

EXPRESIONES LÓGICAS. Parte 2

I. Gracia, P. García, A. López
Junio, 2023

Más de una condición

navideño.py

```
día = int(input('Introduce el día del cumpleaños: '))
mes = int(input('Introduce el mes del cumpleaños: '))
# ¿Nació el día de Navidad?
...
```

¿Podemos combinar la comparación del día y la del mes?

not	lo contrario de
and	cierto sólo si las dos partes son ciertas; falso en cualquier otro caso
or	falso sólo si las dos partes son falsas; cierto en cualquier otro caso

Más de una condición

navideño.py

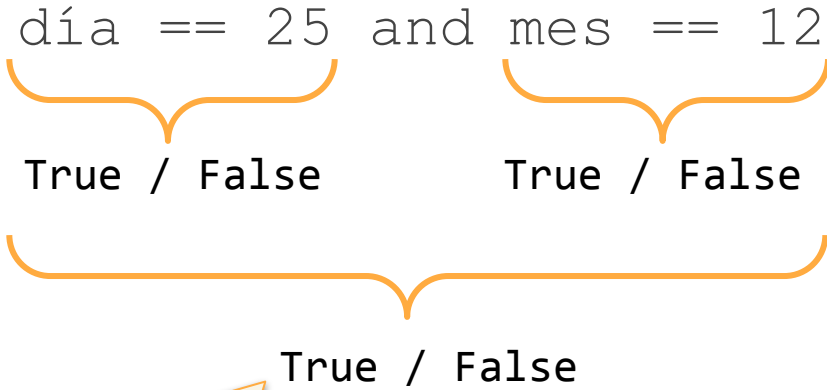
```
día = int(input('Introduce el día del cumpleaños: '))
mes = int(input('Introduce el mes del cumpleaños: '))
# ¿Nació el día de Navidad?
print(f'¿Nació en Navidad? { día == 25 and mes == 12 }');
```

Orden de evaluación

```
día == 25 and mes == 12
```

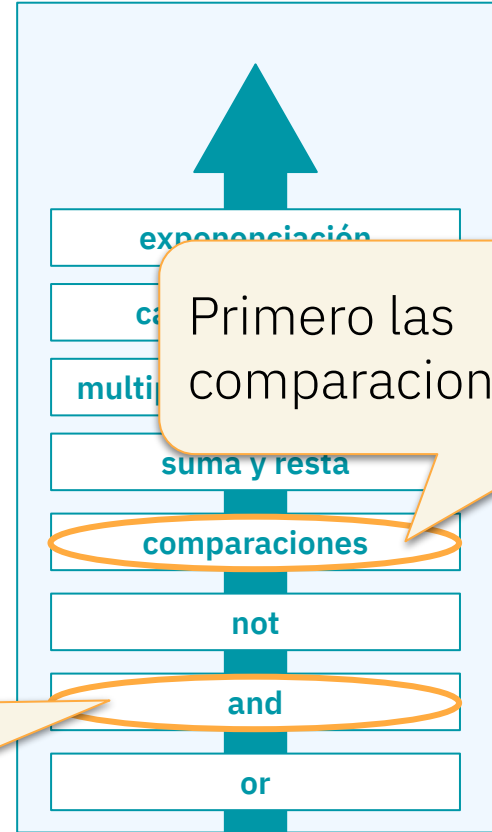
True / False True / False

True / False



Sólo será cierto si ambas partes son ciertas

...y después el and



Primero las comparaciones...

Ejemplo de evaluación

día == 25 and mes == 12

True False

False

VARIABLES	
día	25
mes	11

Al menos 1 parte
es falsa

Combinar condiciones

¿Nació el día de Año Nuevo?



```
día == 1 and mes == 1
```

Sólo será cierta si ambas variables tienen el valor 1



```
día and mes == 1
```

Cuidado, lo que hay a ambos lados del and debe ser de tipo lógico

Combinar condiciones

¿Nació el día de Navidad o el día de Año Nuevo?

```
(día == 25 and mes == 12) or (día == 1 and mes == 1)
```

True / False

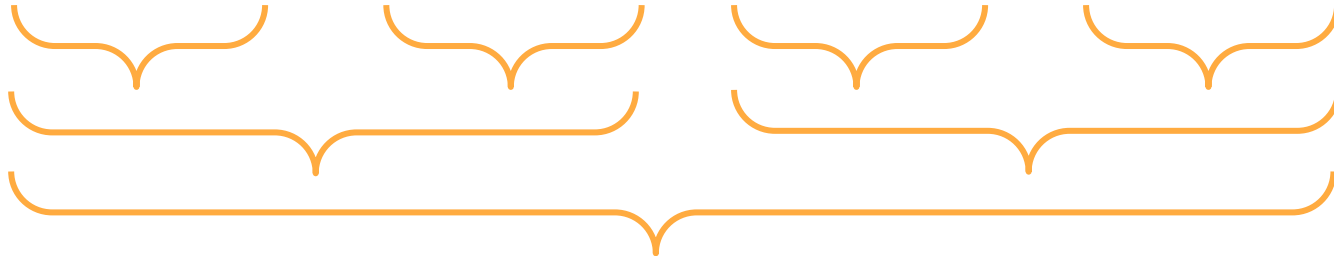
True / False

True / False

Sólo será cierta si alguna de las dos partes es cierta

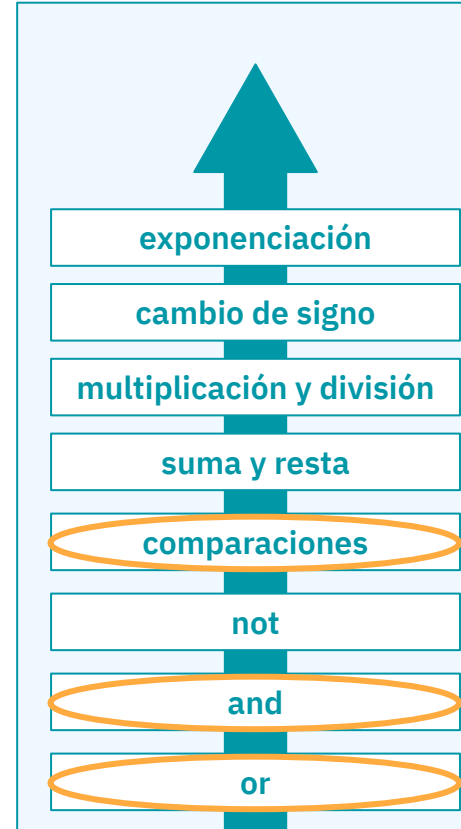
Orden de evaluación

```
día == 25 and mes == 12 or día == 1 and mes == 1
```



```
(día == 25 and mes == 12) or (día == 1 and mes == 1)
```

Paréntesis no necesarios,
pero... ¡dan claridad!



Negación de una condición

¿Nació en el segundo trimestre del año?

```
mes >= 4 and mes <= 6
```

¿Nació en cualquier trimestre del año excepto el segundo?

```
not (mes >= 4 and mes <= 6)
```

True / False

True / False

Sólo será cierta si lo que hay entre paréntesis es falso

Expresiones equivalentes

¿Nació en el segundo trimestre del año?

```
mes >= 4 and mes <= 6
```

```
4 <= mes and mes <= 6
```

Son
equivalentes

¿Nació en cualquier trimestre del año excepto el segundo?

```
not (mes >= 4 and mes <= 6)
```

```
mes < 4 or mes > 6
```

Son
equivalentes

Secuencia de comparaciones

¿A qué equivale esta expresión en Python?

Es válida

```
4 <= mes <= 6
```

```
4 <= mes and mes <= 6
```

Son
equivalentes

¿Y ésta?

```
9 < día < 21
```

```
9 < día and día < 21
```

Son
equivalentes

Ejemplo: Año bisiesto

`bisiesto.py`

```
año = int(input('Introduce el año: '))
```

```
# ¿Es bisiesto?
```

```
es_bisiesto =
```

```
print(f'¿Es bisiesto? { es_bisiesto }')
```

Un año es bisiesto si es múltiplo de 4 pero no de 100, o bien si es múltiplo de 400

Analizar los casos

Un año es bisiesto si es múltiplo de 4 pero no de 100, o bien si es múltiplo de 400

¿año es bisiesto?

año es múltiplo de 400 \Rightarrow **True**

```
año % 400 == 0
```

año es múltiplo de 4 y no es múltiplo de 100 \Rightarrow **True**

```
año % 4 == 0 and año % 100 != 0
```

resto de casos \Rightarrow **False**

Ejemplo: Año bisiesto

bisiesto.py

```
año = int(input('Introduce el año: '))  
# ¿Es bisiesto?  
es_bisiesto = (año % 4 == 0 and año % 100 != 0) or \  
              (año % 400 == 0)  
print(f'¿Es bisiesto? { es_bisiesto }')
```

Ejemplo: Año bisiesto

bisiesto.py

```
año = int(input('Introduce el año: '))  
# ¿Es bisiesto?  
es_bisiesto = (año % 4 == 0 and año % 100 != 0) or \  
              (año % 400 == 0)  
  
print(f'¿Es bisiesto? { es_bisiesto }')
```

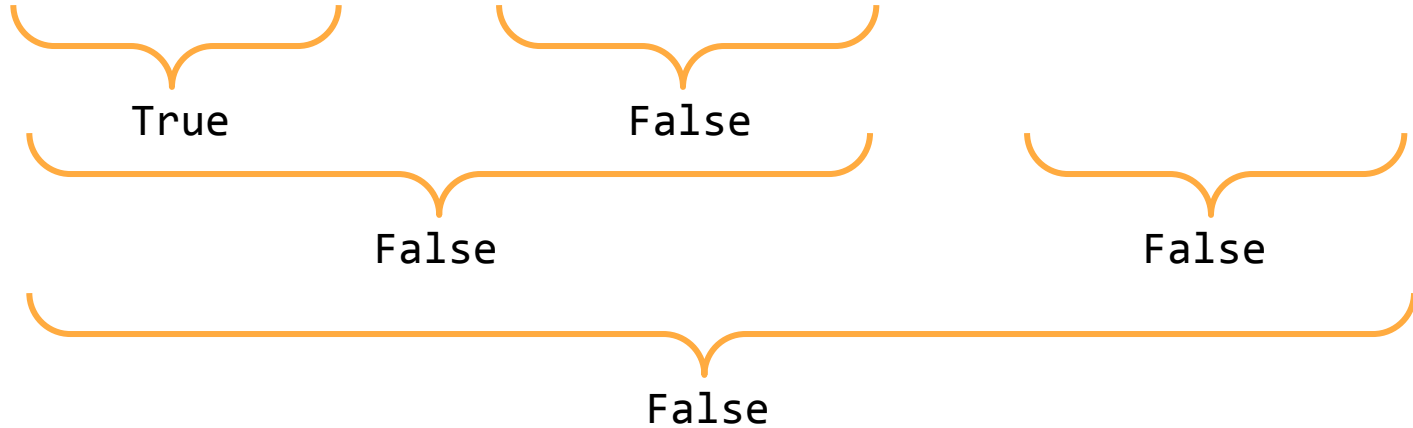
Indica que
la instrucción
continúa en la
siguiente línea

Ejemplo de evaluación

¿Es bisiesto?

VARIABLES	
año	2500

`(año % 4 == 0 and año % 100 != 0) or (año % 400 == 0)`



Isabel Gracia, Pedro García-Sevilla, Angeles López
gracia@uji.es, pgarcia@uji.es, lopeza@uji.es