



El futuro digital  
es de todos

MinTIC

# Formador: Sergio Medina

» **JavaScript**  
**Estructuras de datos y Ciclos**

xxx

## Ciclo No.3 – Semana No.4



Mision  
TIC2022



El futuro digital  
es de todos

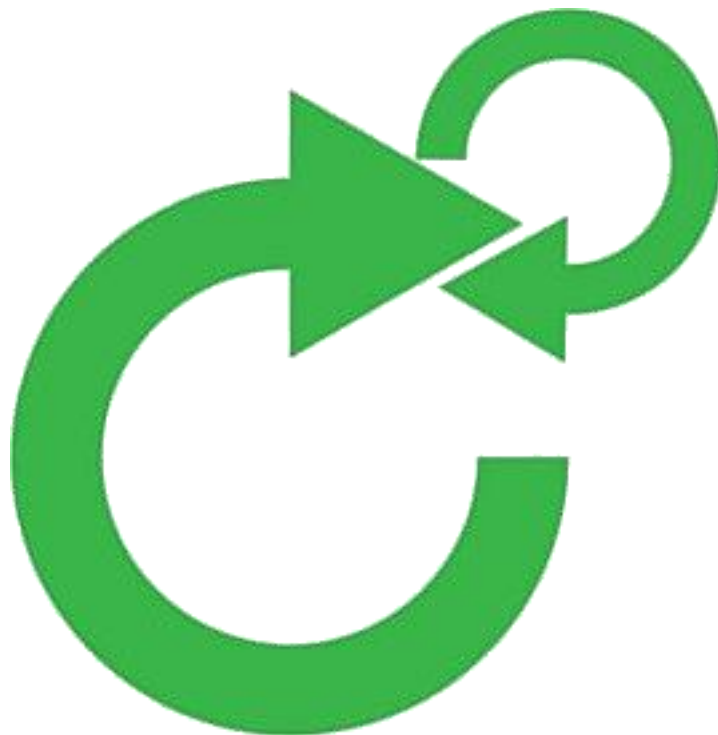
MinTIC



x  
x  
x

# Ciclos

Los bucles o ciclos ofrecen una forma rápida y sencilla de hacer algo repetidamente





# Ciclo for

x  
x  
x

- Un ciclo for se repite hasta que una condición especificada se evalúe como false. El bucle for de JavaScript es similar al bucle for de Java.
- Sintaxis:

**for** ([expresiónInicial]; [expresiónCondicional];  
[expresiónDeActualización])  
instrucción



# Ciclo for

x  
x  
x

- Ejemplo:

```
var n=0;  
for (var i = 0; i < 9; i++) {  
    n += i;  
}  
alert (n);
```



# for...of

x  
x  
x

- La sentencia **for...of** ejecuta un bloque de código para cada elemento de un objeto iterable, como lo son: String, Array.
- Sintaxis:

```
for (variable of iterable) {  
    statement  
}
```





## for...of

- Ejemplo:

```
let cadena = "UNAB";
```

```
for (let valor of cadena) {  
  alert(valor);  
}
```

```
// "U"
```

```
// "N"
```

```
// "A"
```

```
// "B"
```



# Ciclo do...while

x  
x  
x

- La instrucción **do...while (hacer mientras)** se repite hasta que una condición especificada se evalúe como falsa. La condición se evalúa después de ejecutar la sentencia, dando como resultado que la sentencia especificada se ejecute al menos una vez.
- **Sintaxis:**

**do**  
**expresión**  
**while (condición);**



# Ciclo do...while

x  
x  
x

- Ejemplo:

```
let i = 0;  
do {  
    i += 1;  
    alert(i);  
}  
while (i < 5);
```





# Ciclo while

x  
x  
x

- Ejecuta sus instrucciones siempre que una condición especificada se evalúe como true. Dicha condición es evaluada antes de ejecutar la sentencia
- **Sintaxis:**

**while (condición)**  
**expresión**



# Ciclo while

x  
x  
x

- Ejemplo:

```
n = 0;  
x = 0;  
while (n < 3) {  
    n ++;  
    x += n;  
}
```



# Ciclo while



- Evita los bucles infinitos. Debemos asegurarnos que la condición en un bucle eventualmente se convierta en false; de lo contrario, el bucle nunca terminará.
- **Sintaxis:**

```
let a=2;  
while (a<5) {  
  alert(a);  
}
```



# Uso de break



- Se usa la instrucción **break** para terminar un bucle

Sintaxis:

```
for (let i = 0; i < 6; i++) {  
  if (a === 3) {  
    break;  
  }  
}
```



# Uso de break

x  
x  
x

Sintaxis:

```
let n = 0;
```

```
let x = 0;
```

```
while (n<3) {
```

```
    n++;
```

```
    x+= 2;
```

```
    alert(n);
```

```
    if (n===2){
```

```
        break;
```

```
    }
```

```
}
```



## ARREGLOS - ARRAYS



- Un arreglo es un conjunto de datos o una estructura de datos homogéneos que se encuentran ubicados en forma consecutiva en la memoria RAM (sirve para almacenar datos en forma temporal).
- Unidimensionales
- Multidimensionales





# ARREGLOS – Unidimensionales - Vectores

- Es un tipo de datos estructurado que está formado de una colección finita y ordenada de datos del mismo tipo.

Nombre del Arreglo (Obsérvese que todos los elementos de este arreglo tienen el mismo nombre, C)

Índice ( o subíndice) del elemento en el arreglo C



C[0]	C[1]	C[2]	C[3]	C[4]	C[5]	C[6]	C[7]
------	------	------	------	------	------	------	------

12	3	27	16	4	51	8	40
----	---	----	----	---	----	---	----



# ARREGLOS - Multidimensionales

- Es un tipo de datos estructurado que está formado por mas de una colección finita y ordenada de datos del mismo tipo.

columns

filas

10	-3	4
6	7	-2
14	48	-33

ARREGLO BIDIMENSIONAL  
(MATRIZ)



ARREGLO TRIDIMENSIONAL



## ARREGLOS - ARRAYS



Una array o arreglo es una colección de variables del mismo tipo, a la que se hace referencia por un nombre común. En Java, los arrays pueden tener una o más dimensiones, aunque el array unidimensional es el más común.



## Crear un Array



```
let frutas = ["Manzana", "Banana"]
```

```
alert(frutas.length)
```

```
// 2
```



## ARREGLOS - ARRAYS



Acceder a un elemento de Array mediante su índice

let primero = frutas[0] // Manzana

let ultimo = frutas[frutas.length - 1] // Banana



# ARREGLOS - ARRAYS



## Recorrer un Array



```
frutas.forEach(function(elemento, indice,  
array) {  
    alert(elemento, indice);  
})  
// Manzana 0  
// Banana 1
```





## ARREGLOS - ARRAYS



Añadir un elemento al principio de un Array



```
let nuevaLongitud = frutas.unshift('Fresa')  
// Añade "Fresa" al inicio  
// ["Fresa", "Manzana", "Banana"]
```



## ARREGLOS - ARRAYS



Eliminar el primer elemento de un Array



```
let primero = frutas.shift()  
// Elimina "Fresa" del inicio  
// ["Manzana", "Banana"]
```



## ARREGLOS - ARRAYS



Encontrar el índice de un elemento del Array

```
frutas.push('Fresa')  
// ["Manzana", "Banana", "Fresa"]
```

```
let pos = frutas.indexOf('Banana')  
// 1
```



# ARREGLOS - ARRAYS



## Copiar un Array



```
let copiaArray = vegetales.slice();
```



El futuro digital  
es de todos

MinTIC

**Formador:  
Sergio Medina**

» **JavaScript**  
**Estructuras de datos y Ciclos**

xxx

**Ciclo No.3 – Semana No.4**



**Misión  
TIC2022**