EXERCISES QUE TRABAJAREMOS EN EL CUE

- EXERCISE 1: EVALUANDO MÚLTIPLES CONDICIONES CON WHILE
- EXERCISE 2: OPERADOR OR CON WHILE
- EXERCISE 3: RECORRIENDO UN DICCIONARIO

EXERCISE 1: EVALUANDO MÚLTIPLES CONDICIONES CON WHILE

En el siguiente ejercicio crearemos un ciclo con la sentencia **while,** en el cual deben cumplirse múltiples condiciones para que el **while** continúe ejecutándose.

Debemos crear una variable a la que llamaremos `contador`, y le asignaremos valor cero. Esta variable incrementará su valor en una unidad al final de cada iteración del ciclo. Crearemos dos variables adicionales a las que llamaremos `a` y `b`, y les asignaremos el valor 5 y 10, respectivamente. El ciclo **while** se ejecutará siempre y cuando `contador` sea menor que `a` y `b`.

En cada iteración imprimiremos el valor de `contador`, de `a` y de `b`.

Primero declaremos todas las variables según las especificaciones del ejercicio.

```
1 a = 5
2 b = 10
3 contador = 0
```

Ahora escribamos la sentencia del **while**. Para ello utilizaremos el operador **<** (menor que) en cada una de las comparaciones, y el operador **and** para especificar que deben cumplirse ambas condiciones.

En el cuerpo del **while** primero imprimiremos el valor de todas las variables, y luego actualizaremos el valor de contador. Recordemos los dos puntos al final de la primera línea del **while**, así como la indentación en el cuerpo.

```
1 while contador < a and contador < b:
2     print("cuenta: { contador }, a: {a}, b: {b}")
3     contador += 1
```

Ahora escribamos el código completo de la solución, y el resultado:

```
a = 5
b = 10
contador = 0

while contador < a and contador < b:
    print(f"cuenta: { contador }, a: {a}, b: {b}")
    contador += 1

#Resultado:
cuenta: 0, a: 5, b: 10
cuenta: 1, a: 5, b: 10
cuenta: 2, a: 5, b: 10
cuenta: 3, a: 5, b: 10
cuenta: 4, a: 5, b: 10
```

EXERCISE 2: OPERADOR OR CON WHILE

En este ejercicio crearemos un ciclo con la sentencia **while,** en el cual debe cumplirse alguna de las diferentes condiciones para que el **while** continúe ejecutándose. Debemos observar las pequeñas diferencias en la escritura con respecto al ejercicio anterior, y como el resultado es completamente diferente.

Este ejercicio tiene requerimientos similares al anterior, exceptuando las condiciones que deben cumplirse. Debemos crear una variable a la que llamaremos `contador` y le asignaremos valor cero, ésta incrementará su valor en una unidad al final de cada iteración del ciclo. Crearemos dos variables adicionales a las que llamaremos `a` y `b`, y les asignaremos el valor 5 y 10, respectivamente. El ciclo **while** se ejecutará siempre y cuando `contador` sea menor que `a` o `b`.

En cada iteración imprimiremos el valor de `contador`, de `a` y de `b`.

Primero declaremos todas las variables según las especificaciones del ejercicio.

```
a = 5
b = 10
contador = 0
```

Ahora escribamos la sentencia del **while**. Para ello utilizaremos el operador **<** (menor que) en cada una de las comparaciones, y el operador **or** para especificar que debe cumplirse alguna de las condiciones.

En el cuerpo del **while** primero imprimiremos el valor de todas las variables, y luego actualizaremos el valor de contador. Recordemos los dos puntos al final de la primera línea del **while**, así como la indentación en el cuerpo.

```
while contador < a or contador < b:
    print("cuenta: { contador }, a: {a}, b: {b}")
    contador += 1
```

Ahora escribamos el código completo de la solución, y el resultado:

```
a = 5
b = 10
contador = 0

while contador < a or contador < b:
    print(f"cuenta: { contador }, a: {a}, b: {b}")
    contador += 1


#Resultado:

Cuenta: 0, a: 5, b: 10
Cuenta: 1, a: 5, b: 10
Cuenta: 2, a: 5, b: 10
Cuenta: 3, a: 5, b: 10
Cuenta: 4, a: 5, b: 10
Cuenta: 5, a: 5, b: 10
Cuenta: 6, a: 5, b: 10
Cuenta: 7, a: 5, b: 10
Cuenta: 8, a: 5, b: 10
Cuenta: 9, a: 5, b: 10
```

EXERCISE 3: RECORRIENDO UN DICCIONARIO

A continuación, recorreremos los valores de un diccionario utilizando la sentencia **for**.

Debemos recorrer los datos de un diccionario llamado acciones, que contiene tanto los tickets de las acciones, como sus correspondientes precios. El diccionario de las acciones es el siguiente:

```
acciones = {'AAPL': 187.31, 'MSFT': 124.06, 'FB': 183.50}
```

Primero declaremos la variable `acciones`, y le asignamos el diccionario según el enunciado.

```
acciones = {
        'AAPL': 187.31,
        'MSFT': 124.06,
        'FB': 183.50
    }
```

Luego, crearemos el ciclo **for**. Para ello usaremos la palabra **in** para recorrer todos los valores del diccionario. Utilizaremos el método `.items()` en nuestro diccionario para generar una clave y un valor para cada iteración. Los valores que nos devuelve el método `.ítems()` los asignaremos a las variables clave y valor (podemos colocarles cualquier nombre).

En el cuerpo del **for** imprimiremos los valores de clave y valor, que en este caso corresponde al nombre de la acción y su precio. Recordemos los dos puntos al final de la primera línea del **for**, así como de la indentación del cuerpo.

```
for clave, valor in acciones.items() :
    print(clave + " : " + str(valor))
```

Ahora escribamos el código completo de la solución, y el resultado:

```
acciones = {
        'AAPL': 187.31,
        'MSFT': 124.06,
        'FB': 183.50
    }

for clave, valor in acciones.items() :
    print(clave + " : " + str(valor))


#Resultado:

AAPL : 187.31
MSFT : 124.06
FB : 183.5
```