

```
In [3]: # Ejercicio N°1:
# Escribir un programa que pregunte al usuario por las ventas de los últimos 5 días, y muestre por pantalla una serie con los datos de las ventas por producto.

import pandas as pd #Importamos pandas

inicio = int(input('Introduce el día inicial de ventas: '))#Creamos un int para ingresar el día inicial de ventas
fin = int(input('Introduce el día final de ventas: ')) #Terminamos el int con el día final de ventas
producto = {} #Creamos diccionario vacío para que itere el for
numdeunidades = {} #Creamos diccionario vacío para que itere el for
precio = {} #Creamos diccionario vacío para que itere el for
for i in range(inicio, fin+1):#Con este for tenemos iteraciones desde el dato inicial al final
    producto[i] = str(input('Introduce el nombre del producto vendido el día ' + str(i) + ': ')) #Aquí unimos un texto con otro str
    numdeunidades[i] = float(input('Introduce el número de unidades vendidas de ' + producto[i] + ': ')) #Aquí unimos un texto con un float
    precio[i] = float(input('Introduce el precio unitario de ' + producto[i] + ': ')) #Aquí unimos un texto con un float
producto = pd.Series(producto) #Tenemos el producto como una serie
DFproducto = pd.DataFrame(producto) #Convertimos la serie producto en un DataFrame

numdeunidades = pd.Series(numdeunidades) #Tenemos el numdeunidades como una serie
DFnumdeunidades = pd.DataFrame(numdeunidades) #Convertimos la serie numunidadea en un DataFrame

precio = pd.Series(precio) #Tenemos el precio como una serie
DFprecio = pd.DataFrame(precio) #Convertimos la serie precio en un DataFrame

print (pd.concat([DFproducto, DFnumdeunidades, DFprecio], axis = 1, sort= True)) #Imprimimos la tabla sin el 95% de descuento
pd.concat([DFproducto, DFnumdeunidades, DFprecio*0.95], axis = 1, sort= True) #Imprimimos la tabla con el 95% de descuento

-----

#Ejercicio 2
#Escribir un programa que reciba un diccionario con los ingresos mensuales de los clientes de una empresa, y devuelva una serie con el ingreso mínimo, máximo y

import pandas as pd #Importamos pandas

def ingreso_mensual(mensual): #Definimos una función ingreso_mensual con el parámetro mensual
    mensual = pd.Series(mensual) #Tenemos el mensual como una serie
    estadisticos = pd.Series([mensual.min(), mensual.max(), mensual.mean(), mensual.std()], index=['Min', 'Max', 'Media', 'Desviación típica']) #Tenemos los es
    return estadisticos #Nos retorna la serie estadístico con un index

mensual = {'Juan':500, 'María':650, 'Pedro':400, 'Carmen': 850, 'Luis': 500} #Diccionario mensual con sus key y sus valores de los mensuales
print(ingreso_mensual(mensual)) #Imprime la función

-----

#Ejercicio 3
#Escribir programa que genere y muestre por pantalla un DataFrame con los datos de la tabla siguiente:
# Mes Ventas Gastos
#0 Enero 30500 22000
#1 Febrero 35600 23450
#2 Marzo 28300 18100
#3 Abril 33900 35700
#4 Mayo 42500 32450

# A continuación, generar nuevas columnas:
# 1. Columna 1 = Gasto/Ventas*100.
# 2. Columna 2 = Si Ventas>= 30000, imprimir "Meta Superada",
# caso contrario imprimir "Meta no Superada".
# 3. Columna 3 = Si Gasto>Ventas, calcular el sobre gasto.
# 4. Obtener el total de las ventas de los últimos 3 meses
# 5. Obtener el total de gastos de febrero a abril

import pandas as pd #Importamos pandas

#Diccionario con los meses y sus valores de ventas y gastos
db = {
    'Mes':['Enero', 'Febrero', 'Marzo', 'Abril', 'Mayo'],
    'Ventas':[30500, 35600, 28300, 33900, 42500],
    'Gastos':[22000, 23450, 18100, 35700, 32450],
}

#Función generar_data_well con parámetro datos, luego guardamos los keys por separado
def generar_data_well(datos):
    months = datos["Mes"]
    ventas = datos["Ventas"]
    gastos = datos["Gastos"]

#Creamos unas listas vacías de las columnas para luego llenarlas con información
Columna1ListTemp = []
Columna2ListTemp = []
Columna3ListTemp = []
#Creamos un for que recorra en los meses
for index in range(len(months)):
    gasto = gastos[index]
    venta = ventas[index]
    columna1 = gasto / venta * 100 #Hacemos las operaciones que nos pide en la columna 1
    Columna1ListTemp.append(columna1)

    if venta >= 30000: #Con el if creamos la condición que nos pide en la columna 2
        Columna2ListTemp.append('Meta superada')
    else:
        Columna2ListTemp.append('Meta no superada')

    if gasto > venta: #Con el if creamos la condición que nos pide en la columna 3
        Columna3ListTemp.append(gasto - venta)
    else:
        Columna3ListTemp.append('') #Con esto para que por si acaso se aumente un espacio en la tercera columna en caso los index de las demás columnas no

#Llenamos las listas con la información de las columnas
datos["Columna 1"] = Columna1ListTemp
datos["Columna 2"] = Columna2ListTemp
datos["Columna 3"] = Columna3ListTemp
return datos

data_compleja = generar_data_well(db)
contabilidad = pd.DataFrame(data_compleja) #Convertimos en un DataFrame
print(contabilidad)
print("\n") #Creamos un interlineado, espacio.

#Obtenemos el total de las ventas de los últimos 3 meses
last_months = contabilidad.iloc[-3:] #Aquí seleccionamos los últimos 3 meses
total_ventas = last_months["Ventas"].sum()
print("Total de ventas de los últimos 3 meses: " + str(total_ventas))

#Obtenemos el total de gastos de febrero a abril
febrero_hasta_abril = contabilidad[1:4] #Aquí seleccionamos desde febrero hasta abril
total_febrero_hasta_abril = febrero_hasta_abril["Gastos"].sum()
print("Total de gastos de febrero a abril: " + str(total_febrero_hasta_abril))

-----

#Ejercicio 4
#Con el siguiente fichero Data - Riesgos.csv contiene información de riesgo de clientes de un banco. Escribir un programa con los siguientes requisitos:
#1. Generar un DataFrame con los datos del fichero.
#2. Mostrar por pantalla las dimensiones del DataFrame, el número de datos que contiene, los nombres de sus columnas y filas, los tipos de datos de las columnas y filas.
#3. Extraer las 100 primeras columnas y almacenarlas en un contenedor.

import pandas as pd #Importamos pandas

#Cargamos la data desde nuestra PC
from google.colab import files
files.upload()

#Leemos el documento en formato excel no en csv.
data_frame = pd.read_excel('Data - Riesgos.xlsx')
columns_name = " - ".join(data_frame.columns.tolist()) #Convertimos el DataFrame en una lista

#Aquí vamos uniendo los str con información del DataFrame
print("Nombre columnas: " + columns_name)
print("Filas: " + str(data_frame.shape[0]) + " - Columnas: " + str(data_frame.shape[1]))
print("Tipos de datos: ")
print(data_frame.dtypes)
print("\n")

#Trabajamos con las 10 primeras filas
first_row = data_frame.iloc[:10]
print("10 primeras filas: ")
print(first_row)
print("\n") #Creamos un interlineado, espacio.

#Trabajamos con las 10 últimas filas
last_row = data_frame.iloc[-10:]
print("10 últimas filas: ")
print(last_row)
print("\n") #Creamos un interlineado, espacio.

#Verificamos si tiene datos perdidos/nulos
print("Datos perdidos/nulos: ")
print(data_frame.isnull().sum())
print("\n") #Creamos un interlineado, espacio.

#Extraemos las 100 primeras columnas y las almacenamos en un contenedor.
get_rows = data_frame.iloc[:100]
container = get_rows.values.tolist()
```