

# NODE JS

## Clase 7: Java/Javascript - variables - Igualdades



## ... Clase 7

01

Java y  
Javascript

02

Variables

03

Igualdades



Nación  
Servicios



# Introducción a JavaScript

## ... Introducción a JavaScript

**JavaScript** es un lenguaje de programación multiplataforma orientado a objetos utilizado frecuentemente para hacer que las páginas web sean interactivas.



## ... Introducción a JavaScript

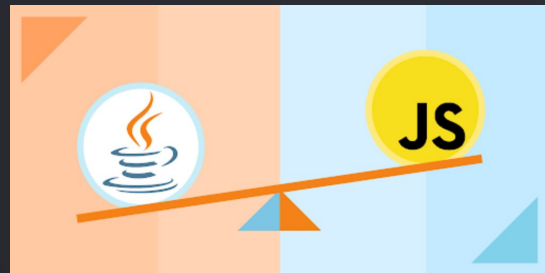
Se caracteriza por tener un tipado débil, y por una gran versatilidad lo que posibilita su uso, tanto para **back** como para **frontEnd**. Siendo el lenguaje por excelencia para este último.



## ... Introducción a JavaScript

Inicialmente se llamó **LiveScript**, dado que tomaba la mayoría de reglas a nivel sintaxis de las expresiones de **Java**, su nombre cambió a **JavaScript**.

Se diferencia de éste, en que no tiene el tipado estático ni la fuerte verificación de tipos.



## ... JavaScript y Java, POO

Mientras que **Java** sostiene el modelo de objetos a partir de clases, **JavaScript** lo hace a partir de **prototipos** proporciona herencia dinámica; es decir, lo que se hereda puede variar en objetos individuales.



## ... JavaScript y Java, POO

**JavaScript** también admite funciones sin requisitos declarativos especiales. Las funciones pueden ser propiedades de objetos, ejecutándose como métodos débilmente tipados.





## ... JavaScript y Java, POO

Todo esto hace que JavaScript sea un lenguaje mucho **más flexible** comparado con Java. No es necesario declarar todas las variables, clases y métodos. No tienes que preocuparte por si los métodos son públicos, privados o protegidos, y no tienes que implementar interfaces

# ¡Hola Mundo!



Para comenzar a escribir JavaScript, escribiremos nuestro primer código en JavaScript, el clásico “Hola mundo!”:

```
alert("Hola Mundo");
```



## ... Clase 7

01

Java y  
Javascript

02

Variables

03

Igualdades



Nación  
Servicios



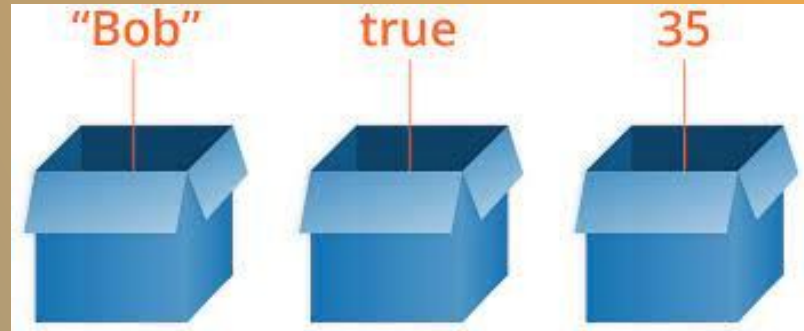
# Variables primitivas

A grayscale photograph of four people (three men and one woman) sitting around a desk in a library, working together. They are looking at papers and a laptop. The background is filled with bookshelves. The text 'Variables primitivas' is overlaid in the center in a bold, orange font.

# Variables primitivas

En **JavaScript**, un primitive (valor primitivo, tipo de dato primitivo) son datos que no son un objeto y no tienen métodos. Hay 6 tipos de datos primitivos:

- string
- number
- bigint
- boolean
- undefined
- symbol.



Algunos autores, pueden tomar También nul, aunque es un caso especial.

## ... Booleanos

Son la unidad básica del universo informático, un **bit**, un valor que puede representar un uno o un cero, verdadero, falso, positivo, negativo.

Puntualmente en **JavaScript**, los veremos representados únicamente por “true” o “false”. Siendo por default, falsos.



## ... Number

En **JavaScript**, los objetos y variables del tipo **Number**, permiten trabajar con variables numéricas, tanto enteras como de punto flotante.



## ... Number

Dentro de los detalles agregados por **JavaScript**, al momento de trabajar con números, aparece la variable **NaN** (Not a Number, “no es un número”) Este valor se retorna en casos donde se esté esperando una variable numérica y la misma no aparezca.



# Asignaciones numéricas



Nación  
Servicios



Veremos como generar las distintas asignaciones matemáticas, a partir de la sintaxis en **JavaScript**:

- `var x=102; //Entero`
- `var y=102.7;//Decimal (float, flotante)`
- `var z=13e4;//exponencial`
- `var n=new Number(16);// Objeto numérico`

**Salidas:** 102 102.7 130000 16

## ... Funciones numéricas

- **MIN\_VALUE**: retorna el mínimo valor.
- **MAX\_VALUE**: retorna el máximo valor.
- **POSITIVE\_INFINITY**: infinito positivo, usado también en caso de desborde.
- **NEGATIVE\_INFINITY**: infinito negativo, usado también en caso de desborde.

## ... Strings (cadenas)

Por definición, strings son secuencias de caracteres alfanuméricos (letras, números y/o símbolos) muy utilizados en programación. En **Javascript**, una cadena siempre estará entre comillas tanto simples, como dobles (**" "**, **' '**).

## ... Strings (cadenas)

Cada vez que manipulamos una cadena, se crea una nueva instancia por dentro de javascript, lo que significa que se generará un nuevo espacio de memoria con una copia del valor de la cadena. Así que tenemos que usar una variable para almacenarlo.

## ... Strings (cadenas)

Es posible interpolar, concatenar, verificar posiciones de caracteres o incluso reemplazar partes de strings. Para esto, emplearemos los métodos:

- **.length**: Devuelve la longitud de la cadena
- **charAt(n)**: Devuelve el carácter ubicado en la posición n
- **indexOf(c)**: Devuelve la posición en la que 'c', aparece por primera vez
- **toUpperCase()/LowerCase**: Convierte a mayúscula y minúscula respectivamente.
- **trim()**: elimina los espacios en blanco al principio o al final de una string

## ... Strings (cadenas)

- **substr()**: Permite extraer parte de una cadena.
- **slice(n,m)**: Similar a substr, solo qué pasamos n y m para establecer carácter inicial y final a extraer.
- **replace()**: Reemplaza parte de una cadena por otra.
- **concat()**: Concatena de manera alternativa a '+'. En este caso, solo debemos pasar las dos cadenas en cuestión.
- **split()**: permite romper una string a partir de caracteres separadores que informaremos al método como parámetro.

A grayscale photograph of four people (two women and two men) sitting at a long table in a library, surrounded by bookshelves. They are looking at papers and a laptop, appearing to be in a collaborative work or study session. The image is dimly lit, with the text overlaid in a bright yellow color.

# Particularidades con variables

## ... Hoisting

En **JavaScript**, cada variable es "elevada" (**hoisting**) a la parte superior de su contexto de ejecución. Este mecanismo mueve las variables a la parte superior de su alcance antes de que se ejecute el código.



## ... Hoisting

Es por esta misma razón que **"es posible usar una variable antes de que se haya declarado"**: en tiempo de ejecución, la variable se elevará (**hoisting**) y todo funcionará correctamente.

# Hoisting en JavaScript



Nación  
Servicios



**Mediante este recurso, es que en JavaScript es posible usar una variable antes de que se haya declarado**

## ... Hoisting

```
var exhibeMensaje = function () {  
  mensaje = 'Alura';  
  console.log (mensaje);  
  var mensaje;  
}
```

Al ejecutar: exhibeMensaje(); // Imprime 'Alura'

## ... **Var**

Las declaraciones de variables, donde sea que ocurran, son procesadas antes de que cualquier otro código sea ejecutado. El ámbito de una variable declarada con la palabra reservada `var` es su contexto de ejecución en curso.

## ... **Var**

Puede ser la función que la contiene o, para las variables declaradas fuera de cualquier función, un ámbito global. Si volvemos a declarar una variable Javascript, está no perderá su valor.

... **let**

let te permite declarar variables limitando su alcance (**scope**) al bloque, declaración, o expresión donde se está usando. a diferencia de la palabra clave var la cual define una variable global o local en una función sin importar el ámbito del bloque.

La otra diferencia entre var y let es que este último se inicializa a un valor sólo cuando un analizador lo evalúa.

## ... **const**

Esta declaración crea una constante cuyo alcance puede ser global o local para el bloque en el que se declara.

Es necesario inicializar la constante, es decir, se debe especificar su valor en la misma sentencia en la que se declara, lo que tiene sentido, dado que no se puede cambiar posteriormente.



## ... Clase 7

01

Java y  
Javascript

02

Variables

03

Igualdades



Nación  
Servicios





## ... Igualdad estricta y regular

El operador de estricta igualdad (**===**) revisa si dos operandos son iguales y produce un resultado Booleano.

A diferencia del operador de igualdad regular (**==**), el operador de estricta igualdad siempre considera que los operandos de distinto tipo de valor son diferentes y nunca similares.

## ... Igualdad estricta y regular

En palabras más sencillas, la doble igualdad, chequea solo el valor de las variables. Mientras que la triple igualdad chequea también los tipos de variables involucrados, para asegurar una igualdad.

## ... Igualdad estricta y regular

```
var numero = 5;
```

```
var cadena = "5";
```

```
console.log(numero == cadena);
```

// Salida: true, (dado que se realiza una conversión automática de tipo)

```
console.log(numero === cadena);
```

// Salida: false (se compara valor y tipo, y son diferentes)

A grayscale photograph of four people (two men and two women) sitting at a long table in a library or study. They are looking at papers and a laptop, appearing to be in a collaborative study or work session. The background is filled with bookshelves packed with books. The image has a dark, semi-transparent overlay.

# Introducción a funciones

## ... Funciones en JavaScript

Las funciones son elementos básicos dentro del desarrollo informático, ya que nos permiten:

- Agrupar código de manera organizada.
- Llevar a cabo operaciones de manera reiterada sin necesidad de reiteración de código.
- Deben brindar solución a un conjunto extenso de casos.

## ... Anatomía de una función en JavaScript

```
function nombreDeLaFuncion(parametro1, parametro2) {  
  var variableLocal = 10;  
  return resultado;  
}  
var resultado = nombreDeLaFuncion(valor1, valor2);
```

## ... Nomenclatura en funciones

### Declaración de función:

Comienza con la palabra clave `function`, seguida del nombre de la función, paréntesis que pueden contener los parámetros separados por comas (opcional) y luego abre una llave `{` para definir el cuerpo de la función.

## ... Nomenclatura en funciones

### Nombre de la función:

Es el identificador único de la función. Puedes usar este nombre para invocar la función desde otras partes del código.



## ... Nomenclatura en funciones

### Parámetros (opcional):

Son variables que actúan como marcadores de posición para los datos que se pasan a la función cuando se la invoca. Los parámetros se definen entre paréntesis y separados por comas. Dentro del cuerpo de la función, puedes utilizar estos parámetros como variables locales.

## ... Nomenclatura en funciones

### Cuerpo de la función:

Aquí es donde defines la lógica de la función. Puedes realizar operaciones, manipular datos, utilizar condicionales, bucles, etc. dentro del cuerpo de la función.

## ... Nomenclatura en funciones

### Instrucción de retorno (opcional):

Si quieres que la función devuelva un valor, puedes utilizar la palabra clave `return` seguida del valor que deseas devolver. Si no hay una declaración de retorno o si no se proporciona un valor, la función devolverá `undefined` por defecto.

## ... Nomenclatura en funciones

### Llamada a la función:

Para ejecutar el código dentro de la función y obtener un resultado (si la función devuelve algo), simplemente escribe el nombre de la función seguido de paréntesis que contengan los valores que deseas pasar como argumentos a los parámetros de la función.

# ¡Muchas gracias!

