FOR bucle finito: "Por cada cosa, haré..."

Los humanos siempre buscamos algo que se llama "economía de la lenguage", un proceso evolutivo enfocado a la minimización del esfuerzo invertido a la hora de hablar, escribir o expresarse en general. Un ejemplo de ello, es el uso de *bucles*.

Imagina que tienes 20 invitados a una fiesta que organizaste y quieres darles un vaso de gaseosa, y para ello le pedirás el favor a un amigo que te ayude a servirlos. Darle la siguiente instrucción sería muy ineficiente:

"toma un vaso, sirve gaseosa y pásaselo al invitado 1, luego toma un vaso, sirve gaseosa y pásaselo al invitado 2, luego, toma un vaso, sirve gaseosa y pásaselo al invitado 3, ... toma un vaso, sirve gaseosa y pásaselo al invitado 20"

Una forma más simple es con palabras que transmiten la idea de bucles:

Por cada **invitado** en **la fiesta**, toma un vaso, sirve gaseosa y pásalo"

Por cada, señala que vamos a repetir acciones, ¿cuántas veces? tantos invitados haya en la fiesta, es decir, habrá x repeticiones como x **objetos** haya en un **grupo**.

Nuestro cerebro ya está entrenado para entender que para saber el número de repeticiones necesito saber qué **elemento** dentro de qué **conjunto** (siempre esas dos).

En Python funciona igual:

```
for persona in invitados:
    print("Toma un vaso")
    print("Sirve gaseosa")
    print(f"Entrega a {persona}")
    print("Vuelve a la cocina")
```

La sintaxis (forma de escribir) es de la siguiente manera:

1. Empieza con **for**, que es como decir "Para cada".

2. Luego especificas una **variable(una caja)** que va a cambiar su **valor(objeto)** con cada iteración del bucle, irá tomando sucesivamente cada objeto del grupo

3. Después usas la palabra **in**, que es como decir "en".

4. Luego nombras el **grupo** (lista, tupla, diccionario, fila, columna, base de datos, etc.) sobre la que estás iterando

5. terminas la línea con **":"**, que abre paso a un bloque de código, donde explicas la acción que realizarás, equivalente a "voy a realizar esta acción".

6. Luego con sangría (tab), colocas los comandos a repetir

Ejercicios:

“Traduzca” de español a Python:

1. Para cada día de la semana, tengo una reunión.
2. Para cada alumno en la clase, asigno una computadora.
3. Para cada número del 1 al 10, imprimo su cuadrado.
4. Para cada empresa, reviso su oferta económica
5. Para cada plato en el lavaplatos, lo lavo y lo seco

While bucle indeterminado: "Mientras una condición sea cierta, algo se repite"

While al igual que For es un bucle que señala que una acción debe ser repetida. En el caso de for, las repeticiones están determinadas por el número de objetos en un conjunto. While, por otro lado, no considera ni objetos ni elementos, considera solo una condición: **Mientras** algo sea cierto (True - boolean), una acción se repetirá, cuando la condición se vuelva False, no se repite más.

"Piensa en un while como una fiesta en la que el DJ está poniendo tu música favorita. La instrucción que le das a tu cuerpo es bastante simple: 'Mientras la música que suena sea de mi agrado, seguiré bailando'. No tienes un número definido de canciones que quieres bailar; simplemente, tu acción de bailar está condicionada a la presencia de esa música que te gusta.

En el mundo de la programación, el bucle while sigue una lógica similar. Le proporcionamos a nuestro programa una condición y le decimos: 'Mientras esta condición sea verdadera, sigue ejecutando las instrucciones que te he dado'.

```
buena_musica=True
while buena_musica:
    print("bailar")
```

Ejercicios:

1. Mientras llueva, me quedo en casa.

2. Mientras la contraseña sea incorrecta, no puedo ingresar.

3. Mientras no haya pico y placa, puedo sacar el carro

4. Mientras mi nivel de azúcar en la sangre sea menor a 140, no requiero tratamiento

5. Mientras no entienda, sigo estudiando