

Guía de Ejercicios Python: Estructuras de Control con Énfasis en IF

Instrucciones Generales

- Cada ejercicio debe ser resuelto en un archivo Python independiente, nombrado según el número del ejercicio (ej: `ejercicio1.py`, `ejercicio2.py`, etc.)
- Todos los ejercicios deben incluir el uso de la estructura de control `if`
- Utiliza comentarios para documentar tu código
- Las soluciones se encuentran al final de esta guía

Ejercicio 1: Variables y Validación de Acceso

Tema: Variables (strings)

Descripción: Crea un programa que simule el acceso a una aplicación de streaming. El programa debe:

1. Solicitar nombre de usuario y contraseña
2. Comparar con credenciales predefinidas
3. Mostrar un mensaje de bienvenida personalizado si las credenciales son correctas
4. Mostrar un mensaje de error si las credenciales son incorrectas
5. Si el usuario está registrado pero la contraseña es incorrecta, dar una pista sobre la contraseña (primera letra)

Tiempo estimado: 10-15 minutos

Ejercicio 2: Constantes y Calculadora de Descuentos

Tema: Constantes y operadores

Descripción: Desarrolla una calculadora para una tienda de ropa en línea que:

1. Defina constantes para diferentes categorías de descuentos (`DESCUENTO_ESTUDIANTE = 0.15`, `DESCUENTO_CLIENTE_FRECUENTE = 0.10`, etc.)
2. Solicite el precio original de un artículo
3. Pregunte si el usuario es estudiante (s/n) y si es cliente frecuente (s/n)
4. Aplique el descuento correspondiente (si aplican varios, usar el mayor)
5. Muestre el precio original, el descuento aplicado y el precio final

Tiempo estimado: 15-20 minutos

Ejercicio 3: Operadores y Verificador de Playlist

Tema: Operadores lógicos y comparación

Descripción: Crea un programa que ayude a un usuario a decidir si una canción es adecuada para su playlist basándose en criterios específicos:

1. Solicita duración de la canción en minutos (float)
2. Solicita género musical (string)
3. Solicita año de lanzamiento (int)
4. Verifica si la canción cumple con los criterios para ser incluida:

- Duración entre 2.5 y 4.5 minutos
- Género sea "rock", "pop" o "indie" (sin distinguir mayúsculas/minúsculas)
- Año de lanzamiento posterior a 2010

5. El programa debe informar si la canción se incluye en la playlist y explicar por qué sí o por qué no

Tiempo estimado: 15-20 minutos

Ejercicio 4: Aritmética y Conversor de Unidades

Tema: Aritmética y operaciones matemáticas

Descripción: Desarrolla un programa que convierta entre diferentes unidades de medida:

1. Muestra un menú con las siguientes opciones:
 - Convertir de kilómetros a millas
 - Convertir de millas a kilómetros
 - Convertir de grados Celsius a Fahrenheit
 - Convertir de grados Fahrenheit a Celsius
2. Según la opción elegida, solicita el valor a convertir
3. Realiza los cálculos correspondientes ($1 \text{ km} = 0.621371 \text{ millas}$, $^{\circ}\text{F} = ^{\circ}\text{C} \times 9/5 + 32$, $^{\circ}\text{C} = (^{\circ}\text{F} - 32) \times 5/9$)
4. Si el usuario introduce un valor negativo para distancias, muestra un mensaje adecuado
5. Para temperaturas, indica si está bajo cero, a temperatura ambiente ($15\text{-}25^{\circ}\text{C}$) o caliente

Tiempo estimado: 20-25 minutos

Ejercicio 5: Listas y Organizador de Biblioteca Digital

Tema: Listas y manipulación de colecciones

Descripción: Crea un programa que ayude a organizar una pequeña biblioteca digital de libros:

1. Define una lista con al menos 5 libros (títulos como strings)
2. Permite al usuario:
 - Ver todos los libros
 - Buscar un libro por nombre (indicar si existe o no)
 - Agregar un nuevo libro (verificando que no exista ya)
 - Eliminar un libro (verificando que exista)
3. Utiliza comparaciones con `if` para las verificaciones
4. Muestra mensajes apropiados en cada operación
5. Si la biblioteca supera los 10 libros, mostrar un mensaje indicando que la versión gratuita está llena

Tiempo estimado: 20-25 minutos

Soluciones

Solución Ejercicio 1: Variables y Validación de Acceso

```

# Definimos las credenciales correctas
usuario_correcto = "estudiante"
contrasena_correcta = "python123"

# Solicitamos credenciales al usuario
usuario_ingresado = input("Ingrese su nombre de usuario: ")
contrasena_ingresada = input("Ingrese su contraseña: ")

# Verificamos las credenciales
if usuario_ingresado == usuario_correcto and contrasena_ingresada == contrasena_correcta:
    print(f"¡Bienvenido/a {usuario_ingresado}! Acceso concedido.")
elif usuario_ingresado == usuario_correcto:
    # El usuario es correcto pero la contraseña no
    print("Contraseña incorrecta.")
    print(f"Pista: la contraseña comienza con '{contrasena_correcta[0]}'")
else:
    print("Usuario no registrado. Por favor, verifica tus credenciales.")

```

Explicación: Este programa utiliza variables de tipo string para almacenar y comparar credenciales. La estructura de control `if-elif-else` permite manejar diferentes escenarios: acceso exitoso, contraseña incorrecta o usuario no registrado. El operador `and` asegura que ambas condiciones se cumplan para el acceso exitoso.

Solución Ejercicio 2: Constantes y Calculadora de Descuentos

```

# Definimos las constantes para los descuentos
DESCUENTO_ESTUDIANTE = 0.15 # 15%
DESCUENTO_CLIENTE_FRECUENTE = 0.10 # 10%
DESCUENTO_PRIMERA_COMPRA = 0.05 # 5%

# Solicitamos el precio del artículo
precio_original = float(input("Ingrese el precio del artículo: $"))

# Preguntamos por las condiciones para descuentos
es_estudiante = input("¿Es usted estudiante? (s/n): ").lower() == 's'
es_cliente_frecuente = input("¿Es usted cliente frecuente? (s/n): ").lower() == 's'
es_primera_compra = input("¿Es su primera compra? (s/n): ").lower() == 's'

# Calculamos el descuento a aplicar
descuento = 0

if es_estudiante:
    descuento = DESCUENTO_ESTUDIANTE
elif es_cliente_frecuente:
    descuento = DESCUENTO_CLIENTE_FRECUENTE
elif es_primera_compra:
    descuento = DESCUENTO_PRIMERA_COMPRA

# Calculamos el precio final
monto_descuento = precio_original * descuento
precio_final = precio_original - monto_descuento

```

```

# Mostramos los resultados
print("\n--- Resumen de la compra ---")
print(f"Precio original: ${precio_original:.2f}")
if descuento > 0:
    print(f"Descuento aplicado: {descuento*100:.0f}% (${monto_descuento:.2f})")
else:
    print("No se aplicaron descuentos")
print(f"Precio final: ${precio_final:.2f}")

```

Explicación: Este programa utiliza constantes para representar los diferentes tipos de descuentos. La estructura `if-elif` determina cuál es el descuento más ventajoso para el cliente. El operador `==` compara si la respuesta del usuario coincide con 's' para determinar si aplica cada tipo de descuento. La notación de formato `:.2f` asegura que los valores monetarios se muestren con dos decimales.

Solución Ejercicio 3: Operadores y Verificador de Playlist

```

# Solicitamos información sobre la canción
duracion = float(input("Ingrese la duración de la canción en minutos: "))
genero = input("Ingrese el género de la canción: ").lower()
anio = int(input("Ingrese el año de lanzamiento: "))

# Definimos los criterios
duracion_adeuada = 2.5 <= duracion <= 4.5
genero_adeuado = genero in ["rock", "pop", "indie"]
anio_adeuado = anio > 2010

# Verificamos si la canción cumple todos los criterios
if duracion_adeuada and genero_adeuado and anio_adeuado:
    print("\n¡La canción es perfecta para tu playlist!")
else:
    print("\nLa canción no cumple con todos los criterios para la playlist:")

    # Explicamos por qué no cumple cada criterio
    if not duracion_adeuada:
        print(f"- La duración ({duracion} min) está fuera del rango ideal (2.5-4.5 min)")

    if not genero_adeuado:
        print(f"- El género '{genero}' no está entre los preferidos (rock, pop, indie)")

    if not anio_adeuado:
        print(f"- El año {anio} es anterior a 2010")

```

Explicación: Este programa utiliza operadores lógicos (`and`), operadores de comparación (`<=`, `>`) y el operador de pertenencia (`in`) para evaluar si una canción cumple con los criterios establecidos. Las variables booleanas (`duracion_adeuada`, `genero_adeuado`, `anio_adeuado`) hacen el código más legible y permiten generar mensajes específicos sobre por qué una canción no cumple los requisitos.

Solución Ejercicio 4: Aritmética y Conversor de Unidades

```
# Mostramos el menú de opciones
print("Conversor de Unidades")
print("1. Kilómetros a Millas")
print("2. Millas a Kilómetros")
print("3. Celsius a Fahrenheit")
print("4. Fahrenheit a Celsius")

# Solicitamos la opción al usuario
opcion = int(input("\nSeleccione una opción (1-4): "))

# Procesamos según la opción elegida
if opcion == 1:
    valor = float(input("Ingrese la distancia en kilómetros: "))
    if valor < 0:
        print("Error: La distancia no puede ser negativa.")
    else:
        resultado = valor * 0.621371
        print(f"{valor} kilómetros equivalen a {resultado:.2f} millas.")

elif opcion == 2:
    valor = float(input("Ingrese la distancia en millas: "))
    if valor < 0:
        print("Error: La distancia no puede ser negativa.")
    else:
        resultado = valor / 0.621371
        print(f"{valor} millas equivalen a {resultado:.2f} kilómetros.")

elif opcion == 3:
    valor = float(input("Ingrese la temperatura en grados Celsius: "))
    resultado = valor * 9/5 + 32
    print(f"{valor}°C equivalen a {resultado:.1f}°F.")

    # Comentario sobre la temperatura
    if valor < 0:
        print("Está bajo cero, ¡qué frío!")
    elif 15 <= valor <= 25:
        print("Está a temperatura ambiente, muy agradable.")
    else:
        print("Hace calor.")

elif opcion == 4:
    valor = float(input("Ingrese la temperatura en grados Fahrenheit: "))
    resultado = (valor - 32) * 5/9
    print(f"{valor}°F equivalen a {resultado:.1f}°C.")

    # Comentario sobre la temperatura
    if resultado < 0:
        print("Está bajo cero, ¡qué frío!")
    elif 15 <= resultado <= 25:
        print("Está a temperatura ambiente, muy agradable.")
```

```

    else:
        print("Hace calor.")

else:
    print("Opción no válida. Por favor seleccione un número del 1 al 4.")

```

Explicación: Este programa utiliza operaciones aritméticas para realizar conversiones entre unidades. La estructura `if-elif-else` maneja las diferentes opciones del menú y también valida los datos de entrada (como evitar distancias negativas). Las fórmulas de conversión se implementan directamente en el código, y se usan operadores como `*`, `/`, `+` y `-` para los cálculos.

Solución Ejercicio 5: Listas y Organizador de Biblioteca Digital

```

# Inicializamos la biblioteca con algunos libros
biblioteca = ["Don Quijote de la Mancha", "Cien años de soledad",
              "El principito", "Harry Potter", "1984"]

# Mostramos el menú
print("Biblioteca Digital")
print("1. Ver todos los libros")
print("2. Buscar un libro")
print("3. Agregar un libro")
print("4. Eliminar un libro")

# Solicitamos la opción al usuario
opcion = int(input("\nSeleccione una opción (1-4): "))

# Procesamos según la opción elegida
if opcion == 1:
    if len(biblioteca) == 0:
        print("La biblioteca está vacía.")
    else:
        print("\nLibros disponibles:")
        for i, libro in enumerate(biblioteca, 1):
            print(f"{i}. {libro}")

elif opcion == 2:
    busqueda = input("Ingrese el nombre del libro a buscar: ")
    encontrado = False

    for libro in biblioteca:
        if busqueda.lower() in libro.lower():
            print(f"Libro encontrado: {libro}")
            encontrado = True

    if not encontrado:
        print("No se encontró ningún libro con ese nombre.")

elif opcion == 3:
    nuevo_libro = input("Ingrese el nombre del nuevo libro: ")

```

```

# Verificamos si el libro ya existe (ignorando mayúsculas/minúsculas)
if any(libro.lower() == nuevo_libro.lower() for libro in biblioteca):
    print("Este libro ya existe en la biblioteca.")
else:
    biblioteca.append(nuevo_libro)
    print(f'"{nuevo_libro}" ha sido agregado a la biblioteca.')

# Verificamos si la biblioteca está llena
if len(biblioteca) > 10:
    print("¡Atención! Has alcanzado el límite de 10 libros de la versión gratuita.")

elif opcion == 4:
    if len(biblioteca) == 0:
        print("La biblioteca está vacía, no hay libros para eliminar.")
    else:
        print("\nLibros disponibles:")
        for i, libro in enumerate(biblioteca, 1):
            print(f"{i}. {libro}")

        indice = int(input("\nIngresa el número del libro a eliminar: ")) - 1

        if 0 <= indice < len(biblioteca):
            libro_eliminado = biblioteca.pop(indice)
            print(f'"{libro_eliminado}" ha sido eliminado de la biblioteca.')
        else:
            print("Número de libro inválido.")

else:
    print("Opción no válida. Por favor seleccione un número del 1 al 4.")

```

Explicación: Este programa utiliza una lista para almacenar los libros de la biblioteca. La función `enumerate()` permite mostrar los libros numerados. Los métodos de lista como `append()` y `pop()` se utilizan para agregar y eliminar elementos. La función `any()` con una expresión generadora permite verificar si un libro ya existe. La sentencia `if` se utiliza en múltiples contextos para validar datos y controlar el flujo del programa.