

2 - Sentencias de selección

Horas de Libre Configuración

Curso 2023- 2024



if



- ▶ No hay carácter especial para delimitar los bloques, simplemente se introduce una sangría (espacios o tabuladores).
- ▶ En Python se recomienda una sangría de 4 caracteres.
- ▶ No hay que encerrar entre paréntesis la condición a evaluar.

if

- Puede ser simple, doble o múltiple.

SIMPLE	DOBLE	MÚLTIPLE
<pre>if edad < 18: menor = True</pre>	<pre>if a < b: menor = a else: menor = b</pre>	<pre>if mes == 1: letra = "E" elif mes == 2: letra = "F" ... else: letra = "X"</pre>

match



- Variante para representar una selección múltiple.

```
match mes:  
    case 1:  
        letra = "E"  
    ...  
    case _:    # caso por defecto
```

match



- ▶ Se pueden combinar casos en uno sólo con el símbolo |.

```
match diasem:  
    case 6 | 7:  
        print("Fin de semana")
```

match



- ▶ También se pueden incluir expresiones para indicar los valores aceptados en el caso.

```
match edad:  
    case n if n < 18:  
        print("Menor de edad")
```

Asignación condicional



- ▶ Permite asignar a una variable un valor en función de una condición.

```
menor = a if a < b else b
```

Cortocircuito en expresiones lógicas

- ▶ Hay veces en la que una condiciones lógica no evalúa por completo todas las condiciones para establecer un valor de verdad.
- ▶ Cuando eso ocurre, se dice que se produce un cortocircuito lógico.

Cortocircuito en expresiones lógicas

- ▶ and: si una condición es falsa, ya lo es toda la expresión.

```
if a >= 0 and a <= 10:
```

- ▶ Si a es un valor negativo, no llega a evaluarse si es menor que 10.

Cortocircuito en expresiones lógicas

- ▶ or: si una condición es cierta, ya lo es toda la expresión.

```
if dia==6 or dia==7:
```

- ▶ Si dia vale 6, ya no llega a evaluarse si es igual a 7.

Cortocircuito en expresiones lógicas



CONSEJO

Haz pruebas a los programas utilizando casos que verifiquen todas las bifurcaciones y condiciones del código.