

Taller de Manipulación de Listas en Python

Ejercicio 1: Sumar Elementos Pares

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

```
lista = [1, 2, 3, 4, 5, 6]

suma_pares = sum (num for num in lista if num % 2 == 0)

print("La suma de los elementos pares es:", suma_pares)
```

Ejercicio 2: Producto de Elementos Impares

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

```
lista = [1, 2, 3, 4, 5, 6]

producto_impares = 1

for num in lista:
    if num % 2 != 0:
        producto_impares *= num

print("El producto de los elementos impares es:", producto_impares)
```

Ejercicio 3: Encontrar Elementos Comunes

Escribe un algoritmo que encuentre los elementos comunes entre dos listas y los muestre.

```
lista1 = [1, 2, 3, 4, 5]

lista2 = [4, 5, 6, 7, 8]

comunes = list(set(lista1) & set(lista2))

print("Elementos comunes:", comunes)
```

Ejercicio 4: Eliminar los Primeros N Elementos

Escribe un algoritmo que elimine los primeros N elementos de una lista.

```
lista = [1, 2, 3, 4, 5, 6, 7, 8]
```

```
N = 3
```

```
lista = lista[N:]
```

```
print("Lista después de eliminar los primeros", N, "elementos:", lista)
```

Ejercicio 5: Crear una Lista con Elementos Elevados al Cubo

Escribe un algoritmo que genere una lista donde cada elemento sea el cubo de los números del 1 al 10.

```
cubos = [num**3 for num in range(1, 11)]
```

```
print("Lista de cubos:", cubos)
```

Ejercicio 6: Ordenar Lista de Palabras por Longitud

Escribe un algoritmo que ordene una lista de palabras por su longitud.

```
palabras = ["python", "es", "un", "lenguaje", "de", "programación"]
```

```
palabras_ordenadas = sorted(palabras, key=len)
```

```
print("Palabras ordenadas por longitud:", palabras_ordenadas)
```

Ejercicio 7: Remover Elementos Negativos

Escribe un algoritmo que elimine todos los elementos negativos de una lista de números.

```
lista = [1, -2, 3, -4, 5, -6]
```

```
lista_positiva = [num for num in lista if num >= 0]
```

```
print("Lista sin elementos negativos:", lista_positiva)
```

Ejercicio 8: Encontrar el Segundo Mayor Número

Escribe un algoritmo que encuentre el segundo mayor número en una lista de números.

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

```
mayor = max(lista)
```

```
lista_sin_mayor = [num for num in lista if num != mayor]
```

```
segundo_mayor = max(lista_sin_mayor)
```

```
print("El segundo mayor número es:", segundo_mayor)
```

Ejercicio 9: Intercalar Elementos de dos Listas

Escribe un algoritmo que intercale los elementos de dos listas de igual longitud.

```
lista1 = [1, 3, 5]
lista2 = [2, 4, 6]
lista_intercalada = [None]*(len(lista1) + len(lista2))
lista_intercalada[0::2] = lista1
lista_intercalada[1::2] = lista2
print("Lista intercalada:", lista_intercalada)
```

Ejercicio 10: Dividir una Lista en dos Partes

Escribe un algoritmo que divida una lista en dos partes iguales. Si la lista tiene un número impar de elementos, la primera parte debe tener un elemento más que la segunda.

```
lista = [1, 2, 3, 4, 5]
mitad = len(lista) // 2
if len(lista) % 2 != 0:
    mitad += 1
lista1 = lista[:mitad]
lista2 = lista[mitad:]
print("Primera parte:", lista1)
print("Segunda parte:", lista2)
```

Ejercicio 11: Crear una Lista de Factoriales

Escribe un algoritmo que genere una lista de factoriales de los números del 1 al 5.

```
import math
factoriales = [math.factorial(num) for num in range(1, 6)]
print("Lista de factoriales:", factoriales)
```

Ejercicio 12: Filtrar Palabras que Contengan una Subcadena

Escribe un algoritmo que filtre una lista de palabras y devuelva solo las palabras que contienen una subcadena específica.

```
palabras = ["hola", "mundo", "holanda", "python", "holístico"]  
  
subcadena = "hol"  
  
palabras_filtradas = [palabra for palabra in palabras if subcadena in palabra]  
  
print("Palabras que contienen 'hol':", palabras_filtradas)
```

Ejercicio 13: Duplicar Elementos de una Lista

Escribe un algoritmo que duplique cada elemento de una lista de números.

```
lista = [1, 2, 3, 4, 5]  
  
lista_duplicada = [num*2 for num in lista]  
  
print("Lista con elementos duplicados:", lista_duplicada)
```

Ejercicio 14: Alternar Mayúsculas y Minúsculas en una Lista de Strings

Escribe un algoritmo que alterne mayúsculas y minúsculas en una lista de strings.

```
palabras = ["python", "es", "genial"]  
  
palabras_alternadas = [palabra.upper() if i % 2 == 0 else palabra.lower() for i, palabra in enumerate(palabras)]  
  
print("Palabras con alternancia de mayúsculas y minúsculas:", palabras_alternadas)
```

Ejercicio 15: Crear una Lista de Números Primos

Escribe un algoritmo que genere una lista de números primos menores a 20.

```
primos = []

for num in range(2, 20):

    es_primo = True

    if num <= 1:

        es_primo = False

    for i in range(2, num):

        if num % i == 0:

            es_primo = False

            break

    if es_primo:

        primos.append(num)

print("Lista de números primos menores a 20:", primos)
```

Ejercicio 16: Crear una Lista de Números Fibonacci

Escribe un algoritmo que genere una lista de los primeros 10 números de Fibonacci.

```
fibonacci = [0, 1]

for i in range(2, 10):

    siguiente = fibonacci[-1] + fibonacci[-2]

    fibonacci.append(siguiente)

print("Lista de los primeros 10 números de Fibonacci:", fibonacci)
```

Ejercicio 17: Remover Elementos por Índice

Escribe un algoritmo que elimine los elementos en posiciones impares de una lista.

```
lista = [1, 2, 3, 4, 5, 6, 7, 8]

lista_pares = [num for i, num in enumerate(lista) if i % 2 == 0]

print("Lista sin elementos en posiciones impares:", lista_pares)
```

Ejercicio 18: Crear una Lista de Potencias

Escribe un algoritmo que genere una lista de los números del 1 al 5 elevados a la cuarta potencia.

```
potencias = [num**4 for num in range(1, 6)]

print("Lista de potencias a la cuarta:", potencias)
```

Ejercicio 19: Crear una Lista de Elementos en Orden Inverso

Escribe un algoritmo que invierta el orden de los elementos en una lista de strings.

```
palabras = ["python", "es", "genial"]

palabras_invertidas = palabras[::-1]

print("Lista en orden inverso:", palabras_invertidas)
```

Ejercicio 20: Remover Palabras Cortas

Escribe un algoritmo que elimine las palabras de menos de 4 caracteres de una lista de strings.

```
palabras = ["python", "es", "genial", "muy", "bueno"]
```

```
palabras_largas = [palabra for palabra in palabras if len(palabra) >= 4]
```

```
print("Palabras con 4 o más caracteres:", palabras_largas)
```