

# While / do while

DigitalHouse>

“

Existen otras dos estructuras dentro de los bucles o ciclos que nos pueden ayudar a resolver nuestros problemas eficientemente en determinadas situaciones: **while** y **do while**.



”

# Estructura básica while

Tiene una estructura similar a la de los condicionales **if/else**, palabra reservada + condición entre paréntesis. Sin embargo, el **while Loop** reevalúa esa condición repetidas veces y **ejecuta su bloque de código** hasta que la condición **deja de ser verdadera**.

```
while (condicion) {  
    //código que se ejecutará en cada repetición  
}  
    // Hace algo para que la condición eventualmente se deje  
    de cumplir  
}
```

# Estructura básica while

Tomando el ejemplo utilizado con el for, veamos como sería utilizando while.

```
{  
  let vuelta = 1  
  while(vuelta <= 5) {  
    console.log('Dando la vuelta número ' + vuelta);  
    vuelta++ //al final de cada vuelta sumara 1 a vuelta  
  };  
}
```



```
Dando la vuelta número 1  
Dando la vuelta número 2  
Dando la vuelta número 3  
Dando la vuelta número 4  
Dando la vuelta número 5
```

# Estructura básica while

Antes de ejecutar el código en cada vuelta, se pregunta si la condición resulta verdadera o falsa.

Si es **verdadera**, continúa con nuestras instrucciones.

Si es **falsa**, detiene el ciclo.

```
let vuelta = 1
while(vuelta <= 5) {
  console.log('Dando la vuelta número ' + vuelta);
  vuelta++;
};
```

# Estructura básica while

Es importante generar el contador al comenzar para evitar caer en lo que se conoce como **loop infinito**.

```
let vuelta = 1
while(vuelta <= 5) {
  console.log('Dando la vuelta número ' + vuelta);
  vuelta++
};
```

El **loop infinito** sucede cuando nuestra condición es constantemente verdadera, lo que resulta en ejecutar nuestro código eternamente. Esto puede causar varios problemas, siendo el más importante que trabe todo nuestro programa.

“

El ciclo do **while** es similar al **while**, pero se diferencia en que, sin importar la condición, la acción se realizará al menos una vez.

”



## Estructura básica do while

A diferencia del ciclo **while**, la condición en este caso se verifica al finalizar el bloque de código. Por lo tanto, sin importar lo que se resuelva, las acciones se realizarán al menos una vez.

```
{  
  let vuelta = 5  
  do{  
    console.log('Dando la vuelta número ' + vuelta);  
    vuelta++ //Se suma 1 a vuelta por lo tanto vuelta = 6  
  } while(vuelta <= 5); //al vuelta ser 6 la condición retorna  
  false y se termina el bloque de código
```

Fuera de esto, el ciclo **do while** es idéntico en funcionalidad al ciclo **while**.



# Los ciclos **en acción**

## While

En cada iteración se verifica si el valor de **vuelta** es menor o igual a 5. Si es así, se ejecuta el **console.log()** y se incrementa el valor de vuelta en 1.

Cuando vuelta deje de ser menor o igual a 5, se corta el ciclo.







Iteración #	Valor de vuelta	¿Vuelta <= 5 ?	Ejecutamos
1	1	true	✓
2	2	true	✓
3	3	true	✓
4	4	true	✓
5	5	true	✓
6	6	false	✗

# Los ciclos **en acción**

## Do While

Por otro lado, en este caso se ejecuta el bloque de código al comenzar, y **luego** comienza a verificar que el valor de **vuelta** es menor o igual a 5.

Cuando vuelta deje de ser menor o igual a 5, se corta el ciclo.

Iteración #	Valor de vuelta	¿Vuelta <= 5 ?	Ejecutamos
1	1	no se verifica	
2	2	true	
3	3	true	
4	4	true	
5	5	true	
6	6	false	

DigitalHouse>