

Ciclos en Java



Ciclos en Java

Los ciclos (o bucles) en Java se utilizan para ejecutar repetidamente un bloque de código mientras se cumpla una condición específica. Java proporciona varios tipos de ciclos: `for`, `while` y `do-while`. En esta lección, exploraremos cada uno de estos ciclos con su sintaxis, ejemplos, ejercicios y aplicaciones del mundo real.

1. Ciclo `while`

Sintaxis

```
while (condicion) {  
    // Bloque de código a ejecutar  
}
```

- **condición:** Evalúa si el ciclo debe continuar o no antes de cada iteración.

Ejemplo: Imprimir los números del 1 al 10

```
public class CicloWhileEjemplo {  
    public static void main(String[] args) {  
        int i = 1;  
        while (i <= 10) {  
            System.out.println(i);  
            i++;  
        }  
    }  
}
```

```
}  
}
```

Ejercicio: Imprimir los números impares del 1 al 20

```
public class NumerosImpares {  
    public static void main(String[] args) {  
        int i = 1;  
        while (i <= 20) {  
            if (i % 2 != 0) {  
                System.out.println(i);  
            }  
            i++;  
        }  
    }  
}
```

2. Ciclo do-while

Sintaxis

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

- **condición:** Evalúa si el ciclo debe continuar o no después de cada iteración. El bloque de código se ejecuta al menos una vez.

Ejemplo: Imprimir los números del 1 al 10

```
public class CicloDoWhileEjemplo {  
    public static void main(String[] args) {  
        int i = 1;  
        do {  
            System.out.println(i);  
            i++;  
        } while (i <= 10);  
    }  
}
```

Ejercicio 3: Imprimir los números del 1 al 10 en orden inverso

```
public class NumerosInversos {  
    public static void main(String[] args) {  
        int i = 10;  
        do {  
            System.out.println(i);  
            i--;  
        } while (i > 0);  
    }  
}
```

3. Ciclo for

Sintaxis

```
for (inicializacion; condicion; actualizacion) {  
    // Bloque de código a ejecutar  
}
```

- **inicialización:** Se ejecuta una vez al principio del ciclo y se utiliza para inicializar la variable de control.
- **condición:** Evalúa si el ciclo debe continuar o no.
- **actualización:** Se ejecuta al final de cada iteración del ciclo y se utiliza para actualizar la variable de control.

Ejemplo: Imprimir los números del 1 al 10

```
public class CicloForEjemplo {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 10; i++) {  
            System.out.println(i);  
        }  
    }  
}
```

Ejercicio: Imprimir los números pares del 1 al 20

```
public class NumerosPares {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 20; i++) {  
            if (i % 2 == 0) {  
                System.out.println(i);  
            }  
        }  
    }  
}
```

Ejercicio Final

Ejercicio: Crear un menú interactivo

Crea un programa que muestre un menú interactivo utilizando un ciclo `do-while`. El menú debe permitir al usuario elegir entre diferentes opciones y realizar acciones según la elección. Por ejemplo, mostrar un mensaje, realizar una operación matemática, o salir del menú.

```
import java.util.Scanner;  
  
public class MenuInteractivo {  
    public static void main(String[] args) {  
        var consola = new Scanner(System.in);  
        var salir = false;  
  
        do {
```

```
System.out.println("Menú:");
System.out.println("1. Mostrar mensaje");
System.out.println("2. Sumar dos números");
System.out.println("3. Salir");
System.out.print("Elige una opción: ");
var opcion = consola.nextInt();

switch (opcion) {
    case 1:
        System.out.println(";Hola, bienvenido al menú interactivo!");
        break;
    case 2:
        System.out.print("Ingresa el primer número: ");
        var num1 = consola.nextInt();
        System.out.print("Ingresa el segundo número: ");
        var num2 = consola.nextInt();
        var suma = num1 + num2;
        System.out.println("La suma es: " + suma);
        break;
    case 3:
        salir = true;
        System.out.println("Saliendo del sistema...");
        break;
    default:
        System.out.println("Opción no válida, intenta de nuevo.");
}

System.out.println();
} while (!salir);
}
```

Resumen

En esta lección, hemos cubierto los ciclos `for`, `while` y `do-while` en Java, proporcionando la sintaxis básica, ejemplos simples, ejercicios y ejemplos del mundo real. Los ciclos son fundamentales en la programación y permiten automatizar tareas repetitivas de manera eficiente.

¡Espero que esta lección te sea muy útil!

Saludos!

Ing. Ubaldo Acosta

Fundador de [GlobalMentoring.com.mx](https://www.globalmentoring.com.mx)