

Ejemplo 1 de Array

```
<body>
  <script>
    // Array definido 1 a 1

    let array1=[]; // let array-1=new Array(); es equivalente
    array1[0]=22;
    array1[1]="Paquito";
    array1[2]=44;

    //Array definido en una línea inicializando elementos

    let myArray=[1,2,"Cancamusa"]; // Valores dentro del array

    alert("Primer elemento del array myArray[0,1,2] myArray[0]:"+ " "+myArray[0]);
    alert("Segundo elemento del array myArray[0,1,2] myArray[1]:"+ " "+myArray[1]);
    alert("Tercer elemento del array myArray[0,1,2] myArray[2]:"+ " "+myArray[2]);
    alert("Elementos del array myArray=[]: "+myArray +"\\n"+ " Longitud del array:"+ " "+myArray.length); // array y
    tamaño
    alert("FIN")

  </script>
</body>
```

Ejemplo 2 de Array

```
<body>
  <script t>
    // Array que almacena los 12 meses del año

    var meses = ["Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre",
    "Diciembre"];

    // Se muestra el nombre de cada mes

    alert(meses[0]);
    alert(meses[1]);
    alert(meses[2]);
    alert(meses[3]);
    alert(meses[4]);
    alert(meses[5]);
    alert(meses[6]);
    alert(meses[7]);
    alert(meses[8]);
    alert(meses[9]);
    alert(meses[10]);
    alert(meses[11]);

  </script>
</body>
```

LOS CONDICIONALES

Se utilizan para ejecutar cierto bloque de código si se cumple una condición especificada. Los condicionales más comunes son **if**, **else if** y **else**.

- 1. if:** Se utiliza para ejecutar un bloque de código si una condición es verdadera. Si la condición no es verdadera, el bloque de código no se ejecuta.
- 2. else if:** Se utiliza para especificar una nueva condición si la primera condición no se cumple

EJEMPLO 1

```
<script>
  let edad = 18;

  if (edad >= 18) {
    document.write("Eres mayor de edad.");
  }
</script>
```

En este ejemplo, si la variable edad es mayor o igual a 18, se imprimirá "Eres mayor de edad."

EJEMPLO 2

```
<script>
  let hora = 14;

  if (hora < 12) {
    document.write("Buenos días.");
  }
  else if (hora < 18) {
    document.write("Buenas tardes.");
  }
  else {
    document.write("Buenas noches.");
  }
</script>
```

En este ejemplo, se evalúa la variable hora. Si es menor que 12, se imprime "Buenos días." Si no, pero es menor que 18, se imprime "Buenas tardes." Si ninguna de las condiciones anteriores se cumple, se imprime "Buenas noches."

EJEMPLO 3

```
<script type="text/javascript">
var numero1 = 5;
var numero2 = 8;

if(numero1 <= numero2) {
  alert("numero1 no es mayor que numero2");
}
if(numero2 >= 0) {
  alert("numero2 es positivo");
}
if(numero1 < 0 || numero1 != 0) {
  alert("numero1 es negativo o distinto de cero");
}
if(++numero1 < numero2) {
  alert("Incrementar en 1 unidad el valor de numero1 no lo hace mayor o igual que numero2");
}
</script>
```

EJEMPLO 4

```
<script>
let edadAna,edadLuis;
// Convertimos a entero las cadenas
edadAna=parseInt(prompt("Introduce la edad de Ana",""));
edadLuis=parseInt(prompt("Introduce la edad de Luis",""));

if (edadAna > edadLuis){
  alert("Ana es mayor que Luis.");
}
else {
  if (edadAna<edadLuis){
    alert("Ana es menor que Luis.");
  }
  else {
    alert("Ana tiene la misma edad que Luis.");
  }
}
alert(" Ana tiene "+edadAna+" años y Luis "+ edadLuis);
</script>
```

EJEMPLO 5

```
<script>
  var numero1 = 5;
  var numero2 = 8;

  if(numero1 <= numero2) {
    alert("numero1 no es mayor que numero2");
  }
  if(numero2 >= 0) {
    alert("numero2 es positivo");
  }
  if(numero1 < 0 || numero1 != 0) {
    alert("numero1 es negativo o distinto de cero");
  }
  if(++numero1 < numero2) {
    alert("Incrementar en 1 unidad el valor de numero1 no lo hace mayor o igual que numero2");
  }
</script>
```

EJEMPLO 6: En este Ejemplo usamos el DOM con **innerHTML**

```
<style>
  .demo {background-color: coral; padding: 16px;}
</style>
<body>
  <h1>Sentencias condicionales</h1>
  <div id="demo"></div>
  <script>
    let edadAna,edadLuis;
    // Convertimos a entero las cadenas
    edadAna=parseInt(prompt("Introduce la edad de Ana",""));
    edadLuis=parseInt(prompt("Introduce la edad de Luis",""));
    if (edadAna > edadLuis){
      document.getElementById ("demo").innerHTML =
        " Ana es mayor que Luis: &nbsp; &nbsp;Ana tiene "+ edadAna + " años y Luis " + edadLuis;
    }
    if (edadAna < edadLuis){
      document.getElementById("demo").innerHTML =
        " Ana es menor que Luis: &nbsp; &nbsp;Ana tiene "+ edadAna + " años y Luis " + edadLuis;
    }
    if (edadAna = edadLuis){
      document.getElementById("demo").innerHTML =
        " Ana tiene la misma edad que Luis: &nbsp; &nbsp;Ana tiene "+ edadAna + " años y Luis " + edadLuis;
    }
    else {
      document.getElementById("demo").innerHTML =
        " Error no has introducido valores numéricos enteros";
    }
  </script>
</body>
</html>
```

Confirm

Es una función en JavaScript que muestra un cuadro de diálogo al usuario con un mensaje y dos botones: "Aceptar" y "Cancelar". La función devuelve true si el usuario hace clic en "Aceptar" y false si hace clic en "Cancelar".

Aquí tienes ejemplos prácticos:

EJEMPLO 1:

```
</script>
  let respuesta = confirm("¿Estás seguro de que quieres eliminar este elemento?");
  if (respuesta === true) {
    alert("Elemento eliminado.");
  } else {
    alert("Eliminación cancelada.");
  }
  alert("FIN");
</script>
```

En este ejemplo, cuando se ejecuta el código, aparecerá un cuadro de diálogo con el mensaje "¿Estás seguro de que quieres eliminar este elemento?"

- Si el usuario hace clic en "Aceptar", la variable respuesta será true, por lo que se imprimirá "Elemento eliminado."
- Si el usuario hace clic en "Cancelar", la variable respuesta será false, por lo que se imprimirá "Eliminación cancelada."

EJEMPLO 2: Cuando el usuario pulsa el botón Aceptar, este método devuelve el valor true; Cancelar devuelve el valor false. Con ayuda de este método el usuario puede decidir sobre preguntas concretas e influir de ese modo directamente en la página:

```
</script>

<script>
let respuesta;
respuesta=confirm ("¿Deseas eliminar a este tolete?");
alert("Seguro que Sí!!"+"\\n"+respuesta);
alert("Bay Bay !!"+"\\n"+"Te vas a arrepentir");
document.write("Seguro que Sí!!"+"\\n"+respuesta+"\\n"+"Bay Bay !!"+"\\n"+"Te vas a arrepentir")
</script>

</script>
```

El bucle for

Se utiliza para repetir un bloque de código un número específico de veces. Se compone de tres partes: la inicialización, la condición de continuación y el incremento o decremento. Aquí está el resumen:

for (inicialización; condición; incremento/decremento): Este bucle ejecuta un bloque de código mientras se cumpla una condición especificada.

EJEMPLO 1

El siguiente es un ejemplo práctico de un bucle for en JavaScript:

```
for (let i = 0; i < 5; i++) {
  alert("El valor de i es: " + i);
}
```

En este ejemplo:

- La inicialización es **let i = 0;**, donde i se establece en 0 al comienzo del bucle.
- La condición es **i < 5;**, donde el bucle se ejecutará mientras **i sea menor que 5.**
- El incremento es **i++**, lo que significa que i se incrementará en 1 en cada iteración del bucle.

El valor de i es: 0
.....
El valor de i es: 4

EJEMPLO 3

```
<script>

  alert("Escribe las potencias de 2 hasta 300");
  let aux=1;
  for (i=2;i<=300;i*=2) {
    alert("2 elevado a "+aux+" es igual a "+i);
    aux++;
  }
  alert("Se han escrito las potencias de 2 menores de 300");

</script>
```

EJEMPLO 2

```
<script>

  alert("Se van a escribir los números pares del 2 al 10");
  for (i=2;i<=10;i+=2) {
    alert(i);
  }

</script>
```

El bucle for es útil cuando sabes exactamente cuántas veces quieres que se ejecute un bloque de código.

EJEMPLO 4

```
<script>

  alert("Ejemplo: Calcular los números pares del 0 al 20 excepto los que sean múltiplos de 3");
  let i;
  let j=0;
  let k=0;
  let l=0;
  let m=0;
  for (i=2;i<=20;i+=2){
    j+=1;
    m=j-k;
    if ((i%3)===0){k+=1; continue;}
    alert("Elemto par no múltiplo de 3 nº"+ " " +m+" "
    +"es:"+ " " +i)
  }
  l=j-k
  alert("Total de elementos pares no múltiplos de 3, de 20 números, son: "+ " " + l)
  alert("Total de elementos pares múltiplos de 3, de 20 números, son:"+ " " + k)
  alert ("FIN")

</script>
```

LA FUNCIÓN prompt()

Se utiliza para mostrar un cuadro de diálogo al usuario que le **permite ingresar datos**. Esta función devuelve el valor ingresado por el usuario como una cadena de texto.

Aquí está el resumen:

Prompt : El método prompt() abre un cuadro de diálogo en pantalla en el que se pide al usuario que introduzca algún dato. Si se pulsa el botón Cancelar, el valor de devolución es false/null.

Pulsando en Aceptar se obtiene el valor true y la cadena de caracteres introducida se guarda para su posterior procesamiento.

EJEMPLO 1

```
let nombre = prompt("Por favor, introduce tu nombre:",
"Usuario");

if (nombre !== null) {
  alert("¡Hola, " + nombre + "! Bienvenido.");
} else {
  alert("Has cancelado la operación.");
}
```

En este ejemplo:

- La función prompt() muestra un cuadro de diálogo al usuario con el mensaje "Por favor, introduce tu nombre:" y un campo de entrada donde el usuario puede escribir su nombre. El segundo parámetro "Usuario" es el valor predeterminado que aparecerá en el campo de entrada.
- El valor ingresado por el usuario se almacena en la variable nombre.
- Se verifica si el usuario ha ingresado un valor (nombre !== null). Si el usuario hace clic en "Aceptar" sin ingresar nada, nombre será una cadena vacía (""). Si el usuario hace clic en "Cancelar", nombre será null.
- Dependiendo de la respuesta del usuario, se imprime un mensaje de bienvenida o un mensaje indicando que la operación ha sido cancelada.

Este es un ejemplo básico de cómo se puede utilizar la función prompt() para interactuar con el usuario y obtener información ingresada por él.

EJEMPLO 4

// ver si un número introducido es par:

```
<script type="text/javascript">
```

```
var numero = prompt("Introduce un número entero");
```

```
var resultado = parImpar(numero);
```

```
alert("El número "+numero+" es "+resultado);
```

```
function parImpar(numero) {
  if(numero % 2 == 0) {
    return "par";
  }
  else {
    return "impar";
  }
}
```

EJEMPLO 2

```
<script>
```

```
let provincia;
```

```
provincia=prompt("Introduzca la provincia ","");
```

```
provincia=confirm("Estás de acuerdo ?")
```

```
alert("Usted ha introducido la siguiente información
"+provincia);
```

```
alert("Operación realizada con éxito");
```

```
</script>
```

EJEMPLO 3

```
<html >
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
```

```
<title>Ejercicio 8 - Funciones básicas</title>
```

```
<script>
```

```
var numero = prompt("Introduce un número entero");
```

```
var resultado = parImpar(numero);
```

```
alert("El número "+numero+" es "+resultado);
```

```
function parImpar(numero) {
```

```
  if(numero % 2 == 0) {
```

```
    return "par";
```

```
  }
```

```
  else {
```

```
    return "impar";
```

```
  }
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

</script>

EL ELEMENTO BREAK

Se utiliza en JavaScript principalmente dentro de bucles (for, while, do-while, etc.) y también dentro de declaraciones switch.

Cuando se encuentra dentro de un bucle, break se usa para salir del bucle inmediatamente, incluso si la condición del bucle aún es verdadera. Esto permite interrumpir la ejecución del bucle antes de que se cumpla la condición de salida natural.

EJEMPLO 1

```
<script>
  for (let i = 0; i < 5; i++) {
    if (i === 3) {
      break; // Sale del bucle cuando i es igual a 3
    }
    document.write(i);
  }
</script>
```

En este ejemplo, el bucle se ejecutará hasta que i sea igual a 3. En ese punto, la declaración break se activa y el bucle se detiene de inmediato, incluso si la condición (i < 5) aún es verdadera.

El resultado de este código sería: **0 1 2**

Después de que i es igual a **3**, **el bucle se detiene** y la ejecución del **programa continúa con la línea de código después del bucle.**

EJEMPLO 2

```
<script>
  // Ejemplo: Pregunta por la clave y permitir tres
  // respuestas incorrectas
  let auxclave=true;
  let numveces=0;

  //Mientras no introduzca la clave y no se pulse
  //Cancelar
  while (auxclave !== "anonimo" && auxclave) {
    auxclave=prompt("Introduce la clave ", "");
    numveces++;
    if (numveces === 3 || auxclave=="HolaCaraCola")
      break;
  }
  if (auxclave=="HolaCaraCola"){
    alert("La clave es correcta");
  }
  else{
    alert("Has excedido los tres intentos: La clave NO es
    correcta");
    alert ("La clave es: HolaCaraCola");
  }
</script>
```

LA DECLARACIÓN continue

Se utiliza dentro de bucles para saltar la iteración actual y continuar con la siguiente iteración del bucle. Básicamente, cuando se encuentra la declaración continue, el bucle salta al siguiente paso sin ejecutar el resto del código dentro del bloque del bucle para esa iteración en particular.

El efecto que tiene la instrucción "continue" en un bucle es el de hacer retornar a la secuencia de ejecución a la cabecera del bucle, volviendo a ejecutar la condición o a incrementar los índices cuando sea un bucle for. Esto permite saltarse recorridos del bucle.

EJEMPLO 1

Ejemplo de continue en un bucle for para imprimir solo los números impares del 1 al 5:

```
<script>
  for (let i = 1; i <= 5; i++) {
    if (i % 2 === 0) {
      continue; // Salta la iteración actual si el número es par
    }
    alert(i);
  }
</script>
```

En este ejemplo:

- La variable i recorre los números del 1 al 5 en el bucle for.
- La condición if (i % 2 === 0) verifica si i es un número par. Si es par, se encuentra la declaración continue, que salta la iteración actual.
- Si i es impar, se imprimirá su valor en la consola.

El resultado de este código sería imprimir solo los números impares del 1 al 5: **1 2 3 4 5**

EJEMPLO 2

```
<script>
```

```
  alert("Ejemplo: Calcular los números pares del 0 al 20 excepto los que sean múltiplos de 3");
```

```
  let i;
  let j=0;
  let k=0;
  let l=0;
  let m=0;
  for (i=2;i<=20;i+=2){
    j+=1;
    m=j-k;
    if ((i%3)===0){k+=1; continue;}
    alert("Elemto par no múltiplo de 3 n°"+ " " +m+" " +"es:"+ " " +i)
  }
  l=j-k
  alert("Total de elementos pares no múltiplos de 3, de 20 números, son: "+" " + l)
  alert("Total de elementos pares múltiplos de 3, de 20 números, son:"+ " " + k)
  alert ("FIN")
```

```
</script>
```

EL BUCLE while

Se utiliza para repetir un bloque de código mientras una condición especificada sea verdadera. El bloque de código se ejecuta repetidamente mientras la condición sea verdadera, y se detiene cuando la condición se evalúa como falsa.

- **do { ... } while (condición):** Este bucle ejecuta un bloque de código una vez y luego repite la ejecución del bloque mientras la condición especificada sea verdadera.

El bucle while es útil cuando no se sabe exactamente cuántas veces se debe ejecutar el bloque de código, pero se sabe que debe continuar mientras se cumpla una condición específica.

EJEMPLO 1

```
<script>
  let i = 1;
  while (i <= 5) {
    alert(i);
    i++;
  }
</script>
```

En este ejemplo:

- La variable i se inicializa con el valor 1.
- La condición i <= 5 verifica si i es menor o igual a 5. Mientras esta condición sea verdadera, el bucle se ejecutará.
- Dentro del bucle, se imprime el valor de i en la consola.

EJEMPLO 2

Ejemplo : Númros pares del 2 al 10

```
<script>
```

```
  alert("números pares del 2 al 10");
  let i=2;
  while (i<=10) {
    alert(i);
    i+=2;
  }
  alert("Ya se han mostrado los números pares del 2 al 10");
```

```
</script>
```

- Después de imprimir el valor, se incrementa i en 1 usando i++.

El resultado de este código sería imprimir los números del 1 al 5 en la consola: 1 2 3 4 5

EL BUCLE do-while

La estructura básica del ciclo **do-while en JavaScript** es la siguiente:

```
do {  
  // Bloque de código a ejecutar  
}  
while (condición);
```

Aquí, el bloque de código dentro de las llaves {} se ejecutará al menos una vez, independientemente de si la condición es verdadera o falsa. Después de ejecutar el bloque de código, JavaScript evaluará la condición en la declaración while.

Si la condición es verdadera, el bloque de código se ejecutará nuevamente. Este proceso se repetirá hasta que la condición sea falsa.

A continuación, te muestro un ejemplo práctico de cómo se utiliza el bucle do-while para imprimir los números del 1 al 5:

EJEMPLO

```
<script>  
  let i = 1;  
  
  do {  
    console.log(i);  
    i++;  
  } while (i <= 5);  
  
</script>
```

En este ejemplo:

- La variable i se inicializa con el valor 1.
- El bloque de código dentro del do se ejecutará al menos una vez sin verificar la condición.
- Dentro del bucle, se imprime el valor de i en la consola.
- Luego, se incrementa i en 1 usando i++.
- Después de ejecutar el bloque una vez, se verifica la condición i <= 5. Si la condición es verdadera, el bucle se repetirá y se ejecutará nuevamente el bloque.

El resultado de este código sería imprimir los números del 1 al 5 en la consola, al igual que en el ejemplo del bucle while. La diferencia clave es que con el bucle do-while, el bloque de código se ejecutará al menos una vez, incluso si la condición es falsa desde el principio.

EJEMPLO 2 : Preguntar por una clave hasta que se introduzca la correcta

```
<script>  
  alert("Preguntar por una clave hasta que se introduzca la correcta=clave");  
  let auxclave;  
  do {  
    auxclave=prompt("introduce la clave ", "");  
  }  
  while (auxclave!=="clave")  
  alert("Has acertado la clave es igual a: "+auxclave);  
</script>
```

