

# PYTHON BÁSICO

**Ing. Henry Alberto Hernández Martínez**

[hahernandezm@udistrital.edu.co](mailto:hahernandezm@udistrital.edu.co)

Ingeniería en telecomunicaciones

Universidad Distrital Francisco José de Caldas



# CONTENIDO

- Introducción
- Estructuras de control
- Elementos básicos de la interfaz gráfica de usuario.



# INTRODUCCIÓN



# ANACONDA

- Es un distribución libre y abierta de los lenguajes Python y R, utilizada en ciencia de datos, aprendizaje automático y está orientado a simplificar el despliegue y administración de los paquetes de software.

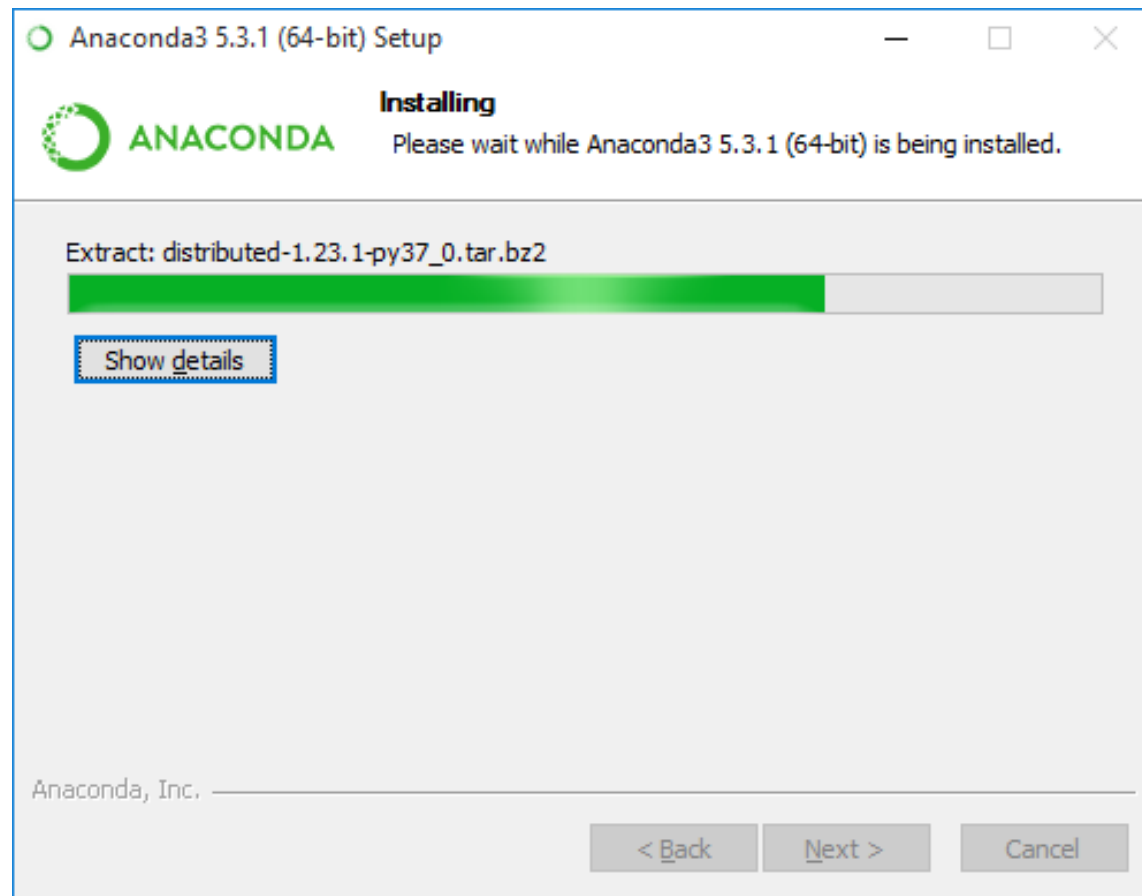


ANACONDA®



# INSTALACIÓN

- Es una aplicación intuitiva que se puede descargar de la pagina y simplemente hay que ejecutarla.



# LIBRERÍAS

- Abra la aplicación (como administrador) ANACONDA PROMPT y digite **pip o conda + install + nombre de la librería a instalar.**

Anaconda Prompt (Anaconda3)

```
(base) C:\Users\██████████>pip install matplotlib  
Requirement already satisfied: matplotlib in c:\program files\anaconda3\lib\site-packages  
Requirement already satisfied: cyclor>=0.10 in c:\program files\anaconda3\lib\site-packages  
Requirement already satisfied: kiwisolver>=1.0.1 in c:\program files\anaconda3\lib\site-packages
```

- Previamente actualice conda digitando **conda update conda.**



# LIBRERÍAS

Digite las siguientes líneas de código en el terminal de Anaconda.

- `conda create -n mi_sesion pip python=3.6`
- `actívale mi_sesion`
- `pip install --ignore-installed --upgrade tensorflow`
- `pip install keras`
- `pip install Theano`
- `pip install pyserial`
- `pip install pysnmp`
- `pip install pysmi`
- `pip install pygame`
- `conda install py-opencv`
- `pip install pymysql`
- `conda install scikit-learn`
- `pip install paho-mqtt`
- `pip install matplotlib`
- `pip install pillow`
- `pip install scapy`
- `conda install pandas`
- `conda install spyder`



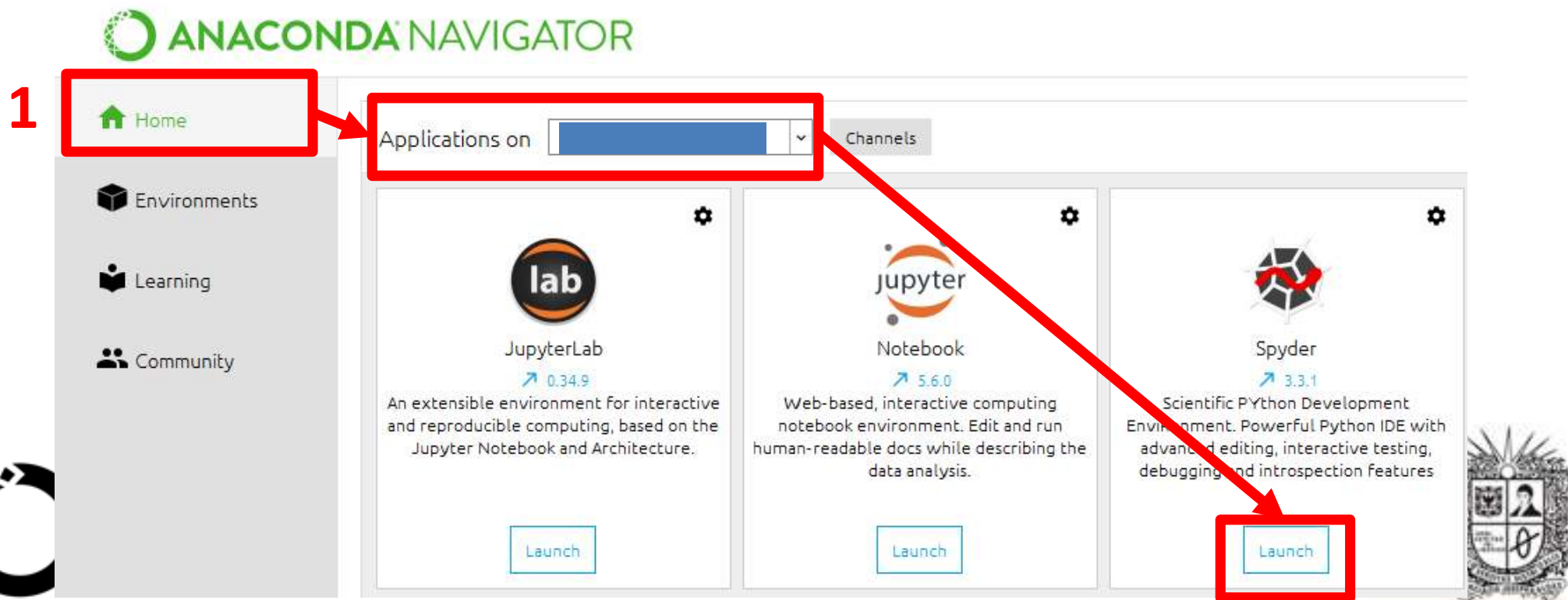
# PRIMER APLICACIÓN EN PYTHON





# PRIMER APLICACIÓN

- Abra ANACONDA NAVIGATOR haga clic en home, seleccione la sesión creada en la opción Applications on y haga clic en el icono Launch de Spyder.



# PRIMER APLICACIÓN

The image shows the Spyder Python IDE interface. The main editor window on the left contains a Python script named `temp.py` with the following content:

```
1 # -*- coding: utf-8 -*-
2 """
3 Editor de Spyder
4
5 Este es un archivo temporal.
6 """
7
8 a=1+3
9
```

The right side of the interface features three panels:

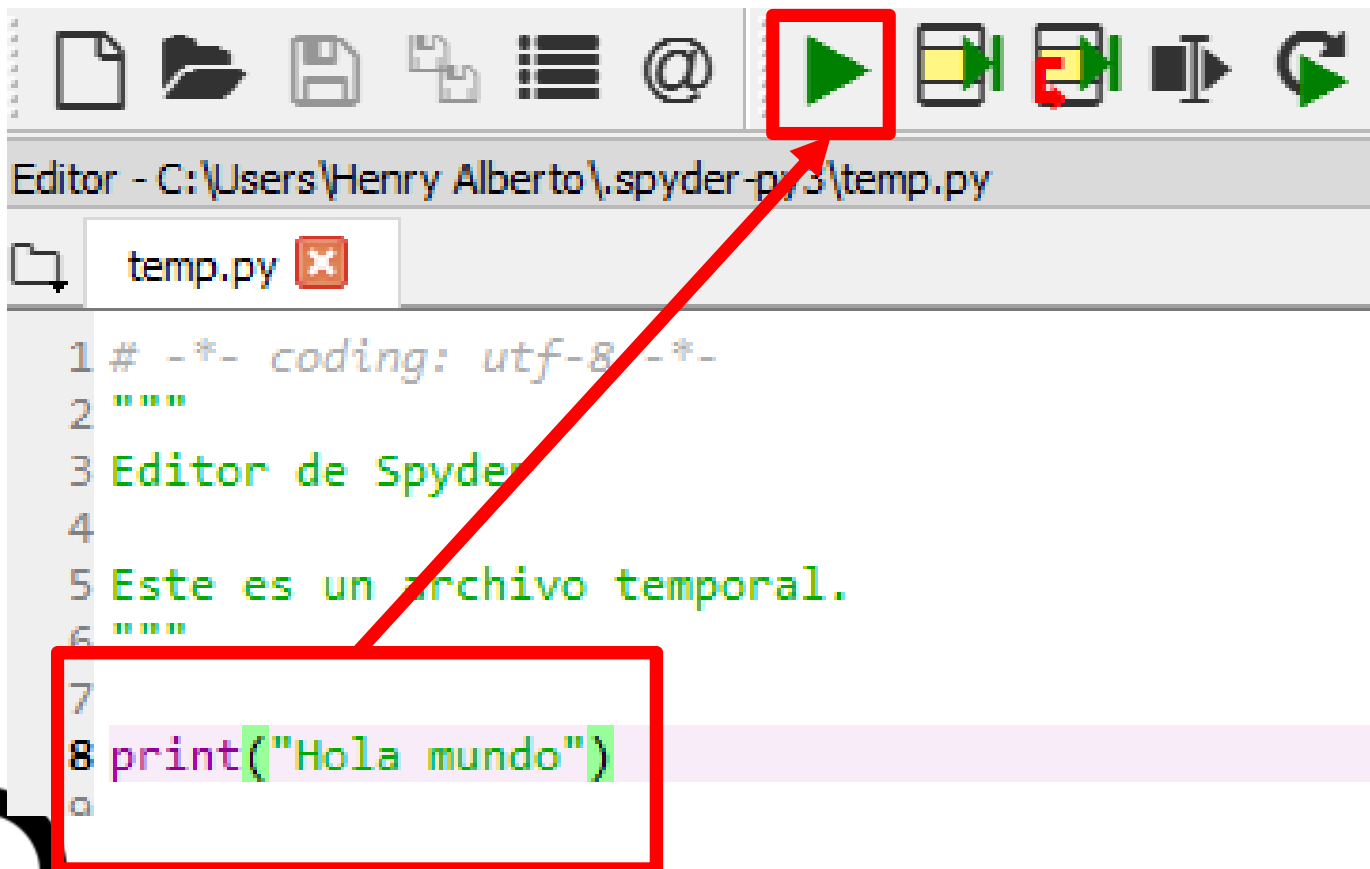
- BARRA DE HERRAMIENTAS** (Toolbar): Located at the top right, it contains icons for file operations (open, save, print), editing (undo, redo), and execution (run, debug, console).
- EXPLORADOR** (Variable Explorer): Located in the middle right, it displays a table of variables. The table has columns for **Nombre** (Name), **Tipo** (Type), **Tamaño** (Size), and **Valor** (Value). It shows a single variable `a` of type `int` with a size of 1 and a value of 4.
- TERMINAL** (IPython Terminal): Located at the bottom right, it shows the execution of the script. The output includes the Python version (3.7.0), IPython version (6.5.0), and the command `runfile('C:/Users/Henry Alberto/.spyder-py3/temp.py', wdir='C:/Users/Henry Alberto/.spyder-py3')`.

The **ÁREA DE TRABAJO** (Work Area) is highlighted with a red border, encompassing the main editor and the variable explorer.



# PRIMER APLICACIÓN

- Digite la línea mostrada (en el área de trabajo) y haga clic en run (observe el terminal).

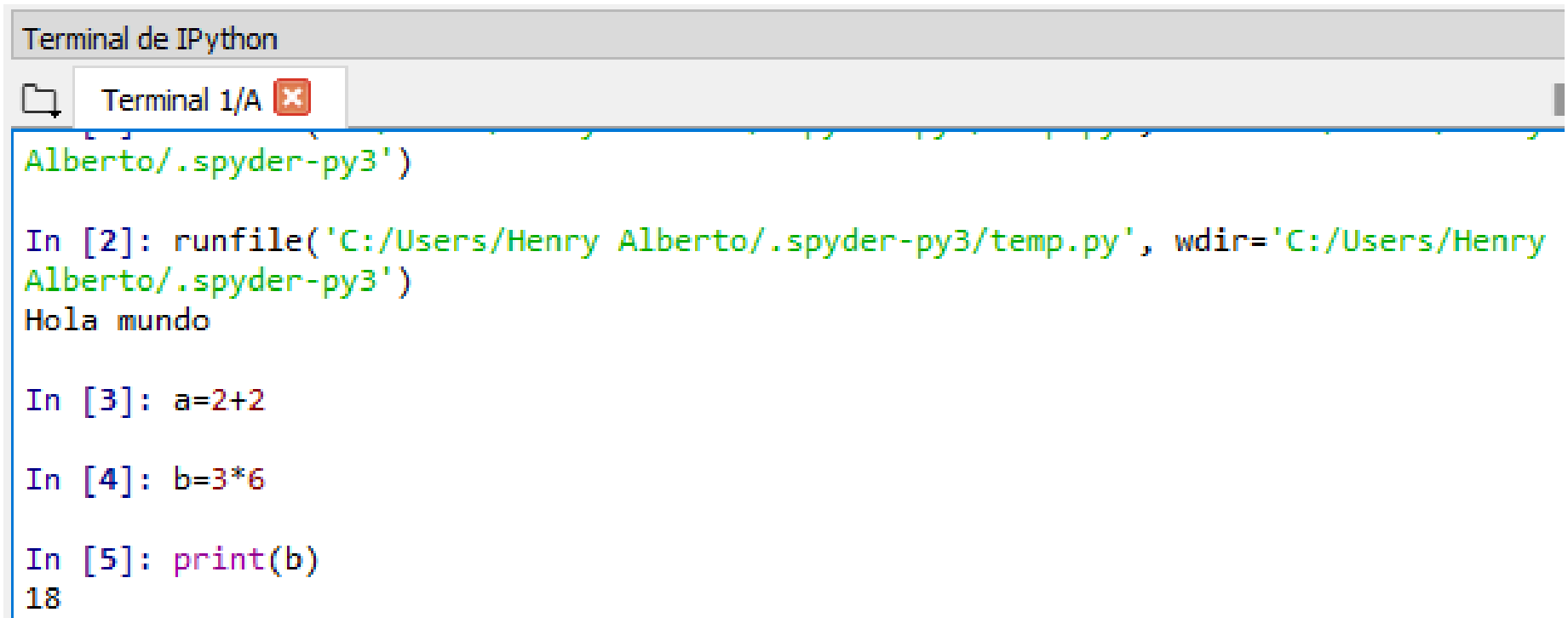


```
1 # -*- coding: utf-8 -*-
2 """
3 Editor de Spyder
4
5 Este es un archivo temporal.
6 """
7
8 print("Hola mundo")
9
```



# PRIMER APLICACIÓN

- Ahora en el terminal haga algunas líneas de código y visualice su forma de trabajo.



The screenshot shows a Spyder IPython terminal window titled 'Terminal de IPython'. The window has a tab labeled 'Terminal 1/A'. The terminal content is as follows:

```
Alberto/.spyder-py3')  
  
In [2]: runfile('C:/Users/Henry Alberto/.spyder-py3/temp.py', wdir='C:/Users/Henry  
Alberto/.spyder-py3')  
Hola mundo  
  
In [3]: a=2+2  
  
In [4]: b=3*6  
  
In [5]: print(b)  
18
```

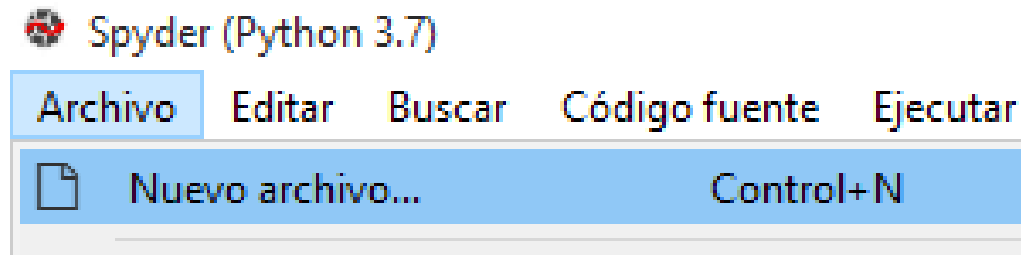


# CARACTERÍSTICAS BÁSICAS DE PYTHON



# CARACTERÍSTICAS

- Cree un archivo nuevo en Spyder.



- Está es la forma de crear un comentario e importar un elemento.

```
1 # -*- coding: utf-8 -*-  
2 #Este es un comentario en PYTHON  
3 """Esta es una forma de crear un  
4 comentario de varias líneas"""  
5 #Una librería se importa con el comando import.  
6 #Por ejemplo  
7 import numpy  
8 numpy.abs(-9)
```



# TIPOS DE VARIABLE

```
6 """PYTHON no tiene un tipo de variable especifico"""  
7 #Por ejemplo  
8 Entero=4  
9 Decimal=2.4  
10 Caracter='b'  
11 Cadena="Hola"  
12 Vector=[1,2,3,4,5]
```

---



# TIPOS DE VARIABLE

```
6 """PYTHON no tiene un tipo de variable especifico"""
7 #Por ejemplo
8 Entero=4
9 Decimal=2.4
10 Caracter='b'
11 Cadena="Hola"
12 Vector=[1,2,3,4,5]
```

Explorador de variables			
Nombre	Tipo	Tamaño	
Cadena	str	1	Hola
Caracter	str	1	b
Decimal	float	1	2.4
Entero	int	1	4
Vector	list	5	[1, 2, 3, 4, 5]

Como se observa PYTHON asigna automáticamente el tipo de variable.





# CONCATENACIÓN

```
15 """Formas de anidar e imprimir variables"""
16 #Imprimir la variable directamente
17 print(Entero)
18 #Convertir las variables a caracter
19 Texto1=Cadena+" "+str(Entero)+" "+str(Decimal)
20 print(Texto1)
21 #Anidar caracteres directamente
22 Texto2="El caracter es: "+Caracter
23 print(Texto2)
24 #Crear una lista
25 Texto3=[Texto1, Texto2]
26 print(Texto3)
```



# OPERACIONES ARITMÉTICAS

```
15 """Formas de anidar e imprimir variables"""
16 #Imprimir la variable directamente
17 print(Entero)
18 #Convertir las variables a caracter
19 Texto1=Cadena+" "+str(Entero)+" "+str(Decimal)
20 print(Texto1)
21 #Anidar caracteres directamente
22 Texto2="El caracter es: "+Caracter
23 print(Texto2)
24 #Crear una lista
25 Texto3=[Texto1, Texto2]
26 print(Texto3)
```

---

```
29 """Operaciones matemáticas básicas"""
30 Suma=5+2
31 Resta=5-3
32 Multiplicacion=2*3
33 Division=3/1
34 print(Suma)
```



# OPERACIONES ARITMÉTICAS

```
36 """Otras operaciones matemáticas"""  
37 Cociente=3//1  
38 Residuo=4%2  
39 base=2  
40 exponente=3  
41 Potenciacion=pow(base,exponente)  
42 Redondear=round(2.55)  
43 lista=[-1,4,5]  
44 Maximo=max(lista)  
45 Minimo=min(lista)
```



# OPERACIONES ARITMÉTICAS

```
36 """Otras operaciones matemáticas"""
```

```
37 Cociente=3//1
```

```
38 Residuo=4%2
```

```
39 base=2
```

```
40 exponente=3
```

```
41 Potenciacion=pow(base,exponente)
```

```
42 Redondear=round(2.55)
```

```
43 lista=[-1,4,5]
```

```
44 Maximo=max(lista)
```

```
45 Minimo=min(lista)
```

```
49 """Operaciones con números complejos"""
```

```
50 a=1+2j
```

```
51 b=2.2-5j
```

```
52 print([a+b, b*a])
```



# FUNCIONES TRIGONOMÉTRICAS

```
54 """Algunas operaciones requieren de la libreria math,  
55 por su complejidad."""  
56 import math as matematicas  
57 #Número PI  
58 pi=matematicas.pi  
59 #Ejemplo funciones trigonométricas  
60 print(matematicas.sin(pi/2))  
61 print(matematicas.cos(pi/2))  
62 print(matematicas.tan(pi/2))  
63 #Funciones trigonométricas inversas  
64 print(matematicas.asin(0.37))  
65 print(matematicas.acos(1))  
66 print(matematicas.atan(0.4))  
67 #Conversiones  
68 #De radianes a grados sexagesimales  
69 print(matematicas.degrees(0.5))  
70 #De grados sexagesimales a radianes  
71 print(matematicas.radians(90))
```



# ESTRUCTURAS DE CONTROL



# ESTRUCTURAS DE CONTROL

```
73 """PYTHON tiene dos estructuras de control de
74 entrada y salida por pantalla"""
75 #Entrada
76 numero=input("Digite un número:")
77 #Salida
78 print("El número es: ",numero)
```



# ESTRUCTURAS DE CONTROL

```
73 """PYTHON tiene dos estructuras de control de
74 entrada y salida por pantalla"""
75 #Entrada
76 numero=input("Digite un número:")
77 #Salida
78 print("El número es: ",numero)

80 """La estrucutra condicional asume que lo que esta
81 adelante un espacio tabular(tab) pertenece a ella."""
82 #Ejemplo
83 if(int(numero)==3):
84     print("Numero Correcto")
85 else:
86     print("Numero Incorrecto")
87 print("Continua la ejecución de la aplicación")
```





# ESTRUCTURAS DE CONTROL

```
89 """En PYTHON funcionan las formas de comparación convencionales como"""
90 """x!=y          x No es igual a y"""
91 """x==y          x Es igual a y"""
92 """x>y           x Es mayor que y"""
93 """x<y           x Es menor que y"""
94 """and           comparación lógica and"""
95 """or            comparación lógica or"""
96 #Ejemplo de implementación con un elif
97 Selector=int(input("Digite una opción: "))
98 if(Selector==0):
99     print("El número es cero")
100 elif(Selector==1):
101     print("El número es uno")
102 elif(Selector<0 or Selector>2):
103     print("El número está fuera de rango")
```



# ESTRUCTURAS DE CONTROL

```
89 """En PYTHON funcionan las formas de comparación convencionales como"""
90 """x!=y          x No es igual a y"""
91 """x==y          x Es igual a y"""
92 """x>y           x Es mayor que y"""
93 """x<y           x Es menor que y"""
94 """and           comparación lógica and"""
95 """or            comparación lógica or"""
96 #Ejemplo de implementación con un elif
97 Selector=int(input("Digite una opción: "))
98 if(Selector==0):
99     print("El número es cero")
100 elif(Selector==1):
101     print("El número es uno")
102 elif(Selector<0 or Selector>2):
103     print("El número está fuera de rango")
```

```
104
105 """Nota: La estructura SWITCH en PYTHON no existe"""
```



# ESTRUCTURAS DE CONTROL

```
107 """Estructuras de Bucle"""
108 #Ejemplos
109 contador=0
110 iteraciones=int(input("Digite la cantidad de iteraciones"))
111 """For incremental"""
112 for i in range(0,iteraciones):
113     print(i)
114 """For decremental"""
115 for i in range(-iteraciones,0):
116     print(i)
117 """While"""
118 while (contador<iteraciones):
119     print(contador)
120     contador=contador+1
```



# LISTAS

```
1 # -*- coding: utf-8 -*-
2 """PYTHON tiene la capacidad de almacenar listas"""
3 Lista=["Sujeto 1", "Sujeto 2", "Sujeto 3"]
4 """La forma más sencilla de publicar es:"""
5 print(Lista)
6 """Sin embargo, puede usar una estructura"""
7 #el comando len se usa para conocer la longitud de la lista
8 for i in range(len(Lista)):
9     print(Lista[i])
```



# LISTAS

```
1 # -*- coding: utf-8 -*-
2 """PYTHON tiene la capacidad de almacenar listas"""
3 Lista=["Sujeto 1", "Sujeto 2", "Sujeto 3"]
4 """La forma más sencilla de publicar es:"""
5 print(Lista)
6 """Sin embargo, puede usar una estructura"""
7 #el comando len se usa para conocer la longitud de la lista
8 for i in range(len(Lista)):
9     print(Lista[i])
10 """La matrices son un caso particular de lista, ya que,
11 son una lista anidada."""
12 #Ejemplo
13 Matriz=[]
14 Numero_Filas=5
15 Numero_Columnas=10
16 for i in range(Numero_Filas):
17     Matriz.append([])
18     for j in range(Numero_Columnas):
19         Matriz[i].append("Dato a asignar en la posición: "
20                             +str(i)+", "+str(j))
21 print(Matriz)
```



# DEFINIR FUNCIONES

```
1 # -*- coding: utf-8 -*-
2 """Definición de funciones"""
3 """Declaración de variables"""
4 a=1
5 """Está función tiene parámetros de entrada"""
6 def funcion1(Variable_Entrada):
7     """Código particular"""
8     """Si desea usar una variable contenida definida en
9         otro punto del código, use el comando global."""
10    global a
11    a=a*Variable_Entrada
12
13 """Está función no tiene parámetros de entrada"""
14 """pero si tiene un parámetro de salida"""
15 def funcion2():
16     global a
17     b=a+3
18     return b
```



# DEFINIR FUNCIONES

```
20 """Definición de la función principal"""
21 if __name__ == "__main__":
22     """Llamado a las distintas funciones"""
23     funcion1(3)
24     w=a*funcion2()
25     print(w)
```



# INTERFAZ GRAFICA





# MI PRIMER INTERFAZ GRAFICA

```
1 # -*- coding: utf-8 -*-
2 """Mi primer aplicación con interfaz grafica"""
3 """importe la libreria Tkinter"""
4 from tkinter import *
5
6 """Función principal"""
7 if __name__ == "__main__":
8     """Crear ventana vacia"""
9     Aplicacion=Tk()
10    """Definición de los objetos a usar"""
11    Aplicacion.title("Primer App")
12    etiqueta=Label(Aplicacion,text="Saludos")
13    boton=Button(Aplicacion,text="OK")
14    """Poner objetos en la ventana"""
15    etiqueta.pack()
16    boton.pack()
17    """La instrucción Mainloop mantiene activa
18    la aplicación"""
19    Aplicacion.mainloop()
```



# CREAR EVENTOS



# CREAR EVENTOS

**Recuerde importar tkinter**

```
22 """Función principal"""
23 if __name__ == "__main__":
24     """Crear ventana vacía"""
25     Aplicacion=Tk()
26     """Definición de los objetos a usar"""
27     Aplicacion.title("Edición")
28     """Crear etiqueta con variable de texto"""
29     T_Etiqueta="Digite algo..."
30     etiqueta=Label(Aplicacion,text=T_Etiqueta)
31     """Crear Botón (clic=nombre de la función
32     asociada)"""
33     boton=Button(Aplicacion,text="Pulse",command=clic)
34     """Crear campo de entrada y/o salida"""
35     T_Salida=" "
36     C_Entrada=Entry(Aplicacion, textvariable=T_Salida)
37     """Poner objetos en la ventana (aparecen
38     en el orden que se pongan en el código)"""
39     etiqueta.pack()
40     boton.pack()
41     C_Entrada.pack()
42     """Bucle infinito"""
43     Aplicacion.mainloop()
```



# CREAR EVENTOS

```
22 """Función principal"""
23 if __name__ == "__main__":
24     """Crear ventana vacía"""
25     Aplicacion=Tk()
26     """Definición de los objetos a usar"""
27     Aplicacion.title("Edición")
28     """Crear etiqueta con variable de texto"""
29     T_Etiqueta="Digite algo..."
30     etiqueta=Label(Aplicacion,text=T_Etiqueta)
31     """Crear Botón (clic=nombre de la función
32     asociada)"""
33     boton=Button(Aplicacion,text="Pulse",command=clic)
34     """Crear campo de entrada y/o salida"""
35     T_Salida=" "
36     C_Entrada=Entry(Aplicacion, textvariable=T_Salida)
37     """Poner objetos en la ventana (aparecen
38     en el orden que se pongan en el código)"""
39     etiqueta.pack()
40     boton.pack()
41     C_Entrada.pack()
42     """Bucle infinito"""
43     Aplicacion.mainloop()
```

Si lo compilo debe aparecer error, ya que, clic no esta definida.



# CREAR EVENTOS

```
3 """importe la libreria Tkinter"""
4 from tkinter import *
5 """sys contiene a try"""
6 import sys
7
8 def clic():
9     """Verifique si el recurso u objeto está disponible"""
10    try:
11        """Leer desde el campo (String por defecto)"""
12        Numero=int(C_Entrada.get())
13        T_Etiqueta="El numero es: "+str(Numero)
14        """Modificar el estado de etiqueta"""
15        etiqueta.config(text=T_Etiqueta)
16    except ValueError:
17        """La excepción se activa si hay al menos un error en el
18        dato del campo de entrada"""
19        T_Etiqueta="Introduzca un dato"
20        etiqueta.config(text=T_Etiqueta)
```



# EDICIÓN DE CARACTERÍSTICAS



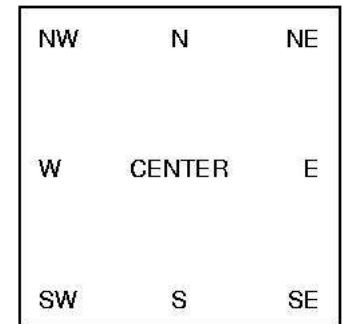
# EDICIÓN

```
8 """Función principal"""
9 if __name__ == "__main__":
10     """Crear ventana vacía"""
11     Aplicacion=Tk()
12     """Ajustar dimensiones de la ventana (Ancho, Alto)"""
13     Aplicacion.geometry("500x600")
14     Aplicacion.title("Características")
15     """Color de fondo y texto Alto, Ancho y ubicación del texto"""
16     etiqueta=Label(Aplicacion,text=T_Etiqueta,bg='red', fg='blue')
17     """Tamaño del boton"""
18     boton=Button(Aplicacion,text="Pulse", width=20, height=10, anchor="ne")
19     """Crear campo de entrada y/o salida"""
20     T_Salida=" "
21     C_Entrada=Entry(Aplicacion, textvariable=T_Salida,bg='yellow')
22     """Poner objetos en la ventana (aparecen
23     en el orden que se pongan en el código)"""
24     etiqueta.pack()
25     boton.pack()
26     C_Entrada.pack()
27     """Bucle infinito"""
28     Aplicacion.mainloop()
```



# EDICIÓN

```
8 """Función principal"""
9 if __name__ == "__main__":
10     """Crear ventana vacía"""
11     Aplicacion=Tk()
12     """Ajustar dimensiones de la ventana (Ancho, Alto)"""
13     Aplicacion.geometry("500x600")
14     Aplicacion.title("Características")
15     """Color de fondo y texto Alto, Ancho y ubicación del texto"""
16     etiqueta=Label(Aplicacion,text=T_Etiqueta,bg='red', fg='blue')
17     """Tamaño del boton"""
18     boton=Button(Aplicacion,text="Pulse", width=20, height=10, anchor="ne")
19     """Crear campo de entrada y/o salida"""
20     T_Salida=" "
21     C_Entrada=Entry(Aplicacion, textvariable=T_Salida,bg='yellow')
22     """Poner objetos en la ventana (aparecen
23     en el orden que se pongan en el código)"""
24     etiqueta.pack()
25     boton.pack()
26     C_Entrada.pack()
27     """Bucle infinito"""
28     Aplicacion.mainloop()
```





# EDICIÓN

```
etiqueta.pack()  
boton.pack()  
C_Entrada.pack()
```

*Al reemplazar la forma de poner los objetos en la ventana de .pack por .place se fija su ubicación.*

```
"""Fija ubicación y tamaño"""  
etiqueta.place(x=70, y=140, width=100, height=30)  
boton.place(x=60, y=40, width=100, height=30)  
C_Entrada.place(x=260, y=240, width=100, height=30)
```

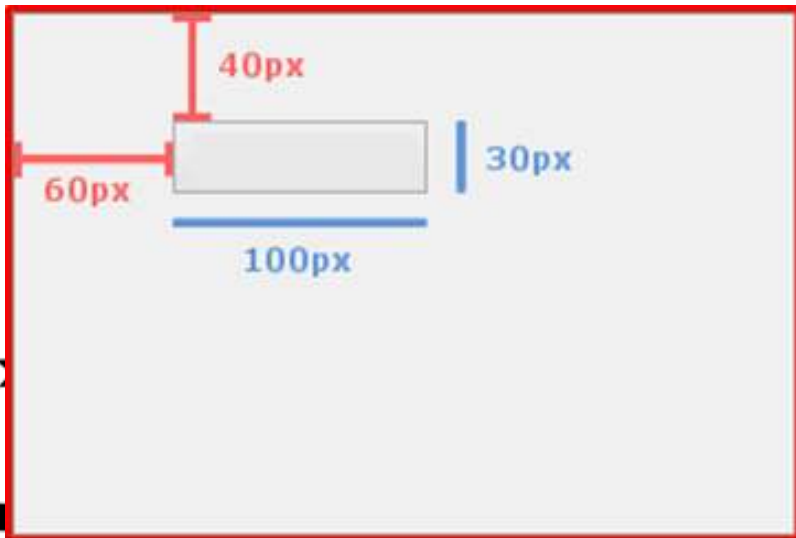


# EDICIÓN

```
etiqueta.pack()  
boton.pack()  
C_Entrada.pack()
```

*Al cambiar la forma de poner los objetos en la ventana con `.pack` por `.place` se puede fijar su ubicación.*

```
"""Fija ubicación y tamaño"""  
etiqueta.place(x=70, y=140, width=100, height=30)  
boton.place(x=60, y=40, width=100, height=30)  
C_Entrada.place(x=260, y=240, width=100, height=30)
```



# GRAFICAS DE TENDENCIA

```
4 from tkinter import *
5 """Matplotlib es una libreria para publicar graficas de tendencia."""
6 import matplotlib.pyplot as plt
7
8 def clic():
9     lista=[1,2,3,4,5]
10    plt.plot(lista)
11    """Etiquetas"""
12    plt.xlabel("Valores eje x")
13    plt.ylabel("Valores eje y")
14    plt.title("Titulo")
15    """Activar grilla"""
16    plt.grid(True)
17    """Mostrar gráfico"""
18    plt.show()
19
20 """Función principal"""
21 if __name__ == "__main__":
22     Aplicacion=Tk()
23     Aplicacion.title("Graficar")
24     boton=Button(Aplicacion,text="Gafique",command=clic)
25     boton.pack()
26     Aplicacion.mainloop()
```



# EJERCICIO



# EJERCICIO

- En una interfaz gráfica hay tres campos de entrada; ID (números enteros), nombre (texto) e ingresos mensuales (decimal con punto). La interfaz tiene la capacidad de almacenar temporalmente la información de un grupo de personas (cuando se cierra la aplicación la información almacenada desaparece), la cual se actualiza al pulsar un botón. Además dispone de otros tres botones; El primero permite ordenar a los usuarios ingresados por ID. El segundo permite seleccionar la forma de ordenamiento (ascendente o descendente, con el método burbuja) y el tercero permite graficar los ingresos mensuales almacenados temporalmente, de acuerdo con el orden dado a la lista de personas ingresadas. Tenga en cuenta que las etiquetas del eje x son el nombre de las personas y del eje y los ingresos mensuales.

