PROGRAMACIÓN Y CIBERSEGURIDAD

PYTHON FOR REALLY BEGINNERS

DE UN ESTUDIANTE A OTRO

Ruben Apablaza Muñoz
-OZonE-

```
# Estas lineas me ayudaron a comprender esos primeros pasos y me ayudaron a
entender porque python se hace tan sencillo.
# Esta es la forma como yo estructure esa primera parte y esa primera
práctica.
# Comparto estos pasos para aquellos que estan recien dando esos baby
steps... una sintaxis sencilla y simple.
# Espero que este primer ejercicio te ayude a entender como aplique yo lo
que veia de diferentes fuentes.
# Solo copia y pega el script en python o en vscode y comienza a entender
como funciona Python.
# Si estas recien comenzando, comentar es aplicar # antes de una linea, por
lo cual
# al momento de ejecutar tus primeros scripts, esas lineas no serán
consideradas en la ejecución.

# El unico consejo que te puedo dar como alguien que aprendio de a poco es,
practica!!!,
# Python no es distinto de cualquier otro lenguaje, como ingles, practica
siempre aunque sean ejercicios pequeños,
# practica para mantener frescos los conocimientos. La IA ayuda hoy en dia,
pero somos nosotros
# los responsables del codigo, hay una delgada linea entre aprender y creer
que estas aprendiendo
# por que la IA hace todo el codigo.
```

```
# S T R I N G S ****************************************
# Las strings se caracterizan por estar entre comillas simples o dobles, y
pueden ser de una sola línea o varias líneas.
print ("Hola, mundo!")
print ("esta es ","una prueba")
print ("esta es","una prueba")
print ("esta es "+"otra prueba")
print ("Y esta también" "es otra prueba")
print ("Y esta también " "es otra prueba")
print ("""esta es
       una prueba de
       varias lineas""")
print ("esta es \nuna prueba de \nvarias lineas")
print ('hola como estas')




# O P E R A D O R E S   A R I T M É T I C O S
*********************************
print (2+2) #suma
print (2-2) #resta
print (2*2) #multiplicación
print (2/2) #división
print (round(2/3, 2)) #redondeo a 2 decimales
print (3//2) #división entera
print (int(2/2)) #convertir a entero
print (2**3) #potencia
print (2**0.5) #raíz cuadrada
print (9**1/3) #raíz cúbica
print (int(9**1/3)) #raíz cúbica convertida a entero
print (2%3) #módulo
```

```python
# V A R I A B L E S   Y   M E T O D O S ********************************
nombre = "Juan Salvador Gaviota el gran maestro"
edad = 30
promedio = 8.58

print (nombre) #imprimir variable
print (nombre) #convertir a mayúsculas
print (nombre.upper()) #convertir a mayúsculas
print (nombre.lower()) #convertir a minúsculas
print ("\n") #imprimir salto de línea
print (nombre.capitalize()) #convertir a mayúsculas la primera letra
print (nombre.title()) #convertir a mayúsculas la primera letra de cada
palabra
print ("\n"*3) #imprimir 3 saltos de línea
print (nombre.replace("a", "o")) #reemplazar una letra por otra
print (nombre.replace("a", "o").upper()) #reemplazar una letra por otra y
convertir a mayúsculas

print (len(nombre)) #numero de caracteres del string
print (len(nombre.replace(" ", ""))) #numero de caracteres sin espacios del
string

print ("\n*------------------ EJERCICIO ------------------*\n")

print ("Mi nombre es", nombre+",", "tengo", edad, "años y mi promedio es",
(round(promedio, 1)), "en la universidad") #imprimir varias variables
```

```
# U S E R   I N P U T
**********************************************************************
********************
input_nombre = input("¿Cuál es tu nombre? ")  # Solicitar nombre al usuario
input_edad = int(input("¿Cuál es tu edad? "))  # Solicitar edad al usuario
input_nota1 = float(input("¿Cuál es tu nota 1? "))  # Solicitar nota1 al
usuario
input_nota2 = float(input("¿Cuál es tu nota 2? "))  # Solicitar nota2 al
usuario
suma_notas = round((input_nota1) + (input_nota2) ,1)  # Sumar las notas
print(f"Hola {input_nombre}, tu edad es {input_edad} y la suma de tus notas
es {suma_notas:}")  # Imprimir el resultado con formato

print("\n")
```

```python
# F U N C I O N E S - block de código reutilizable (como un programa dentro de otro programa) ********************************************
    # Cuando se DEFINE una función las variables que se definen se llaman PARAMETROS
    # Cuando se LLAMA a una función las variables que se definen se llaman ARGUMENTOS

def salto_de_linea():
    print("\n")  # Función que imprime un salto de línea

def saludar(nombre):
    """Función que saluda a una persona."""
    print(f"Hola, {nombre}!") #Se define la función con un parámetro 'nombre'

saludar("Juan")  # Llamar a la función con un argumento

salto_de_linea() # se llama a la función salto_de_linea para imprimir un salto de línea

def sumar(a, b):
    return a + b

a= int(input("Ingrese el primer número: "))
b= int(input("Ingrese el segundo número: "))

sumar(a, b)

print (sumar(a, b))

#o podria hacerlo de esta manera:

def sumar2():
    a = int(input("Ingrese el primer número: "))
    b = int(input("Ingrese el segundo número: "))
    print ("La suma de los dos numeros es:", a + b)

sumar2()

salto_de_linea()
```

```python
# E X P R E S I O N E S   B O O L E A N A S   Y   O P E R A D O R E S   D E   C O M P A R A C I Ó N **********
    #El resultado de una expresión booleana es True o False
    #Los operadores de comparación son: ==, !=, >, <, >=, <=
    # "and" y "or" son operadores lógicos

BOOL1 = True
BOOL2 = False
BOOL3 = 5 > 3

print("BOOL1:", BOOL1)  # True porque BOOL1 es True
print("BOOL2:", BOOL2)  # False porque BOOL2 es False
print("BOOL3:", BOOL3)  # True porque 5 es mayor que 3
print (BOOL1 and BOOL2)  # False porque BOOL2 es False / con and si uno es False, el resultado es False
print (BOOL1 or BOOL1)  # True porque BOOL1 es True / con or si uno es True, el resultado es True
print (BOOL1 or BOOL3)  # True porque BOOL1 es True y BOOL3 es True, osea los dos son True.

BOOL4 = 5 == 5 # True porque 5 es igual a 5
BOOL5 = 5 != 3 # True porque 5 no es igual a 3
BOOL6 = 5 < 3 # False porque 5 no es menor que 3
BOOL7 = 5 > 3 # True porque 5 es mayor que 3
BOOL8 = 5 <= 5 # True porque 5 es menor o igual a 5
BOOL9 = 5 >= 3 # True porque 5 es mayor o igual a
```

```python
# C O N D I C I O N A L E S (if, elif, else)
**************************************************

def beber():

    edad = int(input("¿Cuál es tu edad? "))

    if edad >= 18:
        print("Puedes beber alcohol")
    elif edad >= 16:
        print("Puedes beber cerveza")
    else:
        print("No puedes beber alcohol ni cerveza")

beber()

# juntando con opeadores logicos

def beber2():

    edad2 = int(input("¿Cuál es tu edad? "))
    dinero = int(input("¿Cuánto dinero tienes? "))

    if edad2 >= 18 and dinero >= 1000:
        print("Puedes beber alcohol y tienes dinero suficiente")
    else:
        print("No puedes beber alcohol o no tienes dinero suficiente")

beber2()

#se puede mejorar y jugar con mas opciones pero por ahora lo dejo asi
```

```python
# L O O P S  ( B U C L E S ) (while, for)
**************************************************

# BUCLE FOR

#El bucle for itera sobre cada elemento de la lista vocales y lo imprime en
pantalla.

vocales = ['a', 'e', 'i', 'o', 'u']
for x in vocales:
    print(f"La vocal es: {x}")

# Otro ejemplo de bucle for sería:

for i in range(5):  # Itera desde 0 hasta 4
    print(f"El número es: {i}")

#aplicado a al comando ping:
import os
for ip in range (1,254):
    ping = os.system(f"ping 192.168.0.{ip} -c 1")
    if ping == 0:
        print(f"192.168.0.{ip} está activo")
    else:
        print(f"192.168.0.{ip} no responde")

        # IMPORTANTE: Ejecutar como administrador para que funcione
correctamente el comando ping en Windows.

# BUCLE WHILE

# El bucle while ejecuta el bloque de código mientras la condición sea
verdadera.

i = 1
while i < 10:  # Mientras i sea menor que 10
    print(f"El número es: {i}")
    i += 2  # Incrementa i en 2 en cada iteración


password = ""
while password != "1234":
    password = input("Ingrese la contraseña: ")
    if password == "1234":
        print("Contraseña correcta")
```

```
    else:
        print("Contraseña incorrecta, intente nuevamente")

print ("Acceso concedido")
```

```
# C A L C U L A D O R A  S I M P L E
**************************************************

num1= float(input("Ingrese el primer número: "))
operacion = input("Ingrese la operación a realizar (+, -, *, /): ")
num2 = float(input("Ingrese el segundo número: "))

while True:
    if operacion == "+":
        resultado = num1 + num2
        print(f"El resultado de {num1} + {num2} es: {resultado}")
        break
    elif operacion == "-":
        resultado = num1 - num2
        print(f"El resultado de {num1} - {num2} es: {resultado}")
        break
    elif operacion == "*":
        resultado = num1 * num2
        print(f"El resultado de {num1} * {num2} es: {resultado}")
        break
    elif operacion == "/":
        if num2 != 0:
            resultado = num1 / num2
            print(f"El resultado de {num1} / {num2} es: {resultado}")
            break
        else:
            print("Error: División por cero no permitida. Intente
nuevamente.")
            num2 = int(input("Ingrese el segundo número (diferente de cero):
"))
    else:
        print("Operación no válida. Intente nuevamente.")
        operacion = input("Ingrese la operación a realizar (+, -, *, /): ")
```

```python
# L I S T A S ******************************
# Se caracterizan por estar entre corchetes y pueden contener diferentes tipos de datos.

frutas  = ("manzana", "banana", "naranja", "kiwi", "pera") # esto es una tupla, no una lista

frutas = "manzana", "banana", "naranja", "kiwi", "pera"  # esto es una cadena de texto, no una lista

frutas = ["manzana", "banana", "naranja", "kiwi", "pera"]  # esto es una lista


print(frutas)  # Imprime la lista completa
print(frutas[2])  # Imprime el primer elemento de la lista
print(frutas[0:3])  # Imprime los elementos desde el índice 0 hasta el 2 (sin incluir el 3)
print(frutas[-1])  # Imprime el último elemento de la lista
print(frutas[1:])  # Imprime los elementos desde el índice 1 hasta el final
print(frutas[:2])  # Imprime los elementos desde el inicio hasta el índice 1 (sin incluir el 2)

frutas.append("banana")  # Agrega "banana" al final de la lista
print(frutas)  # Imprime la lista actualizada

print(len(frutas))  # Imprime la longitud de la lista
print(frutas.count("banana"))  # Cuenta cuántas veces aparece "banana" en la lista
print(frutas.index("kiwi"))  # Imprime el índice del primer elemento "kiwi" en la lista
print("pera" in frutas)  # Verifica si "pera" está en la lista

frutas.sort()  # Ordena la lista alfabéticamente
print(frutas)  # Imprime la lista ordenada

frutas.insert(2, "uva")  # Inserta "uva" en el índice 2
print(frutas)  # Imprime la lista actualizada

frutas.remove("banana")  # Elimina la primera aparición de "banana" de la lista
print(frutas)  # Imprime la lista actualizada

frutas.pop()  # Elimina el último elemento de la lista
print(frutas)  # Imprime la lista actualizada
```

```python
frutas.pop(1)  # Elimina el elemento en el índice 1
print(frutas)  # Imprime la lista actualizada

verduras = ["lechuga", "tomate", "cebolla", "pimiento", "zanahoria"]

frutas_y_verduras = frutas + verduras  # Combina las listas de frutas y verduras
print(frutas_y_verduras)  # Imprime la lista combinada

frutas_y_verduras.sort()  # Ordena la lista combinada alfabéticamente
print(frutas_y_verduras)  # Imprime la lista combinada ordenada

notas = [["Ruben", 8.5], ["Ana", 9.0], ["Luis", 7.5], ["Marta", 9.5]]
print(notas)  # Imprime la lista de notas
nota_ruben = notas[0][1]  # Accede a la nota de Rubén. El primer índice [0] accede a la lista de Rubén, y el segundo índice [1] accede a su nota.
print (nota_ruben)  # Imprime la nota de Rubén

notas[0][1] = 9.0  # Actualiza la nota de Rubén a 9.0
print(notas)  # Imprime la lista de notas actualizada

lista_anidada = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
print(lista_anidada[1][1]) # Imprime el elemento en la segunda fila y segunda columna de la lista anidada

lista_vacia = []  # Crea una lista vacía
print(lista_vacia)  # Imprime la lista vacía

lista_vacia.append("nuevo elemento")  # Agrega un nuevo elemento a la lista vacía
print(lista_vacia)  # Imprime la lista con el nuevo elemento
```

```python
# T U P L A S **********************************

# Las tuplas son similares a las listas, pero son inmutables (no se pueden modificar una vez creadas) y van entre paréntesis.

coordenadas_geograficas = (40.7128, -74.0060)  # Tupla con coordenadas geográficas (latitud, longitud)
color_rojo = (255, 0, 0)  # Tupla con valores RGB para el color rojo
color_verde = (0, 255, 0)  # Tupla con valores RGB para el color verde
color_azul = (0, 0, 255)  # Tupla con valores RGB para el color azul

# SON INMUTABLES, NO SE PUEDEN MODIFICAR

# coordenadas_geograficas[0] = 41.0  # Esto generaría un error porque las tuplas son inmutables

# color_rojo[0] = 200  # Esto generaría un error porque las tuplas son inmutables
# color_rojo.pop()  # Esto generaría un error porque las tuplas son inmutables

str(color_rojo)  # Convierte la tupla a una cadena de texto
print(color_rojo [0])  # Imprime el primer elemento de la tupla color_rojo

# Las tuplas se pueden usar para almacenar datos que no necesitan ser modificados, como coordenadas geográficas o colores RGB.
# Las tuplas son útiles cuando se necesita garantizar que los datos no cambien accidentalmente.
# Las tuplas también se pueden usar como claves en diccionarios, ya que son inmutables.
```

```python
# D I C C I O N A R I O S **********************************

# llaves y valores, se caracterizan por estar entre llaves y los valores se acceden por su llave.
# ejemplo de diccionario con información de una persona:

persona = {
    "nombre": "Juan",
    "edad": 30,
    "ciudad": "Madrid",
    "profesion": "Ingeniero"
}
print(persona)  # Imprime el diccionario completo
print(persona["nombre"])  # Imprime el valor asociado a la llave "nombre"
print(persona["edad"])  # Imprime el valor asociado a la llave "edad"
print(persona["ciudad"])  # Imprime el valor asociado a la llave "ciudad"
print(persona["profesion"])  # Imprime el valor asociado a la llave "profesion"

# Agregar un nuevo par llave-valor al diccionario
persona["pais"] = "España"
print(persona)  # Imprime el diccionario actualizado

persona["edad"] = 31  # Actualiza el valor asociado a la llave "edad"
print(persona)  # Imprime el diccionario actualizado

print(persona.get("nombre"))  # Imprime el valor asociado a la llave "nombre" usando el método get


frutas = {"manzana": 2, "banana": 3, "naranja": 4}  # Diccionario con frutas y sus cantidades
print(frutas)  # Imprime el diccionario completo

frutas_y_verduras = {"FRUTAS": ["manzana", "banana", "naranja"], "VERDURAS": ["lechuga", "tomate", "cebolla"]}  # Diccionario con listas de frutas y verduras
print(frutas_y_verduras)  # Imprime el diccionario completo
print(frutas_y_verduras["FRUTAS"])  # Imprime la lista de frutas
print(frutas_y_verduras["FRUTAS"][1])  # Imprime el indice 1 de la lista de frutas, en este caso "banana"
frutas_y_verduras["VERDURAS"].append("zanahoria")  # Agrega "zanahoria" a la lista de verduras
print(frutas_y_verduras)  # Imprime el diccionario actualizado
```

```python
# S T R I N G S   A V A N Z A D A S **********************************

nombre = "Ruben Apablaza"

print(nombre[0])  # Imprime el primer carácter de la cadena
print(nombre[-1])  # Imprime el último carácter de la cadena
print(nombre[5:8])  # Imprime los caracteres desde el índice 5 hasta el 7 (sin incluir el 8)
print(nombre[:5]) # Imprime los caracteres desde el inicio hasta el índice 3 (sin incluir el 5)

oracion = "Hola como te va en el dia de hoy"
print(oracion.split())  # Divide la cadena en una lista de palabras, usando el espacio como separador por defecto
print(oracion.split("e"))  # Divide la cadena en una lista de palabras usando la letra "e" como separador
print(oracion.split(" "))  # Divide la cadena en una lista de palabras usando el espacio

oracion_separada = oracion.split()

oracion_unida = "kakita".join(oracion_separada)  # Une la lista de palabras en una cadena, usando "kakita" como separador
print(oracion_unida)  # Imprime la cadena unida

direccion_ip = "192-168-0-45"
print (direccion_ip)

direccion_ip = direccion_ip.replace("-", ".")  # Reemplaza los guiones por puntos
print (direccion_ip)

nueva_oracion = "ella me dijo 'anda a la feria'" # Usa comillas simples dentro de comillas dobles
print(nueva_oracion)  # Imprime la nueva oración con comillas simples

nueva_oracion2 = 'ella me dijo "anda a la feria"'  # Usa comillas dobles dentro de comillas simples
print(nueva_oracion2)  # Imprime la nueva oración con comillas dobles

nueva_oracion3 = "ella me dijo \"anda a la feria\""  # Escapa las comillas dobles con una barra invertida
print(nueva_oracion3)  # Imprime la nueva oración con comillas dobles
```

```python
print(nueva_oracion.strip())  # Elimina los espacios en blanco al inicio y al final de la cadena
print(nueva_oracion.lower())  # Convierte la cadena a minúsculas
print(nueva_oracion.upper())  # Convierte la cadena a mayúsculas
print(nueva_oracion.title())  # Convierte la primera letra de cada palabra a mayúsculas
print(nueva_oracion.capitalize())  # Convierte la primera letra de la cadena a mayúsculas
print(nueva_oracion.startswith("ella"))  # Verifica si la cadena comienza con "ella"
print(nueva_oracion.endswith("feria'"))  # Verifica si la cadena termina con "feria"
print(nueva_oracion.find("dijo"))  # Busca la posición de la subcadena
print(nueva_oracion.index("dijo"))  # Busca la posición de la subcadena, genera error si no se encuentra
print(nueva_oracion.count("a"))  # Cuenta cuántas veces aparece la letra "a" en la cadena
print(nueva_oracion.replace(" ", " KK "))  # Reemplaza los espacios por "KK"
print("f" in nueva_oracion)  # Verifica si la letra "f" está en la cadena, devuelve True o False
print ("F" in nueva_oracion)  # Verifica si la letra "F" está en la cadena, devuelve True o False, ojo con la mayúscula y minúscula

letra = "A"
palabra = "hola"
print(letra in palabra)  # Verifica si la letra "A" está en la palabra "hola", devuelve False porque no hay coincidencia

letra2 = "a"
palabra = "hola"
print(letra2 in palabra)  # Verifica si la letra "A" está en la palabra "hola", devuelve True porque hay coincidencia

print(letra.lower() in palabra)  # Convierte la letra "A" a minúscula y verifica si está en la palabra "hola", devuelve True porque hay coincidencia

ingrese_su_respuesta = input("Ingrese ACUERDO o DESACUERDO")

if ingrese_su_respuesta.upper().strip() == "ACUERDO": #no importa si el usuario ingresa en mayúsculas o minúsculas y # elimina los espacios al inicio y al final
    print("Usted está de acuerdo")
elif ingrese_su_respuesta.upper() == "DESACUERDO":
    print("Usted está en desacuerdo")
```

```python
pelicula = "El señor de los anillos: La comunidad del anillo"
print (f"La película es {pelicula}") # esta es una forma de formatear cadenas de texto, usando f-strings
print ("La película es {}".format(pelicula))  # Otra forma de formatear cadenas de texto, usando el método format
print ("La película es %s" % pelicula)  # Otra forma de formatear cadenas de texto, usando el operador %
print ("La película es " + pelicula)  # Otra forma de formatear cadenas de texto, usando el operador +

# todas sirven para formatear cadenas de texto, pero las f-strings son las más recomendadas por su simplicidad y legibilidad.
```