

EJERCICIO 91

Dada una lista de edades, utiliza una función lambda para filtrar las edades que sean mayores o iguales a 18 años (es decir, filtrar adultos).

EJERCICIO 92

Dada una lista de puntuaciones de exámenes, utiliza una función lambda para clasificar cada puntuación en "Aprobado" o "Reprobado". Una puntuación de 60 o más es considerada "Aprobado".

EJERCICIO 93

Dada una lista de precios de productos, utiliza una función lambda para aplicar un descuento del 10% a los productos cuyo precio sea mayor a 100. Para los productos cuyo precio sea menor o igual a 100, aplicar un descuento del 5%.

EJERCICIO 94

Dada una lista de temperaturas en grados Celsius, utiliza una función lambda para convertir estas temperaturas a grados Fahrenheit. Usa la fórmula $F = C \times 1.8 + 32$. Luego, clasifica las temperaturas en Fahrenheit como "Frío" (menos de 50°F), "Templado" (entre 50°F y 85°F) o "Calor" (más de 85°F).

EJERCICIO 95

Dada una lista de ingresos anuales, utiliza una función lambda para clasificar estos ingresos en "Bajo" (menos de 30,000), "Medio" (entre 30,000 y 70,000) o "Alto" (más de 70,000).

EJERCICIO 96

Tienes una lista de diccionarios que representan diferentes productos. Cada producto tiene un nombre, precio y cantidad en stock. Filtra la lista para obtener solo los productos cuyo precio sea mayor a 100 y cuya cantidad en stock sea mayor a 10. Luego, ordena los productos filtrados por precio en orden descendente usando una función lambda.

```
productos = [  
    {'nombre': 'Producto A', 'precio': 150, 'cantidad': 30},  
    {'nombre': 'Producto B', 'precio': 85, 'cantidad': 50},  
    {'nombre': 'Producto C', 'precio': 120, 'cantidad': 5},  
    {'nombre': 'Producto D', 'precio': 200, 'cantidad': 20},  
    {'nombre': 'Producto E', 'precio': 75, 'cantidad': 40},  
    {'nombre': 'Producto F', 'precio': 110, 'cantidad': 15}  
]
```

EJERCICIO 97

Dada una lista de diccionarios que contienen datos de empleados (nombre, salario y departamento), calcula un nuevo campo 'impuesto' que sea el 10% del salario, y crea una nueva lista de diccionarios que solo contenga el nombre y el impuesto de cada empleado. Usa funciones lambda.

```
empleados = [  
    {'nombre': 'Juan', 'salario': 50000, 'departamento': 'Ventas'},  
    {'nombre': 'Ana', 'salario': 60000, 'departamento': 'Marketing'},  
    {'nombre': 'Luis', 'salario': 75000, 'departamento': 'IT'},  
    {'nombre': 'Marta', 'salario': 45000, 'departamento': 'RRHH'}  
]
```

EJERCICIO 98

Dado un DataFrame con datos de estudiantes (nombre, matemáticas, ciencias, inglés), aplica una función lambda que calcule la media de las tres materias para cada estudiante y añade una nueva columna 'promedio'.

```
import pandas as pd  
  
datos = {  
    'nombre': ['Carlos', 'Lucía', 'Marcos', 'Ana'],  
    'matemáticas': [90, 85, 78, 92],  
    'ciencias': [88, 90, 80, 85],  
    'inglés': [85, 88, 90, 86]  
}  
  
df = pd.DataFrame(datos)
```

EJERCICIO 99

Usa pandas para crear un DataFrame de ventas que incluya columnas de producto, cantidad y precio. Agrupa por producto y calcula el ingreso total (cantidad * precio) usando funciones lambda.

```
import pandas as pd  
  
datos_ventas = {  
    'producto': ['A', 'B', 'A', 'C', 'B', 'A', 'C'],  
    'cantidad': [10, 5, 20, 15, 10, 5, 10],  
    'precio': [100, 200, 100, 150, 200, 100, 150]  
}  
  
df_ventas = pd.DataFrame(datos_ventas)
```

EJERCICIO 100

Dado un DataFrame con datos de sensores (sensor_id, temperatura, humedad, presion), filtra las filas donde la temperatura es mayor a 25 y la presión es menor a 1010, y selecciona solo las columnas 'sensor_id' y 'temperatura' usando una función lambda.

```
import pandas as pd

datos_sensores = {
    'sensor_id': [1, 2, 3, 4, 5],
    'temperatura': [22, 27, 30, 24, 26],
    'humedad': [55, 65, 60, 58, 63],
    'presion': [1012, 1008, 1005, 1011, 1009]
}

df_sensores = pd.DataFrame(datos_sensores)
```

EJERCICIO 101

Dado un DataFrame con datos de rendimiento de empleados (nombre, proyectos_completados, horas_trabajadas, satisfacción), calcula una métrica personalizada que sea una combinación ponderada de las tres columnas. La fórmula de la métrica es:

$$\text{métrica} = (0.5 \times \text{proyectos_completados}) + (0.3 \times \text{horas_trabajadas}) + (0.2 \times \text{satisfacción})$$

Añade una nueva columna 'métrica' al DataFrame.

```
import pandas as pd

datos_rendimiento = {
    'nombre': ['Pedro', 'María', 'Luis', 'Ana'],
    'proyectos_completados': [5, 7, 3, 8],
    'horas_trabajadas': [40, 35, 50, 30],
    'satisfacción': [80, 90, 70, 85]
}

df_rendimiento = pd.DataFrame(datos_rendimiento)
```

EJERCICIO 102

Dado un DataFrame con datos de ventas (producto, ventas_mensuales, ventas_anuales), normaliza las columnas 'ventas_mensuales' y 'ventas_anuales' al rango [0, 1] usando una función lambda. La normalización se realiza usando la fórmula:

$$\text{normalizado} = \frac{x - \min}{\max - \min}.$$

```
import pandas as pd

datos_ventas = {
    'producto': ['A', 'B', 'C', 'D'],
    'ventas_mensuales': [150, 200, 100, 250],
    'ventas_anuales': [1800, 2400, 1200, 3000]
}

df_ventas = pd.DataFrame(datos_ventas)
```

EJERCICIO 103

Dado un DataFrame con datos de acciones (fecha, cierre, volumen), crea un nuevo DataFrame que contenga la fecha, el precio de cierre y la diferencia porcentual diaria del precio de cierre usando funciones lambda.

```
import pandas as pd

datos_acciones = {
    'fecha': ['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04'],
    'cierre': [150, 155, 153, 160],
    'volumen': [1000, 1100, 1050, 1150]
}

df_acciones = pd.DataFrame(datos_acciones)
```

EJERCICIO 104

Dado un DataFrame con datos de transacciones (cliente, monto, fecha), agrupa por cliente y calcula el total de gastos, el número de transacciones y el gasto promedio por transacción usando funciones lambda.

```
import pandas as pd

datos_transacciones = {
    'cliente': ['Juan', 'Ana', 'Luis', 'Ana', 'Juan', 'Luis', 'Ana'],
    'monto': [150, 200, 100, 250, 300, 150, 100],
    'fecha': ['2023-01-01', '2023-01-02', '2023-01-03',
             '2023-01-04', '2023-01-05', '2023-01-06', '2023-01-07']
}

df_transacciones = pd.DataFrame(datos_transacciones)
```

EJERCICIO 105

Dado un DataFrame con datos de productos (producto, precio, cantidad_vendida), filtra las filas donde el precio es mayor que el promedio y la cantidad vendida es mayor que 50 usando funciones lambda.

```
import pandas as pd

datos_productos = {
    'producto': ['A', 'B', 'C', 'D', 'E'],
    'precio': [150, 85, 120, 200, 75],
    'cantidad_vendida': [30, 50, 5, 20, 40]
}

df_productos = pd.DataFrame(datos_productos)
```