

Hacer en los siguientes ejercicios

Arrays unidimensionales y matrices

Ejercicio 1: Función Sencilla para Crear una Matriz

Escribe una función que cree una matriz de tamaño $n \times n$ con números consecutivos.

```
def crear_matriz(n):  
    return [[j + i*n for j in range(1, n+1)] for i in range(n)]  
  
if __name__ == '__main__':  
    n = 3  
  
    matriz = crear_matriz(n)  
  
    for fila in matriz:  
        print(fila)
```

Ejercicio 2: Función para Sumar Filas de una Matriz

```
def sumar_filas(matriz):  
    return [sum(fila) for fila in matriz]  
  
if __name__ == '__main__':  
    matriz = [  
        [1, 2, 3],  
        [4, 5, 6],  
        [7, 8, 9]  
    ]  
  
    suma_filas = sumar_filas(matriz)  
  
    print(f"Suma de los elementos de cada fila: {suma_filas}")
```

Ejercicio 3: Función para Sumar Columnas de una Matriz

```
def sumar_columnas(matriz):  
    return [sum(fila[i] for fila in matriz) for i in range(len(matriz[0]))]  
  
if __name__ == '__main__':  
    matriz = [  
        [1, 2, 3],  
        [4, 5, 6],  
        [7, 8, 9]  
    ]  
    suma_columnas = sumar_columnas(matriz)  
    print(f'Suma de los elementos de cada columna: {suma_columnas}')
```

Ejercicio 4: Función para Transponer una Matriz

```
def transponer_matriz(matriz):  
    return [[fila[i] for fila in matriz] for i in range(len(matriz[0]))]  
  
if __name__ == '__main__':  
    matriz = [  
        [1, 2, 3],  
        [4, 5, 6],  
        [7, 8, 9]  
    ]  
    matriz_transpuesta = transponer_matriz(matriz)  
    for fila in matriz_transpuesta:  
        print(fila)
```

Ejercicio 5: Función para Multiplicar Dos Matrices

```
def multiplicar_matrices(A, B):  
    resultado = [[sum(a*b for a, b in zip(A_fila, B_columna)) for B_columna in zip(*B)] for A_fila in A]  
    return resultado  
  
if __name__ == '__main__':  
    A = [  
        [1, 2, 3],  
        [4, 5, 6],  
        [7, 8, 9]  
    ]  
    B = [  
        [9, 8, 7],  
        [6, 5, 4],  
        [3, 2, 1]  
    ]  
    producto = multiplicar_matrices(A, B)  
    for fila in producto:  
        print(fila)
```

Ejercicio 6: Función para Filtrar Números Pares de una lista

```
def filtrar_pares(lista):  
    return [num for num in lista if num % 2 == 0]  
  
if __name__ == '__main__':  
    lista = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
    pares = filtrar_pares(lista)  
    print(f"Números pares: {pares}")
```

Ejercicio 7: Función con Parámetros para Encontrar Máximo y Mínimo en una lista

```
def max_min(lista):  
    return max(lista), min(lista)  
  
if __name__ == '__main__':  
    lista = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
    maximo, minimo = max_min(lista)  
    print(f"El número máximo es: {maximo}")  
    print(f"El número mínimo es: {minimo}")
```

Ejercicio 8: Función para Crear una Lista de Números Primos < a 20

```
def numeros_primos(n):  
    primos = []  
    for num in range(2, n):  
        es_primo = True  
        for i in range(2, int(num ** 0.5) + 1):  
            if num % i == 0:  
                es_primo = False  
                break  
        if es_primo:  
            primos.append(num)  
    return primos  
  
if __name__ == '__main__':  
    primos = numeros_primos(20)  
    print(f"Números primos menores a 20: {primos}")
```

Ejercicio 9: Función para Calcular Promedio de una Lista

```
def calcular_promedio(lista):  
    return sum(lista) / len(lista)  
  
if __name__ == '__main__':  
    lista = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
    promedio = calcular_promedio(lista)  
    print(f"El promedio de los números es: {promedio}")
```

Ejercicio 10: Función para Encontrar la Mediana de una lista

```
def encontrar_mediana(lista):  
    lista_ordenada = sorted(lista)  
    n = len(lista_ordenada)  
    mitad = n // 2  
    if n % 2 == 0:  
        return (lista_ordenada[mitad - 1] + lista_ordenada[mitad]) / 2  
    else:  
        return lista_ordenada[mitad]  
  
if __name__ == '__main__':  
    lista = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
    mediana = encontrar_mediana(lista)  
    print(f"La mediana de los números es: {mediana}")
```

Ejercicio 11: Función para Contar Palabras en una Cadena de texto

```
def contar_palabras(cadena):  
    palabras = cadena.split()  
    return len(palabras)
```

```
if __name__ == '__main__':  
    cadena = " es un lenguaje de programación genial"  
    num_palabras = contar_palabras(cadena)  
    print(f"El número de palabras es: {num_palabras}")
```

Ejercicio 12: Función para Crear una Lista de Números Aleatorios entre [a y b]

```
import random  
  
def generar_numeros_aleatorios(n, a, b):  
    return [random.randint(a, b) for _ in range(n)]  
  
if __name__ == '__main__':  
    n = 10  
    a = 1  
    b = 100  
    numeros_aleatorios = generar_numeros_aleatorios(n, a, b)  
    print(f"Lista de números aleatorios: {numeros_aleatorios}")
```

Ejercicio 13: Función para Encontrar Elementos Únicos

```
def encontrar_unicos(lista):  
    return list(set(lista))  
  
if __name__ == '__main__':  
    lista = [1, 2, 2, 3, 4, 4, 5]  
    unicos = encontrar_unicos(lista)  
    print(f"Elementos únicos: {unicos}")
```

Ejercicio 14: Función para Calcular la Suma de Diagonales de una Matriz

Calcule la suma de las diagonales principal y secundaria de una matriz cuadrada.

```
def suma_diagonales(matriz):  
    n = len(matriz)  
    diagonal_principal = sum(matriz[i][i] for i in range(n))  
    diagonal_secundaria = sum(matriz[i][n - 1 - i] for i in range(n))  
    return diagonal_principal, diagonal_secundaria  
  
if __name__ == '__main__':  
    matriz = [  
        [1, 2, 3],  
        [4, 5, 6],  
        [7, 8, 9]  
    ]  
    diag_principal, diag_secundaria = suma_diagonales(matriz)  
    print(f"Suma de la diagonal principal: {diag_principal}")  
    print(f"Suma de la diagonal secundaria: {diag_secundaria}")
```

Ejercicio 15: Función para Crear una Lista de 10 Números Fibonacci

```
def numeros_fibonacci(n):  
    fibonacci = [0, 1]  
    for i in range(2, n):  
        siguiente = fibonacci[-1] + fibonacci[-2]  
        fibonacci.append(siguiente)  
    return fibonacci
```

```
if __name__ == '__main__':  
    fibonacci = numeros_fibonacci(10)  
    print(f"Lista de los primeros 10 números de Fibonacci: {fibonacci}")
```