

1. Instalación de Python 3.9 y las bibliotecas.

En la página web de Python (www.python.org) iremos a *Downloads* para descargar la versión más reciente para *Windows*, *Mac* o *Linux*, como se ve en la figura 1.

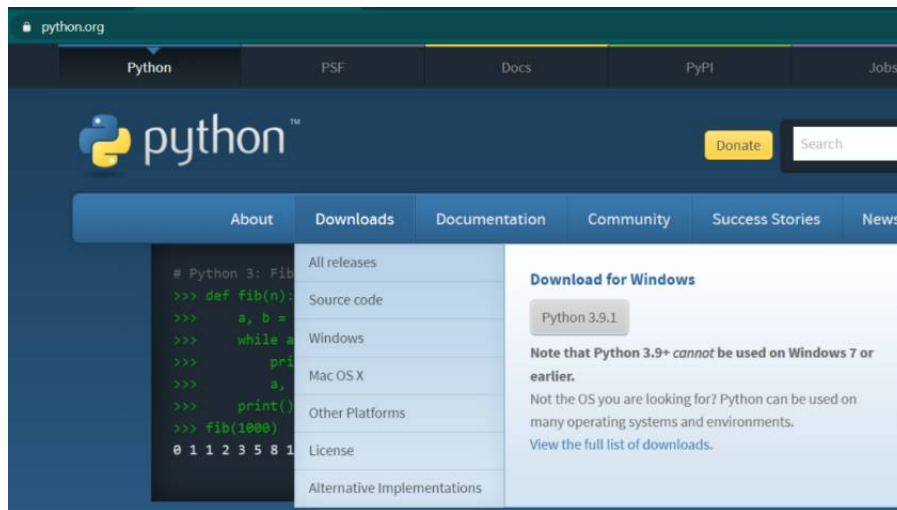


Figura 1. Descarga de la versión más reciente de Python.

Deberá ejecutarse el archivo .exe que se descargó e instalar el lenguaje, verificando que el casillero *Add Python 3.9 to PATH* esté marcado con una paloma, como se ve en la figura 2.



Figura 2. Ventana de instalación de Python con casillero marcado.

Esto es importante, porque más adelante instalaremos las bibliotecas y esta ruta (*PATH*) será direccionada automáticamente para ubicar los archivos de las bibliotecas. Continuar con la instalación haciendo clic en *Install Now*. Una vez que el programa queda instalado se mostrará la imagen de la figura 3. En esta ventana se puede tener acceso a un tutorial en línea (*online tutorial*), la documentación del lenguaje (*documentation*), que información hay

de nuevo en esta versión (*what's new*) o información sobre el uso de Python en Windows (*using Python on Windows*).

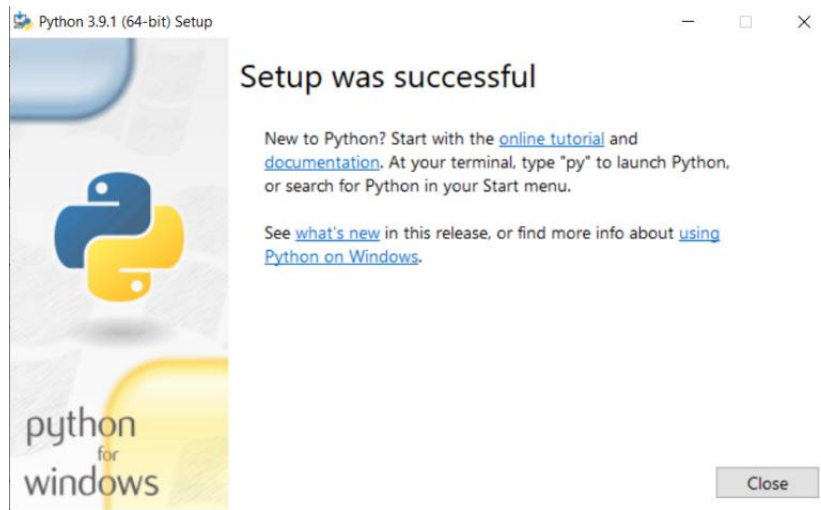


Figura 3. Python instalado.

Para ingresar al lenguaje, desde el menú Inicio de Windows, deben hacer clic en *IDLE (Python 3.9 64-bit)*, como se observa en la figura 4.

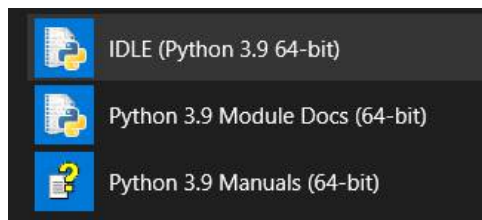


Figura 4. Acceso a Python 3.9

Se mostrará el *Shell de Python* desde donde iniciaremos un nuevo archivo para crear los programas, según se ve en la figura 5.

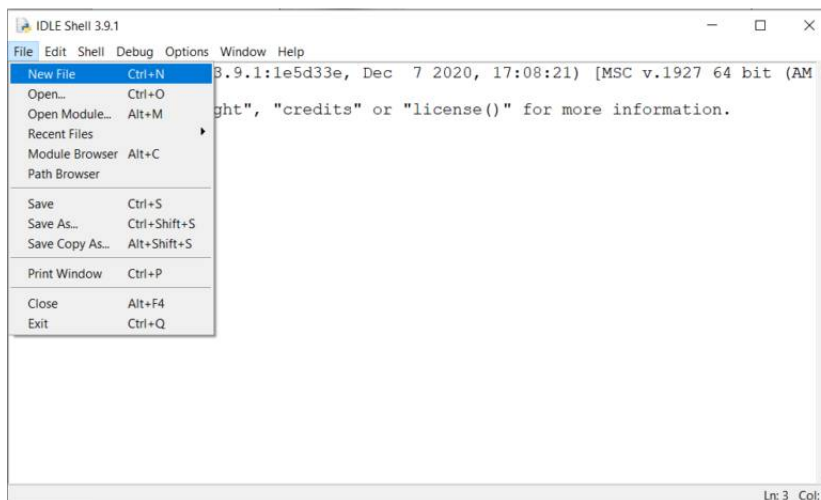


Figura 5. Shell de Python y menú File para crear un nuevo archivo.

La ventana donde se escribirá el programa se muestra en la figura 6.

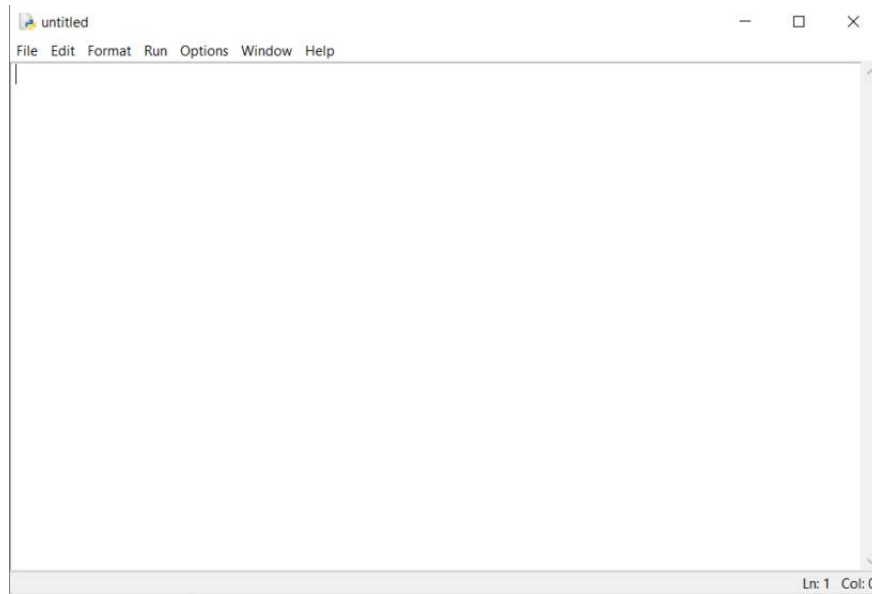


Figura 6. Ventana para escritura del programa.

A continuación, mostraremos como se instala la biblioteca *Numpy*, y de una forma similar, se instalarán las bibliotecas *Pandas*, *Tabulate*, *EasyGUI* y *MatPlotLib*, que son las que usaremos en este curso.

Vamos a utilizar la línea de comandos (*command.com*) o símbolo del sistema de *Windows*, el cual se puede llamar desde la búsqueda escribiendo *command*. En forma textual se escribirá el comando *pip install numpy*, el cual ejecutará el programa de instalación de aplicaciones de *Python: p(ython) i(nstall) p(rogram)*, como se ve en la figura 7, iniciará la descarga de los archivos y los ubicará en la ruta (*path*) correcta para ser invocados por *Python* mediante la instrucción *import*.

```

Microsoft Windows [Versión 10.0.19041.685]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

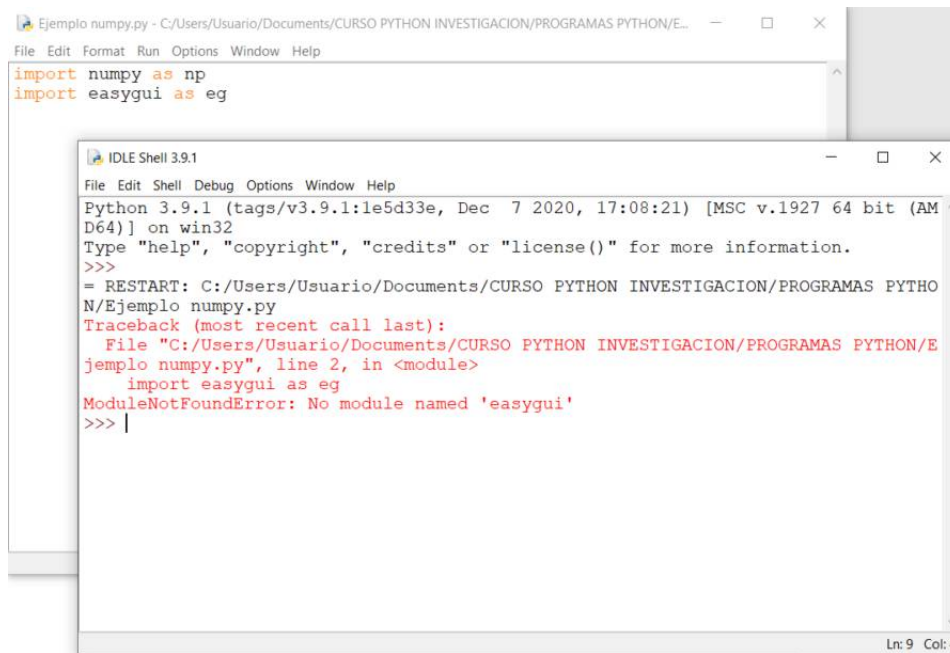
C:\Users\Usuario>pip install numpy
Collecting numpy
  Downloading numpy-1.19.5-cp39-cp39-win_amd64.whl (13.3 MB)
    |#####| 13.3 MB 3.3 MB/s
Installing collected packages: numpy
Successfully installed numpy-1.19.5
WARNING: You are using pip version 20.2.3; however, version 20.3.3 is available.
You should consider upgrading via the 'c:\users\usuario\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.

C:\Users\Usuario>

```

Figura 7. Instalación de Numpy con pip.

Mientras no se instalen las bibliotecas externas, no pueden ser utilizadas en los programas de *Python*, como se ve en la figura 8, donde se invoca a la biblioteca *EasyGUI* cuando aún no se instala. Puede observarse como no hay mensaje de error para la biblioteca *Numpy*.



```

Ejemplo numpy.py - C:/Users/Usuario/Documents/CURSO PYTHON INVESTIGACION/PROGRAMAS PYTHON/E...
File Edit Format Run Options Window Help
import numpy as np
import easygui as eg

IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Usuario/Documents/CURSO PYTHON INVESTIGACION/PROGRAMAS PYTHON/Ejemplo numpy.py
Traceback (most recent call last):
  File "C:/Users/Usuario/Documents/CURSO PYTHON INVESTIGACION/PROGRAMAS PYTHON/Ejemplo numpy.py", line 2, in <module>
    import easygui as eg
ModuleNotFoundError: No module named 'easygui'
>>> |
Ln: 9 Col: 4

```

Figura 8. Mensaje de error por invocar EasyGUI con import sin estar instalado.

Una vez instaladas todas las bibliotecas externas mencionadas anteriormente, están listas para usarse en un programa, las cuales se llamarán mediante la instrucción *import* o sus variantes: *from biblioteca import módulo as alias*.

Ejemplo 1.

Se tienen los datos de dos variables X y Y , los cuales se requieren para generar una interpolación de *Lagrange* al no presentar equidistancia entre los valores de X . La fórmula de interpolación de *Lagrange* se expresa como:

$$y_{int}(x_{int}) = \sum_{i=1}^n Y_i \prod_{j=1, j \neq i}^n \frac{x_{int} - X_j}{X_i - X_j}$$

Con $i \neq j$ para $i, j = 1, 2, \dots, n$

Elaborar un programa que permita encontrar un valor interpolado de Y a partir de un valor x_{int} . Usar un entorno *GUI* y presentar la gráfica de los puntos. Se muestran los datos y su gráfica generada con Mathcad en la figura 9.

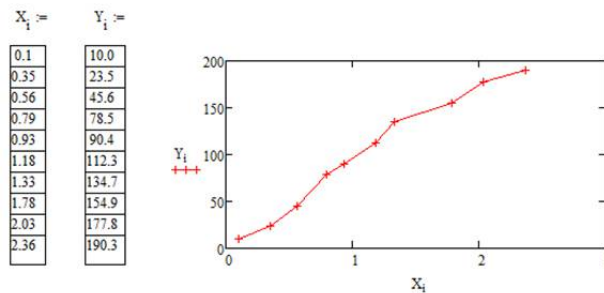


Figura 9. Datos y gráfica para aplicar interpolación.

El programa se muestra a continuación:

```

import numpy as np
import easygui as eg
import matplotlib.pyplot as py
from tabulate import tabulate as tab

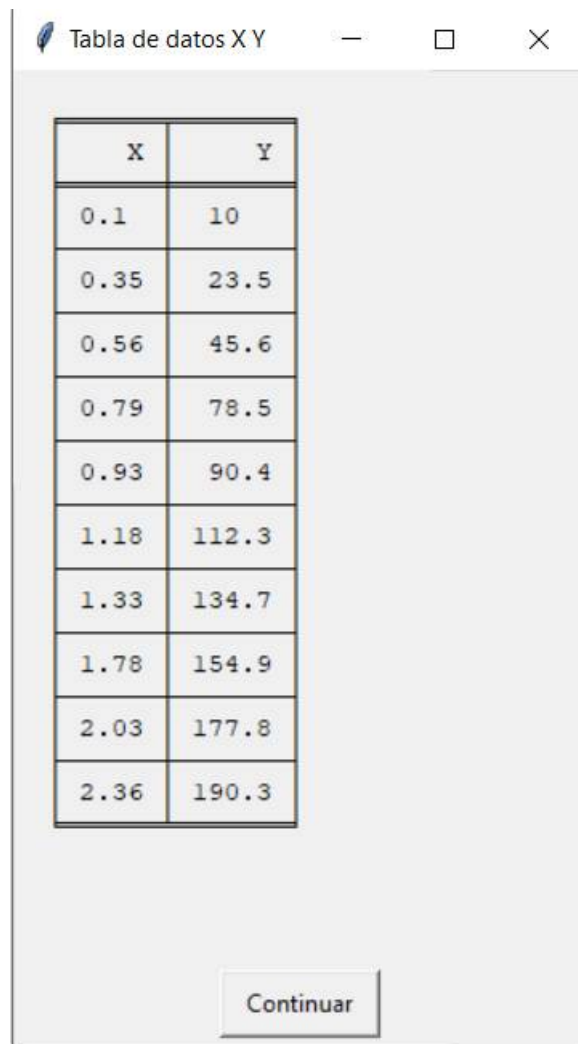
X=[0.10,0.35,0.56,0.79,0.93,1.18,1.33,1.78,2.03,2.36]
Y=[10.0,23.5,45.6,78.5,90.4,112.3,134.7,154.9,177.8,190.3]

def Entrada():
    xint=eval(eg.enterbox("Ingrese el valor de X a interpolar"))
    return xint
def Tabla(X,Y,msg):
    tabla=[(X,Y)]
    cabeza=["X","Y"]
    texto="Tabla de datos X Y"
    txttabla=tab(np.transpose(tabla),cabeza,tablefmt="fancy_grid")
    eg.msgbox(txttabla+"\n"+msg,texto,"Continuar")
def Lagrange(X,Y,xint):
    n=len(X)
    suma=0
    for i in range(0,n):
        producto=1
        for j in range(0,n):
            if i!=j:
                producto=producto*(xint-X[j])/(X[i]-X[j])
        suma=suma+Y[i]*producto
    return suma
def Grafica(X,Y,xint,yint):
    py.plot(X,Y,"-",xint,yint,"o")
    py.show()
def Resultados(xint,yint):
    txt="El valor de Y para "+str(xint)+" es: "+str(yint)
    Tabla(X,Y,txt)
    Grafica(X,Y,xint,yint)
def Principal():
    otro="S"
    while otro=="S" or otro=="s":
        msg=""
        Tabla(X,Y,msg)
        xint=Entrada()
        yint=Lagrange(X,Y,xint)
        Resultados(xint,yint)
        otro=eg.enterbox("Otro (S/N):")

Principal()

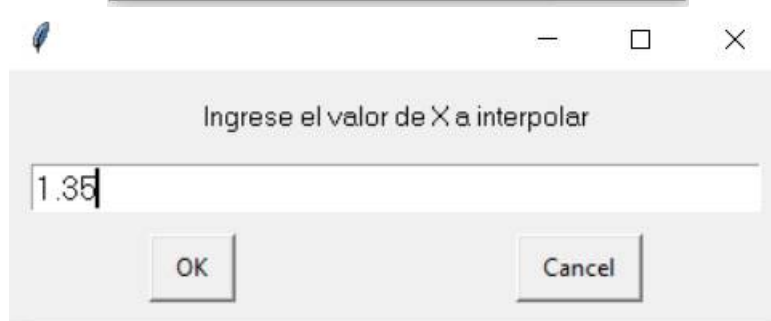
```

La corrida del programa se presenta a continuación:



A Python GUI window titled "Tabla de datos X Y" with standard window controls (minimize, maximize, close). It contains a table with two columns, X and Y, and a "Continuar" button at the bottom.

X	Y
0.1	10
0.35	23.5
0.56	45.6
0.79	78.5
0.93	90.4
1.18	112.3
1.33	134.7
1.78	154.9
2.03	177.8
2.36	190.3



A Python GUI window titled "Ingrese el valor de X a interpolar" with standard window controls. It features a text input field containing the value "1.35" and two buttons: "OK" and "Cancel".

Tabla de datos X Y

X	Y
0.1	10
0.35	23.5
0.56	45.6
0.79	78.5
0.93	90.4
1.18	112.3
1.33	134.7
1.78	154.9
2.03	177.8
2.36	190.3

El valor de Y para 1.35 es:
137.95323221184424

Continuar

