

¿Qué es Python?

Python es un lenguaje de programación popular. Fue creado por Guido van Rossum y lanzado en 1991.

Se utiliza para:

- desarrollo web (lado del servidor),
- desarrollo de software,
- matemáticas,
- secuencias de comandos del sistema.

¿Qué puede hacer Python?

- Python se puede usar en un servidor para crear aplicaciones web.
- Python se puede usar junto con el software para crear flujos de trabajo.
- Python puede conectarse a sistemas de bases de datos. También puede leer y modificar archivos.
- Python se puede usar para manejar big data y realizar operaciones matemáticas complejas.
- Python se puede utilizar para la creación rápida de prototipos o para el desarrollo de software listo para la producción.

¿Por qué Python?

- Python funciona en diferentes plataformas (Windows, Mac, Linux, Raspberry Pi, etc.).
- Python tiene una sintaxis simple similar al idioma inglés.
- Python tiene una sintaxis que permite a los desarrolladores escribir programas con menos líneas que otros lenguajes de programación.
- Python se ejecuta en un sistema de interpretación, lo que significa que el código se puede ejecutar tan pronto como se escribe. Esto significa que la creación de prototipos puede ser muy rápida.
- Python se puede tratar de forma procesal, orientada a objetos o funcional.

Bueno saber

- La versión principal más reciente de Python es Python 3, que usaremos en este tutorial. Sin embargo, Python 2, aunque no se

actualiza con nada más que actualizaciones de seguridad, sigue siendo bastante popular.

- En este tutorial, Python se escribirá en un editor de texto. Es posible escribir Python en un entorno de desarrollo integrado, como Thonny, Pycharm, Netbeans o Eclipse, que son particularmente útiles cuando se administran colecciones más grandes de archivos de Python.

Sintaxis de Python en comparación con otros lenguajes de programación

- Python fue diseñado para facilitar la lectura y tiene algunas similitudes con el idioma inglés con influencia de las matemáticas.
- Python usa nuevas líneas para completar un comando, a diferencia de otros lenguajes de programación que a menudo usan punto y coma o paréntesis.
- Python se basa en la sangría, usando espacios en blanco, para definir el alcance; como el alcance de los bucles, funciones y clases. Otros lenguajes de programación a menudo usan corchetes para este propósito.

RESUMEN DE PYTHON(E)

COMENTARIOS LINEALES Y MULTILINEALES(estos no afectan en nada)

```
PYTHON > primero.py
1  #python permite el tipado dinámico-comentario linea
2  ...
3  el tipado es que una misma variable puede contener diferentes tipos
4  de datos_comentario multilinea
5  ...
6  |
7
```

PEPITO

OPERACIONES MATEMÁTICAS

```
PYTHON > primero.py > ...  
1 #operaciones matemáticas  
2  
3 resultado=3**3*(13/5-(2*4))  
4 print(resultado)  
5  
6
```

```
[Running] python -u "c:\Users\ACER\Desktop\ELMER\PYTHON\primero.py"  
-145.8
```

OPERACIONES RELACIONALES

```
PYTHON > primero.py > ...  
1 #operaciones relacionales:< menor  
2 # mayor>  
3 # <=menor igual  
4 # mayor igual>=  
5 # diferente!=igual  
6 # ==  
7 a=20  
8 b=80  
9 c=100  
10 resultado=a+b==c  
11 print(resultado)  
12
```

```
[Done] exited with code=0 in 0.633 seconds
```

```
[Running] python -u "c:\Users\ACER\Desktop\ELMER\PYTHON\primero.py"  
True
```

OPERACIONES LÓGICOS

```
PYTHON > primero.py > ...  
1 #operaciones lógicos  
2 # negacion=not  
3 # or(disyuncion)=or suma  
4 # and(conjuncion)=and multiplicacion  
5  
6 # false=0  
7 # true=1  
8 a=10  
9 b=15  
10 c=20  
11 resultado=((a>b) and (b<c))  
12 print(resultado)
```

PEPITO

```
[Done] exited with code=1 in 0.668 seconds

[Running] python -u "c:\Users\ACER\Desktop\ELMER\PYTHON\primero.py"
False
```

OPERADORES DE ASIGNACION

```
ENTRADA
#operadores de asignacion
a=0
a+=5
print(a)#se hace de manera similar con otros operadores
SALIDA
[Running] python -u "c:\Users\ACER\Desktop\ELMER\PYTHON\primero.py"
5
```

SALIDA DE DATOS

```
ENTRADA
#SALIDAS
nombre="Elmeco"
edad=17

print("hola",nombre,"tienes",edad,"años")#primera opcion
print("hola {} tienes {} años".format(nombre,edad))#segunda opcion
print(f"hola{nombre}tienes{edad}años")#tercera opcion
SALIDA
[Running] python -u "c:\Users\ACER\Desktop\ELMER\PYTHON\primero.py"
hola Elmeco tienes 17 años
```

ENTRADA DE DATOS POR CONSOLA

```
#ENTRADA DE DATOS

nombre=input("digite su nombre: ")
print(f"hola{nombre}")
digite su nombre:ELMER
holaELMER

Process finished with exit code 0
```

Almacenando numeros

```
numero=int(input("digite un número:"))
print(f"el numero es {numero}")
resultado

digite un número:5
el numero es 5
```

Process finished with exit code 0

FUNCIONES INTEGRADAS

```
#FUNCIONES INTEGRADAS
n=len("python247")
#round=funcion de redondear
#abs=valor absoluto
#bin=binario
#int=entero
#len=contar los caracteres
print(n)
salida
9
```

CAPÍTULO 1:

ELEMENTOS BASICOS

Ejercicio 1:

- Escribir la siguiente expresión en forma de expresión algorítmica

$$\frac{a^3 \times (b^2 - 2ac)}{2b}$$

```
#EJERCIO-1
a=float(input("a->"))
b=float(input("b->"))
c=float(input("c->"))
resultado=(a**3*(b**2-2*a*c))/(2*b)
print(f"el resultado es:{resultado}")
SALIDA
a->4
b->5
c->6
el resultado es:-147.2
```

Ejercicio 2:

- Determinar la solución lógica de la siguiente operación:

$$((3+5 \times 8) < 3 \text{ and } ((-\frac{6 \times 4}{3}) + 2 < 2)) \text{ or } (a > b)$$

PEPITO

```
#EJERCIO-2
a=float(input("a->"))
b=float(input("b->"))
resultado=((3+5*8)<3 and ((-6/3*4)+2<2))or(a>b)
print(f"el resultado es:{resultado}")
```

SALIDA

```
a->10
b->11
el resultado es:False
```

EJERCICIO 3

```
#EJERCIO3=HACER UN PROGRMA QUE INTERCAMBIE VALOR DE 2 VARIABLES
a=input("a->")
b=input("b->")
```

```
a,b=b,a
print(f"el nuevo valor de a es:{b} ")
print(f"el nuevo valor de b es:{a} ")
```

SALIDA

```
a->3
b->4
el nuevo valor de a es:3
el nuevo valor de b es:4
```

EJERCICIO 4

Ejercicio 4:

- Hacer un programa para ingresar el radio de un círculo y se reporte su área y la longitud de la circunferencia.



$$\begin{aligned}\text{área} &= \pi * r^2 \\ \text{longitud} &= 2 * \pi * r\end{aligned}$$

```
#EJERCICIO-4
import math

radio=float(input("radio->"))
area=math.pi*radio**2
longitud=2*math.pi*radio
print(f"el area es:{area:.2f}")#2f es para mostrar solo dos decimales3
print(f"la longitud de la circunferencia es:{longitud:.2f}")
salida
radio->3
```

```
el area es:28.27
la longitud de la circunferencia es:18.85
```

EJERCICIO 5

Ejercicio 5:

- Una tienda ofrece un descuento del 15% sobre el total de la compra y un cliente desea saber cuánto deberá pagar finalmente por su compra.



```
#EJERCICIO-5S
precio=float(input("Digite el precio:"))
descuento=precio*15/100#o puede ser 0.15
precio_final=precio-descuento

print(f"El precio a pagar es ${precio_final:.2f}")
salida
Digite el precio:20
El precio a pagar es $:17.00
```

CAPITULO 2

```
#CONDICIONALES
numero=int(input("digite un numero:"))
if numero>0:#si el número
    print("el numero es positivo")
elif numero==0:#neutro
    print("el numero es 0")
else:#si no es....
    print("el número es negativo")

print("FELICITACIONES")#YA NO PERTENECE AL
salida
digite un numero:-5
el número es negativo
FELICITACIONES
```

```
#condicionales combinados
edad=int(input("Digite su edad:"))
```

```

if 0<edad<100:#si la edad mayor a cero y menor a 100
    print("Edad correcta")
    if edad>=18:#si la edad
        print("Es mayor de edad")

else:
    print("Edad incorrecta")
SALIDA
Digite su edad:17
Edad correcta

```

Ejercicio 1:

- Hacer un programa que pida 2 números y se de cuenta cuál de ellos es par, o si ambos lo son.

```

#EJERCICIO 1
num1=int(input("Digite un número:"))
num2=int(input("Digite un número:"))

if num1%2==0 and num2%2==0:#estamos sacando el residuo
    print("Ambos números son pares")
elif num1%2==0 and num2%2!=0:
    print(f"{num1} es par")
elif num1%2!=0 and num2%2==0:
    print(f"{num2} es par")

else :
    print("Ambos números son impares")
SALIDA EN LA CONSOLA
Digite un número:100
Digite un número:25
100 es par

```

```

#EJERCICIO 2
#HACER UN PROGRAMA QUE PIDA 3 NUMEROS Y DETERMINE CUAL ES EL MAYOR

a1=int(input("Digite un número:"))
a2=int(input("Digite un número:"))
a3=int(input("Digite un número:"))

if a1>=a2 and a1>=a3:

```



```

    print(f"El numero mayor es: {a1}")
elif a2>=a1 and a2>=a3:
    print(f"El número mayor es: {a2}")
elif a3>=a1 and a3>=a2:
    print(f"El número mayor es: {a3}")
SALIDA EN CONSOLA:

```

```

Digite un número:84
Digite un número:52
Digite un número:41
El numero mayor es: 84

```

#EJERCICIO 3

CREAR UN PROGRAMA QUE RECONOZCA SI ES UNA VOCAL

```

letra=input("Digite un carácter:").lower()#transforma a mayuscula

```

```

if letra=='a' or letra=='e' or letra=='i' or letra=='o' or letra=='u':
    print("Si es una vocal")

```

```

else:
    print("No es una vocal")

```

SALIDA:

```

Digite un carácter:g
No es una vocal

```

Ejercicio 4:

- Construir un programa que simule el funcionamiento de una calculadora que puede realizar las cuatro operaciones aritméticas básicas (suma, resta, multiplicación y división). El usuario debe especificar la operación con el primer carácter del nombre de la operación.

#EJERCICIO 4

```

num1= float(input("Digite un número:"))
num2= float(input("Digite un número:"))

```

```

operacion=input("Digite la operacion: ").upper()#para convertir d e minúcula
a mayuscula

```

```

if operacion=='S':

```

```

    suma=num1+num2
    print(f"\nLa suma es:{suma}")#\n para hacer un salto de linea
elif operacion=='R':
    resta= num1-num2
    print(f"\nLa resta es:{resta}")
elif operacion=='P' or operacion=='M':
    multiplicacion:num1*num2
    print(f"\nEl producto es:{multiplicacion}")
elif operacion=='D':
    division=num1/num2
    print(f"\nLa division es:{division:.2f}")
else:
    print(f"\nSe equivocó de la operación-----vuelva a intentarlo")
SALIDA:
Digite un número:4
Digite un número:5
Digite la operacion: d

La division es:0.80

```

```

#EJERCICIO 5
'''HACER UN PROGRMA QUE SIMLULE UN CAJERO AUTOMÁTICO CON UN SALDO INICIAL DE
DE $1000 Y TENDRÁ EL SIGUIENTE MENÚ
1:INGRSAR DINERO
2:RETIRAR DINERO
3:MOSTRAR DINERO DISPONIBLE
4:SALIR'''
saldo=1000
print("\t.:MENÚ:.")
print("1.Ingresar dinero en la cuenta")
print("2.Retirar dinero de la cuenta")
print("3.Mostar dinero de la cuenta")
print("4.Salir")
opcion=int(input("Digite un aopcion del menú: "))
print()

if opcion==1:
    extra=float(input("Cuanto dinero desea ingresar a la cuenta->"))
    saldo+=extra
    print(f"el dinero de la cuenta :{saldo}")
elif opcion==2:

```

PEPITO

```
retirar=float(input("Cuanto de dinero desea retirar->"))
if retirar>saldo:
    print("No tiene esa cantidad de dinero")
else:
    saldo-=retirar
    print(f"Dinero en la cuenta:{saldo}")
elif opcion==3:
    print(f"Dinero en la cuenta{saldo}")
elif opcion==4:
    print("Gracias por su preferencia-----vuelva pronto")
else:
    ("Error-----opcion inválida")
SALIDA:
    .:MENÚ:.
    .:MENÚ:.
1.Ingresar dinero en la cuenta
2.Retirar dinero de la cuenta
3.Mostar dinero de la cuenta
4.Salir
Digite un aopcion del menú: 2

Cuanto de dinero desea retirar->1024
No tiene esa cantidad de dinero
```

```
#CAPITULO 3_LISTAS
lista=["Lunes","Martes","Miércoles","Jueves","Viernes"]
print(lista[0:5])#indice 0 [0:3]solo llega una posicion antes
SALIDA:
['Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes']
```

```
#tuplas:ya no se puede agregar otro valor
tupla=(4,"hola",8,8,7,8)
lista=list(tupla)#de tupla lista /de lista a tupla :(tuple)
print(lista)
salida
[4, 'hola', 8, 8, 7, 8]
```

```
#conjuntos
'''no pueden haber duplicados'''
conjunto=set()#set=conjunto

conjunto={1,23,4,5,6,"elmec"}#dentro de esto no puede haber otras
colecciones
```

PEPITO

```
conjunto.add(7)#para agregar un elemento
conjunto.discard(23)#para eliminar un elemento
conjunto.clear()#para vaciar
print(7 in conjunto)
salida:
False
```

```
# operaciones con conjuntos
#a|b=union de conjuntos
#a&b interseccion de conjuntos
#a-b = diferencia de conjuntos
#c=a ^b =diferencia simetrica
#print(a.issubset(b))buscar si es subconjunto
a=frozenset({1,2,4,5,6})#frozenset=inmutable...ya no se puede agregar nada
b={6,9,1,2,3,5}
c={1,2,3,4,5,6}

print(a.isdisjoint(b))#disconexo=isdisjoint
SALIDA
False
```

PEPITO