



Estructuras de repetición - For



Consigna:

- Programar una función que muestre los primeros 10 números naturales por la consola.

- A tener en cuenta:

$$\mathbb{N} = \{0, 1, 2, 3, 4, 5, \dots, n\}$$

(Tomemos al cero como un natural)

Consigna:

```
function imprimeNaturales() {  
    console.log(0);  
    console.log(1);  
    console.log(2);  
    console.log(3);  
    console.log(4);  
    console.log(5);  
    console.log(6);  
    console.log(7);  
    console.log(8);  
    console.log(9);  
    console.log(10);  
}
```

Una forma muy estúpida de resolverlo.

Posible resolución

```
function imprimeNaturales()
{
    let numero = 0;
    console.log(numero++);
    console.log(numero++);
    console.log(numero++);
    console.log(numero++);
    console.log(numero++);
    console.log(numero++);
    console.log(numero++);
    console.log(numero++);
    console.log(numero++);
    console.log(numero++);
    console.log(numero++);
}
```

Otra forma estúpida de resolverlo



Estructuras de Repetición

- Cuando tenemos un bloque de código (o una sola línea) que la queremos repetir varias veces, podemos utilizar las estructuras de repetición.
- For
- While
- Do – While

For

- Con for creamos **bucles**.
- El cabezal del for está compuesto por tres expresiones opcionales, encerradas en paréntesis y separadas por puntos y comas, seguidas de una sentencia ejecutada en un bucle.
- **for** ([expresion_inicial]; [condición]; [expresion_final])
 {
 // bloque del for
 }

Ejemplo de For

```
for (let i = 0; i < 5; i++)  
{  
  console.log(i);  
}
```

¡¡Probemos este código en la consola!!

Ejemplo de For clásico (usando un contador)

- **for** ([expresion_inicial]; [condición]; [expresion_final])
- Expresión inicial: Podemos declarar una variable numérica, y asignarle un valor de inicio.
- Condición: Es la condición (expresión booleana) que debe cumplir la expresión inicial para que el bucle se siga ejecutando.
- Expresión final: Expresión para ser evaluada al final de cada iteración del bucle. Esto ocurre antes de la siguiente evaluación de la condición. Generalmente se usa para actualizar o incrementar la variable contador.



Palabras técnicas...

- A cada **repetición** que realice nuestro bucle se le suele llamar **iteración**.
- En caso que usemos un for que contenga una variable numérica en sus expresiones, a dicha variable se le suele llamar **contador** del for.

Otro ejemplo (similar al anterior...)

```
for (let i = 1; i <= 5; i++)  
{  
  console.log(i);  
}
```

Tiene el mismo número de repeticiones que el for anterior,
sólo que los valores del contador van del 1 al 5, en lugar del 0 al 4

Volviendo al ejercicio inicial:

- Programar una función que muestre los primeros 10 números naturales por la consola. $\mathbb{N} = \{0, 1, 2, 3, 4, 5, \dots, n\}$

```
function imprimeNaturales() {  
  for (let i = 0; i <= 10; i++) {  
    console.log(i);  
  }  
}
```



Ejercicio:

- Realizar el mismo ejercicio que acabamos de hacer (Mostrar los primeros 10 naturales)
- Pero usado un **for** con un cabezal distinto.

Una posible solución:

```
// Una forma:
```

```
function imprimeNaturales() {  
  for (let i = 0; i < 11; i++) {  
    console.log(i);  
  }  
}
```

```
// Otra forma distinta:
```

```
function mostrarNaturales() {  
  for (let i = 1; i <= 11; i++) {  
    console.log(i - 1);  
  }  
}
```

```
script>
```



Próxima clase:

- Haremos ejercicios
- Veremos otra estructura de repetición: While