Trabajo Práctico VII

Estructuras repetitivas incondicionales

Las estructuras repetitivas nos facilitan la ejecución reiterada de un mismo bloque de código; en el caso de las incondicionales, la cantidad de veces que se repite el bloque será un número fijo. En Python, la instrucción ***for*** *nos permite crear este tipo de estructuras. Su sintaxis es la siguiente:*

```
for [variable de iteración] in [elementos a repetir]:
    [código a repetir]
```

En donde:

- ***for*** *es una palabra reservada que utilizamos para indicar que vamos a repetir un bloque de código.*
- *La* ***variable de iteración*** *tomará el valor actual de los* ***elementos a repetir*** *en cada iteración del bucle.*
- *Los* ***elementos a repetir*** *son el conjunto de valores sobre el que haremos la repeticiones.*
- *El* ***código a repetir*** *es el bloque de instrucciones que se ejecutará en cada iteración del bucle.*

Por ejemplo, para mostrar en pantalla los primeros 100 números enteros positivos, comenzando desde 0, podríamos escribir:

```
for n in range(101):
    print(n)
```

En este caso, ***n*** *es la variable de iteración, y los* ***elementos a repetir*** *son los números del 0 al 100, que obtenemos con la función predefinida* ***range(101)****. El* ***código a repetir*** *en este caso es la instrucción* ***print(n)****, que simplemente muestra en pantalla el valor actual de la variable de iteración.*

Alternativamente, la función ***range*** *tiene una variante con dos parámetros, por si necesitamos comenzar con un valor que no sea cero. Por ejemplo, para mostrar los números enteros del 50 al 100, podríamos escribir:*

```
for n in range(50,101):
    print(n)
```

Notar que la función ***range*** *no incluye al número máximo del rango indicado.*

---

1. Cree un script para mostrar los primeros 100 números enteros positivos, comenzando desde el 1.

2. Modifique el script del ejercicio anterior para que se muestren sólo los números pares. Para saber si un número es par, utilice el operador de módulo (**%**).

3. Cree un script para calcular el resultado de sumar los números desde el 75 al 150.

4. Cree un script que le solicite al usuario ingresar un número entero, y muestre en pantalla el factorial de dicho número. NOTA: puede obviar la validación en este ejercicio, pero recuerde que la función *range* no incluye al valor máximo enviado como parámetro.
`factorial de n = n! = 1 * 2 * 3 * … * (n - 1) * n`

5. Cree un script que le solicite al usuario ingresar 10 números enteros, y por cada uno, informarle si el mismo es positivo, negativo, o cero.

6. Cree un script que le solicite al usuario ingresar 10 números, y una vez ingresados, le muestre en pantalla cuál es el máximo, y en qué posición lo ingresó. Por ejemplo, si el usuario ingresa los números 2, 63, -3, 20, 55, 89, 7, 32, 9, y 33, se le debería mostrar el mensaje *"El mayor número ingresado es 89, y lo ingresaste en la posición 6"*. NOTA: las posiciones posibles comienzan desde 1.

7. Extienda el script del ejercicio anterior para que también informe el número mínimo ingresado, y su posición.

8. Un cliente ha solicitado un programa que le permita ingresar los mililitros de lluvia caídos diariamente en una semana, para que el programa le informe en pantalla el promedio de precipitación de esa semana. El cliente también desea saber cuál fue el día en que más llovió en la semana.
A modo ilustrativo, un reporte generado por el programa debería verse como sigue, luego de haber leído las precipitaciones de los 7 días de la semana:

```
El promedio de precipitaciones fue de XX mls. diarios.
El día de más precipitaciones fue el xxxxxx (nombre del día).
```

Tenga en cuenta que la numeración de los días de la semana comienza con el 1 para el día domingo.

Codifique el programa para dar solución a lo solicitado por el cliente.