Taller de Manipulación de Listas en Python

Ejercicio 1: Suma de Elementos

Escribe un algoritmo que sume todos los elementos de una lista de números enteros.

```
lista = [1, 2, 3, 4, 5]
suma = sum(lista)
print("La suma de los elementos es:", suma)
```

Ejercicio 2: Producto de Elementos

Escribe un algoritmo que calcule el producto de todos los elementos de una lista de números enteros.

```
lista = [1, 2, 3, 4, 5]
producto = 1
for numero in lista:
    producto *= numero # producto = producto*numero
print ("El producto de los elementos es:", producto)
```

Ejercicio 3: Contar Elementos Específicos

Escribe un algoritmo que cuente cuántas veces aparece un número específico en una lista.

```
lista = [1, 2, 3, 4, 5, 3, 3]
numero_a_contar = 3
contador = lista.count(numero_a_contar)
print(f"El número {numero_a_contar} aparece {contador} veces en la lista.")
```

Ejercicio 4: Encontrar el Máximo y Mínimo

Escribe un algoritmo que encuentre el valor máximo y mínimo en una lista de números.

```
lista = [1, 2, 3, 4, 5]
maximo = max(lista)
minimo = min(lista)
```

```
print("El valor máximo es:", maximo)

print("El valor mínimo es:", minimo)
```

Ejercicio 5: Eliminar Elementos Duplicados

Escribe un algoritmo que elimine los elementos duplicados de una lista.

```
lista = [1, 2, 3, 4, 5, 3, 3]

lista_sin_duplicados = list(set(lista)) #

print("Lista sin duplicados:", lista_sin_duplicados)
```

Explicación como funciona set:

1. **Convertir la lista en un conjunto (set):** La función set(lista) convierte la lista en un conjunto. Los conjuntos en Python son colecciones desordenadas de elementos únicos, lo que significa que cualquier duplicado en la lista original será eliminado.
2. **Convertir el conjunto de nuevo en una lista (list):** La función **list(**set(lista)**)** convierte el conjunto resultante de nuevo en una lista. Esto es necesario porque algunas operaciones que se pueden realizar en listas no están disponibles para conjuntos.

Ejercicio 6: Intercambiar Primer y Último Elemento

Escribe un algoritmo que intercambie el primer y el último elemento de una lista.

```
lista = [1, 2, 3, 4, 5]

lista [0], lista [-1] = lista [-1], lista [0]

print ("Lista después de intercambiar el primer y último elemento:", lista)
```

explicación:

1. Acceso a los elementos: lista[0] accede al primer elemento de la lista y lista[-1] accede al último elemento de la lista. El índice -1 se refiere al último elemento en Python.
2. Asignación múltiple: Python permite la asignación múltiple, que en este caso se utiliza para intercambiar los valores de los elementos en una sola línea. La expresión lista[0], lista[-1] = lista[-1], lista[0] toma los valores actuales del primer y último elemento y los intercambia.

Ejercicio 7: Invertir una Sublista

Escribe un algoritmo que invierta una sublista dentro de una lista más grande.

```
lista = [1, 2, 3, 4, 5]

sublista = lista[1:4] #extraigo desde 1 a 4 [2,3,4]

sublista.reverse() #invierto  [4,3,2]

lista[1:4] = sublista #inserto [4,3,2]

print("Lista después de invertir la sublista:", lista) #resultado [1, 4, 3, 2, 5]
```

Ejercicio 8: Contar Elementos Pares e Impares

Escribe un algoritmo que cuente cuántos números pares e impares hay en una lista.

```
lista = [1, 2, 3, 4, 5]

pares = len([num for num in lista if num % 2 == 0])

impares = len([num for num in lista if num % 2 != 0])

print("Números pares:", pares)

print("Números impares:", impares)
```

Explicación:

1. [**num for num in lista if num % 2 == 0**]: Esta es una lista por comprensión (list comprehension) que recorre cada elemento (num) en lista y selecciona solo aquellos que son pares (donde num % 2 == 0).
2. len(...): La función len se usa para contar el número de elementos en la lista generada por la comprensión de listas.
3. Desglose de la comprensión de listas:
    a. num for num in lista: Itera sobre cada número en lista.
    b. if num % 2 == 0: Incluye num en la nueva lista solo si num es par.
    c. Para lista = [1, 2, 3, 4, 5], la comprensión de listas [num for num in lista if num % 2 == 0] genera [2, 4].
    d. Con len([2, 4]) cuento el resultado de la lista y da 2.
    e. Expresión: num (el valor que deseas incluir en la nueva lista)
    f. Bucle for: for num in lista (recorre cada elemento en la lista original)
    g. Condición if: if num % 2 == 0 (incluye el elemento en la nueva lista solo si cumple la condición)

Ejercicio 9: Concatenar Listas de Strings

Escribe un algoritmo que concatene (unir) todas las cadenas de una lista de strings en una sola cadena larga.

```
frutas = ["manzana", "banana", "cereza", "dátil", "fresa"]

cadena_concatenada = ''.join(frutas)

print("Cadena concatenada:", cadena_concatenada) #el resultado es una palabra manzanabananacerezadátilfresa
```

Ejercicio 10: Filtrar Números Mayores a un Valor Dado

Escribe un algoritmo que filtre los números mayores a un valor dado de una lista.

```
lista = [1, 2, 3, 4, 5]

valor_dado = 3

filtrados = [num for num in lista if num > valor_dado]

print("Números mayores a", valor_dado, ":", filtrados) #resultado >3 son [4,5]
```

Ejercicio 11: Encontrar la Suma de una Sublista

Escribe un algoritmo que encuentre la suma de los elementos en una sublista específica.

```
lista = [1, 2, 3, 4, 5]
sublista = lista[1:4]  # Sublista del índice 1 al 3
suma_sublista = sum(sublista)
print("La suma de la sublista es:", suma_sublista) #resultado 9
```

Ejercicio 12: Calcular el Promedio de una Lista

Escribe un algoritmo que calcule el promedio de los elementos de una lista.

```
lista = [1, 2, 3, 4, 5]
promedio = sum(lista) / len(lista)
print("El promedio de la lista es:", promedio)
```

Ejercicio 13: Filtrar Números Pares

Escribe un algoritmo que filtre y devuelva solo los números pares de una lista.

```
lista = [1, 2, 3, 4, 5]
numeros_pares = [num for num in lista if num % 2 == 0]
print("Números pares:", numeros_pares)
```

Ejercicio 14: Generar una Lista de Cuadrados

Escribe un algoritmo que genere una lista de los cuadrados de los números del 1 al 10.

```
cuadrados = [num**2 for num in range(1, 11)]
print("Lista de cuadrados:", cuadrados)
```

Ejercicio 15: Comprobar la Existencia de un Elemento

Escribe un algoritmo que compruebe si un número dado existe en una lista y devuelva True o False.

```
lista = [1, 2, 3, 4, 5]
numero_a_comprobar = 3
existe = numero_a_comprobar in lista
print(f"¿El número {numero_a_comprobar} existe en la lista?:", existe)
```

Ejercicio 16: Crear una Lista de Números Negativos

Escribe un algoritmo que genere una lista de números negativos del -1 al -10.

```
negativos = [-num for num in range(1, 11)]
print("Lista de números negativos:", negativos)
```

Ejercicio 17: Concatenar Dos Listas

Escribe un algoritmo que concatene dos listas y muestre la lista resultante.

```
lista1 = [1, 2, 3]
lista2 = [4, 5, 6]
lista_concatenada = lista1 + lista2
print("Lista concatenada:", lista_concatenada)
```

Ejercicio 18: Generar una Lista de Pares Ordenados

Escribe un algoritmo que genere una lista de pares ordenados (tuplas) a partir de dos listas de igual longitud.

```
lista1 = [1, 2, 3]
lista2 = ['a', 'b', 'c']
pares_ordenados = list(zip(lista1, lista2))
print("Lista de pares ordenados:", pares_ordenados) # los pares ordenados son [(1, 'a'), (2, 'b'), (3, 'c')]
```

Ejercicio 19: Generar una Lista de los Primeros N Números Naturales

Escribe un algoritmo que genere una lista con los primeros N números naturales, donde N es un valor dado.

```
N = 10
primeros_n_numeros = list(range(1, N+1)) #también puedes hacer una tupla primeros_n_numeros = tupla(range(1, N+1))
print("Primeros", N, "números naturales:", primeros_n_numeros)
```

Ejercicio 20: Crear una Lista de Palabras en Mayúsculas

Escribe un algoritmo que convierta todas las palabras de una lista de strings a mayúsculas, luego a minúsculas, luego con mayúscula inicial y por último que invierta cada palabra.

```
palabras = ["HoLa", "MUNDO", "pYThon"]

palabras_mayusculas = [palabra.upper() for palabra in palabras]

print("Palabras en mayúsculas:", palabras_mayusculas)


palabras = ["HoLa", "MUNDO", "pYThon"]

palabras_minusculas = [palabra.lower() for palabra in palabras]

print("Palabras en mayúsculas:", palabras_minusculas)


palabras = ["Hola", "MUNDO", "pYThon"]

palabras_capitalizadas = [palabra.capitalize() for palabra in palabras]

print("Palabras con mayúscula inicial:", palabras_capitalizadas)


#capitalize() y title() hacen lo mismo.

palabras = ["Hola", "MUNDO", "pYThon"]

palabras_titulo = [palabra.title() for palabra in palabras]

print("Palabras con formato de título:", palabras_titulo) #hace lo mismo que capitalize


palabras = ["Hola", "MUNDO", "pYThon"]

palabras_swapcase = [palabra.swapcase() for palabra in palabras]

print("Palabras con cambio de mayúsculas y minúsculas:", palabras_swapcase)
```