# 1. CONCEPTOS BÁSICOS

**Acerca del lenguaje**

- Múltiples entornos de ejecución en JS:
- Usado mediante un navegador web
- Lenguaje interpretado
- Tipado dinámico - La variable no conservará un tipo de dato único
- Débilmente tipado: No corrige el tipo de dato
- Las variables son nombres que apuntan a un espacio en la memoria virtual
- Las variables no deben empezar por números
- Math métodos

**Tipos de datos**

- Los tipos de datos son anotaciones para el compilador
- Primitivos:  number, defined, string, symbol, boolean, null
- El sistema analiza el código para definir el tipo de dato
- Los tipos de datos primitivos son inmutables es decir no son alterables

```
let number = "Hola"; //inmutable

number = 20;
```

- Cada tipo de dato le corresponde su valor en objeto, si usamos una función de estas se convertirá en el tipo de dato de la función usada.

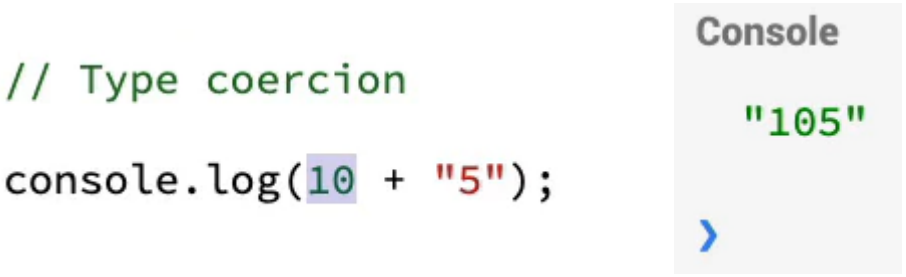
```
/*
string String()
number Number()
boolean Boolean()
undefined
symbol Symbol()
null

BigInt()
*/
```

- Hay métodos como parseInt, String, parseFloat, que son conversiones explícitas porque podemos ver cómo se ejecutan

**Coerción de tipos:**

- Existen conversiones implícitas, no son visibles y no las llamamos, son automáticas ej:  JS decide cómo convertir el dato




- Para evitar comportamientos inesperados hay que convertir explícitamente

**Booleanos:**

Estos son los únicos datos que se toman como false , de resto siempre serán tomados como true


**Truthy y Falsy**

En JavaScript y a lo largo del curso me escucharás usar dos conceptos que de hecho son bastante divertidos de pronunciar, los valores Truthy y Falsy.

Decimos que un valor es Falsy cuando su representación booleana es falso, como mencioné en el tema anterior, los valores Nan, null, 0, -0, “”, y false son los considerados falsy.

Los valores truthy por su parte, son todos aquellos que no sean falsy, es decir que su representación booleana sea verdadero.

En muchos contextos del lenguaje, decir que retorna verdadero o falso no es correcto si no están retornando un booleano, por eso solemos usar las expresiones truthy para referirnos a cualquier valor verdadero, no solamente true, y falsy, para referirnos a cualquier valor falso, no solamente false.

Cuando el intérprete necesita saber si un valor es truthy o falso hace un proceso llamado type coercion, del que hablaremos más adelante, que en términos simples

significa que hará una conversión implícita, si lo simplificamos más significa que el lenguaje convertirá el valor a verdadero para evaluar si es truthy o falsy. Esta conversión es, digamos, momentánea, el valor original o la variable no cambian su valor, javaScript sólo obtendrá su representación booleana para saber si es truthy o falsy, sin modificar el valor original.

**Operadores de comparación**
Permiten formular expresiones booleanas que retornan true o false, hay 8 en JS:

```
== Igual
=== Estricrictamente igual

!= Desigualdad
!== Desigualdad estricta

x > y     Mayor que
x < y     Menor que

>= Mayor o igual que
<= Menor o igual que
```

**Operadores lógicos:**
Permiten formular expresiones que retornan cualquier tipo de valor, en JS hay 4:

```
&& El operador AND
|| El operador OR
! El operador de negación o NOT
?? El operador nullish coalescing // fusión de nulos o unión nulosa
```

**Condicionales:**

JavaScript

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

**Ciclos:**
Ayudan a ejecutar un mismo bloque de código muchas veces

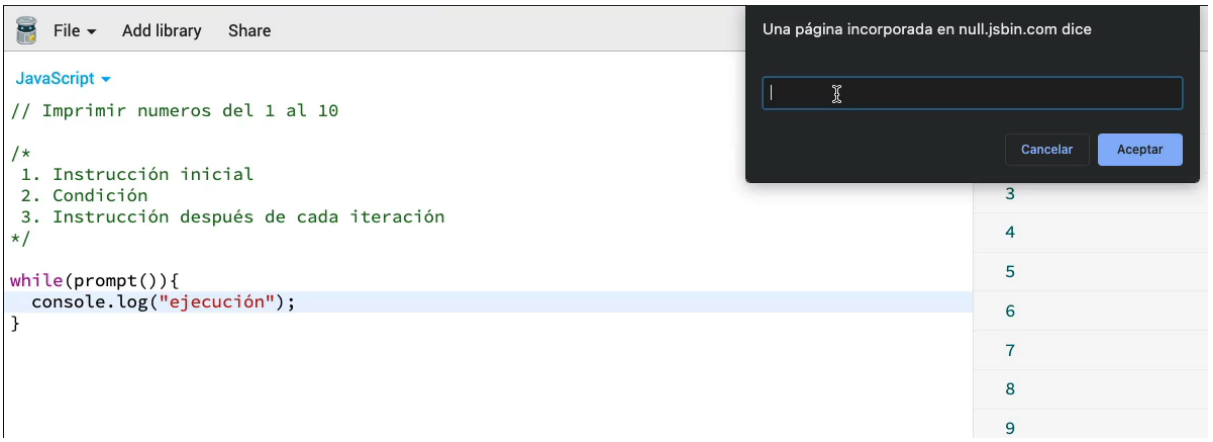
```javascript
// Imprimir numeros del 1 al 10

/*
 1. Instrucción inicial
 2. Condición
 3. Instrucción después de cada iteración
*/

for(let i = 1; i <= 10; i++){
  console.log(i);
  if(i % 2 != 0){ continue; }

  console.log("Es par");
}

console.log("Hola");
```




```javascript
do{
  console.log("ejecución");
}while(prompt());
```

**undefined, null y NaN**

- Undefined: Es un tipo de dato que se le asigna a un valor cuando no ha sido declarado o no tiene valor.
- Null: Objeto que indica la ausencia de valor, es asignable
- NaN: Representa Not a Number, se usa cuando hay un error con numeros