

A graphic consisting of four stacked, 3D rectangular blocks in purple, orange, yellow, and blue, with an orange arrow pointing to the right integrated into the middle of the stack.

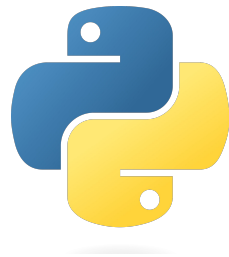
Agencia de
Aprendizaje
a lo largo
de la vida

CODO A CODO INICIAL

Clase 24

Python 12

Bucle For II





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

Clase 25

Python 13 - Listas I

- Definición.
- Creación: Sintaxis, tipos de elementos.
- Acceso a elementos.
- Modificación de listas.

Les damos la bienvenida

Vamos a comenzar a grabar la clase

Procesar y manipular datos

Se puede emplear el **bucle for** para procesar y manipular datos de manera iterativa cuando el número de iteraciones se conoce de antemano. En este contexto, el bucle for resulta útil para interactuar con el usuario a través de la entrada y salida estándar, como la captura de datos mediante **input()** y la presentación de resultados mediante **print()**, permitiendo una ejecución flexible y adaptable según las necesidades del programa.

Procesar y manipular datos | Ejemplo I

Este programa simula la carga de 5 importes de productos de un supermercado, mostrando el total al final.

```
total = 0

for i in range(5):
    importe = float(input("Ingrese el importe: "))
    total = total + importe

print(f"El total de los importes ingresados es: {total}")
```

```
Ingrese el importe: 4
Ingrese el importe: 9
Ingrese el importe: 12
Ingrese el importe: 15
Ingrese el importe: 6
El total de los
importes ingresados es:
46.0
```

Procesar y manipular datos | Ejemplo II

```
usuarioValido = "admin"
contraseniaValida = "1234"

for intento in range(3, 0, -1): # Comienza
    en 3, termina en 1, decrementando en 1
    usuario = input("Usuario: ")
    contrasenia = input("Contraseña: ")

    if usuario == usuarioValido and
    contrasenia == contraseniaValida:
        print("Inicio de sesión exitoso.")
        break # Salir del bucle si el
        inicio de sesión es exitoso
```

(continúa...)

Este programa simula un inicio de sesión con tres intentos, donde el usuario es "admin" y la contraseña es "1234".

Comienza definiendo un usuario y contraseña válidos (A).

El bucle For permitirá hasta 3 intentos de inicio de sesión. En cada iteración del bucle, solicita al usuario que ingrese su nombre de usuario y contraseña (B).

Si los datos ingresados coinciden con el usuario y contraseña válidos, se imprime "Inicio de sesión exitoso" y se finaliza el bucle utilizando un **break** (C). Si bien break está desaconsejado en este caso es necesario dado que se utiliza for en vez de while para este caso.

Procesar y manipular datos | Ejemplo II

```
else:
    print("Usuario o contraseña
incorrectos.")
    print(f"Intentos restantes: {intento
- 1}")
    if intento == 1:
        print("Se han agotado los
intentos. Cierre del programa.")
```

D**E**

Si los datos ingresados son incorrectos, se imprime "Usuario o contraseña incorrectos" y se reduce en uno el número de intentos restantes **(D)**. Cuando los intentos se agotan (es decir, intentos == 1), se imprime "Se han agotado los intentos. Cierre del programa." y se termina la ejecución del programa **(E)**.

```
Usuario: Codo a Codo
Contraseña: 123
Usuario o contraseña incorrectos.
Intentos restantes: 2
Usuario: admin
Contraseña: 1234
Inicio de sesión exitoso.
```


Patrones y secuencias

Con **for** también es posible crear secuencias numéricas y patrones visuales. Para crear **secuencias numéricas**, podemos usar **range()** para generar secuencias ascendentes o descendentes.

Para construir **patrones visuales** podemos emplear bucles for anidados para controlar las filas y columnas del patrón. Esto nos permite imprimir caracteres o símbolos en posiciones específicas dentro del patrón, creando formas geométricas variadas y visualmente interesantes.

Secuencias numéricas | Ejemplo

Este programa crea una secuencia de números pares separados por guiones hasta alcanzar el número ingresado por teclado.

```
maximo = int(input("Ingrese el número máximo para la secuencia: "))  
  
print("Secuencia de números pares:")  
for numero in range(2, maximo + 1, 2):  
    print(numero, end="-")
```

Ingrese el número máximo para la secuencia: 11
Secuencia de números pares:
2-4-6-8-10-

Para pensar:

¿Cómo podemos evitar que se imprima el último guión?

Patrones visuales | Ejemplo

Este código genera un cuadrado de asteriscos con 5 filas y 5 columnas, donde cada fila contiene 5 asteriscos. La expresión `print("* ", end="")` imprime un asterisco seguido de un espacio en la misma línea, y `print()` al final de cada fila crea un salto de línea para pasar a la siguiente fila.

```
for fila in range(1, 6):  
    for columna in range(1, 6):  
        print("* ", end="")  
    print()
```



```
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *
```

Para pensar:

¿Cómo podríamos lograr que se generen estas figuras?

```
* * * * *  
*       *  
*       *  
*       *  
*       *  
* * * * *
```

```
*  
* *  
* * *  
* * * *  
* * * * *
```

Bucle For con Listas

En Python, una **lista** es una estructura de datos que puede contener múltiples elementos, como números, cadenas, u otros objetos. El **bucle For** de Python está diseñado específicamente para iterar sobre colecciones, como las listas.


Por ejemplo, un bucle for puede ser utilizado para iterar sobre los elementos de una lista hasta que se alcance el largo de la lista o encontrar un elemento específico.

Bucles for con Listas | Ejemplo I

Este código recorre la lista **nombres** utilizando un bucle **for**. El bucle se ejecuta copiando, en cada iteración, un elemento de la lista en la variable “nombre”. Esto se repite una vez por cada elemento de la lista, comenzando en el primero y finalizando en el último.

```
# Definir una lista de nombres
nombres = ["Juan", "María", "Pedro", "Ana"]

# Mostrar los nombres usando un bucle for
for nombre in nombres:
    print(nombre)
```



Juan
María
Pedro
Ana

Bucles for con Listas | Ejemplo II

Este código utiliza un bucle for para recorrer una lista de números y calcula la suma de esos números:

```
# Definir una lista de números
numeros = [10, 20, 30, 40, 50]
# Inicializar la suma
suma = 0

# Recorrer la lista y sumar los números
usando un bucle for
for numero in numeros:
    suma = suma + numero

# Mostrar la suma
print("La suma de los números es:", suma)
```

→ La suma de los números es: 150

Para pensar:

¿Qué modificaciones le harías a este programa para que también imprima el promedio de los valores de la lista?

Desafíos

Desafío 1: Patrones visuales

Desarrolla un programa que permita mostrar los siguientes patrones. Luego crear una función para cada uno utilizando For y un menú que permita seleccionar cuál imprimir.



Triángulo escaleno



Rectángulo de 8 x 4



Rombo



Flecha

Desafío 2: Secuencia numérica

Desarrolla un programa que permita crear una secuencia numérica. El usuario ingresará tres valores: inicio, fin y salto. Antes de iniciar el bucle, validar lo siguiente:

- El valor de **inicio** debe ser un número positivo.
- El valor de **fin** debe ser mayor que el de inicio.
- El valor de **salto** no puede ser cero o negativo.

Asumimos que el operador sólo ingresa números enteros.

Desafío 3: Sumador de números pares e impares

Desarrolla un programa que permita ingresar 15 números positivos. Al finalizar se deberá mostrar:

- La **suma** de los números pares.
- El **promedio** de los números impares.
- El número **par más alto**.
- El número **impar más bajo**.

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