

▼ Condicionales

En esta sección veremos la sentencia condicional `if` y las distintas variantes que puede asumir, pero antes de eso introduciremos algunas cuestiones generales de escritura de código.

▼ Operadores de comparación

A continuación se muestra la tabla de los operadores de comparación en Python:

Operador	Símbolo
Igualdad	<code>==</code>
Desigualdad	<code>!=</code>
Menor que	<code><</code>
Menor o igual que	<code><=</code>
Mayor que	<code>></code>
Mayor o igual que	<code>>=</code>

Estos operadores devuelven valores booleanos `True` o `False`. Veamos un ejemplo:

```
1 x=5 # Le damos un valor a la x es 5

1 x == 5 # preguntamos si la x es igual a 5 eso quiere decir los dos signos de igual juntos
    True

1 x == 7# preguntamos si x es igual a 7
    False

1 5 < x# 15 es menor que x
    False

1 x < 10
    True
```

Podemos escribir condiciones más complejas usando los operadores lógicos `and`, `or` y `not`:

```
1 (5 < x) or (x > 10)    # se recomienda el uso de paréntesis
2 # la formula de arriba dice se cumple que 5 es menor que x O que x es mayor que 10?
3
    False

1 (5 < x) and (not (x > 10))# se cumple que 5 es menor que x Y x NO es mayor de 10?
    False

1 (5 < x) and (x < 10) #se cumple que 5 es menor que x Y x es menor que 10
    False
```

Esta última comparación también se pueden escribir en Python de la siguiente forma:

```
1 5 < x < 10
    False
```

▼ ¿Qué es cierto?

Python considera `False` a todo lo que está en la siguiente tabla:

Elemento	Representación
Booleano	False
Nulo	None
Cero entero	0
Cero flotante	0.0
Cadena vacía	''
Tupla vacía	()
Diccionario vacío	{}
Conjunto vacío	set()

✓ Cualquier otra cosa se considera True

Ejemplo de verificación de lista vacía:

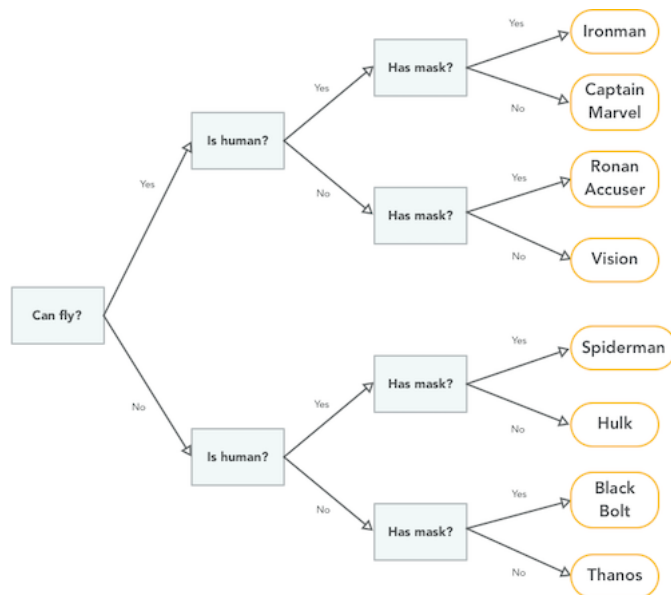
```
1 lista = []
2
3 if lista:
4     print('Hay algo en la lista')
5 else:
6     print('La lista está vacía')

La lista está vacía
```

▼ Ejercicios

▼ Ejercicio 1

Simula el siguiente diagrama de flujo de personajes Marvel utilizando sentencias condicionales:



```
1 # Escriba aquí su solución
2 #Si introducimos:
3 puede_volar = True
4 es_humano = False
5 tiene_mascara = False
6 #Debe devolver: "Vision"
7
8 if puede_volar == True:# si puede volar es verdadero
9     if es_humano == True: # y es humano
10         if tiene_mascara == False: #y tiene mascara
11             print("Capitan Marvel")# escribe capitan Marvel
12         else:# si no tiene mascara
13             print("Iron man")# escribe Iron Man
14     if es_humano != True:#
15         if tiene_mascara == True:
16             print ("Ronan Accuser")
17         else:
18             print ("Vision")
19 if puede_volar == False:
```

```

20 if es_humano == True:
21     if tiene_mascara == True:
22         print ("Spiderman")
23     else:
24         print ("Hulk")
25 else:
26     if tiene_mascara == True:
27         print ("Black Bolt")
28     else:
29         print ("Thanos")
30
31
32

```

Vision

▼ Ejercicio 2

Dada una variable `year` con un valor entero, comprueba si dicho año es bisiesto o no.

- Pista: Un año es bisiesto en el calendario Gregoriano, si es divisible entre 4 y no divisible entre 100. O si es divisible entre 400.
- Puedes hacer la comprobación en [esta lista de años bisiestos](#).

```

1 # Escriba aquí su solución
2 anyo = 2020
3 (anyo%4==0) and (anyo%100!=0)# entre parentesis el año % 4 el resto es igual == a 0 y el año %100 el resto no es !=0
4

```

True

▼ Ejercicio 3

Determina si una cadena de texto dada es un [palíndromo](#).

```

1 #Input:
2 #PALINDROMO es un texto que se lee igual de izquierda a dcha. y de dcha. a izq
3 sentence = 'Ella te da detalle'
4 # Escriba aquí su solución
5 #primero vamos a quitar espacios en blanco
6 sentence_sin_espacios = sentence.replace(" ", "")
7 # ponemos sentence, que sería el texto igual al texto. la orden replace entre parentesis:abrimos comillas y dejamos espacio
8 #en blanco, abrimos comillas y no dejamos nada, así le decimos que sustituya espacios en blanco por nada, y cerramos parentesis
9 sentence
10

```

'Ella te da detalle'

```

1

```

```

1 # Ahora le daremos la vuelta al texto
2 # nombre de la frase sentence abrimos corchetes[inicio, en esté caso en blanco pq es desde el principio:final tb en blanco: dos opcior
3 #1 para que lea de uno en uno los caracteres de izq a dcha. y -1 para leer de uno en uno de dcha a izq.
4 sentence_sin_espacios_al_reves = sentence_sin_espacios[::-1]
5 print (sentence_sin_espacios)

```

Ellatedadetalle

```

1 #Ahora tenemos que poner todo en mayusculas
2 sentence_sin_espacios_al_reves_en_mayusculas =sentence_sin_espacios_al_reves.upper()
3 # print (sentence)

```

```

1 #Ahora tenemos que poner sentence la frase original en Mayusculas y sin espacios
2 sentence_original_mayusculas=sentence.upper()
3 print(sentence_original_mayusculas)
4 #Ahora le quito los espacios a la original tb
5 sentence_original_mayusculas_sin_espacios=sentence_original_mayusculas.replace(" ", "")
6 print (sentence_original_mayusculas_sin_espacios)

```

ELLA TE DA DETALLE
ELLATEDADETALLE

```

1 #POR ULTIMO COMPARAR EL ORIGINAL CON LA FRASE QUE LE HEMOS DADO LA VUELTA
2 (sentence_original_mayusculas_sin_espacios)==(sentence_sin_espacios_al_reves_en_mayusculas)
3
4

```

```
.
```

```
5
```

```
True
```

```
1 Cuando queremos coger solo algunos elementos de una lista
2 lista=("Hola","me","llamo","Juan")# una lista con elementos de texto
3 lista[:]#coge todos los elementos
4 lista[1:]#coge desde el segundo elemento hasta el final
5

('me', 'llamo', 'Juan')
```

[Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.](#)

✓ 0 s completado a las 17:55

