

A graphic on the left side of the slide consisting of four stacked horizontal bars in purple, orange, yellow, and blue. The text "Agencia de Aprendizaje a lo largo de la vida" is written across these bars. The orange bar has a white arrow pointing to the right.

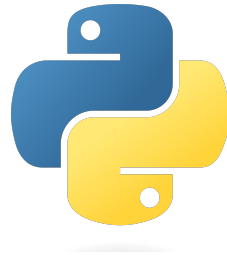
Agencia de
Aprendizaje
a lo largo
de la vida

CODO A CODO INICIAL

Clase 23

Python 11

Bucle For I



Clase 22

Python 10 - Bucle While II

- Bucle While para procesamiento y manipulación de datos
- Creación de patrones y secuencias con bucle While
- Bucle While con Listas

Clase 23

Python 11 - Bucle For I

- Sintaxis.
- Función range():
Secuencias numéricas.
- Iteración en Cadenas
- Iteración sobre Secuencias
- Sentencia break
- Sentencia continue

Clase 24

Python 12 - Bucle For II

- Bucle For para procesamiento y manipulación de datos
- Creación de patrones y secuencias con bucle For
- Bucle For con Listas
- Bucle For con Cadenas

Les damos la bienvenida

Vamos a comenzar a grabar la clase

Bucle For

El bucle **for** es una estructura de control que se utiliza para iterar sobre una secuencia de elementos, como una lista, cadena, rango o cualquier otro objeto iterable. Su sintaxis es simple y compacta, y consta de tres partes: la palabra clave **for**, una variable de iteración que toma el valor de cada elemento en la secuencia en cada iteración, la palabra clave **in**, y la secuencia sobre la que se va a iterar. El bucle for recorre cada elemento de la secuencia y ejecuta un bloque de código para cada uno de ellos, lo que lo hace ideal para realizar tareas repetitivas o para procesar cada elemento de una colección.

Bucle For | La función range()

Esta estructura se utiliza cuando sabemos la cantidad de repeticiones a efectuar. Tiene el siguiente formato:

```
for i in range(inicio, fin, paso):  
    sentencia1  
    sentencia2  
primer sentencia fuera del for
```

- **i:** variable que incrementa su valor en **paso** unidades en cada iteración.
- **inicio:** Es el valor inicial de i
- **fin:** El ciclo se repite mientras i sea menor que fin.
- **paso:** valor en que se incrementa i en cada iteración.

Bucle For | La función range()

Este ejemplo imprime los números del 1 al 5:

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



1
2
3
4
5

La función **range()** genera una secuencia de números en un rango específico. Se utiliza comúnmente en bucles for para iterar sobre una secuencia de números. En la sintaxis de range() contamos con las siguientes alternativas:

- range(inicio, fin, paso)
- range(inicio, fin): Se asume que paso = 1.
- range(fin): Se asume que inicio = 0 y paso = 1.

La función range() no incluye el último valor en la secuencia. Es decir, el rango va desde el valor inicial hasta el valor final menos uno. Es por eso que range(1, 6) generará los números del 1 al 5, pero no incluirá el 6 en la secuencia.

Bucle For | Ejemplos de uso de la función range()

Ejemplo con 3 parámetros:

```
for num in range(5, 16, 2):  
    print(num, end=" ")
```



5 7 9 11 13 15

Ejemplo con 2 parámetros:

```
for num in range(3, 11):  
    print(num, end=" ")
```



3 4 5 6 7 8 9 10

Ejemplo con 1 parámetro:

```
for num in range(10):  
    print(num, end=" ")
```



0 1 2 3 4 5 6 7 8 9

Bucle For | Ejemplo

Con el bucle **for** no necesitamos usar **contadores**, ya que la variable del ciclo puede asumir esa función. Esto permite escribir algunos programas de una manera más compacta.

Este programa permite ingresar 5 valores por teclado, obtener su suma y su promedio:

```
suma = 0
for cont in range(5):
    num= int(input("Ingrese un número: "))
    suma = suma + num

print(f'La suma es: {suma}')
print(f'El promedio es: {suma/(cont+1)}')
```



```
Ingrese un número: 2
Ingrese un número: 6
Ingrese un número: 9
Ingrese un número: 15
Ingrese un número: 3
La suma es: 35
El promedio es: 7.0
```

Bucle For | Iteración sobre cadenas

La **iteración sobre cadenas** se refiere al proceso de recorrer cada carácter individual en una cadena de texto utilizando un bucle. El **bucle for** permite iterar sobre una cadena, recorriendo cada carácter de la cadena uno por uno, comenzando desde el primer carácter hasta el último.

```
cadena = "Python"  
  
for letra in cadena:  
    print(letra)
```



P
y
t
h
o
n

Este bucle recorre cada carácter de la cadena "Python" e imprime cada letra en una línea separada.

Bucle For | Iteración sobre listas

La **iteración sobre listas** se refiere al proceso de recorrer cada elemento de una lista uno por uno para realizar ciertas operaciones o tareas con cada elemento. Mediante el uso de **bucles for**, cada elemento de la lista se asigna a una variable temporal en cada iteración del bucle. Esto permite acceder a cada elemento y realizar acciones específicas con ellos.

```
numeros = [2, 4, 6, 8, 10]  
  
for num in numeros:  
    print(num*2)
```



```
4  
8  
12  
16  
20
```

Este bucle recorre cada elemento de la lista de números e imprime el doble de cada uno en una línea separada.

Estructuras repetitivas | Break

Break permite salir de un bucle **while** o **for** en el momento que se cumpla alguna condición.

```
suma = 0

for cont in range(15):
    print(cont)
    suma = suma + cont
    if cont == 3:
        break

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

```
0
1
2
3
La suma es: 6
```

break no se considera una buena práctica. Si un bucle **for** requiere un break, quizás pueda resolverse el proceso usando en su lugar **while**.

Estructuras repetitivas | Continue

Continue en un bucle **for** permite omitir el resto del código dentro del bucle en una iteración específica y pasar a la siguiente iteración:

```
for contador in range(1, 6):  
    if contador == 3:  
        print("Se omite la iteración  
cuando contador es 3.")  
        continue  
    print("Iteración:", contador)  
  
print("Fin del bucle.")
```

```
Iteración: 1  
Iteración: 2  
Se omite la iteración cuando  
contador es 3.  
Iteración: 4  
Iteración: 5  
Fin del bucle.
```

Desafíos

Desafío 1: Suma de Números Naturales

Desarrolla una función que calcule la suma de los números naturales hasta un número dado utilizando un bucle for. La suma de los números naturales hasta el número n se define como la suma de todos los números enteros positivos desde 1 hasta n .

Por ejemplo, la suma de los números naturales hasta 6 es $1 + 2 + 3 + 4 + 5 + 6 = 21$.

La función debe recibir un número entero n y devolver la suma de los números naturales hasta n .

Desafío 2: Tabla de multiplicar

Escribir un programa que permita obtener la tabla de multiplicar de un valor ingresado por el usuario. Utilizar for. Una salida posible es:

```
Ingresa un número para obtener la tabla de multiplicar: 4
Tabla de multiplicar del 4:
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
```

Desafío extra:

¿Cómo podrías modificar el programa anterior para que se impriman todas las tablas desde el 1 al 10 hasta un número dado?

Desafío 3: Múltiplos de un número

Escribir un programa que muestre todos los múltiplos de un número dado dentro de un rango definido por otro número. El programa debe solicitar al usuario ingresar dos números enteros: el número base del cual se calcularán los múltiplos y el límite superior del rango. Luego imprimirá todos los múltiplos del número base que estén dentro del rango especificado, incluyendo el límite superior.

```
Ingresa el número base: 4
Ingresa el límite superior del rango: 25
Los múltiplos de 4 dentro del rango del 1 al 25 son:
4 8 12 16 20 24
```

Material extra

Artículos de interés

Material extra:

- [Bucle for en Python](#) | IONOS
- [Explicación del búcle for de Python](#) | FreeCodecamp.org
- [Bucle for](#) | El libro de Python

Videos:

- [Bucles for en Python](#) | Tutoriales sobre Ciencia y Tecnología
- [Tablas de multiplicar con bucle for](#) | Tutoriales sobre Ciencia y Tecnología

No te olvides de dar el presente

Recordá:

- Revisar la Cartelera de Novedades.
- Hacer tus consultas en el Foro.
- Realizar el Ejercicio de Repaso.

Todo en el Aula Virtual.

Muchas gracias por tu atención.

Nos vemos pronto