PROGRAMACIÓN EN EXCEL PYTHON O.O CON EXCEL

Formas de Programación en Excel

Excel permite la programación y automatización mediante varias formas y herramientas. A continuación, se presentan las principales formas de programación en Excel.

1. Funciones y Fórmulas Integradas de Excel

Excel dispone de una amplia variedad de funciones predefinidas que permiten realizar cálculos, búsquedas y análisis de datos sin escribir código.

Ejemplos:

- =SUMA(A1:A10): Suma los valores del rango A1:A10.
- =SI(A1 >100, "Mayor que 100", "Menor o igual a 100"): Evalúa una condición y retorna un valor.

2. Macros y VBA (Visual Basic for Applications)

VBA es un lenguaje de programación integrado en Excel que permite crear macros y automatizar tareas. Es útil para automatización de tareas repetitivas, creación de funciones personalizadas e interacción con otros programas.

Ejemplo:

```
Sub MensajeHola()
    MsgBox "¡Hola, mundo!"
End Sub
```

healvare@utp.edu.co

3. Power Query

Power Query es una herramienta que permite la extracción, transformación y carga de datos (ETL) sin escribir código. Es ideal para importar, limpiar y combinar datos de diversas fuentes.

4. Power Pivot y DAX

Power Pivot permite la creación de modelos de datos avanzados, mientras que DAX (Data Analysis Expressions) es un lenguaje de fórmulas para cálculos complejos.

Ejemplo DAX:

 =SUM(Sales[Amount]): Suma todos los valores de la columna Amount de la tabla Sales.

5. Power BI Integration

Excel se puede integrar con Power BI para crear informes y dashboards interactivos basados en datos de Excel. Es ideal para visualización avanzada de datos y creación de informes dinámicos.

6. Office Scripts (Excel Online)

Office Scripts permite escribir scripts en JavaScript o TypeScript para automatizar tareas en Excel Online. Es una alternativa moderna a VBA.

7. Complementos y Add-Ins

Se pueden desarrollar complementos personalizados en VBA, JavaScript o .NET para extender las funcionalidades de Excel, como funciones personalizadas o herramientas de análisis avanzado.

8. Python en Excel

Desde 2024, Excel permite la integración de Python, lo que abre posibilidades para análisis avanzado de datos, machine learning y visualización de datos.

9. ¿Cuál elegir?

La elección del método de programación en Excel depende del tipo de tarea a realizar. VBA sigue siendo la opción más extendida para automatización básica, mientras que Python y Power BI ofrecen soluciones avanzadas para análisis de datos y visualización.

- Si el objetivo es automatizar tareas simples, VBA es una buena opción.
- Si se requiere análisis avanzado de datos, Power Query y DAX son recomendados.
- Para crear dashboards e informes interactivos, Power BI es ideal.
- Si se necesita análisis de datos y machine learning, Python es el camino.
- Si se trabaja en Excel Online, Office Scripts es el enfoque moderno.

Ejemplos de Python en Excel:

Primer ejemplo de Python en Excel

Como entrar a Excel:

- 1. Entrar a Excel en la nube a través de https://excel.cloud.microsoft o en la aplicación de escritorio si se cuenta con una versión de Excel posterior a 2024. Esta guía se elabora en la versión de Excel Microsoft 365 en la nube.
- 2. Iniciar sesión con una cuenta de Microsoft valida. Puede hacerlo a través del botón verde "Iniciar sesión" en la esquina superior derecha si se encuentra en la versión de Excel en la nube.
- 3. Dar clic en crear un libro en blanco.
- 4. Ya debe estar dentro del libro de Excel.

Utilizar Python en Excel: Para el primer ejemplo, vamos a crear un programa sin librerias adicionales, que sume el contenido de las celdas desde A1 hasta A5.

1. Llenar las celdas con números a su elección.

healvare@utp.edu.co

Hernando Alvarez R

- 2. En una celda aparte debe insertar la fórmula de Excel: "=PY(".
- 3. Deberá notar como la celda cambió de apariencia. Ahora puede insertar en ella código del lenguaje Python.
- 4. Si le hace clic derecho a la celda, verá la opción "Abrir editor" o "Show Editor" si está usando Excel en inglés. Dar clic para abrir el editor completo de Python dentro de Excel.
- 5. En esta celda podemos incluir el siguiente código:

```
#Para este ejemplo no se van a usar librerias de Python
1
    #Este programa suma el contenido de las celdas A1 hasta A5
2
    #Crear una clase
3
    class Lista numeros:
4
        def __init__(self,lista):
5
            self.lista = []
6
             for i in lista.values:
7
                 self.lista.append(i)
8
        def sumatoria(self):
9
             suma = 0
10
             for i in self.lista:
11
                 suma += i
12
            return suma
13
    primera_columna = Lista_numeros(xl("A1:A5"))
14
15
    primera_columna.sumatoria()
16
```

El anterior Código corresponde a la creación de una clase con Python o.o. Esta clase tiene una función que suma los 5 primeros valores de la columna A.

- 6. Una vez añadido el código. Presione el comando "ctrl + enter".
- 7. Deberá ver como en la celda se muestra la suma de las celdas anteriores.
- 8. Ahora, si se cambian los valores de las celdas antes mencionadas, el output de la celda de Python deberá variar automáticamente

Segundo ejemplo de Python en Excel

Para el segundo ejemplo, vamos a crear un programa que construya una gráfica de barras.

Para ello usaremos las librerías Pandas y Matplotlib.

1. Llenar las celdas A9 hasta A13 con los nombres de las categorías a su elección.

2. Llenar las celdas B9 hasta B13 con las frecuencias de cada categoría que será analizada en el gráfico de barras.

Observemos tabla de ejemplo:

J11		×X
	А	В
6		
7		
8		
9	Luis	10
10	Hernando	20
11	Ospina	20
12	Alvarez	40
13	Miguel	50

3. En una celda aparte, crear una celda de Python e insertar el siguiente código:

```
#Programa que crea una grafica de barras en Excel.
1
2
    #Importar librerias
3
4
    import matplotlib.pyplot as plt
\mathbf{5}
    import pandas as pd
6
7
    #Configuración de la grafica
8
9
    df_x = pd.DataFrame(xl("A9:A13"))
10
    df_y = pd.DataFrame(xl("B9:B13"))
11
^{12}
    df_x = df_x.iloc[:, 0]
13
    df_y = df_y.iloc[:, 0]
14
15
    plt.bar(df_x,df_y)
16
17
    # Añadir etiquetas y título
^{18}
    plt.xlabel('Nombres')
19
    plt.ylabel('Cantidad')
^{20}
    plt.title('Gráfico de Barras')
21
^{22}
```

healvare@utp.edu.co

Hernando Alvarez R

23	# Mostrar el	gráfico
24	plt.show()	

4. Podrá observar el gráfico de barras dentro de la celda que seleccionó. Observemos el gráfico resultante de la tabla de ejemplo



Gráfico de Barras