

Variables con JS

Declaración

Se declaran con la palabra **var** (antigua pero aún usada), o **let** seguida del nombre de la variable:

```
let nombre_gato;
```

Les podemos asignar directamente a un valor:

```
let nombre_gato = "Isidoro";
```

Podemos declarar varias variables en una línea.

```
let nombre_gato = "Isidoro", vidas_restantes = 3;
```

Variables con JS

Consideraciones

- No es necesario declarar una variable de tipo `var` con su palabra prefijo, pero es una buena práctica.
- Si es de tipo `let` siempre ha de incluir la palabra prefijo.
- No es necesario especificar el tipo de variable.
- Redefinir una variable de tipo `var` no implica que pierda su valor.
- Una variable de tipo `let` no puede ser redefinida
- Podemos especificar el tipo de variable:

```
let nombre_gato      = new String;  
let vidas_restantes = new Number;  
let tiene_pedigri    = new Boolean;  
let hijos_gato       = new Array;  
let gato             = new Object
```

Entrada y salida de datos en JavaScript

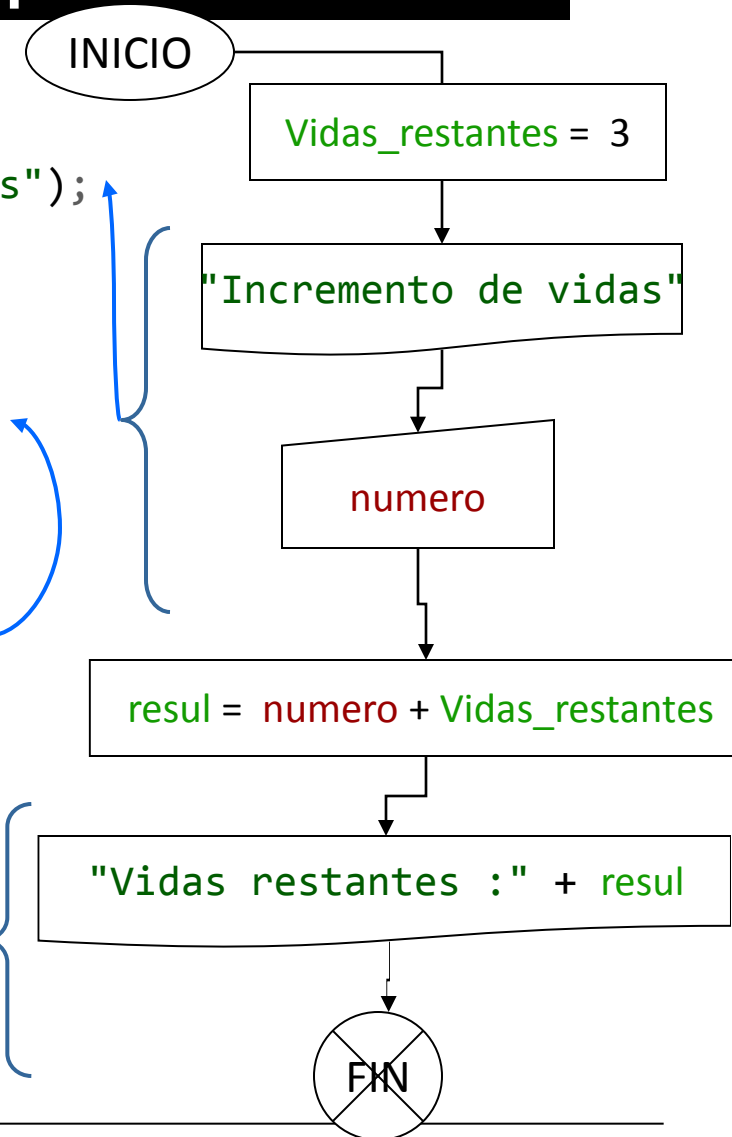
Entrada de datos

```
let numero= window.prompt("Incremento de vidas");
```

Salida de datos

```
window.alert("El numero introduit es"+numero);
```

```
let vidas_restantes =3;  
  
let numero= window.prompt("incremento de vidas");  
  
let resul = numero+vidas_restantes;  
  
alert("Vides restantes:"+resul);  
console.log("Vides restantes:"+resul);
```



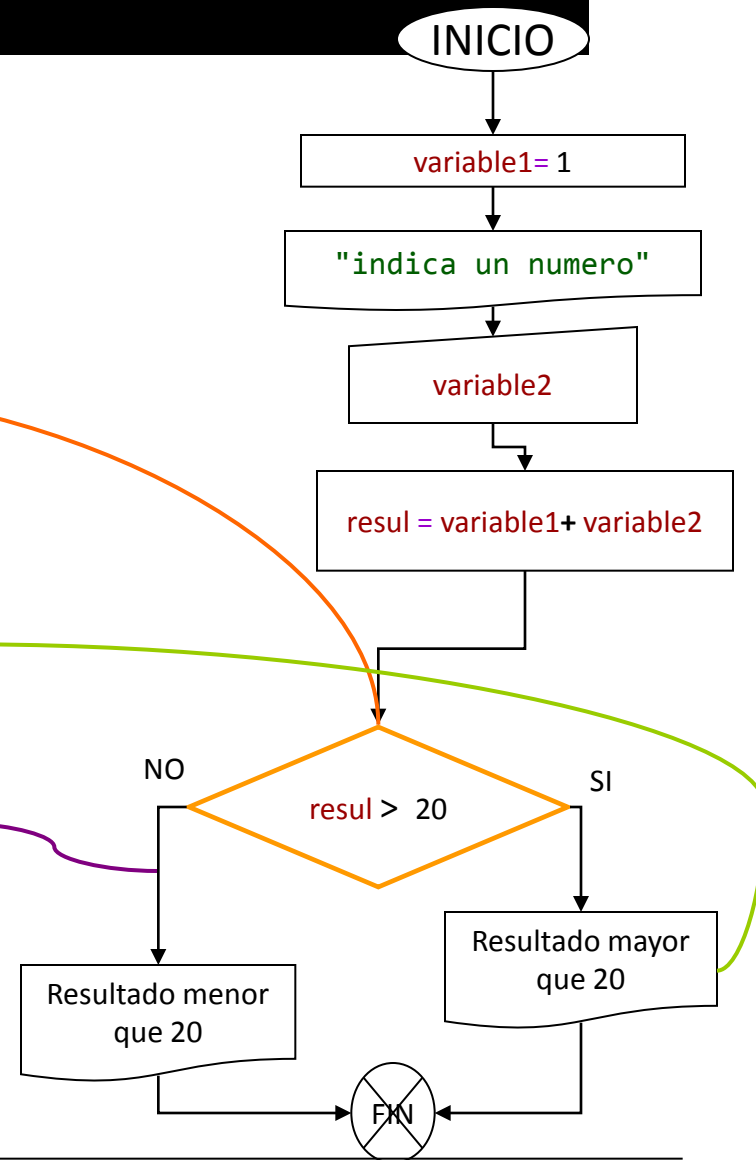
Estructuras de control

```
let variable1 =10;
let variable2=5;

variable2 = window.prompt("indica un numero");

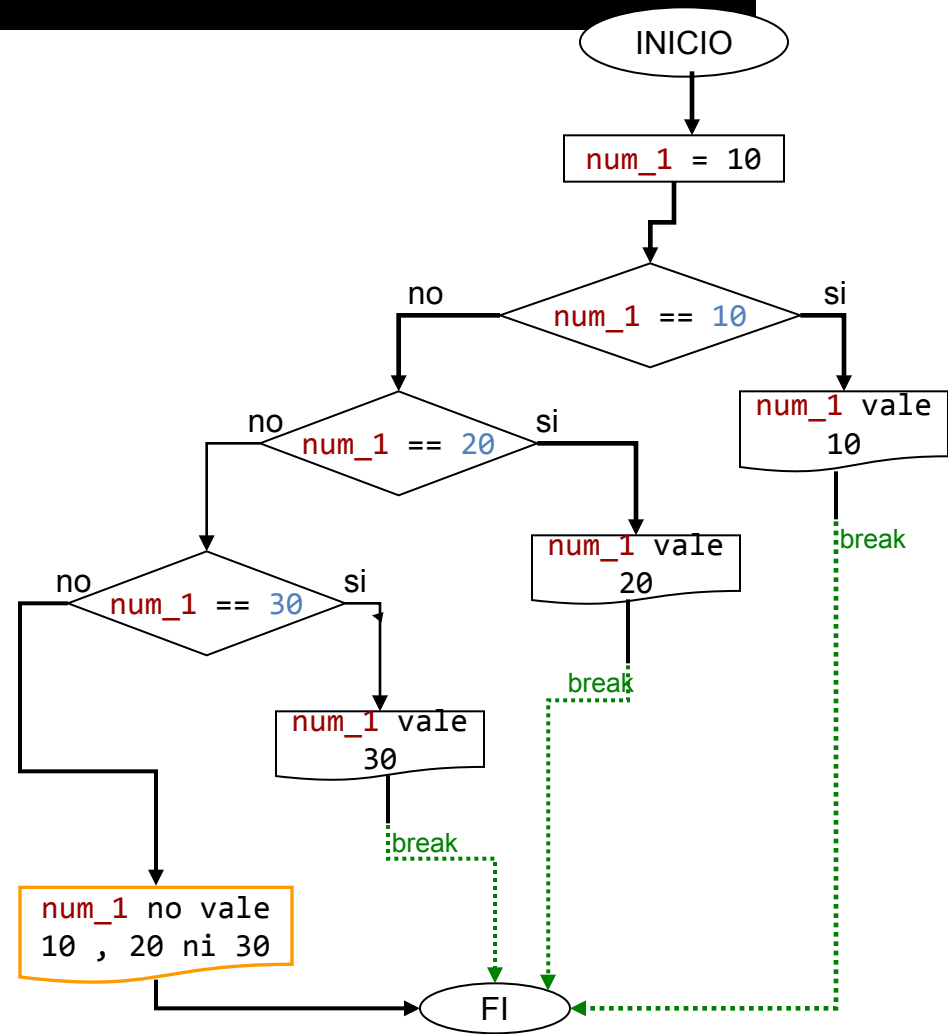
let resul = variable1+ variable2;

if( resul >= 20 ){
    alert("Resultado mayor o igual a 20");
} else{
    alert("Resultado menor que 20");
}
```



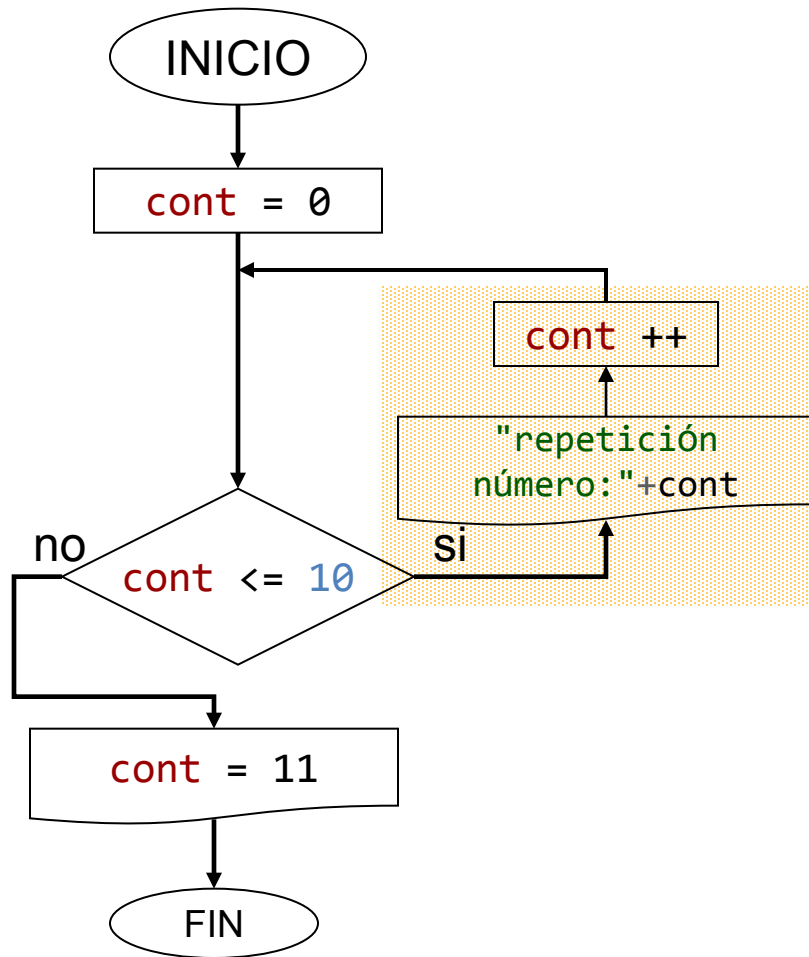
Estructuras de control

```
num_1=10;  
switch( num_1 ){ //valor a comparar  
case 10:  
    alert(num_1+" vale 10");  
    break;  
case 20:  
    alert(num_1+" vale 20");  
    break;  
case 30:  
    alert(num_1+" vale 30");  
    break;  
default:  
    alert(num_1+" no vale 10,20 ni 30");  
}
```



• Sin el **break**, se ejecutan las acciones del siguiente **case**.

Estructuras de repetición While y For



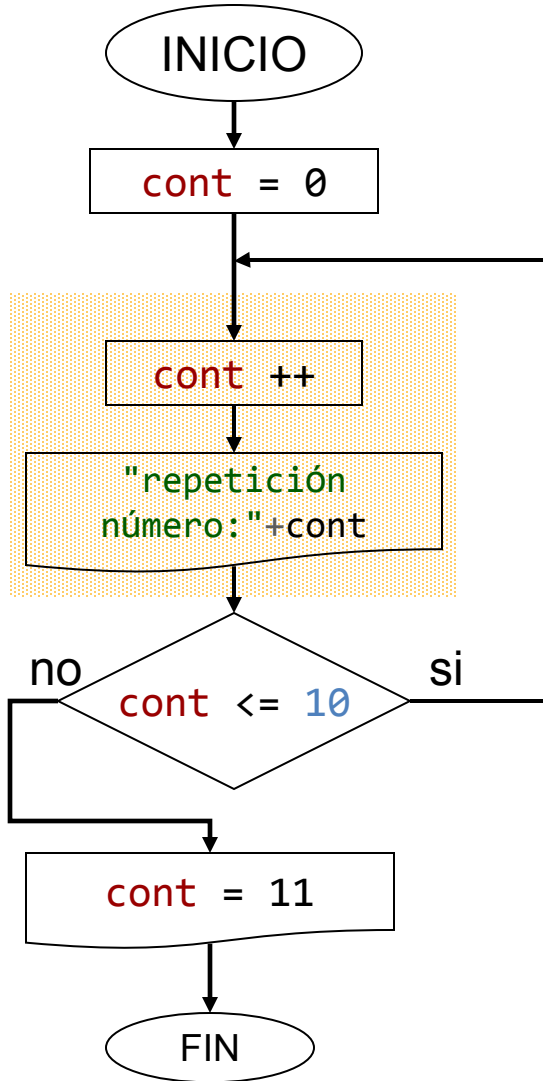
While

```
let cont = 0;
while(cont <= 10){
  console.log("repetición número:"+cont);
  cont++;
}
console.log("Valor final:"+cont);
```

For

```
for(let cont=0; cont <= 10; cont++){
  console.log("repetición número:"+cont);
}
```

Estructuras de repetición Do While



Do While

```
let cont=0;  
do{  
    console.log("repetición número:"+cont);  
    cont++;  
}while(cont <=10);  
console.log("Valor final:"+cont);
```

Estructuras de salto Continue y Break

```
let i=0;
while (i<12){

    document.write("<br /> Iteración número:"+i);
    i=++i;
    if(i * i > 6){
        if(i==3){
            continue;
        }
        document.write("<br />" + i + "Iteración finalizada");
        break;
    }
}
```


Operadores

Aritméticos

- suma (+)
- resta (-)
- resto de la división (%)
- multiplicación (*)
- división (/)

x es el resultado de sumar
5 mas el valor de x

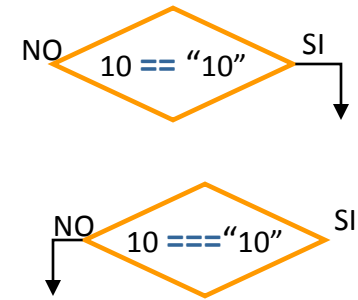
x +=5

x resultado de sumar
el valor de x mas 1

x++

De Comparación

- $x == y$ Retorna si x tiene el mismo valor que y . ($5=="5"$ && $5==5$)
- $x === y$ Retorna si x es igual a y . ($5===5$)
- $x != y$ Retorna si x tiene un valor diferente a y
- $x !== y$ Retorna si x es completamente diferente a y
- $>, <, >=, <=$ Más gran,de más pequeño, más grande o igual, más pequeño o igual.



Lógicos

- AND (&&) Retorna SI , si las dos comparaciones son ciertas.
- OR (||) Retorna SI , si alguna de las dos comparacionse es cierta.
- NEGACION (!) Si el resultado es SI, retorna NO y viceversa.