

## 7. Sentencias condicionales (I) .

### 7.1 La sentencia condicional If

La sentencia if permite la ejecución de una serie de instrucciones en función del resultado de una expresión lógica. El resultado de evaluar una expresión lógica es siempre verdadero (true) o falso (false). Es muy simple, en lenguaje natural sería algo como "si esta condición es verdadera entonces haz esto, sino haz esto otro".

El formato de la sentencia if es el siguiente:

```
if (condición) {
```

```
    instrucciones a ejecutar si la condición es verdadera
```

```
} else {
```

```
    instrucciones a ejecutar si la condición es falsa
```

```
}
```

```
public class ComparacionCadena {  
  
    public static void main(String[] args) {  
        String miFruta = "naranja";  
  
        if ("naranja".equals(miFruta)) {  
            System.out.println("iguales");  
        } else {  
            System.out.println("distintas");  
        }  
    }  
}
```

¿Podrías adivinar cuál es el resultado del programa? Efectivamente, se muestra la palabra “iguales” ya que la comparación "naranja".equals(miFruta) devuelve true y, por tanto, se ejecuta el primer bloque de código, o sea, System.out.println("iguales");.

Veamos ahora un ejemplo algo más interesante, en el que el usuario debe responder a la pregunta que le hace el ordenador.

```
import java.util.Scanner;

public class SentenciaIf {
    public static void main(String[] args) {

        String respuesta;
        Scanner s = new Scanner(System.in);

        System.out.print("¿Cuál es la capital de Portugal? ");

        respuesta = s.nextLine();

        if (respuesta.equals("Lisboa")) {
            System.out.println("¡La respuesta es correcta!");
        } else {
            System.out.println("Lo siento, la respuesta es incorrecta.");
        }
    }
}
```

En el programa se le pregunta al usuario cuál es la capital de Portugal. La respuesta introducida por el usuario se almacena en la variable respuesta. A continuación viene la sentencia condicional if.

Llegado a este punto, el programa evalúa la expresión respuesta.equals("Lisboa"). Observa que para comparar dos cadenas de caracteres se utiliza equals().

Vamos a ver otro ejemplo, esta vez con números. El usuario introducirá un número por teclado y el programa dirá si se trata de un número positivo o negativo.

```
String linea;
Scanner s = new Scanner(System.in);
int entero;

System.out.print("Por favor, introduce un número entero: ");

linea = s.nextLine();
entero = Integer.parseInt( linea );

if (entero < 0) {
    System.out.println("El número introducido es negativo.");
} else {
    System.out.println("El número introducido es positivo.");
}
```

## 7.2. Operadores de comparación

Un carácter suelto como una letra o un signo de puntuación se puede almacenar en una variable de tipo char. El carácter debe ir entrecomillado utilizando las comillas simples (').

| OPERADOR | NOMBRE            | EJEMPLO | DESCRIPCIÓN              |
|----------|-------------------|---------|--------------------------|
| ==       | igual             | a == b  | a es igual a b           |
| !=       | distinto          | a != b  | a es distinto de b       |
| <        | menor que         | a < b   | a es menor que b         |
| >        | mayor que         | a > b   | a es mayor que b         |
| <=       | menor o igual que | a <= b  | a es menor o igual que b |
| >=       | mayor o igual que | a >= b  | a es mayor o igual que b |

**Ejercicio 7.1**

Escribe un programa que pida dos números y muestre el mayor de ellos.

**Ejercicio 7.2**

Escribe un programa que pida tres números y muestre el mayor de ellos.

**Ejercicio 7.3**

Realiza un programa que diga si un número introducido por teclado es divisible entre 7.

**Ejercicio 7.4**

Escribe un programa que dada una hora determinada (horas y minutos), calcule los segundos que faltan para llegar a la medianoche.