LA PROGRAMACIÓN: INICIATE EN UN MUNDO APASIONANTE

# 1. FUNDAMENTOS DE LA PROGRAMACIÓN

Vamos a empezar a programar...

# Tipos de datos

Hablábamos que programar era mover datos de un sitio a otro. Vamos a ver qué tipos de datos tenemos en JavaScript (el lenguaje de programación que hemos elegido para este curso)

### Números

No os asustéis, que las matemáticas no son complejas. Podemos usar números enteros positivos y negativos. Pueden ser número muy largos, para los que JavaScript usa una notación científica. También podemos usar números decimales.

### Cadenas

Conjunto de caracteres. Las definimos entre comillas (simples ' o dobles "). También suele usarse en su término en inglés, *strings*.

### Booleanos

Es la representación que tiene el sistema para lo verdadero y lo falso. Es el principio de la lógica. Para nombrarlos, usamos las palabras en inglés *true* y *false*.

### Tipos especiales

**Indefinido**: usamos la palabra en inglés *undefined* para indicar que un valor no está definido

**Nulo**: usamos la palabra en inglés *null* para indicar que no hay ningún valor

**No es un número** (Not a Number - NaN): indica una indeterminación matemática, por ejemplo, la que resulta de dividir 0 entre 0

**Infinito** (Infinity): indica el valor infinito

# Operaciones con datos

En esta sección hablaremos de las operaciones que podemos hacer con los tipos definidos anteriormente.

## Operaciones con números.

**Suma**. Ejemplo: `3 + 5`

**Resta**. Ejemplo: `4 - 7`

**Multiplicación**: usamos el símbolo `*` para indicar multiplicación. Ejemplo: `4 * 8`

**División**: usamos el símbolo `/` para indicar división. Ejemplo: `50 / 5`

**Módulo**: operación que calcula el resto de la división entera, usando el símbolo %. Ejemplo: `45 % 6`

**Precedencia de operadores:** como en matemáticas, las operaciones precedentes son la multiplicación y la división. Si queremos cambiar esta precedencia, usamos paréntesis `(` y `)` para agrupar las operaciones preferentes

## Operaciones con cadenas

**Concatenación**. Consiste en unir dos cadenas de texto, siendo el resultado una única cadena con la yuxtaposición de las cadenas originales. Usaremos el operador `+` para indicar la concatenación de cadenas.

**Acceso a las letras de una cadena**. Usamos el operador corchetes (square brackets en inglés) indicando el índice de la letra de la cadena que queremos recuperar. Los índices comienzan en 0 para la primera posición de la cadena, y van aumentando de uno en uno hasta llegar a la última posición. Ejemplo: `"pelota"[1]` tendrá como resultado una `"e"`

**Operaciones con booleanos.**

**Negación**. Usamos el operador `!` que se antepone al valor que queremos negar. Ejemplo: `!true` da como resultado `false`

**And lógico**. Usamos el operador `&&`. Sólo si ambas expresiones son verdaderas, dará como resultado verdadero. Ejemplo: `true && false` da como resultado `false`

**Or lógico**. Usamos el operador `||` Sólo si ambas expresiones son falsas, dará como resultado falso. Ejemplo: `true || false` da como resultado `true`

# Variables

Una variable  es un cajón con una etiqueta donde almacenamos datos de un programa. Para definir una variable en JavaScript, usamos la palabra reservada `var` seguida del nombre que le queremos dar a la variable. En este cajón, podemos meter cualquier tipo de dato: numérico, cadena o booleano. Al declarar una variable, podemos indicar su valor usando el signo `=`. Ejemplo: `var palindromo = "oso"`.

Podemos volver a asignar (reasignar) un valor a una variable usando también el signo `=`. Ejemplo: `palindromo = "radar"`.

Como ejemplo para guiar el curso, vamos a tomar el escenario de una baraja de cartas. En este caso, usaremos una variable carta con el contenido de una cadena con la carta indicada. Pero para transmitir esta idea a nuestro programa, tenemos que realizar una simplificación: usaremos dos caracteres en la cadena que representa a una carta, uno con el palo y otro con el valor. Para el palo, en vez de usar el texto completo usaremos solamente la primera letra. Ejemplo: `var carta = "1c"`. Por tanto, para acceder al valor y al palo de una carta, podremos hacerlo con el operador corchetes: `var valor = carta[0]; var palo = carta[1].`

# Comparaciones

Dicen que las comparaciones son odiosas: ¡vamos a verlo!

## Igualdad y desigualdad

Usaremos el operador `===` para las comparar que dos valores (números, cadenas o booleanos) sean iguales, ya sean literales (valores tal cual) o variables. Para comparar que dos valores sean distintos usamos `!==`. Existen también los operadores `==` y `!=` pero desaconsejamos su uso porque devuelven resultados positivos al comparar entre valores de tipo distinto, como números y cadenas.

## Mayor y menor

Para comparar números podemos utilizar los operadores `>` (mayor que) y `<` (menor que). Ejemplo: `3 < 4` devolverá un valor verdadero. También podemos incluir el valor comparado con los operadores `>=` (mayor o igual que) y `<=` (menor o igual que).

## Falsy y truthy

En JavaScript tenemos dos tipos especiales que son falsy y truthy. Falsy representa un valor falso, que puede ser el booleano false, la cadena vacía `""` o el número `0`. Truthy, por contra, representa un valor verdadero que puede ser el booleano true, una cadena no vacía y un número distinto de `0`.

Ejemplo: `3 && true` devuelve verdadero ya que tanto `3` como `true` son valores truthy.

Ejemplo: `true && ""` devuelve falso porque `""` es un valor falsy.