

PROGRAMACIÓN DE ALGORITMOS

PGY1121

Experiencia de Aprendizaje 2

DuocUC



ESCUELA DE
INFORMÁTICA Y
TELECOMUNICACIONES



Objetivos de la sesión

OBJETIVO 1



Utilizar el lenguaje de programación Python para dar respuesta a diferentes casos propuestos.

OBJETIVO 2

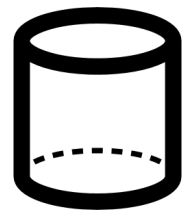


Utilizar las diferentes operadores y sentencias de lectura y escritura para resolver enunciados dados.

Introducción a Python

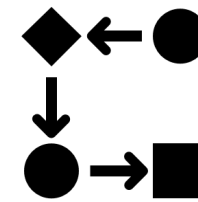


Es un **lenguaje** de programación que tiene las siguientes características:



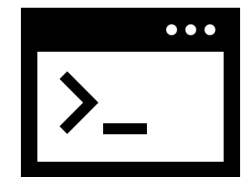
INTERPRETADO

Definido como interpretado y orientado a objetos.



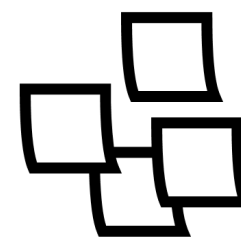
SENCILLO

Sencillo de leer y escribir, dado que es muy similar al lenguaje humano.



MULTIPLATAFORMA

Es un lenguaje multiplataforma de código abierto, lo que implica que es gratuito, permitiendo desarrollar software sin límites.



VARIADO

Posee una gran cantidad de librerías y funciones ya definidas, en su extensa biblioteca.



SOPORTE

Tiene un gran soporte gracias a su comunidad.

Origen de Python



El nombre asignado a este lenguaje es creado por **Guido van Rossum**, en consideración al grupo de comedia llamado Flying Circus de Monty Python.

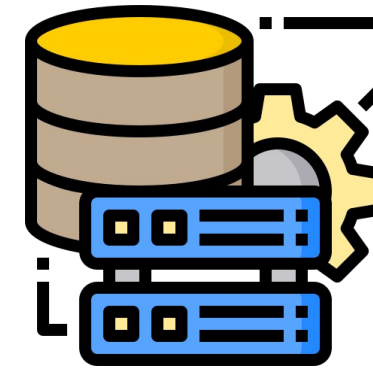


Origen de Python: https://es.wikipedia.org/wiki/Historia_de_Python

Python y su interacción



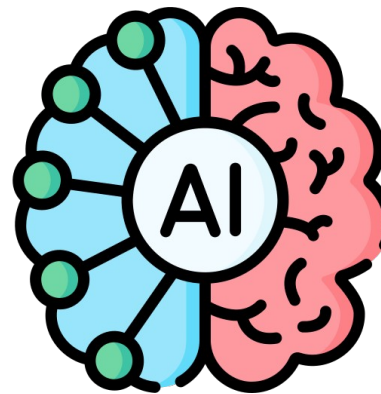
Desarrollo web



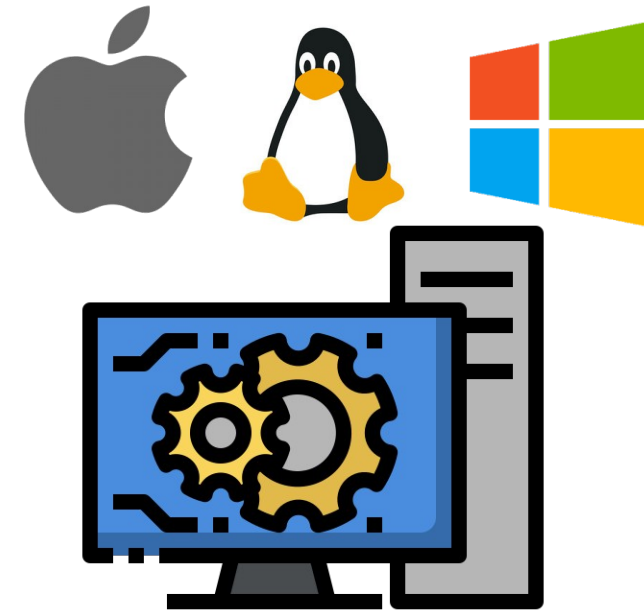
Ciencia de datos



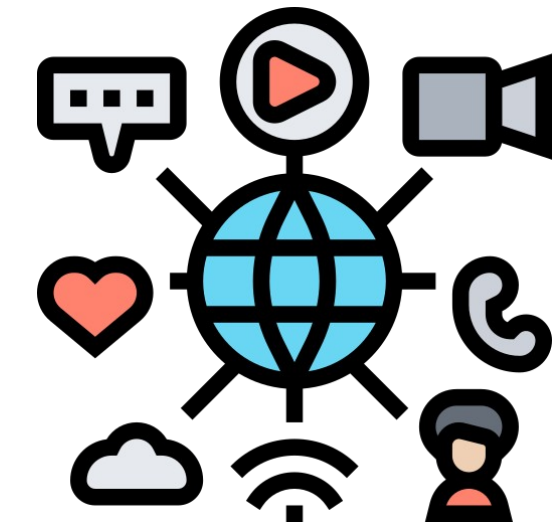
Desarrollo de juegos



Inteligencia artificial y aprendizaje automático



Sistemas operativos



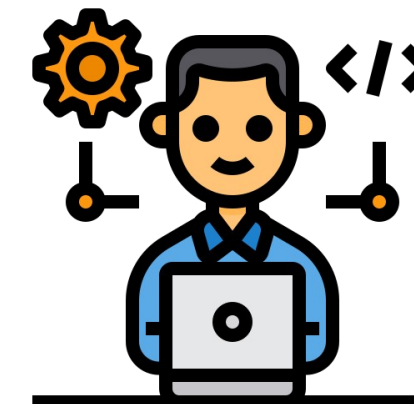
Aplicaciones de web scraping



Aplicaciones empresariales



Sector educativo



Desarrollo de software



GUI de escritorio



Python y su interacción



Busca otras instancias donde Python permite desarrollar aplicaciones y comenta con tus compañeros y docente



¿Dónde practicaremos Python?



Ver más:

<https://colab.research.google.com/notebooks/welcome.ipynb?hl=es>

Otras Aplicaciones como alternativa:

» Descargar Python:
<https://www.python.org/downloads/>

» Descargar Visual Studio Code:
<https://code.visualstudio.com/>

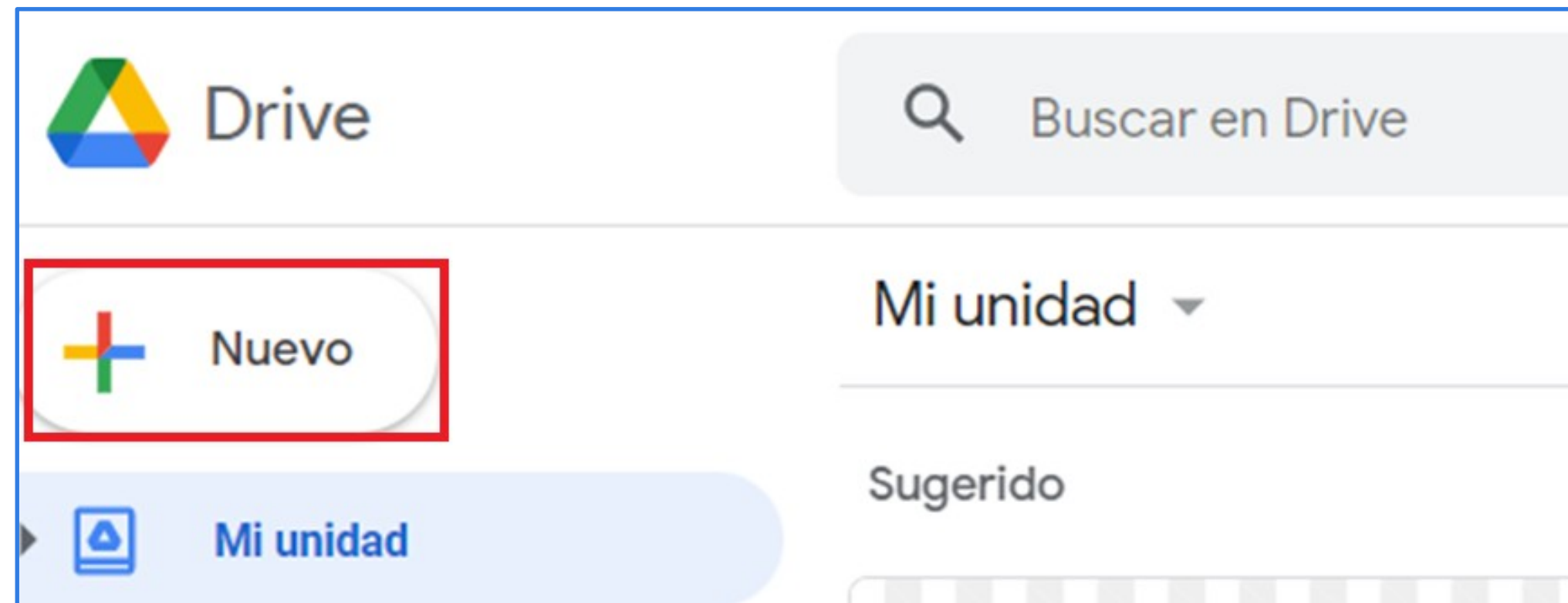
Para poner en práctica las instrucciones de Python, utilizaremos Colaboratory o "**Colab**" de **Google** Research.

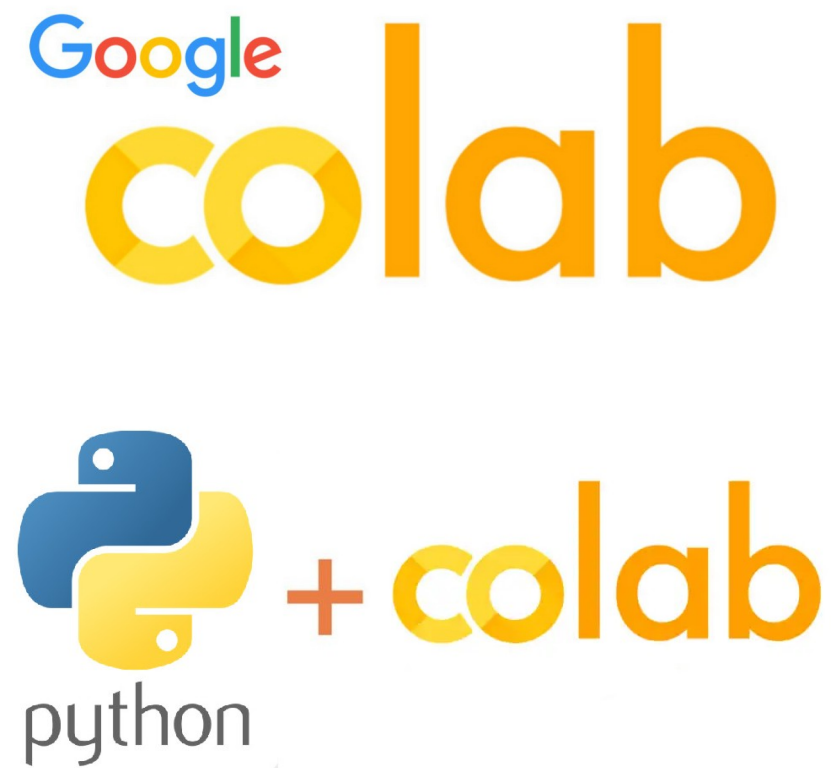
1. Con esto cualquier usuario podrá escribir y ejecutar código de Python en el navegador.
2. Entre sus ventajas no requiere configuración, da acceso gratuito a GPUs y permite compartir contenido fácilmente.
3. Se aplica para diversas tareas como de aprendizaje automático, análisis de datos y educación.
4. Los códigos que contengan mensajes o textos, evitaremos incluir las tildes, con el fin de evitar caracteres especiales.

Google Colab

Pasos relevantes:

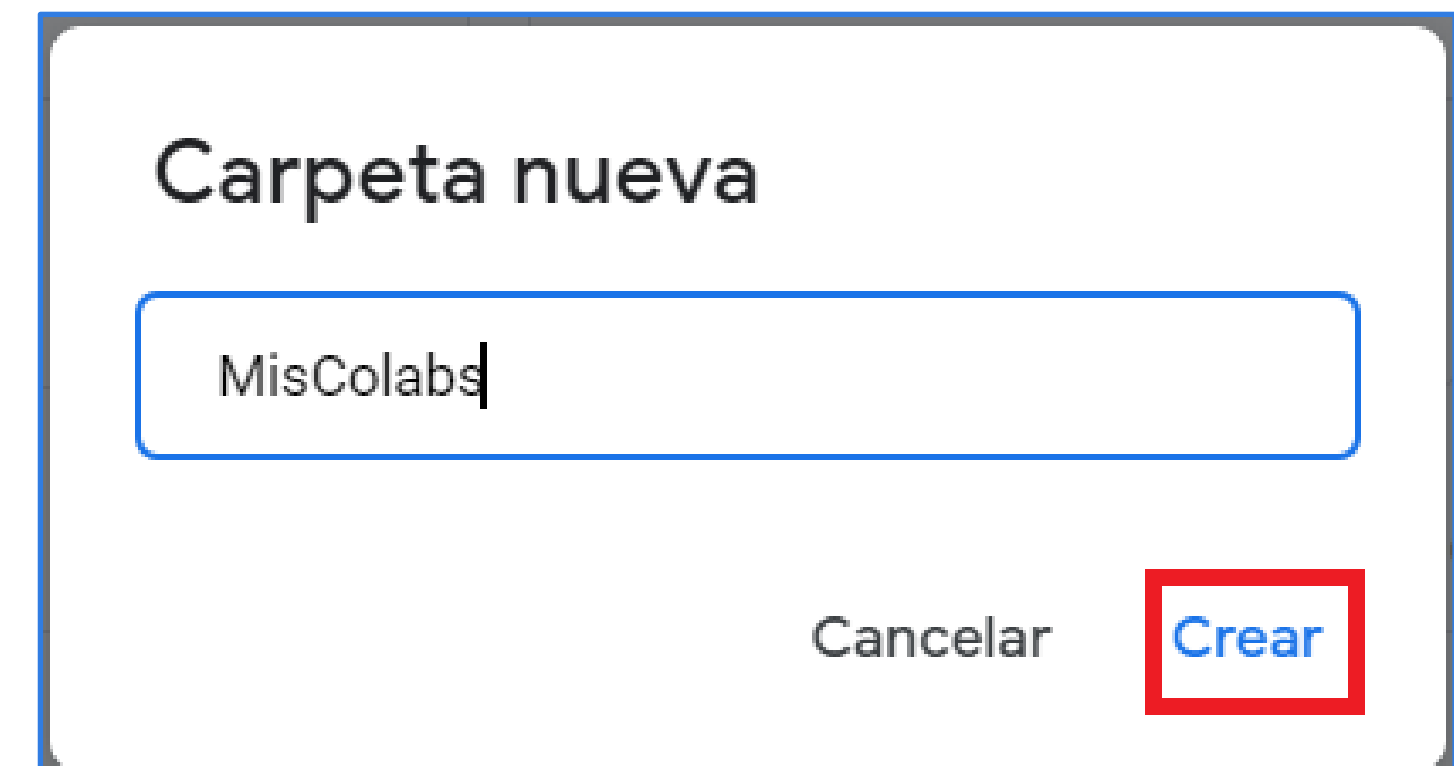
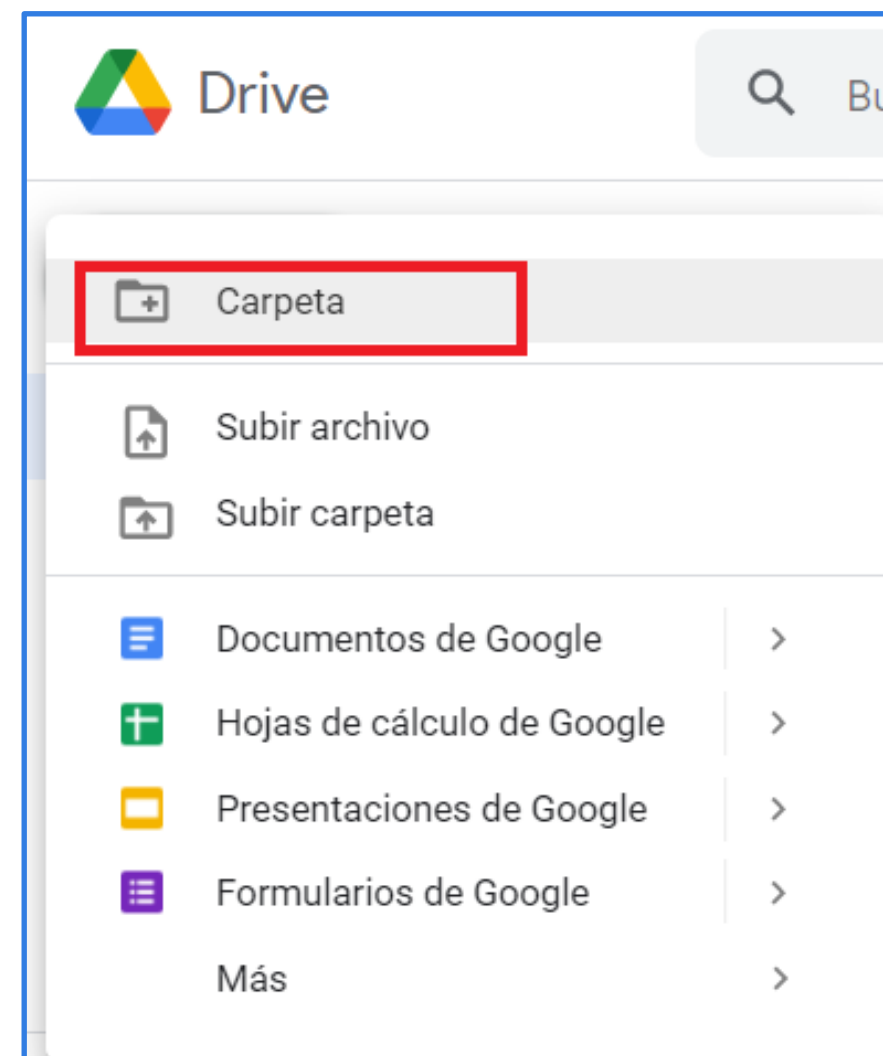
1. Tener una cuenta de [Google](#)
2. Acceder al servicio de [Google Drive](#)
3. Seleccionar [Nuevo](#)





Sigamos los pasos:

4. luego **Carpeta** (dar nombre, ejemplo: “MisColabs”)



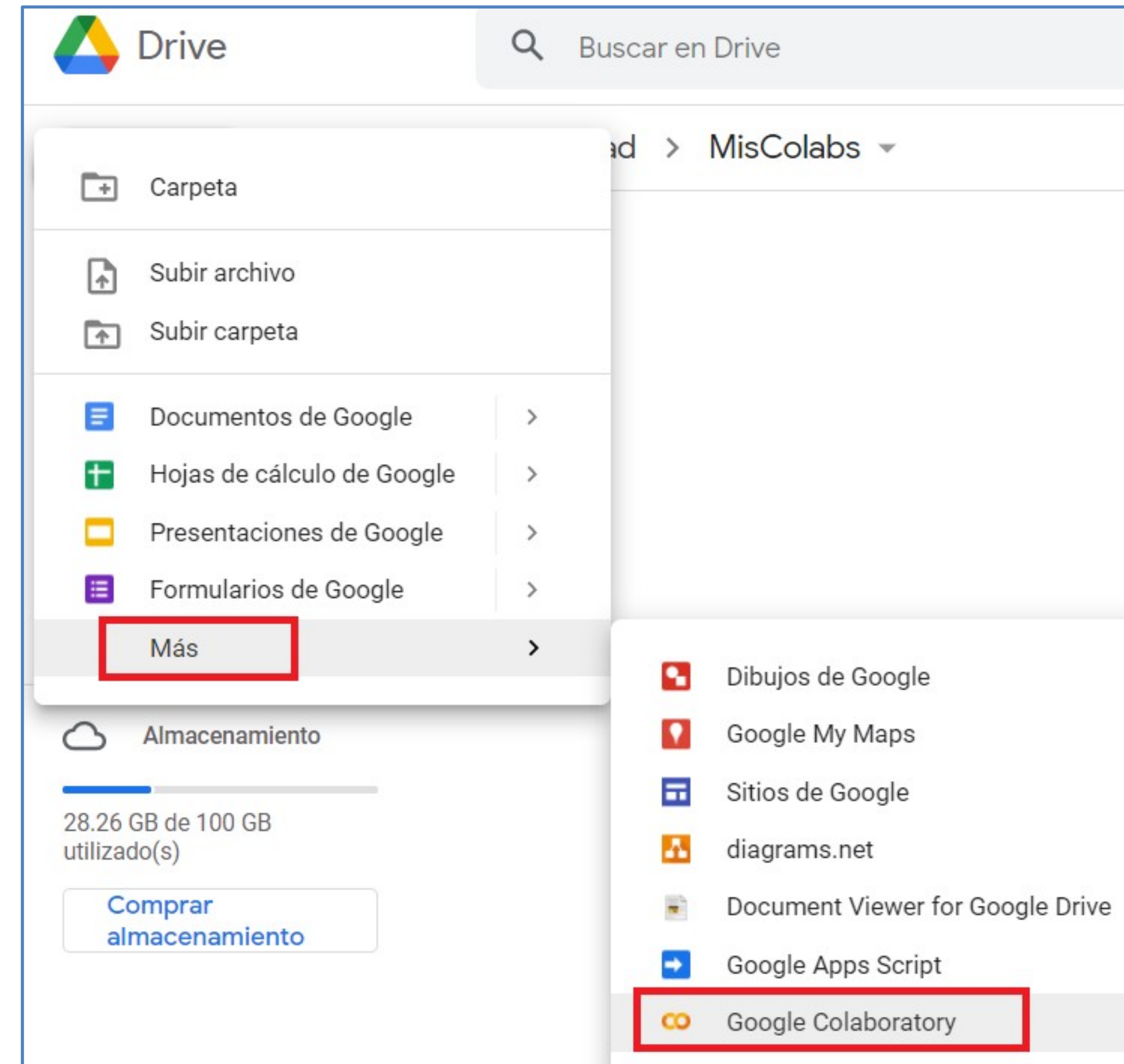
Google Colab

Google
colab

python + colab

Sigamos los pasos:

6. Acceder a la carpeta, luego presionar **Nuevo**.

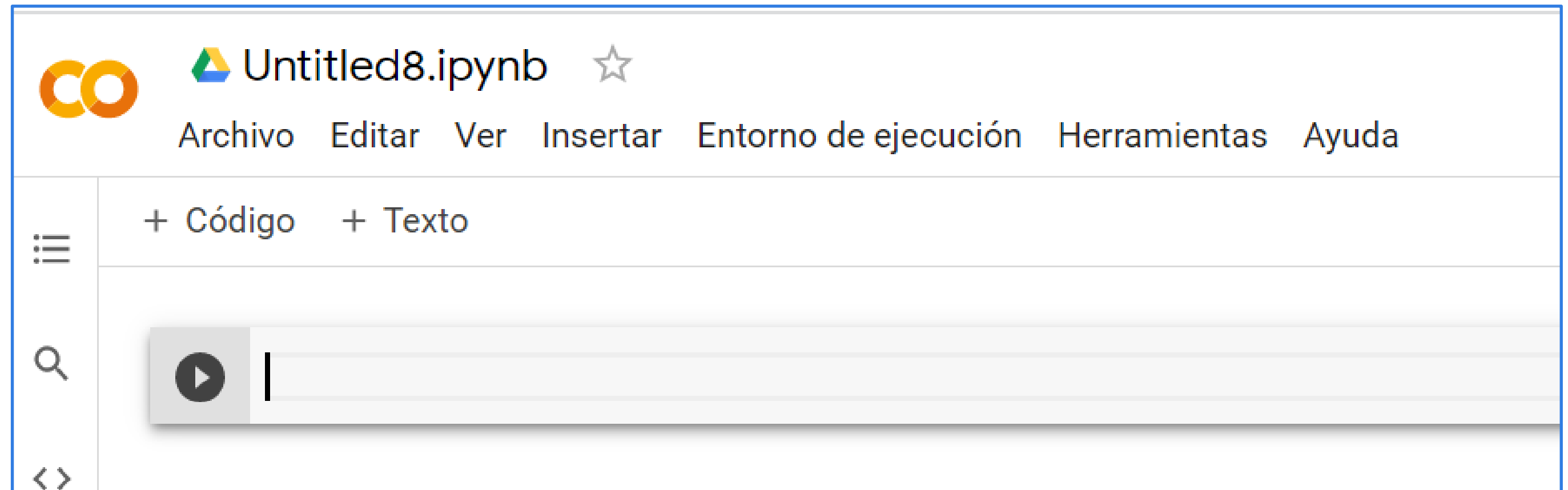


Google Colab



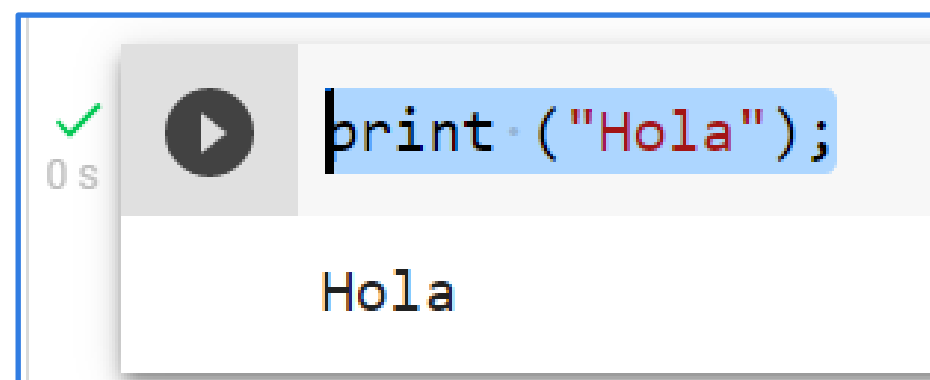
El resultado será:

Ya puedes escribir tu primera línea de código



Escribe tu primer código: `print ("Hola");`

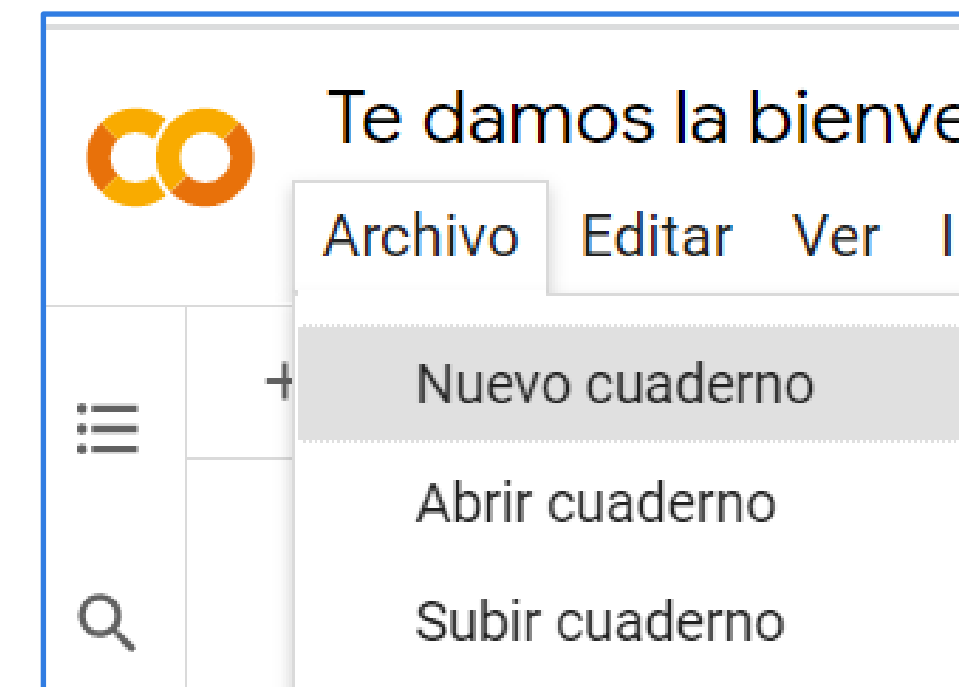
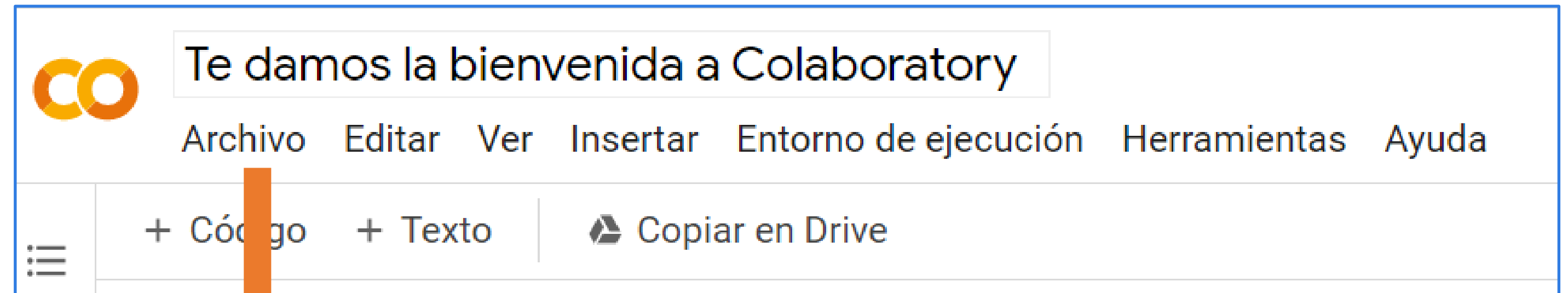
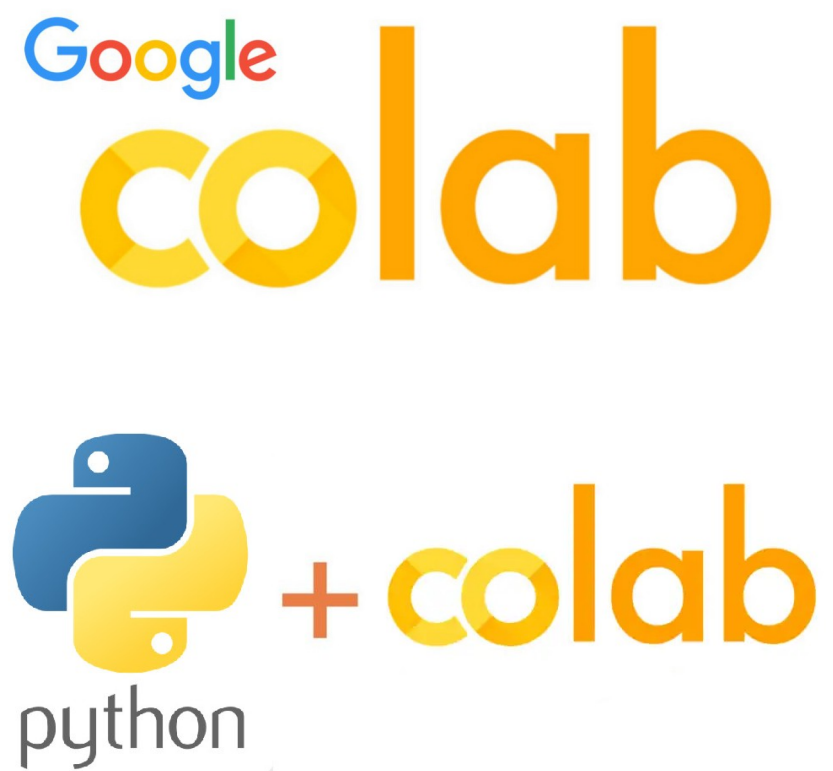
Para ejecutar, presiona



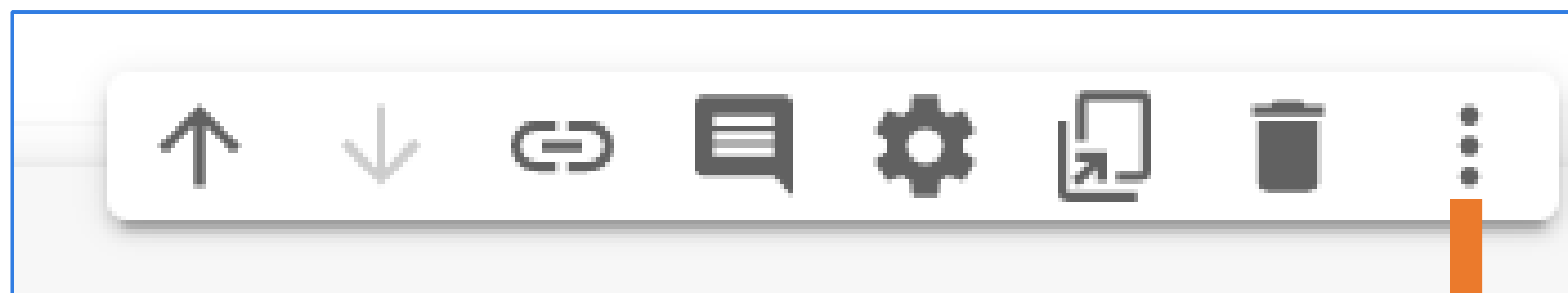
Para guardar el cuaderno con tu código, solo presiona la opción [Archivo](#) y encontrarás alternativas de grabar.

Google Colab

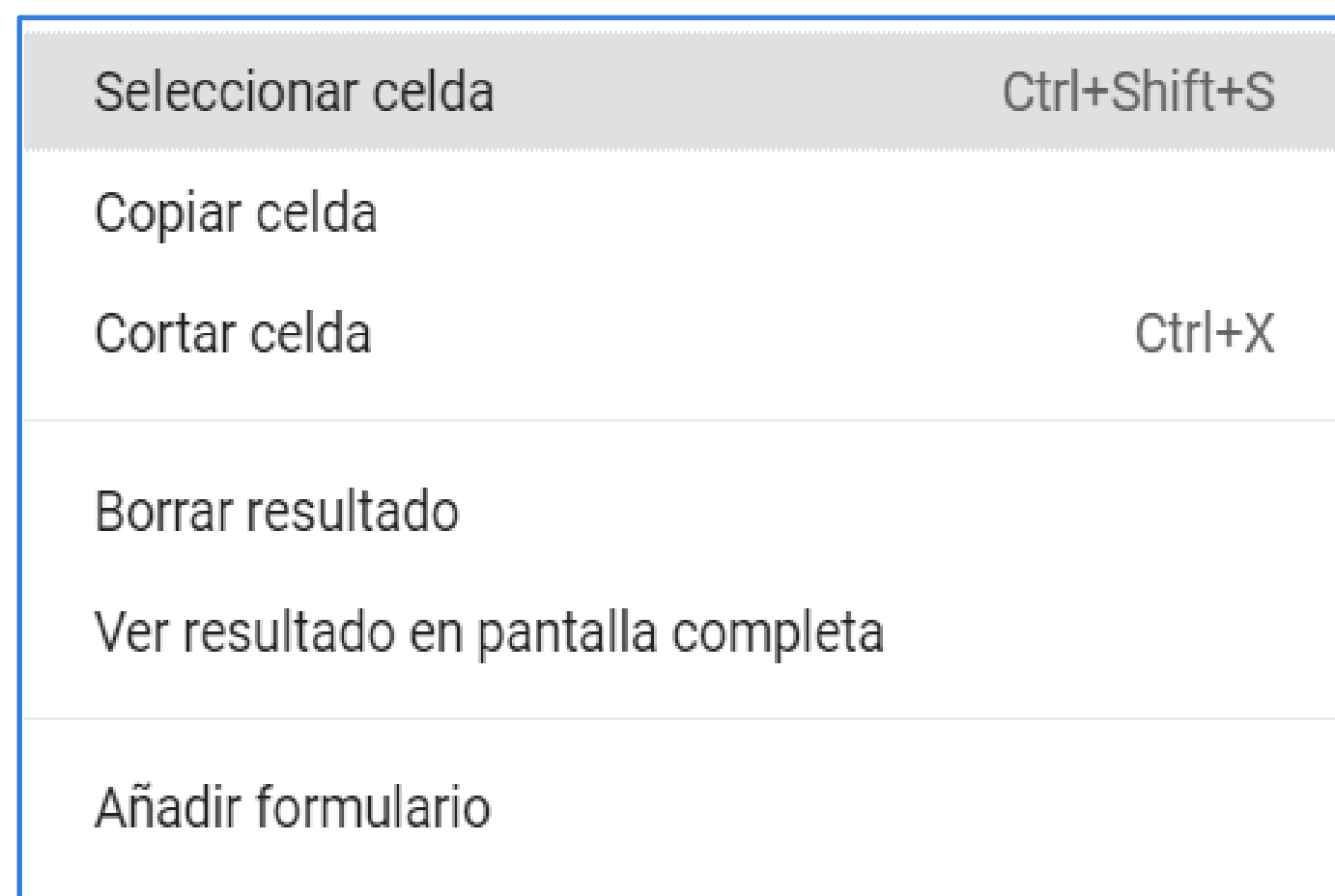
¿Quieres utilizar otro cuaderno?



Cada reglón de código tiene la barra de herramientas



Si seleccionas en más acciones, se despliega el siguiente cuadro.



Sigue explorando Google Colab con tu docente.

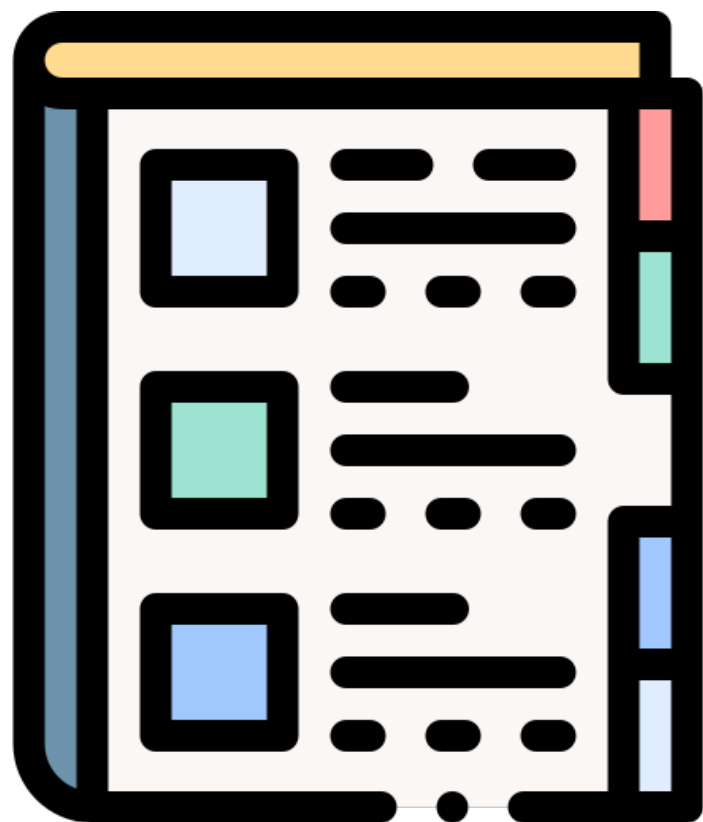


Google
colab

python + colab

Link de Preguntas frecuentes:
<https://research.google.com/colaboratory/faq.html>

Instrucciones de Python



Variables

- Utilizar nombres de tipo nemotécnico y en minúsculas.
- Para nombres compuestos, se utiliza el guión bajo.
- Deben considerar un espacio en blanco entre el operador =

Ejemplos:

- `num = 12`
- `nombre = "Juan"`

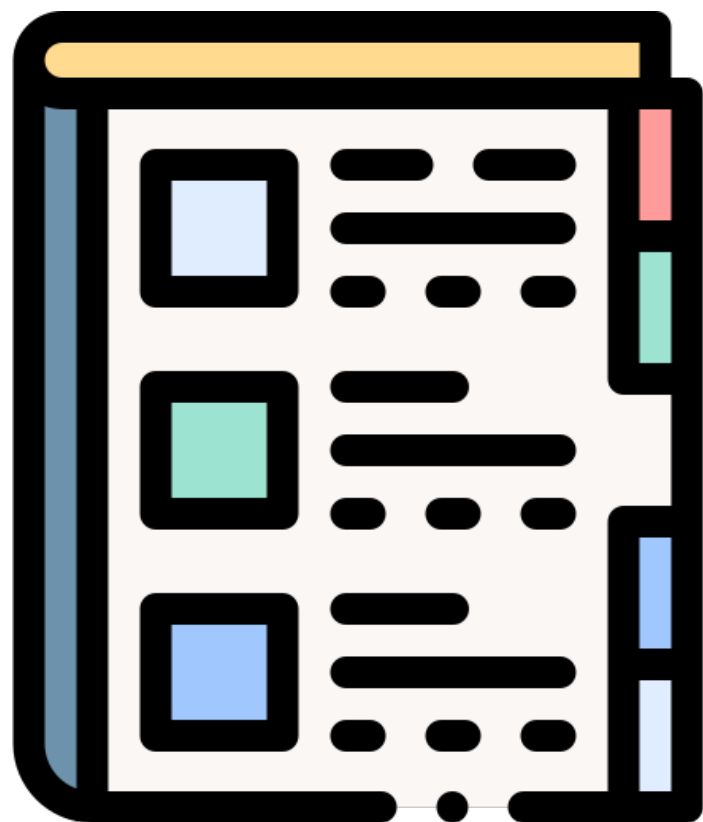
Constante

- Utilizar nombres de tipo nemotécnico y en mayúsculas.
- Para nombres compuestos, utilizar guiones bajos.

Ejemplos:

- `H = "Hola"`

Instrucciones de Python



Tipos de datos

Entero

☾ int

☾ Ejemplo: num = 23

Decimal

☾ float

☾ Ejemplo: num1 = 3.1416

Caracter

☾ chr

☾ Ejemplo: letra = 'M'

Cadena de Texto

☾ str

☾ Ejemplo: palabra = "Hola a todos"

Booleano

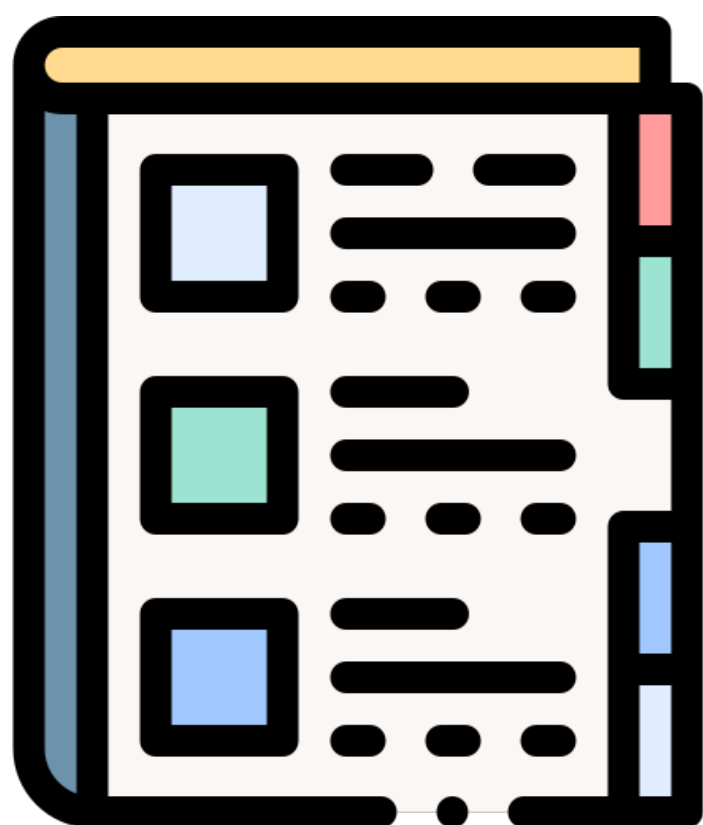
☾ bool

☾ Ejemplo: estado = True

Instrucciones de Python



Ejemplos:



Ingreso de Datos

Para ejecutar esta instrucción se utiliza la función `input()`, la cual recepciona un valor y lo entrega a una variable.

Mostrar Valores

Para mostrar texto y los valores de las variables, se utilizará la función `print()`

```
nombre = input("Ingrese su nombre: ")  
print(f"Su nombre es {nombre}")
```

```
nombre = input("Ingrese su nombre: ")  
print(f"Su nombre es: {nombre}")
```

➔ Ingrese su nombre: Juan
Su nombre es: Juan

```
edad = int(input("Ingrese su edad: "))  
print(f"La edad es: {edad}")
```

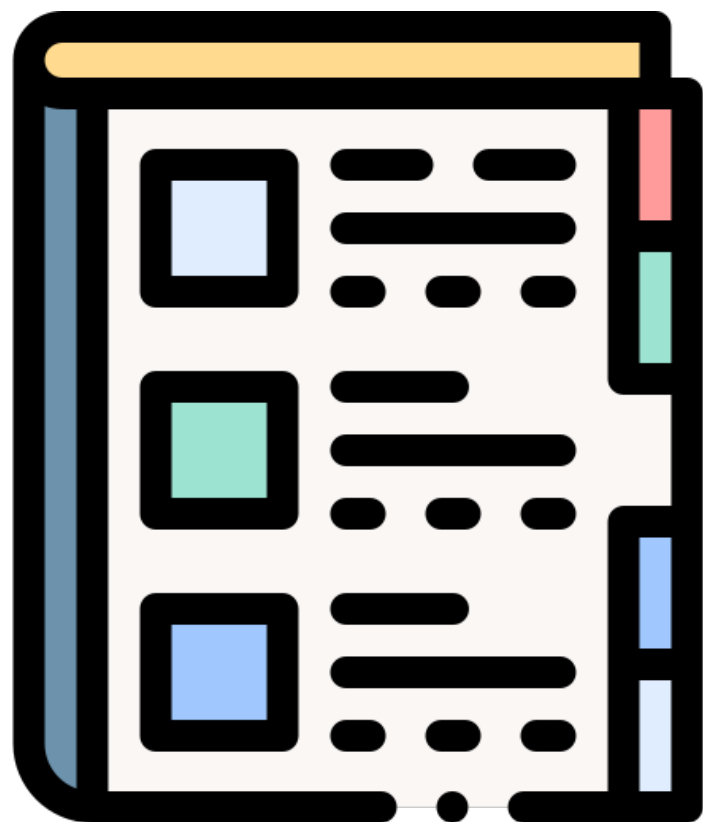
```
[6] edad = int(input("Ingrese edad: "))  
print (f"La edad es: {edad}")
```

Ingrese edad: 99
La edad es: 99

Los códigos que contengan mensajes o textos, evitaremos incluir las tildes, con el fin de evitar caracteres especiales.

Instrucciones de Python

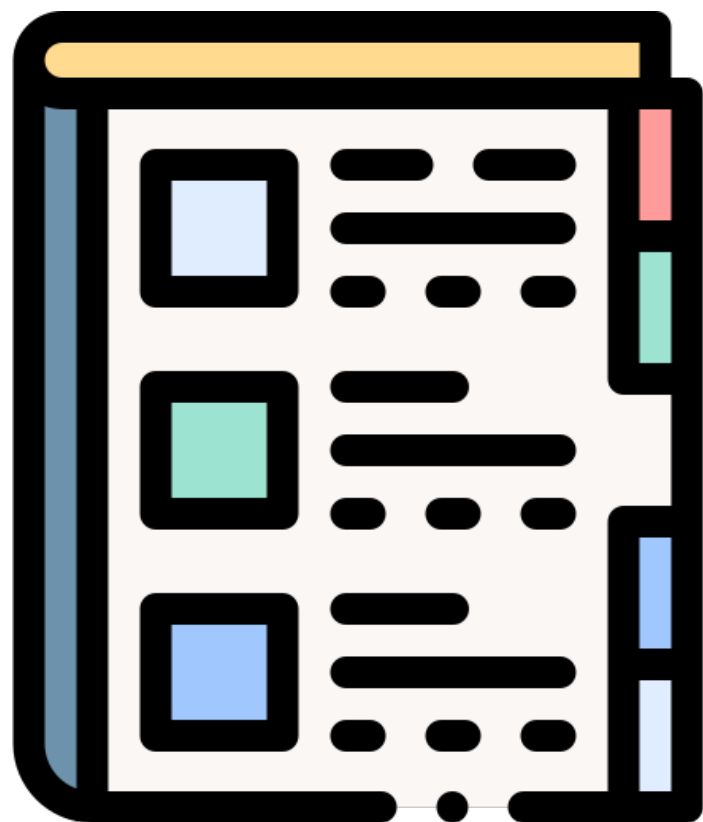
Operadores matemáticos



Símbolo	Significado	Ejemplo	Resultado
+	Suma	<code>a = 10 + 5</code>	a es 15
-	Resta	<code>a = 12 - 7</code>	a es 5
-	Negación	<code>a = -5</code>	a es -5
*	Multiplicación	<code>a = 7 * 5</code>	a es 35
**	Exponente	<code>a = 2 ** 3</code>	a es 8
/	División	<code>a = 12.5 / 2</code>	a es 6.25
//	División entera	<code>a = 12.5 / 2</code>	a es 6.0
%	Módulo	<code>a = 27 % 4</code>	a es 3



Instrucciones de Python



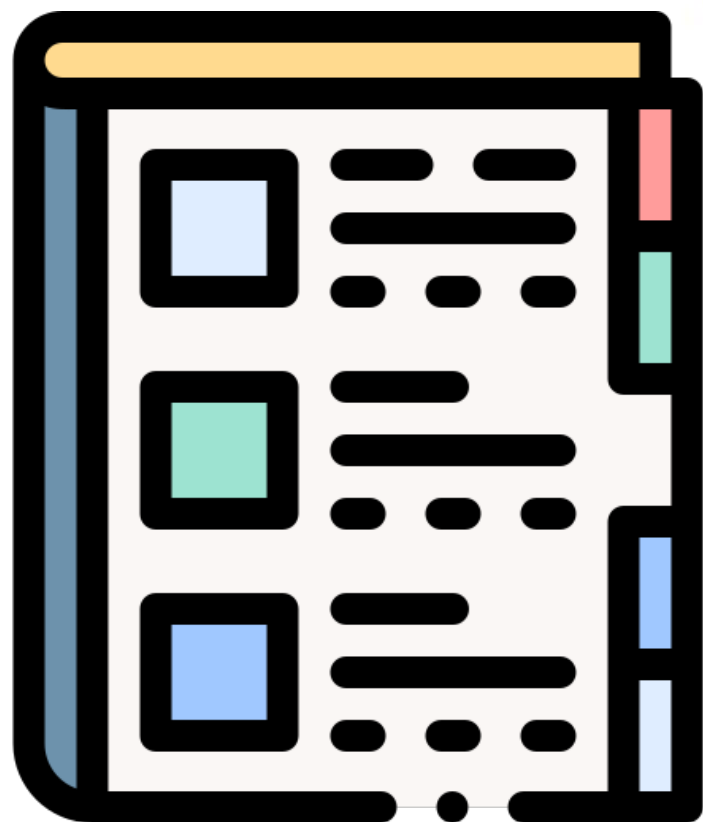
Operadores relacionales

Símbolo	Significado	Ejemplo	Resultado
<code>==</code>	Igual que	<code>5 == 7</code>	Falso
<code>!=</code>	Distinto que	<code>rojo != verde</code>	Verdadero
<code><</code>	Menor que	<code>8 < 12</code>	Verdadero
<code>></code>	Mayor que	<code>12 > 7</code>	Falso
<code><=</code>	Menor o igual que	<code>12 <= 12</code>	Verdadero
<code>>=</code>	Mayor o igual que	<code>4 >= 5</code>	Falso



Instrucciones de Python

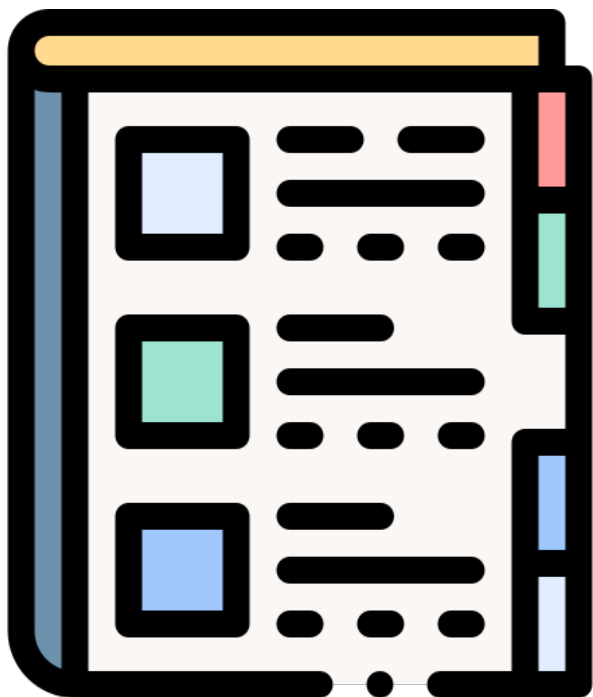
Operadores lógicos



Operador	Ejemplo	Resultado*
and (y)	5 == 7 and 7 < 12	0 y 0
	9 < 12 and 12 > 7	1 y 1
	9 < 12 and 12 > 15	1 y 0
or (o)	12 == 12 or 15 < 7	1 o 0
	7 > 5 or 9 < 12	1 o 1
xor	4 == 4 xor 9 > 3	1 o 1
(o excluyente)	4 == 4 xor 9 < 3	1 o 0

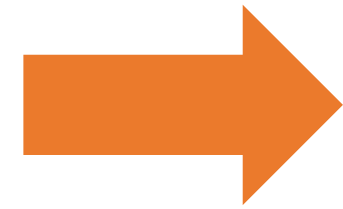


Instrucciones de Python



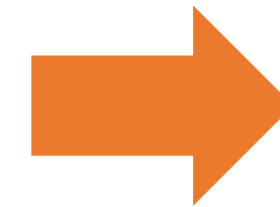
Comentarios en líneas de código

```
'''a = 2  
b = 3  
c = 4'''
```



Las `'''` al inicio y al final del bloque, permiten delimitar el bloque de comentarios de varias líneas.

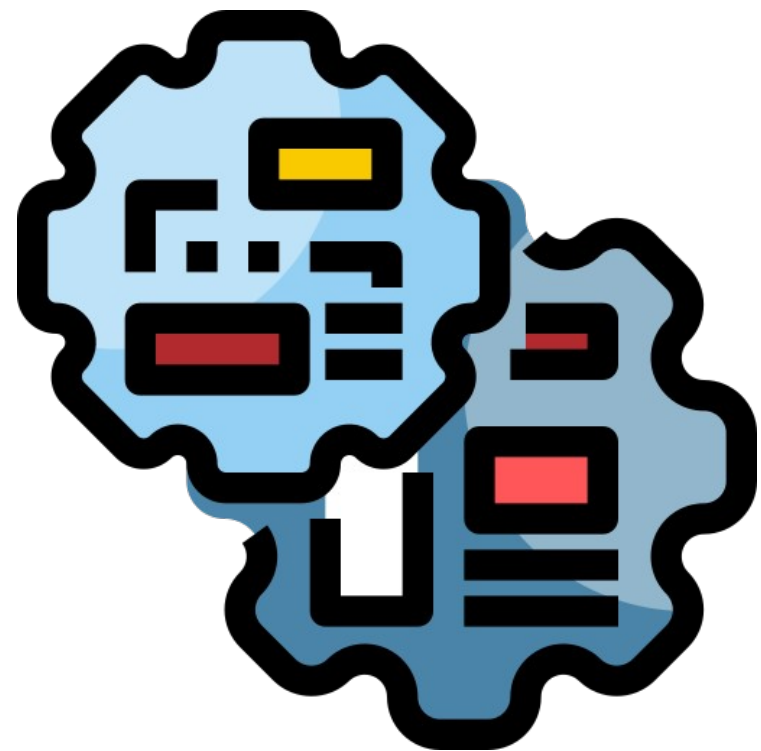
```
#if a == b and c == 4:
```



El símbolo `#`, al inicio de la línea permite comentar esa línea de código.

Ejercicios

Instrucciones: Desarrolle los enunciados, aplicando los contenidos vistos en la clase.



Ejercicio 1

Mostrar por pantalla los siguientes mensajes:

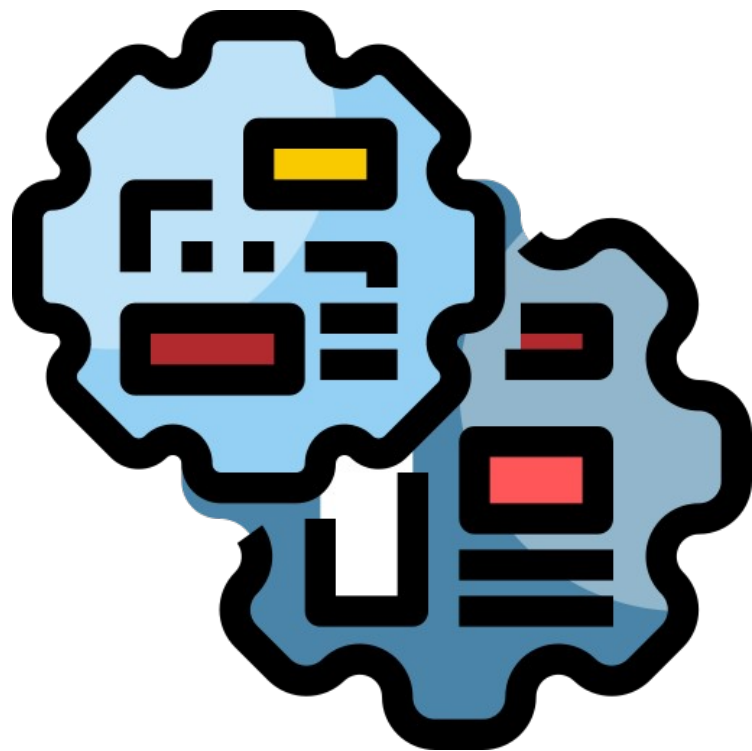
- “Bienvenido al mundo de la programación”.
- “Para comenzar, ingresa tu nombre”.

Ejercicios

Instrucciones: Desarrolle los enunciados, aplicando los contenidos vistos en la clase.

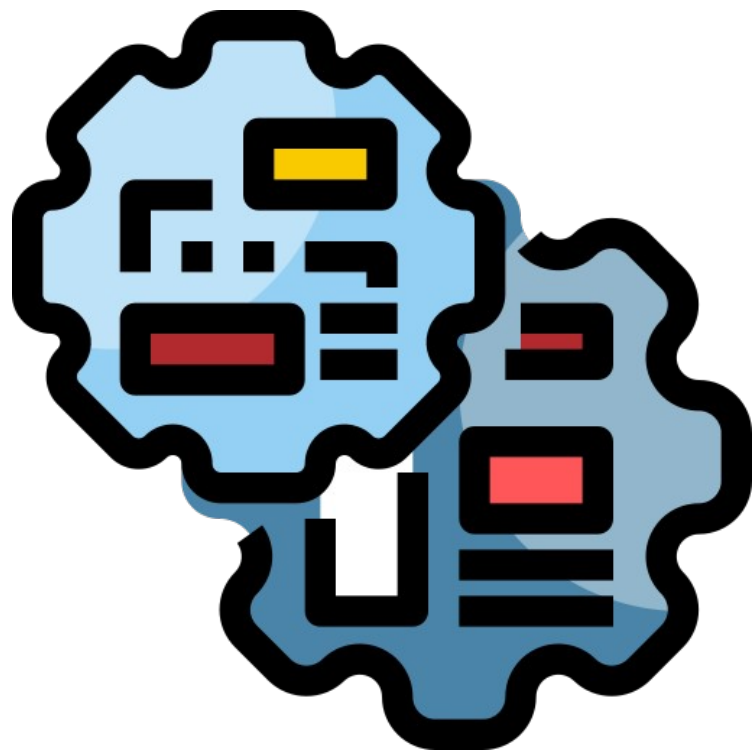


```
#1A  
print ("Bienvenido al mundo de la programación")  
#1B  
print ("Para comenzar, ingresa tu nombre :")
```



Ejercicios

Instrucciones: Desarrolle los enunciados, aplicando los contenidos vistos en la clase.



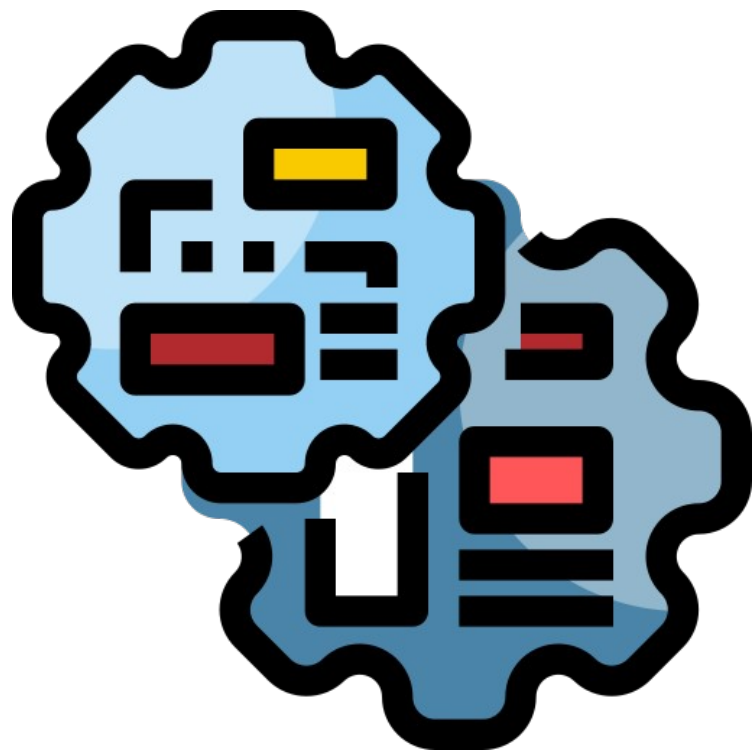
Ejercicio 2

Almacenar el nombre en una variable llamada “nom”, luego mostrar el siguiente mensaje usando `print (f”)`: “Bienvenido {nom}”

Debe imprimir el nombre ingresado por pantalla.

Ejercicios

Instrucciones: Desarrolle los enunciados, aplicando los contenidos vistos en la clase.



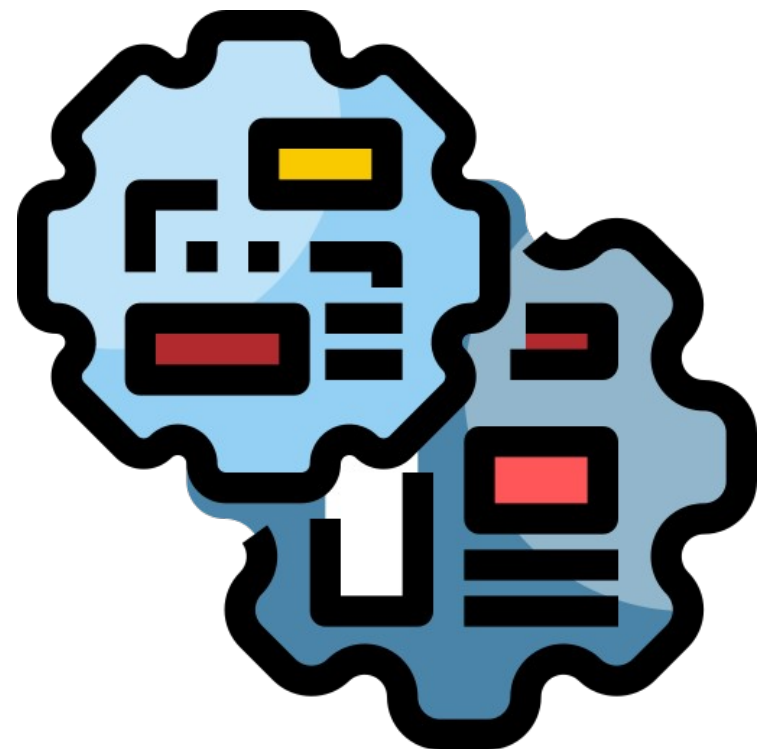
Alternativa de Solución: Ejercicio 2

#2

```
nom = input ("Ingrese su nombre: ")  
print (f"Bienvenido {nom}")
```


Ejercicios

Instrucciones: Desarrolle los enunciados, aplicando los contenidos vistos en la clase.



Ejercicio 3

Utilizando un ingreso de variables de tipo numérica con `int(input())`, se pide el valor de X para resolver la siguiente ecuación:

$$\frac{x^2 + 3x + 1}{4}$$

Muestre el resultado con un mensaje por pantalla.

Ejercicios

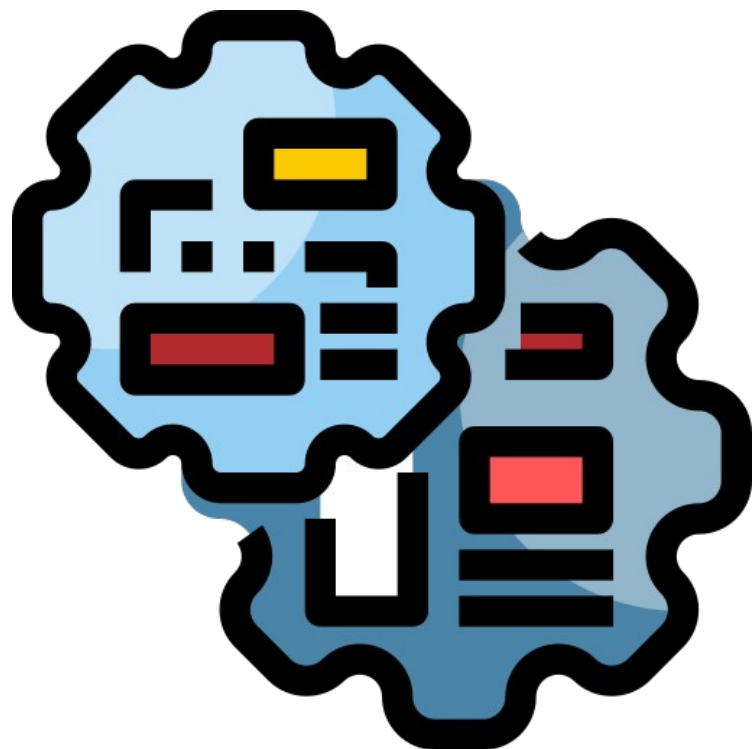
Alternativa de Solución: Ejercicio 3

Instrucciones: Desarrolle los enunciados, aplicando los contenidos vistos en la clase.



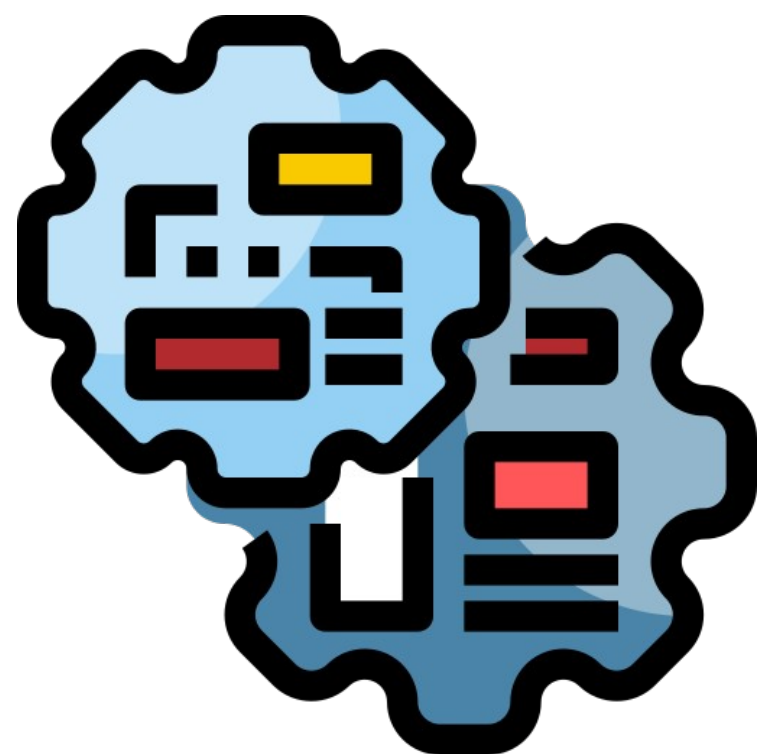
#3

```
x = int(input("Ingrese el valor de X: "))  
resultado = ((x**2)+3*x+1)/4  
print (f"El resultado de la ecuación es: {resultado}")
```



Ejercicios

Instrucciones: Desarrolle los enunciados, aplicando los contenidos vistos en la clase.



Ejercicio 4

Consultar datos de personas y mostrarlos de la siguiente forma:

NOMBRE: JUAN LOPEZ

RUT: 11.111.111-1

CORREO: JUAN.LOPEZ@GMAIL.COM

TELEFONO: 99999999

**Se recomiendan los siguientes comandos de impresión.
Ver siguiente slide**

Ejercicios

Instrucciones: Desarrolle los enunciados, aplicando los contenidos vistos en la clase.



1. Salto de línea “\n”

Ejemplo: `print (“Hola a todos \n soy informático”).`

Resultado:

Hola a todos
soy informático

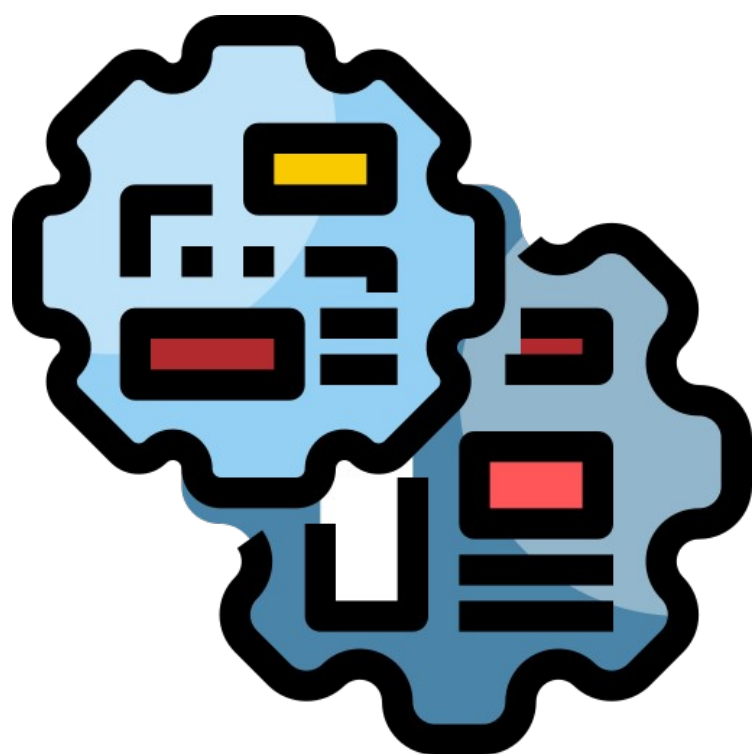
2. Tabulación “\t”

Ejemplo: `print(“Hol\ta \t todos\t\tsoy informático”).`

Resultado: Hola a todos soy informático.

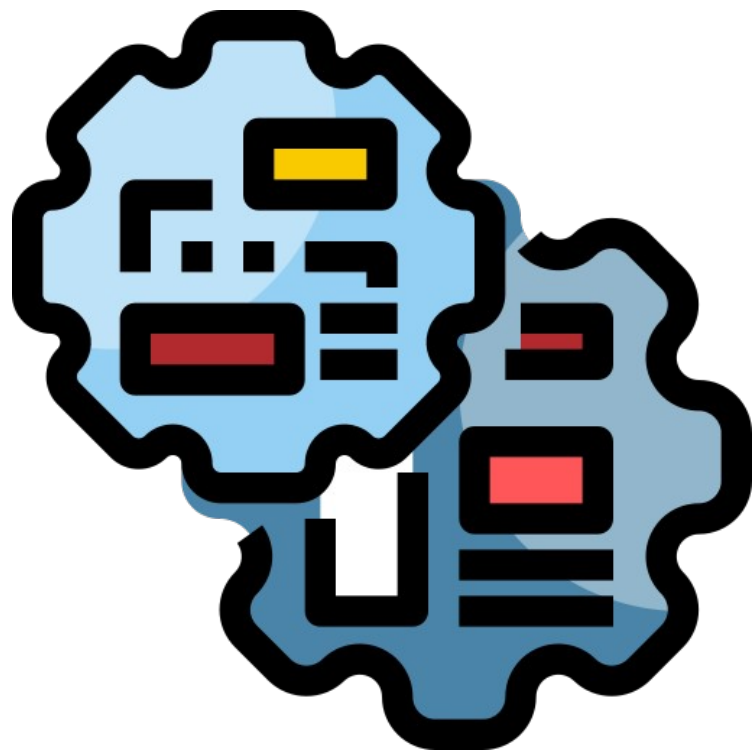
“El tabulador, la tecla Tab o tecla tabuladora del teclado, se utiliza para avanzar hasta el siguiente ‘tab stop’ ”:

→ → → → → → → → → 1 2 3 4 5 6 7 8 9



Ejercicios

Instrucciones: Desarrolle los enunciados, aplicando los contenidos vistos en la clase.



Alternativa de Solución: Ejercicio 4

```
#4
nom = input ("Ingrese nombre:  ")
rut = input ("Ingrese rut:      ")
correo = input ("Ingrese correo:  ")
telefono = int(input ("Ingrese telefono: "))

print (f"NOMBRE:\t\t{nom}")
print (f"RUT:\t\t{rut}")
print (f"CORREO:\t\t{correo}")
print (f"TELEFONO:\t{telefono}")
```


Sentencias de Control

Las sentencias de control, se utilizan para tomar decisiones lógicas. Estas evalúan una condición y en función de su resultado realizan una opción u otra.



Recurso de apoyo:
<https://www.youtube.com/watch?v=iFBcCDDDsms>

Sentencias de Control

Sentencia if

Esta sentencia es una expresión booleana que se evalúa como verdadera (True) o falsa (False), ¿lo recuerdas?, esta estructura la vimos en la Experiencia 1.

La sentencia **if** considera el uso de dos puntos (:) al final de su condición y la o las instrucciones que contenga deben estar indentadas.

Sintaxis:

```
if condición:  
    instrucción 1  
    instrucción 2  
    .....
```

Ejemplo:

```
▶ a = 3  
  b = 2  
  if a > b:  
      print("a es mayor que b")  
  
a es mayor que b
```



Sentencias de Control

Sentencia else

Esta sentencia se utiliza cuando la expresión del if no se cumple y también considera el uso de dos puntos (:) al final de su condición y la o las instrucciones que contenga también deben estar indentadas.

Sintaxis:


if condición:

instrucción 1
instrucción 2
.....

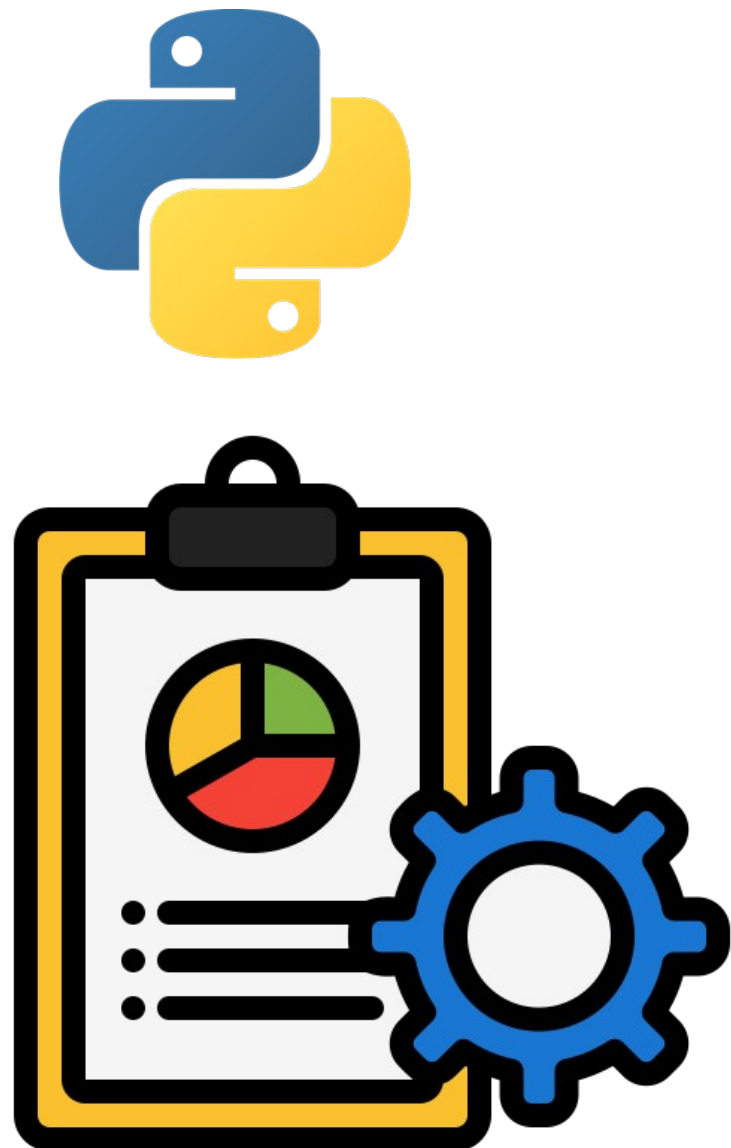
else:

instrucción 1
instrucción 2
.....

Ejemplo:

```
 a = 2  
b = 3  
if a > b:  
    print("a es mayor que b")  
else:  
    print("b es igual o mayor que a")
```

```
 b es igual o mayor que a
```



Sentencias de Control

Sentencia elif

La sentencia **if_else** puede tener un indeterminado número de condiciones para evaluar, no olvidar el uso de dos puntos (:) al final de su condición y la o las instrucciones que contenga deben estar indentadas.

Sintaxis:

if condición:

 instrucción 1
 instrucción 2

elif condición:

 instrucción 1
 instrucción 2

else:

 instrucción 1
 instrucción 2

.....

Cambia los valores de a y b,
para chequear los otros
resultados

Ejemplo:

```
a = 2
b = 3
if a > b:
    print("a es mayor que b")
elif a < b:
    print("b es mayor que a")
else:
    print("a y b son iguales")
```

b es mayor que a



Ejercicios

Ejercicio 1:

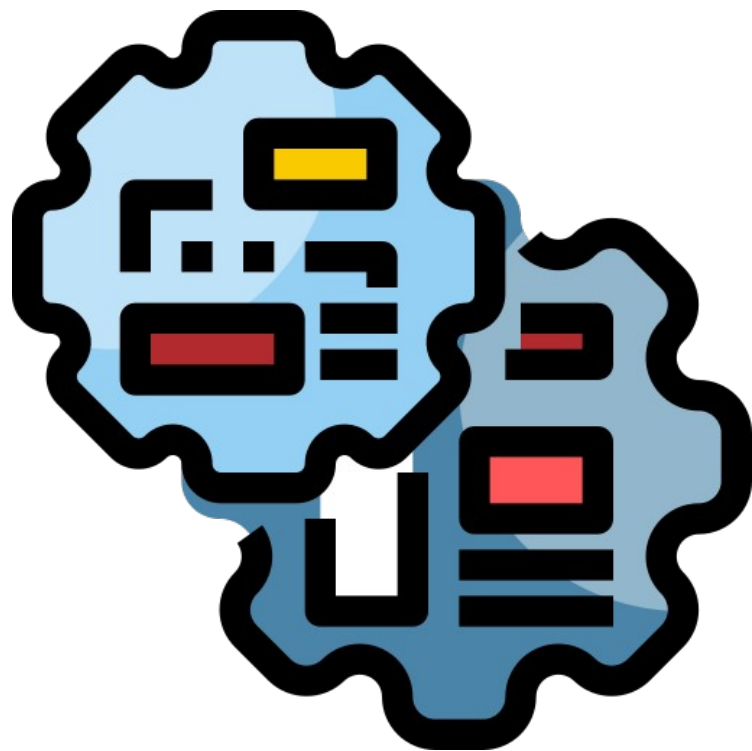
Se pide que construya un programa en Python que consulte la edad de una persona, y de acuerdo a lo ingresado indique si la persona tiene mayoría de edad o no.

Ejercicio 2:

Debe crear un sistema que permita la validación de usuario y contraseña de un empleado en la Empresa XY, los únicos dos usuarios conectados son:

a. User_1: pedro
Pass_1:1234

b. User_2: angel
Pass_2: a4s1



Ejercicios

Ejercicio 3:

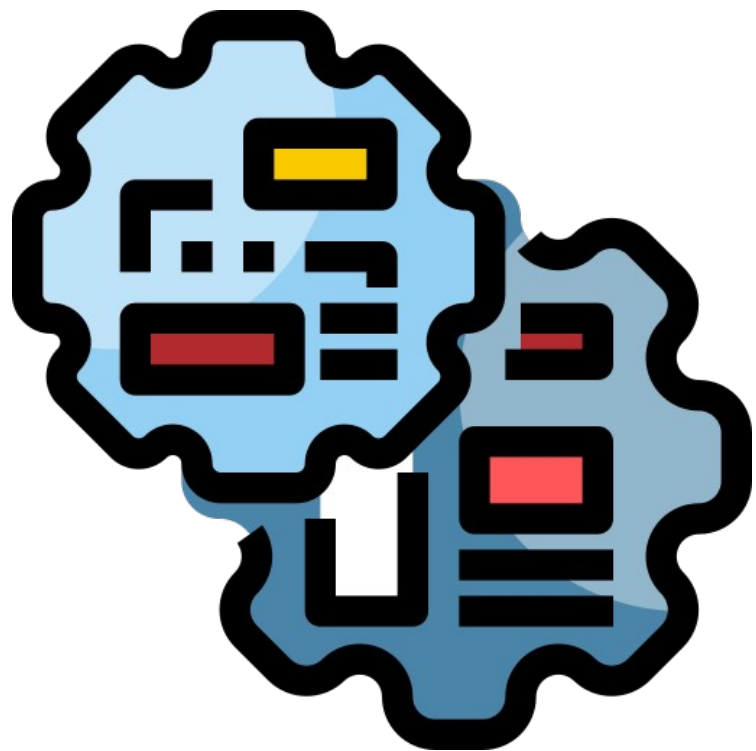
Ingrese 3 notas por teclado (valide que sean entre 1 y 7) y calcule su promedio. Si la nota resultante es mayor o igual a 4.0 entonces indique que está aprobado, en caso contrario notifique que está reprobado.

Ejercicio 4:

Genere un convertidor de:

- Dólar australiano a pesos chilenos
- Peso Argentino a peso chileno
- Yen a pesos chilenos

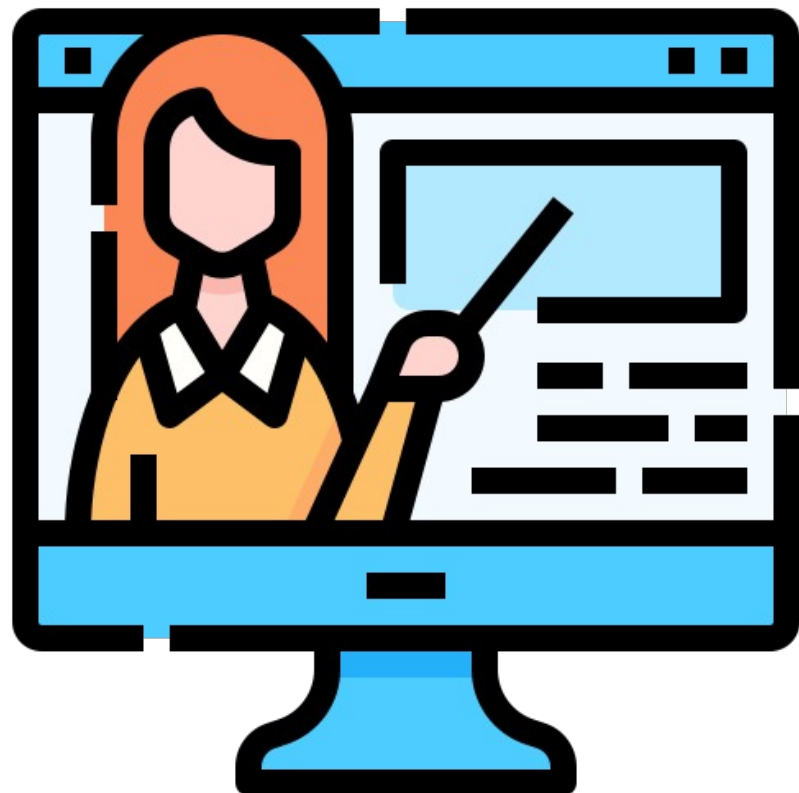
Considere que los valores son variables.

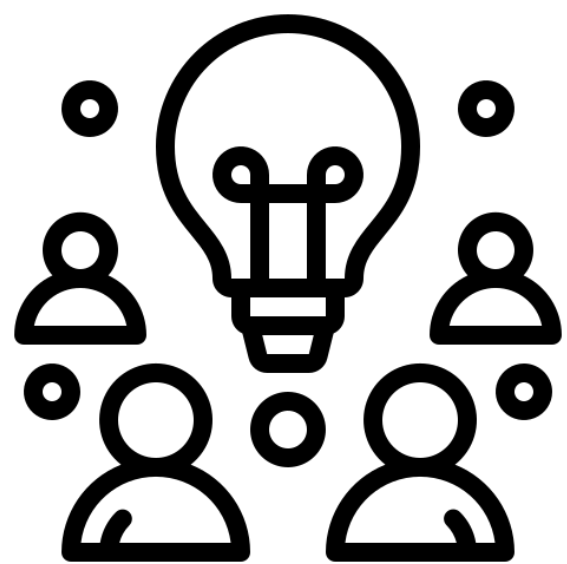


Curso de Python

Para potenciar aún más tus conocimientos sobre Python, te invitamos a participar de un [curso gratis de Cisco](#).

Revisa las instrucciones en **Material estudiantes**, el archivo **Instrucciones** [Curso Gratis de Python](#).





¿Cumplimos el objetivo de esta sesión?
¿Qué debo profundizar o ejercitar?
¿Cuál es el desafío para la siguiente sesión?