

Introducción a Javascript

Contenidos de la Clase

- Variables y constantes
- Prompt y Alert
- Reglas de Escritura
- Operadores Lógicos
- Operadores de Comparación
- Operadores Matemáticos
- Funcionamiento de los condicionales
- Vincular JS con HTML

Variables

Las variables se pueden utilizar para almacenar diferentes tipos de datos, los que vimos hasta ahora son:

- Números
- Cadenas de caracteres
- Booleanos

Para declarar una variable en JavaScript, se puede utilizar la palabra “var” o “let”.

Nota: La palabra reservada “var” ya no se recomienda, por eso intentaremos utilizar siempre “let”, ya que tiene un alcance más claro y predecible.

Variables - Nomenclatura

Las reglas para asignar nombres de variables en JavaScript son las siguientes:

1. Los nombres de variables pueden contener letras, números, guiones bajos (_) o símbolos de dólar (\$). De estos símbolos, el único con el que no se puede comenzar el nombre es con un número.
2. Los nombres de las variables son sensibles a mayúsculas y minúsculas.
3. Los nombres de variables deben ser descriptivos y significativos.
4. Evite utilizar nombres de palabras reservadas en JavaScript.
5. Los nombres de las variables preferiblemente deben estar en camelCase.

Juguemos a revisar nombres de variables...

¿Este nombre es correcto?



variables.js

```
let $color_principal1
```

¿Este nombre es correcto?



```
variables.js  
  
let $color_principal1
```

Los nombres de variables sí pueden comenzar con el símbolo \$ (dólar).

¿Este nombre es correcto?



```
variables.js  
  
let _usuarioNombre
```


¿Este nombre es correcto?



```
variables.js  
  
let _usuarioNombre
```

Los nombres de variables sí pueden comenzar con el símbolo _ (guion bajo).

¿Este nombre es correcto?



```
variables.js  
  
let 5toValor
```

¿Este nombre es correcto?



```
variables.js

let 5toValor
```

Los nombres de variables no pueden comenzar con un número.

¿Este nombre es correcto?



variables.js

```
let sumaFinal
```

¿Este nombre es correcto?



```
variables.js  
  
let sumaFinal
```

Los nombres de variables sí pueden comenzar con una letra minúscula.

¿Este nombre es correcto?



variables.js

```
let /estatura_usuario
```

¿Este nombre es correcto?



NO

```
variables.js  
  
let /estatura_usuario
```

Los nombres de variables no pueden contener símbolos propios de las operaciones matemáticas.

¿Este nombre es correcto?



variables.js

```
let color-1
```


¿Este nombre es correcto?



```
variables.js  
  
let color-1
```

Los nombres de variables no pueden contener símbolos propios de las operaciones matemáticas.

**Pensemos qué valor final se
guarda en cada variable**

¿Qué valor contiene la variable?

variables.js

```
let total = 5+3;
```



¿Qué valor contiene la variable?

variables.js

```
let total = 5+3;
```



¿Qué valor contiene la variable?

variables.js

```
let num1 = 7;  
let total = num1 * 2;
```



¿Qué valor contiene la variable?

variables.js

```
let num1 = 7;  
let total = num1 * 2;
```

14

¿Qué valor contiene la variable?

variables.js

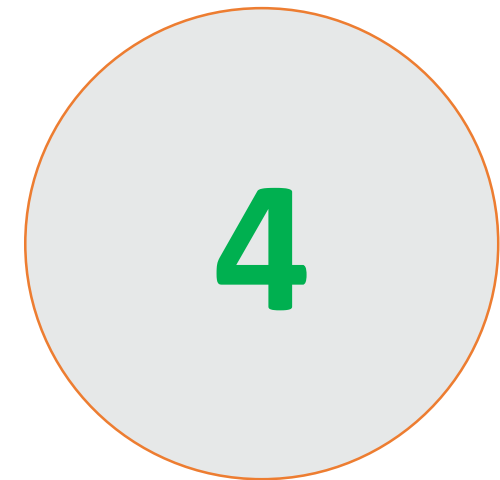
```
let num1 = 7;  
let num2 = 3;  
let total = num1 - num2;
```



¿Qué valor contiene la variable?

variables.js

```
let num1 = 7;  
let num2 = 3;  
let total = num1 - num2;
```



¿Qué valor contiene la variable?

variables.js

```
let a = "ho";  
let b = "la";  
let c = 3;  
let total = a + b + c;
```



¿Qué valor contiene la variable?

variables.js

```
let a = "ho";  
let b = "la";  
let c = 3;  
let total = a + b + c;
```



hola3

¿Qué valor contiene la variable?

variables.js

```
let a = "70";  
let b = "40";  
let total = b + a;
```



¿Qué valor contiene la variable?

variables.js

```
let a = "70";  
let b = "40";  
let total = b + a;
```

4070

¿Qué valor contiene la variable?

variables.js

```
let a = "hola";
```

```
let a = 4;
```

```
let a = a + 2;
```

```
let total = a;
```



¿Qué valor contiene la variable?

variables.js

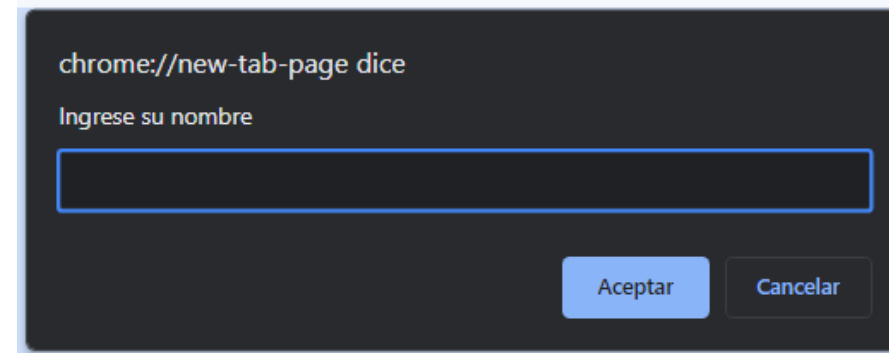
```
let a = "hola";  
let a = 4;  
let a = a + 2;  
let total = a;
```



Prompt y Alert

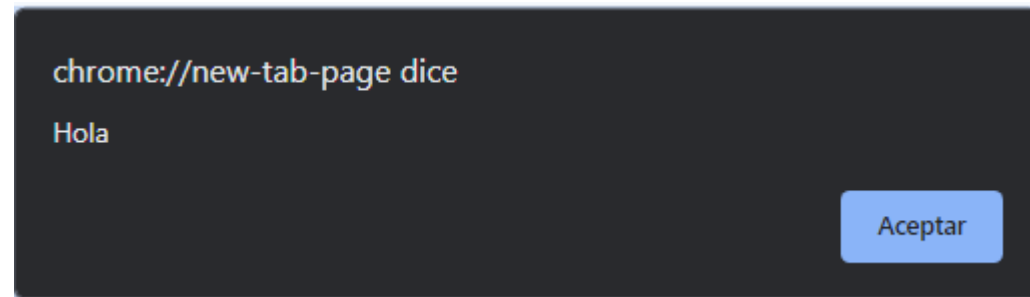
prompt() es un método de JavaScript que se usa para mostrar un cuadro de diálogo con un mensaje que solicita al usuario que ingrese algún texto o información.

```
> prompt("Ingrese su nombre")
```



alert() muestra un diálogo de alerta con un mensaje opcional, y aguardará hasta que el usuario cierre la ventana de diálogo.

```
alert("Hola")
```



Operadores Lógicos

&& : Operador **AND** (Y). Devuelve “true” si ambos operandos son verdaderos.

|| : Operador **OR** (O). Devuelve “true” si al menos uno de los operandos es verdadero.

! : Operador **NOT** (NO). Devuelve el valor inverso del operando.

Estos operadores se utilizan principalmente en las estructuras de control de flujo, como los condicionales.

p	q	p && q
F	F	F
V	V	V
F	V	F
V	F	F

p	q	p q
F	F	F
V	V	V
F	V	V
V	F	V

p	! p
F	V
V	F

Operadores de Comparación

`==` : Compara si dos valores son iguales sin tener en cuenta el tipo de dato.

`===` : Compara si dos valores son iguales y tienen el mismo tipo de dato.

`!=` : Compara si dos valores no son iguales sin tener en cuenta el tipo de dato.

`!==` : Compara si dos valores no son iguales y tienen diferentes tipos de dato.

`<` : Compara si el primer valor es menor que el segundo valor.

`>` : Compara si el primer valor es mayor que el segundo valor.

`<=` : Compara si el primer valor es menor o igual que el segundo valor.

`>=` : Compara si el primer valor es mayor o igual que el segundo valor.

Los operadores de comparación devuelven siempre un valor booleano: verdadero (“true”) o falso (“false”).

Operadores Matemáticos

Operador	Propósito	Ejemplo	Atajo de
+=	Suma el valor de la derecha al valor de la variable de la izquierda y retorna el nuevo valor	x = 3; x += 4;	x = 3; x = x + 4;
-=	Resta el valor de la derecha, del valor de la variable de la izquierda y retorna el nuevo valor.	x = 6; x -= 3;	x = 6; x = x - 3;
*=	Multiplica el valor de la variable en la izquierda por el valor en la derecha y retorna el nuevo valor.	x = 2; x *= 3;	x = 2; x = x * 3;
/=	Divide el valor de la variable en la izquierda por el valor de la derecha y retorna el nuevo valor.	x = 10; x /= 5;	x = 10; x = x / 5;

Condicionales

Condicional: tipo de estructura de control de flujo que permite que el programa ejecute diferentes bloques de código según una condición booleana evaluada como verdadera o falsa.

```
let edad = 18;  
if (edad >= 18) {  
  console.log("Sos mayor de edad");  
}  
else{  
  console.log("Sos menor de edad");  
}
```

Condicionales

A veces vamos a necesitar tener un condicional dentro de otro, o condicionales anidados.

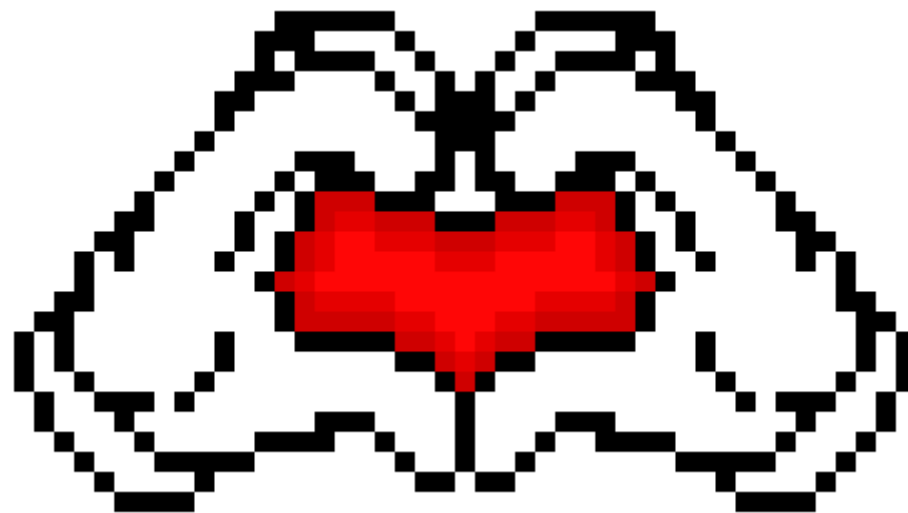
```
let nota = 8;  
if (nota >= 9) {  
  console.log("Promocionado");  
} else if (nota >= 6) {  
  console.log("Aprobado");  
} else {  
  console.log("Insuficiente");  
}
```

Veamos cómo vincular un archivo JS al HTML

¿Preguntas?



¡Nos vemos la próxima clase!



BA MULTIPLICA 2.0

jóvenes  jóvenes



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

