# Ciclo 1
# Fundamentos de programación con Python
# Sesión 13: Implementación de estructuras de control ciclo (for)

Programa Ciencias de la Computación e Inteligencia Artificial
Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda
Bogotá

# Agenda

1. Introducción
2. Colección (range)
3. Ejemplos
4. Ejercicios

# 1. Introducción

Un bucle for es un bucle que repite el bloque de instrucciones un número predeterminado de veces. El bloque de instrucciones que se repite se suele llamar cuerpo del bucle y cada repetición se suele llamar iteración.

```
for <elem> in <iterable>:
    cuerpo del bucle
```

# 1. Introducción

- Esta estructura de control tiene dos propósitos primordiales que no siempre son soportados por todo lenguaje de programación:

    1. Como una forma compacta de escribir un ciclo mientras (while).
    2. Para iterar sobre los elementos de una colección de elementos.

- Esta estructura es usualmente utilizada cuando se conocen los valores inicial y final de la variable que es utilizada en la condición de parada.

# 1. Introducción

- Un ciclo para (for) puede ser usado para obtener uno a uno los elementos de una colección de elementos y poder realizar con cada uno de ellos el mismo bloque de operaciones.
- Un esquema textual que en Python representa dicho ciclo (for) es el que se da en el siguiente fragmento de código.

```
<bloque_prev>
for <elemento> in <coleccion>:
    <bloque>
<bloque_sigui>
```

# 1. Introducción

- El fragmento <bloque_prev> es el bloque de instrucciones previas que han sido ejecutadas antes del ciclo.
- El fragmento <elemento> es la variable que se usa para ir recorriendo (iterando) sobre los elementos de la colección.
- El fragmento <coleccion> es la colección de elementos que ser´a recorrida (iterada) con el ciclo.
- El fragmento <bloque> es el bloque de instrucciones principal del ciclo que se ejecuta con cada uno de los elementos de la colección.
- El fragmento <bloque_sigui> es el bloque de instrucciones que se ejecutan después de terminar de ejecutar el ciclo.

# 2. Colección Range

- Existen varias colecciones que se pueden iterar en Python, una de ellas es la colección (range).
- Una colección (range) es una colección de números en un intervalo, definido por valor inicial, un valor final y un valor de incremento/decremento usado a partir del valor inicial para determinar que valores quedan en el rango.
- Si no se da el valor de inicio, éste se fija en cero (0) y si no se da valor de incremento/decremento, este se fija en uno (1).

# 2. Colección (range)

Los rangos se combinan perfectamente con la instruccion (for):

- Se define una variable que se utilizará para recorrer cada número en el rango.

```
for i in range(-5, 6, 1):
    print(i)

for I in range(10,0,-1):
    print(i)

for i in range(0, 55, 5):
    print(i)
```

# 3. Ejemplo

- Crear un algoritmo que muestre los primeros 10 números de la sucesión de Fibonacci. La sucesión comienza con los números 0 y 1 y, a partir de éstos, cada elemento es la suma de los dos números anteriores en la secuencia: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55…

# 3. Ejemplo

- Dado un número entero positivo, mostrar su factorial. El factorial de un número se obtiene multiplicando todos los números enteros positivos que hay entre el 1 y ese número.

# 3. Ejemplo

- Solicitar al usuario que ingrese una frase y luego imprimir la cantidad de vocales que se encuentran en dicha frase.

# 3. Ejemplo

- Escribir un programa que solicite al usuario una cantidad y luego itere la cantidad de veces dada. En cada iteración, solicitar al usuario que ingrese un número. Al finalizar, mostrar la suma de todos los números ingresados.

# 4. Ejercicios

- Escribir un programa que permita al usuario ingresar 6 números enteros, que pueden ser positivos o negativos. Al finalizar, mostrar la sumatoria de los números negativos y el promedio de los positivos.

  No olvidar que no es posible dividir por cero, por lo que es necesario evitar que el programa arroje un error si no se ingresaron números positivos.

# 4. Ejercicios

- Escribir un programa que permita al usuario ingresar dos años y luego imprima todos los años en ese rango, que sean bisiestos y múltiplos de 10.

**Nota:** para que un año sea bisiesto debe ser divisible por 4 y no debe ser divisible por 100, excepto que también sea divisible por 400.

El futuro digital es de todos

MinTIC

# Preguntas

UNIVERSIDAD SERGIO ARBOLEDA

MisiónTIC2022