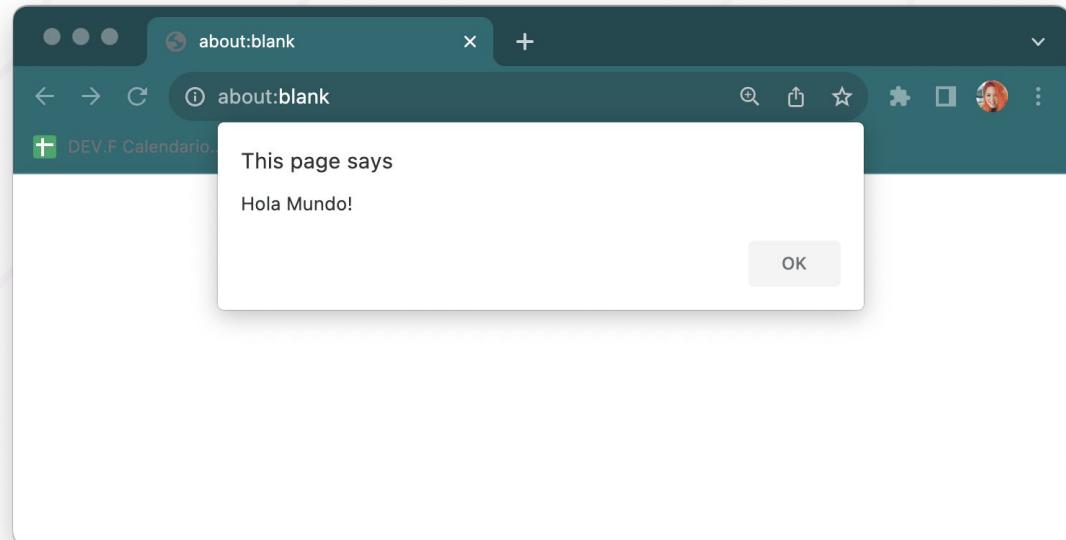
```
1 document.write('Hola Mundo!');
```



alert() - Salida en Navegador en una ventana



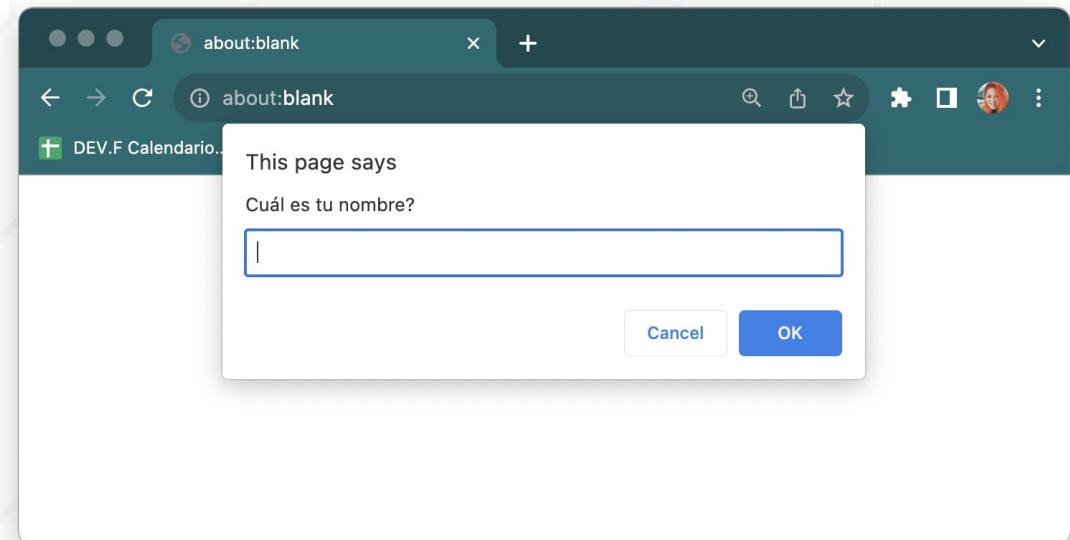
```
1 alert('Hola Mundo!');
```



prompt() - Entrada de Datos



```
1 prompt('Cuál es tu nombre?');
```



Conectando archivos JS a HTML



dev.f

Conectando archivos JS a HTML

Existen 2 formas de usar JavaScript dentro de un documento HTML

Dentro del propio documento HTML



```
1 <script>
2     /* Tu código JavaScript */
3 </script>
```

A través de un archivo externo



```
1 <script src="./app.js"></script>
```

Pero en qué parte?

En la mayoría de los casos, es buena idea colocar las etiquetas de <script> antes de cerrar el </head>. Utilizando el atributo **defer** para que nuestro JavaScript se ejecute una vez que todo esté cargado.

```
● ● ●  
1  <!DOCTYPE html>  
2  <html lang="en">  
3  <head>  
4      <meta charset="UTF-8">  
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6      <title>Document</title>  
7      <script src="./app.js" defer></script>  
8  </head>  
9  <body>  
10  
11 </body>  
12 </html>
```

Comentarios



dev.f

Formas de Comentar en JavaScript



```
1 /* Este es un comentario en JavaScript.  
2 Puede ser multilinea, por medio de / y *  
3 puedo indicar en donde comienza el comentario  
4 * Es comun en documentacion el uso de viñetas usando "*"  
5 */
```



```
1 // Este es un comentario de una sola linea  
2 // Estos empiezan con dos diagonales
```

Tipos de Dato, Variables y Operadores



dev.f

“Dentro del mundo de la computación, solo existen datos. Podemos leer datos, modificar datos, crear datos - pero eso que no sea datos, no puede ser mencionado. Toda esta información es guardada como largas secuencias de bits y es básicamente solo eso”

- Eloquent JS

Variables



dev.f

Variables

Una variable es un símbolo que permite identificar a un elemento.

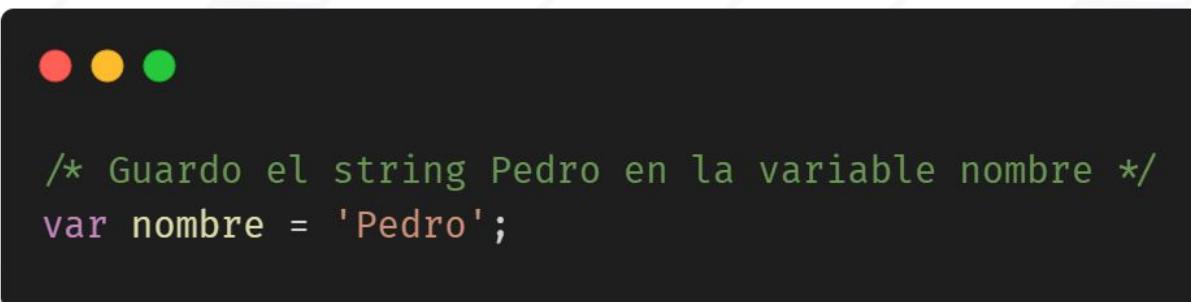
$$X = 1$$

VARIABLE

VALOR

Variables

Normalmente se utilizan para guardar o almacenar información para su uso posterior.



```
/* Guardo el string Pedro en la variable nombre */
var nombre = 'Pedro';
```

Variables

Por ejemplo, cuando nosotros agregamos un contacto en nuestro teléfono, le asignamos un nombre y lo guardamos para llamar a la persona después.



¿Cómo puedo nombrar una variable?



Tips: Nombrando Variables

Es importante dar un nombre **descriptivo** a nuestra variable.

Por ejemplo, es mejor: **temperatura**; que sólo t o tmp;

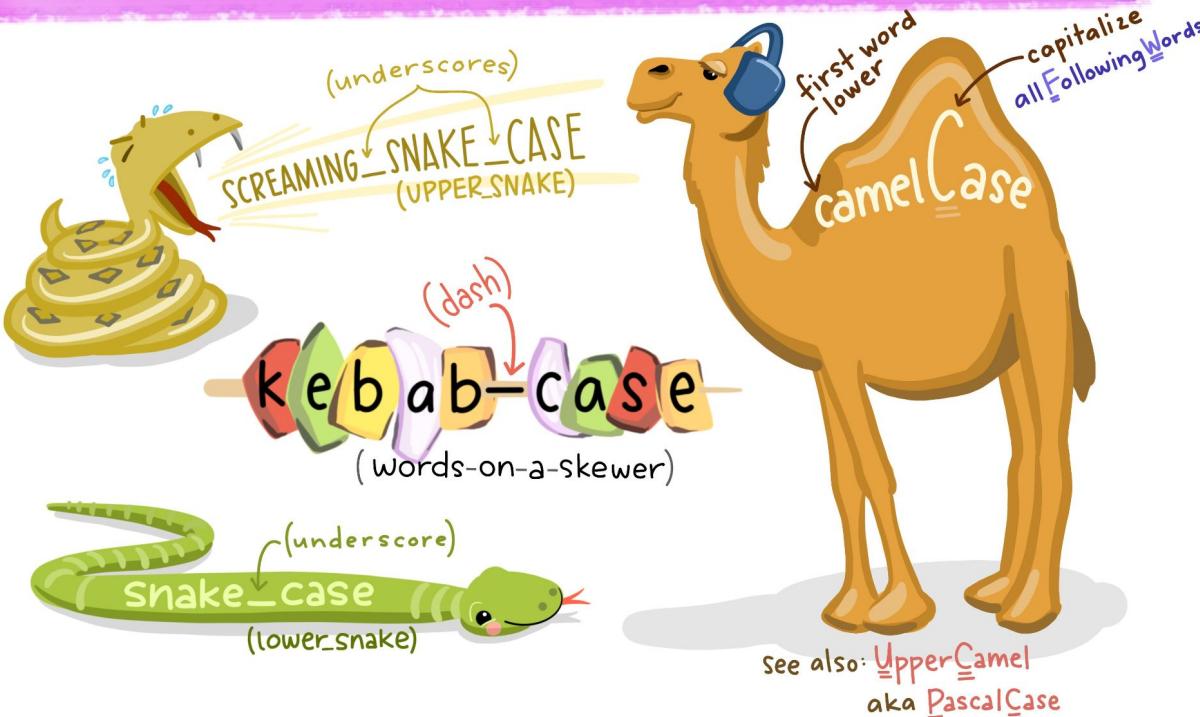
En JavaScript escribimos los nombres de variables usando **camelCase**.

Es decir, primera letra minúscula, y siguientes palabras comienzan con mayúscula, sin usar espacios.



Dato Cultural: ¿Qué otras formas existen?

in that case...



Referencia: Twitter de [Allison Horst](#)

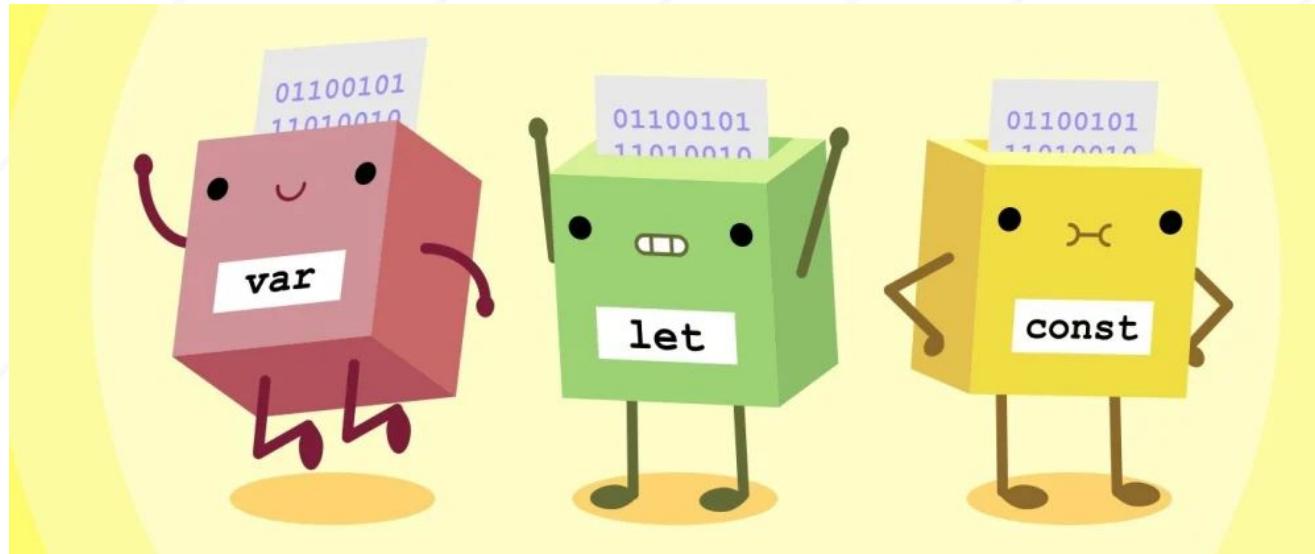
Var, let y const en JS



dev.f

Var vs let vs const en JS

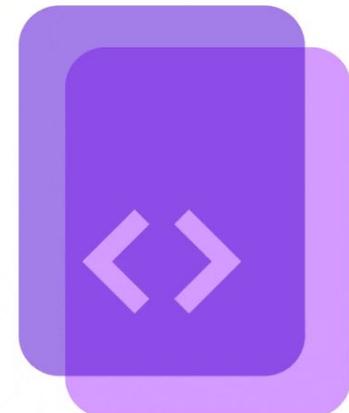
Var, let y const no son más que diferentes maneras de asignar una variable en JS las cuales podemos usar para diferentes casos.



Var vs let vs const en JS

Vamos a ver la diferencia entre cada tipo de asignación, tomando en cuenta los siguientes parámetros:

1. **Scope:** si es por función o por bloque.
2. **Re-asignación:** ¿es posible que cambie de valor?



Tipos de Datos



dev.f

“Strings”

El tipo String de JavaScript se utiliza para representar datos textuales.

Normalmente se representan encerrando el valor textual entre comillas simples o dobles.

```
"Soy un String";  
'También soy un String';
```

Numbers

Number es un objeto primitivo envolvente que permite representar y manipular valores numéricos como 37 o -9.25.

Valores de otro tipo pueden ser convertidos a números usando la función `Number()`

```
123      // one-hundred twenty-three  
123.0    // same  
123 === 123.0 // true
```

También pueden contener valores NaN ("Not a Number" o No es un número).

Booleans

En términos de informática, un boolean es un dato lógico que solo puede tener los valores true (verdadero) o false (falso).



```
/* Tipo Boolean */
true; // Verdadero
false; // Falso
```

Juegan un papel fundamental en la lógica de programación, ya que podemos establecer acciones dependiendo si se cumple o no cierta condición.

Null

El tipo Null tiene el valor: **nulo**.

```
var miVariable = null  
  
console.log(miVariable) // null
```

Un valor nulo significa que es un valor desconocido, indefinido o no inicializado

Undefined

Una variable a la cual no se le haya asignado valor tiene el valor **indefinido**.

```
var saludo;  
console.log(typeof saludo); // "undefined"
```

Symbol

Symbol es un tipo de datos cuyos valores son **únicos** e **inmutables**. Dichos valores pueden ser utilizados como identificadores (claves) de las propiedades de los objetos. Cada valor del tipo **Symbol** tiene asociado un valor del tipo String o Undefined que sirve únicamente como descripción del símbolo.

Array

Los arrays son objetos similares a una lista. Podemos acceder a sus valores por medio del índice en el que se ubican.



```
1 const friends = ['Monica', 'Phoebe', 'Rachel', 'Joey', 'Chandler', 'Ross'];
```

Object

Un Objeto es un valor en memoria al que podemos acceder por un identificador.

```
1 const yaxche = {  
2     nombre: 'Yaxche',  
3     cantidadPerros: 2,  
4     instagram: '@yaxchemanrique',  
5     esMilenial: true,  
6     trabajo: 'Sensei',  
7 };
```

En JavaScript los objetos pueden ser vistos como una colección de propiedades.

Operadores



dev.f

Operadores

Asignación (=)

Suma (+)

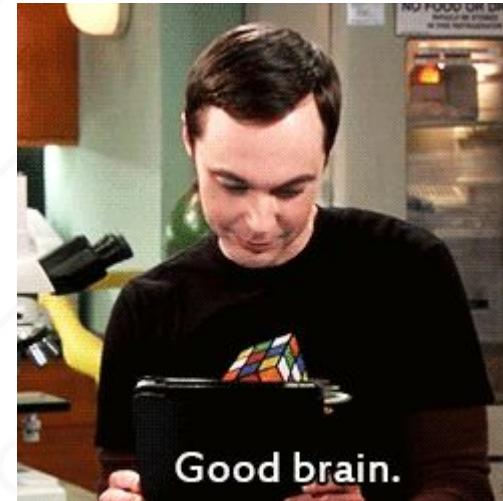
Resta (-)

Multiplicación (*)

División (/)

Residuo de una división (%)

Exponencial (**)



Incremento y Decremento

Incremento:

- **++**

- **+ =**

Decremento:

- **--**

- **- - =**

Comparación | Booleanos

Comparación:

- `>`
- `<`
- `<=`
- `>=`

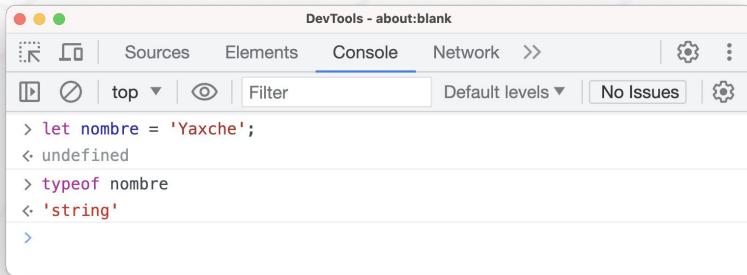
Igualdad:

- `==`
- `=====`
- `!=`
- `!= =`

Lógicos:

- `&&`
- `||`

Operador `typeof`

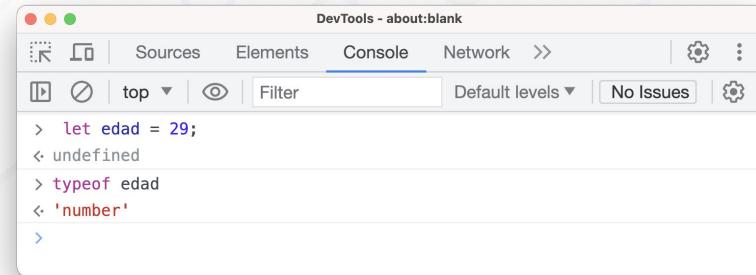


DevTools - about:blank

Sources Elements Console Network > | ⚙️ ⋮

Default levels ▾ No Issues ⚙️

```
> let nombre = 'Yaxche';
< undefined
> typeof nombre
< 'string'
>
```



DevTools - about:blank

Sources Elements Console Network > | ⚙️ ⋮

Default levels ▾ No Issues ⚙️

```
> let edad = 29;
< undefined
> typeof edad
< 'number'
>
```

Actividad 1.

Intro a JavaScript

1. En la consola del navegador, crea un pequeño programa que le pida al usuario que ingrese un número, multiplica ese número por 2 y muestra en salida el resultado de esa operación. Guarda el código en un archivo js.
2. En la consola del navegador escribe un programa que reciba 2 números e imprime en pantalla la resta de ambos. Guarda el código en un archivo js.
3. Escribe un programa que calcule el área de un círculo con un radio $r = 5.3$. Imprime el área en pantalla. (Fórmula: $\pi \times r^2$)
4. Escribe un programa que calcule el área de un rombo, sabiendo que $p = 5\text{cm}$ y $D=8\text{cm}$, $d=6\text{cm}$ (Fórmula: $a=D \times d / 2$)