



EXERCISES QUE TRABAJAREMOS EN EL CUE

- EXERCISE 1: EVALUANDO SI UN NÚMERO ES POSITIVO
- EXERCISE 2: UTILIZANDO LA SENTENCIA ELSE
- EXERCISE 3: EVALUANDO CON ELIF
- EXERCISE 4: ITERANDO UNA LISTA
- EXERCISE 5: USANDO FOR ELSE
- EXERCISE 6: CONOCIENDO WHILE
- EXERCISE 7: SENTENCIA BREAK
- EXERCISE 8: SENTENCIA CONTINUE

EXERCISE 1: EVALUANDO SI UN NÚMERO ES POSITIVO

Creemos una sentencia **if** que compruebe si un número es positivo; y en caso de serlo, imprima un mensaje indicándolo.

Para poder resolver el ejercicio debemos hacer una comparación y chequear si el número es mayor a cero, es decir, positivo. Respetando la sintaxis, se colocan los dos puntos luego de la primera línea del **if**. Luego, escribiremos qué sucederá en caso de ser verdadera la comparación, y debemos respetar la indentación como corresponde, por lo que se dejarán cuatro espacios en blanco, y luego escribiremos la instrucción utilizando la función **print()**.

```
1 # Si el número es positivo, imprimimos un mensaje apropiado
2
3 num = 3
4 if num > 0 :
5     print(num, "es un número positivo.")
```

EXERCISE 2: UTILIZANDO LA SENTENCIA ELSE

Ahora escribamos un programa que chequee si el número es mayor o igual a cero; o si por el contrario, es negativo, emitir un mensaje en pantalla según sea el caso.

Primero chequearemos si el número es mayor o igual a cero, de forma similar al ejemplo anterior, pero sustituiremos la comparación por el operador `>=` (mayor o igual).

```
1 num = 3
2 if num >= 0:
3     print(num, "es positivo o cero")
```

Luego, debemos escribir qué sucederá en caso de que la comparación anterior sea falsa; es decir, que el número es menor a cero = negativo. Para ello utilizamos la sentencia **else**, recordando que se colocan los dos puntos luego del **else**, y que se respeta la indentación en el cuerpo de la instrucción de éste.

```
1 else :
2     print(num, "es negativo")
```

El código completo de nuestro programa queda así:

```
1 num = 3
2 if num >= 0:
3     print(num, "es positivo o cero")
4 else :
5     print(num, "es negativo")
6
7 #Resultado:
8 3 es positivo o cero
```

EXERCISE 3: EVALUANDO CON ELIF

Creemos un programa que verifique si un número es positivo, cero o negativo. Para ello debemos chequear primero si es positivo, y de no ser así, si es igual a cero. En caso de que no cumpla ninguna de las 2 verificaciones, ya sabemos que el número es negativo. Utilizaremos una sentencia **if-elif-else**.

Primero chequeamos si es positivo, tal como hicimos previamente.

```
1 num = 3
2 if num > 0 :
3     print(num, "es positivo")
```

Luego, verificaremos si el número es igual a cero con el **elif** y el operador **==** (igual que). Recordemos los dos puntos después de la primera línea, y respetar la indentación en el cuerpo de las instrucciones.

```
1 elif num == 0 :  
2     print(num, "es igual a cero")
```

Y por último escribiremos qué sucede si ninguna de las verificaciones es verdadera, es decir, si el número es negativo con un **else**, igual a como hicimos anteriormente.

```
1 else :  
2     print(num, "es negativo")
```

El código completo de nuestro programa queda así:

```
1 num = 3  
2  
3 if num > 0 :  
4     print(num, "es positivo")  
5  
6 elif num == 0 :  
7     print(num, "es igual a cero")  
8  
9 else :  
10    print(num, "es negativo")  
11  
12  
13 #Resultado:  
14 3 es positivo
```

EXERCISE 4: ITERANDO UNA LISTA

En este ejemplo utilizaremos la sentencia **for** para recorrer una lista de números y sumarlos. Para ello, primero creamos la lista de números con los números: 6, 5, 3, 8, 4, 2, 5, 4 y 11.

```
1 numeros = [6, 5, 3, 8, 4, 2, 5, 4, 11]
```

Luego, declararemos una variable a la que llamaremos **suma** y asignaremos el valor cero. Esta la usaremos para almacenar el resultado de la suma en cada iteración del ciclo **for**. Al terminar el ciclo, el valor que tenga **suma** será el resultado de la suma de todos los números de la lista.



```
1 suma = 0
```

Ahora escribiremos la instrucción del **for** que recorrerá toda la lista. Usaremos la palabra **in**, que asignará el valor actual de la lista a la variable **val** en cada iteración. En el cuerpo de instrucciones actualizaremos el valor de **suma**, sumaremos el valor actual de **suma** con el valor de **val**, y se lo asignaremos de nuevo a **suma**. Recordemos que según la sintaxis debemos poner puntos luego de la primera línea de instrucción del **for**, y también respetar la indentación en el cuerpo de las instrucciones.

```
1 for val in numeros:
2     sum = sum+val
```

Finalmente, después de la instrucción **for**, imprimiremos en pantalla el resultado de la suma, es decir el valor de la variable **suma**.

```
1 print("La suma es", suma)
```

El código completo de nuestro programa queda así:

```
1 numeros = [6, 5, 3, 8, 4, 2, 5, 4, 11]
2
3 suma = 0
4
5 for val in numeros:
6     suma = suma + val
7
8 print("La suma es", suma)
9
10
11 #Resultado:
12 La suma es 48
```

EXERCISE 5: USANDO FOR ELSE

Ahora requerimos imprimir todos los números contenidos en una lista, y cuando la hayamos recorrido toda debemos imprimir un mensaje que indique que no quedan más elementos en la misma.

Para resolver este problema, primero recorreremos la lista haciendo uso de una instrucción **for**, y luego utilizaremos la sentencia **else** que indicará que ya recorrimos toda la lista para imprimir el mensaje solicitado.

Creamos la lista:

```
1 digitos = [0, 1, 5]
```

Ahora la recorremos usando la sentencia **for** y la palabra **in**, asignaremos cada valor de la lista a la variable **i**. Luego, imprimiremos su valor en cada iteración. Recordemos los dos puntos al final de la primera línea del **for**, y de la indentación para escribir el cuerpo de las instrucciones.

```
1 for i in digitos:
2     print(i)
```

Luego de la sentencia **for** escribimos la sentencia **else**. Recordemos los dos puntos luego de la primera línea y de la indentación del cuerpo.

```
1 else:
2     print("No quedan elementos en la lista.")
```

El código completo de nuestro programa queda así:

```
1 digitos = [0, 1, 5]
2
3 for i in digitos:
4     print(i)
5
6 else:
7     print("No quedan elementos en la lista.")
8
9 #Resultado:
10 0
11 1
12 5
13 No quedan elementos en la lista.
```

EXERCISE 6: CONOCIENDO WHILE

En este ejercicio sumaremos los números enteros del 1 al 10. Para resolverlo usaremos el ciclo **while**, un contador al cual llamaremos **i**, y una variable que contendrá el resultado de la suma a la que llamaremos **suma**.

Como necesitamos sumar los números del 1 al 10, el contador lo inicializaremos con el valor 1. Luego, en cada ciclo le sumaremos una unidad. El ciclo se detendrá una vez el valor de **i** sea superior a 10. Es decir que el **while** continuará mientras el contador **i** sea menor o igual a 10.

Inicializamos las variables **i** y **suma**.

```
1 suma = 0
2 i = 1
```

Ahora escribimos la sentencia **while**. Recordemos que la primera línea debemos terminarla con los dos puntos, y que el cuerpo de la instrucción debe respetar la indentación.

```
1 while i <= 10:
2     suma = suma + i
3     i = i+1      # actualizar contador
```

Finalmente, luego de terminado el **while**, imprimimos el resultado de la suma.

```
1 print("La suma es", sum)
```

El código completo de nuestro programa queda así:

```
1 suma = 0
2 i = 1
3
4 while i <= 10:
5     suma = suma + i
6     i = i+1      # actualizar contador
7
8 print("La suma es", suma)
9
10
11 #Resultado:
12 La suma es 55
```

EXERCISE 7: SENTENCIA BREAK

En este ejercicio vamos a detener la ejecución de un ciclo cuando una condición se cumpla. Para ello haremos uso de la sentencia **break**.

Imprimiremos en pantalla las letras de la palabra murciélago, sin embargo, detendremos la ejecución del ciclo cuando encontremos una letra **c**. Luego, imprimiremos la palabra fin.

Primero creemos el ciclo con la sentencia **for** y la palabra **in**. Recordemos los dos puntos al final de la primera línea, y la indentación en el cuerpo de las instrucciones.

```
1 for val in "murcielago":
```

Dentro del cuerpo de **for** lo primero que haremos será revisar si la variable **val** es igual a "c", y si es así, ejecutaremos la sentencia **break**.

```
1     if val == "c":
2         break
3     print(val)
```

Finalmente, imprimiremos "fin" según lo especificado.

```
1 print("Fin")
```

El código completo de nuestro programa queda así:

```
1 for val in "murcielago":
2     if val == "c":
3         break
4     print(val)
5
6 print("Fin")
7
8 #Resultado:
9
10 m
11 u
12 r
13 Fin
```

EXERCISE 8: SENTENCIA CONTINUE

En este ejercicio requerimos ejecutar un ciclo, el cual saltará una iteración si se cumple una condición preestablecida. Para ello utilizaremos la sentencia **continue**.

Debemos imprimir en pantalla las letras de la palabra cadena, exceptuando las apariciones de la letra "a". Luego, imprimiremos la palabra fin.



Primero creemos el ciclo con la sentencia **for** y la palabra **in**. Recordemos los dos puntos al final de la primera línea, y la indentación en el cuerpo de las instrucciones.

```
1 for val in "cadena":
```

Dentro del cuerpo de **for** lo primero que haremos será chequear si la variable **val** es igual a "a", y si es así ejecutaremos la sentencia **continue**.

```
1 if val == "a":
2     continue
3     print(val)
```

Finalmente, imprimiremos "fin" según lo especificado.

```
1 print("Fin")
```

El código completo de nuestro programa queda así:

```
1 for val in "cadena":
2     if val == "a":
3         continue
4     print(val)
5
6 print("Final")
7
8
9 #Resultado:
10 c
11 d
12 e
13 n
14 Final
```