

OPTATIVO 1 – PYTHON I

Prof. Ing. Aaron Zárate



Ciclo while en Python

Los ciclos en Python son estructuras de control que repiten una serie de instrucciones hasta que se cumple una condición específica.

En Python tenemos dos tipos de estructuras para ejecutar ciclos. Ciclo while y ciclo for. Comencemos con el ciclo while. El ciclo while repite una serie de instrucciones mientras la condición a evaluar sea verdadera.

```
# Sintaxis ciclo while
while condicion:
    # Bloque de código a ejecutar
```

```
# Ejemplo ciclo while
# Imprimir de 1 a 3
contador = 1
while contador <= 3:
    print(contador)
    contador += 1
```

SALIDA

1
2
3

Diagrama de flujo ciclo while en Python

```
# Ejemplo ciclo while  
# Imprimir de 1 a 3  
contador = 1  
while contador <= 3:  
    print(contador)  
    contador += 1
```

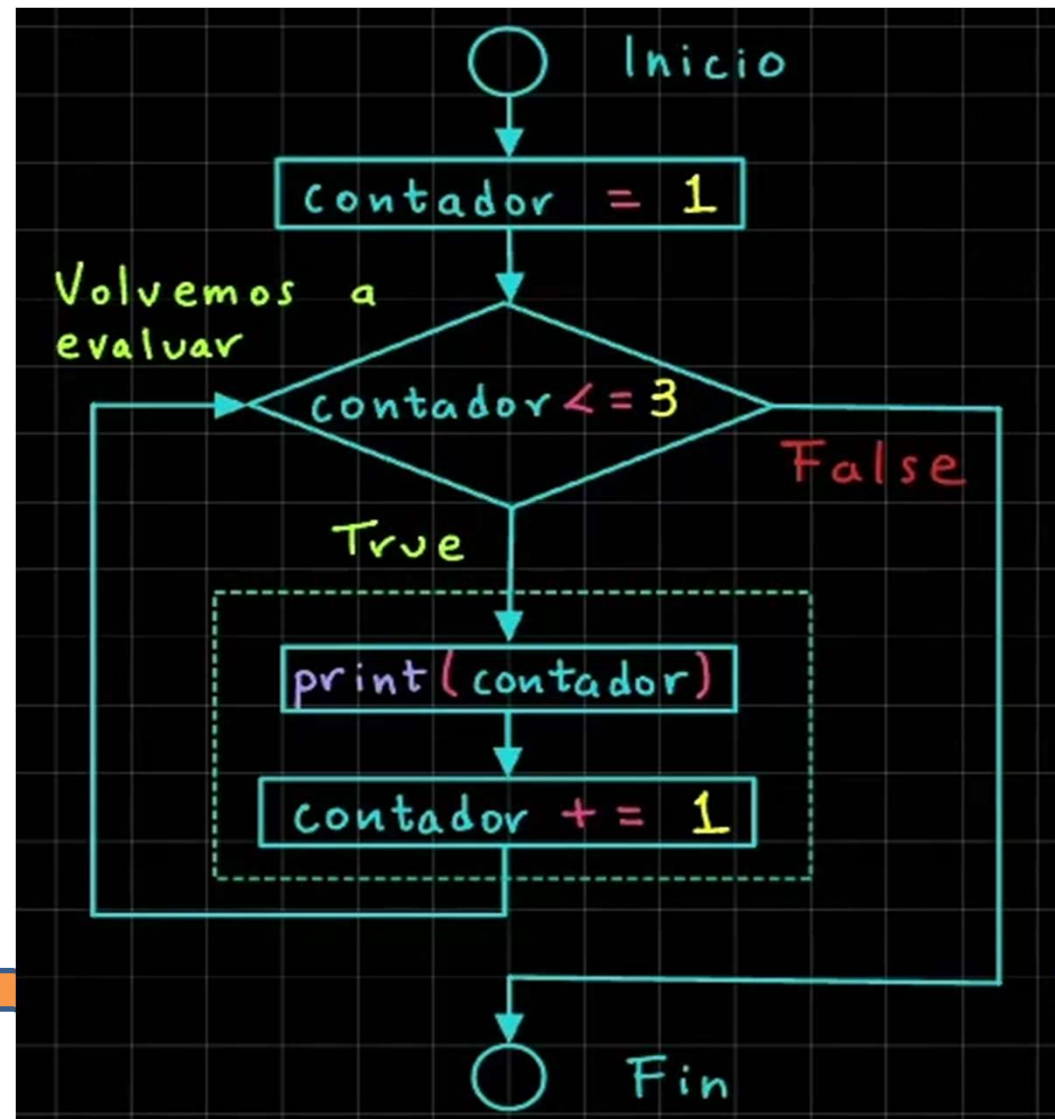
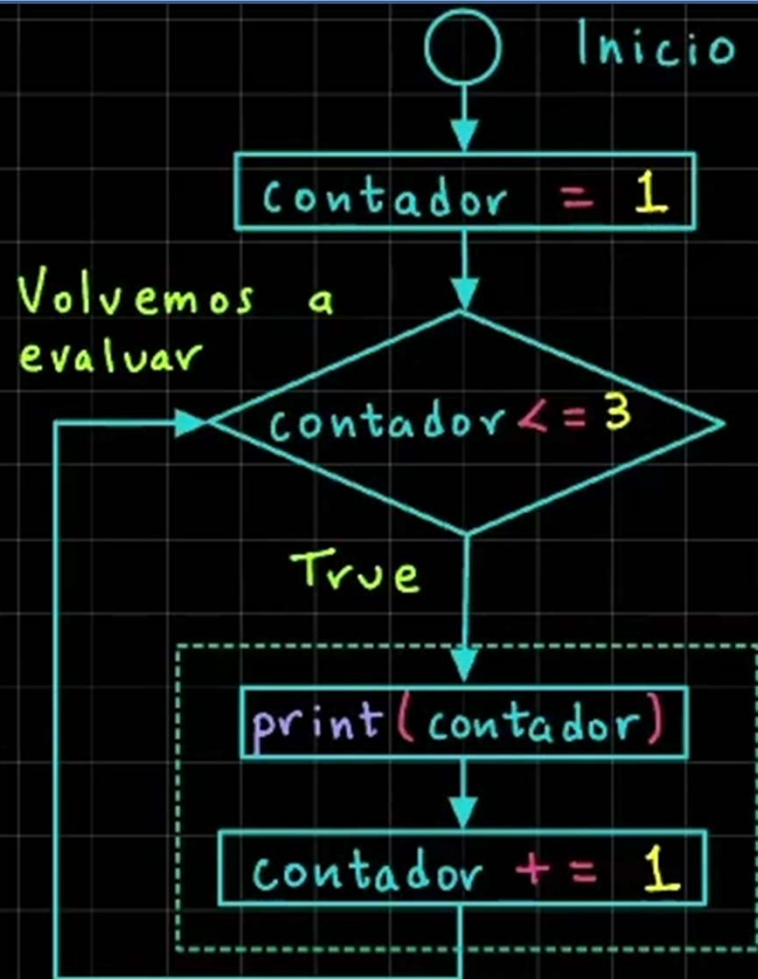


Diagrama de flujo ciclo while en Python



Valor Contador = ~~1~~ 2

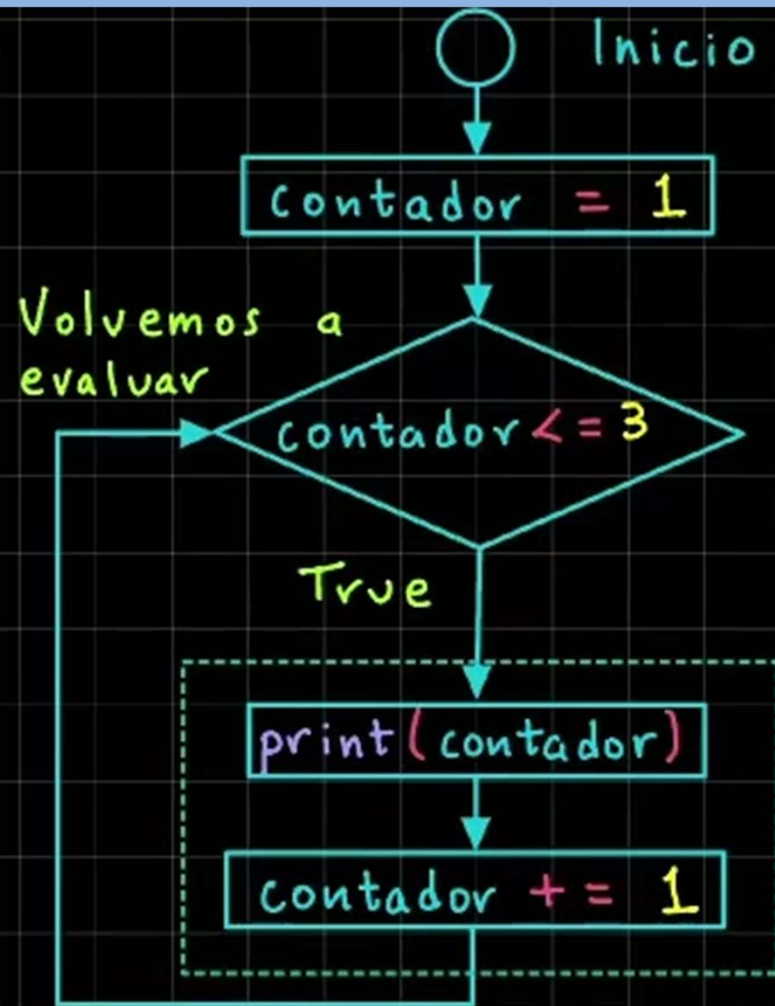
Resultado Condición:

1 <= 3 → True

Salida a Consola:

→ 1

Diagrama de flujo ciclo while en Python



Valor Contador = ~~2~~ 3

Resultado Condición:

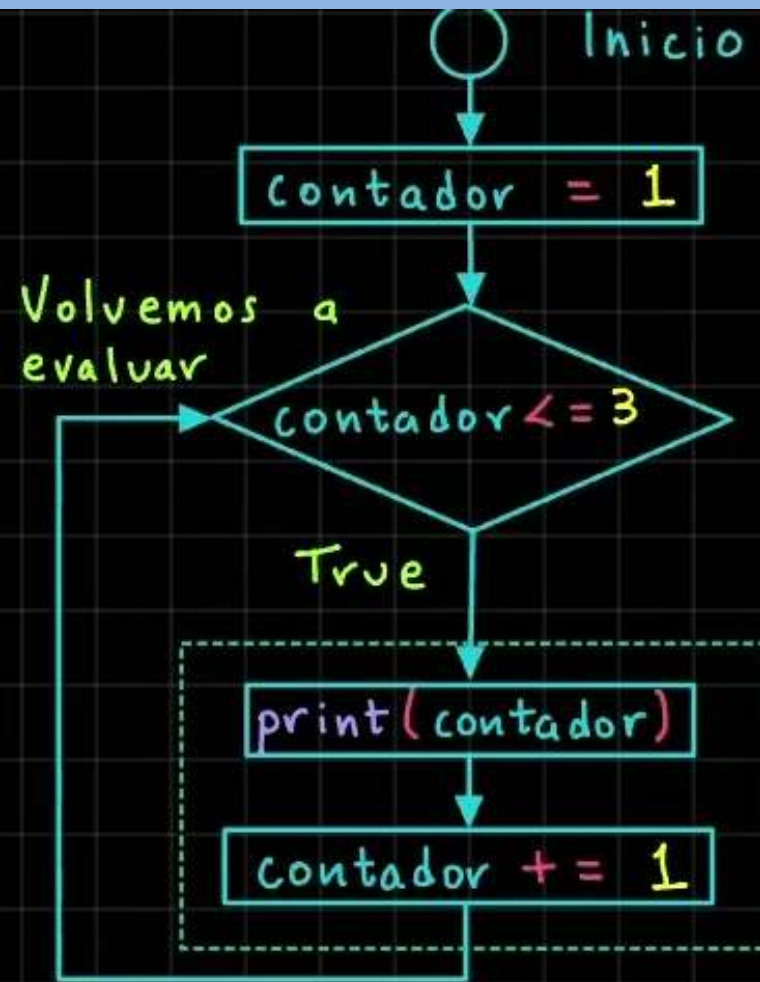
$2 \leq 3 \rightarrow \text{True}$

Salida a Consola:



1
2

Diagrama de flujo ciclo while en Python



Valor contador = ~~3~~ 4

Resultado Condición:

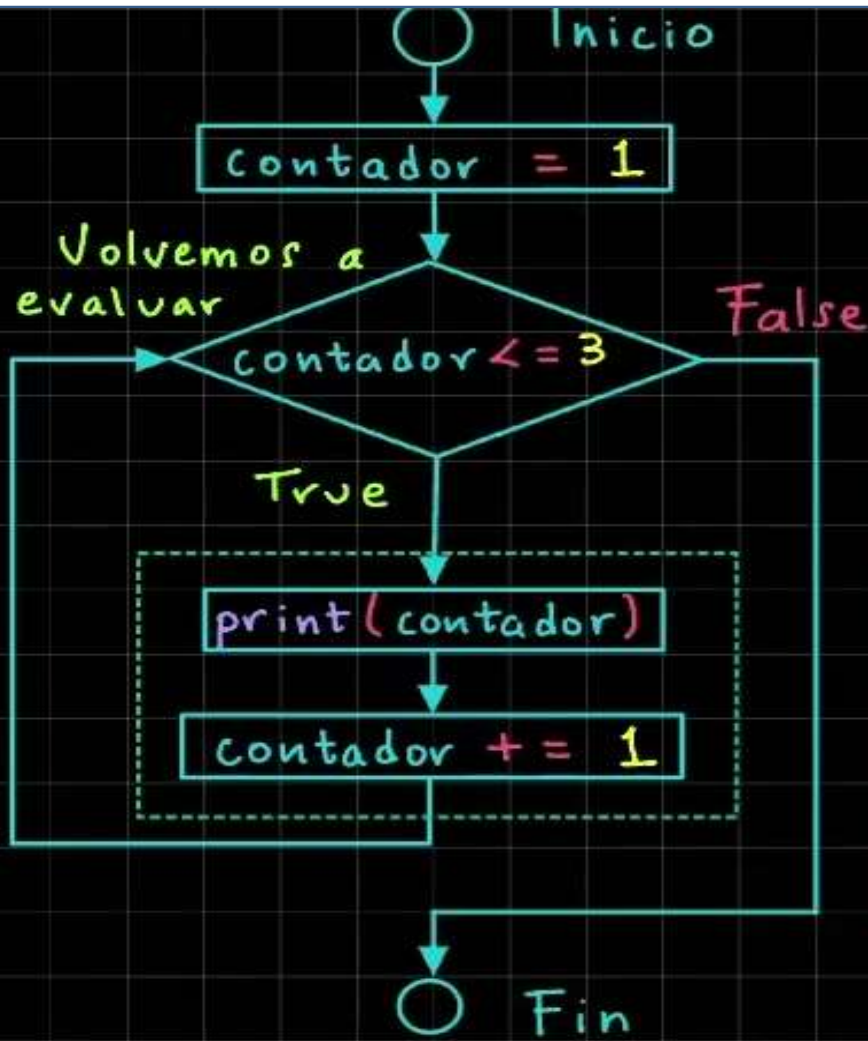
$3 \leq 3 \rightarrow \text{True}$

Salida a Consola:



1
2
3

Diagrama de flujo ciclo while en Python



Variable contador = 4

Resultado Condición:

`4 <= 3 → False`

Salida a Consola:

1
2
3

Ciclo for en Python

El ciclo for itera o recorre una secuencia de valores, por ejemplo los caracteres de una cadena, una lista, etc, y ejecuta un bloque de código por cada elemento de la secuencia.

```
# Sintaxis ciclo for
for variable in secuencia:
    # Bloque de código a ejecutar
```

```
# Ejemplo ciclo for
cadena = 'Hola Mundo'
for letra in cadena:
    print(letra, end=' ')
```

SALIDA a CONSOLA

H o l a M u n d o

Ciclo for en Python

str

0	1	2	3	4	5	6	7	8	9
"H"	"o"	"l"	"a"	" "	"M"	"u"	"n"	"d"	"o"



ciclo_for.py ×

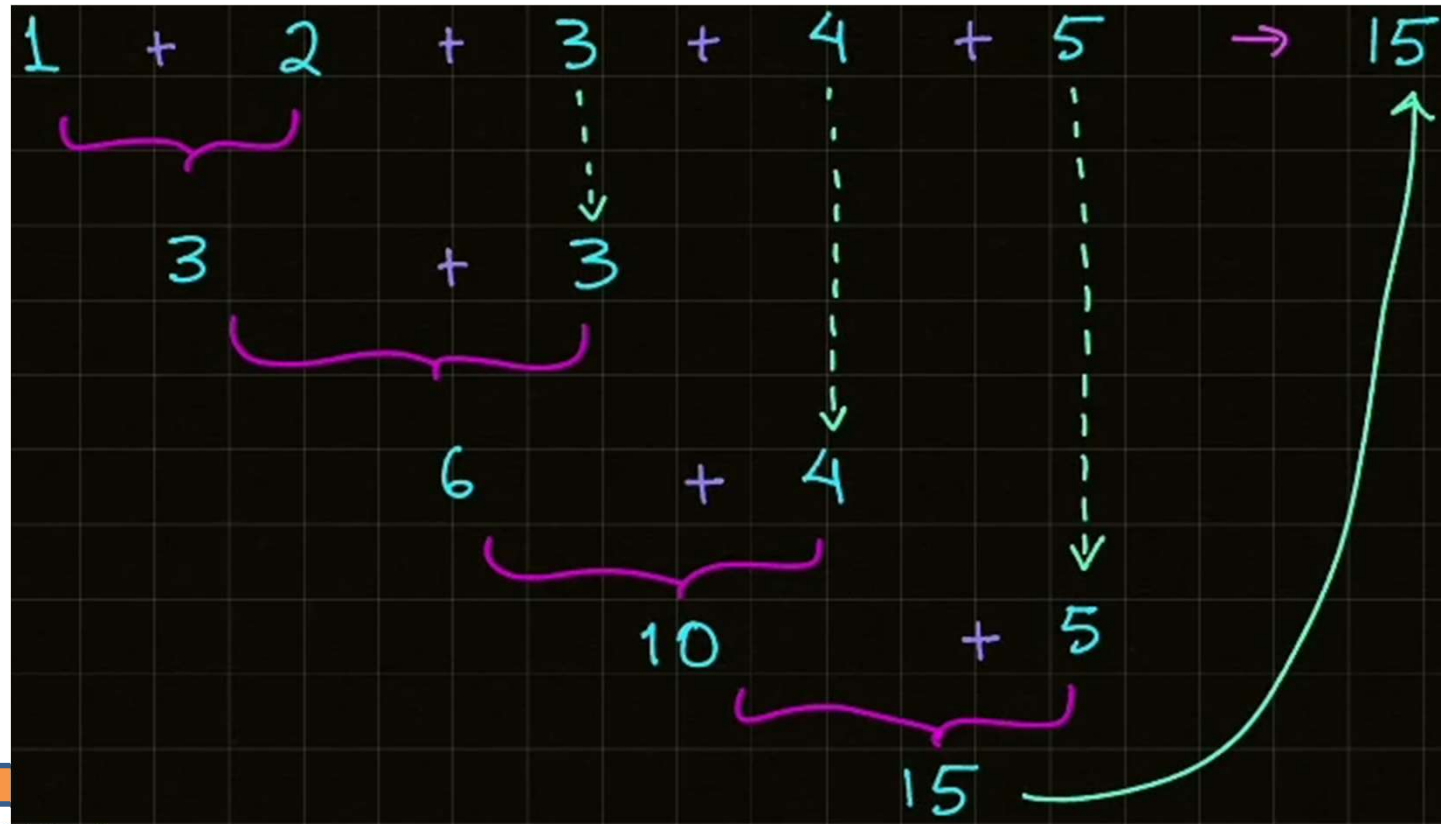
list

0	1	2
"Plátano"	"Fresa"	"Mango"

```
1 print("*** Ciclo for ***")
2 cadena = "Hola Mundo"
3 #interar o recorrer los caracteres
4 for letra in cadena:
5     print(letra, end=" ")
6
7 print("\n")
8 frutas = ["Banana", "Frutilla", "Pera", "Manzana"]
9 for fruta in frutas:
10     print(fruta, end=" ")
```

Ejercicio Acumulador Suma

Realizar la suma de los primeros 5 números utilizando un ciclo while



Ejercicio acumulador Suma

```
print("*** Suma Acumulativa ***")
#Sumar los primeros 5 numeros
maximo=5
numero=1
acumulador_suma=0
#Empezamos a iterar
while numero <= maximo:
    acumulador_suma += numero
    numero += 1
    print(f"\nResultado de la suma acumulada: {acumulador_suma}")
```

Menú iterativo

```
1 print("*** Sistema de administraciond e cuentas ***")
2 salir = False
3 while not salir:
4     print(f"""Menu:
5     1. Crear Cuenta
6     2. Eliminar Cuenta
7     3. salir""")
8     opcion = int(input("Escoje una opcion: "))
9     if opcion == 1:
10        print("Creando tu cuenta...\n")
11    elif opcion ==2:
12        print("Eliminando tu cuenta...\n")
13    elif opcion == 3:
14        print("Saliendo del sistema. Hasta pronto!\n")
15        salir= True
16    else:
17        print("Opcion Invalida proporciona una nueva opcion")
18 else:
19    print("Terminando el sistema de administracion de cuentas")
```

Cajero automático

Aplicación de cajero automático. Las funciones principales del cajero automático son:

Depositar.

Retirar.

Consultar saldo.

El saldo puede tener un valor inicial por ejemplo de 1000\$. Si realizas un retiro se resta de tu saldo, si realizas un deposito se suma a tu saldo

Cajero automático Parte1

```
aplicacion_cajero_automatico.py ×  
  
1  print('*** Aplicación de Cajero Automático ***')  
2  
3  saldo = 1000  
4  salir = False  
5  while not salir:  
6      print(f'''Operaciones que puedes realizar:  
7          1. Consultar Saldo  
8          2. Retirar  
9          3. Depositar  
10         4. Salir''')  
11     opcion = int(input('Escoje una opción: '))
```


Cajero automático Parte2

```
12     if opcion == 1:
13         print(f'Tu saldo actual es: ${saldo:.2f}\n')
14     elif opcion == 2:
15         retiro = float(input('Ingresa el monto a retirar: '))
16         # Validacion
17         if retiro <= saldo:
18             saldo -= retiro # saldo = saldo - retiro
19             print(f'Tu nuevo saldo es: ${saldo:.2f}\n')
20         else:
21             print(f'No cuentas con el saldo suficiente. Saldo actual es: ${saldo:.2f}\n')
22     elif opcion == 3:
23         deposito = float(input('Ingresa el monto a depositar: '))
24         saldo += deposito # saldo = saldo + deposito
25         print(f'Tu nuevo saldo es: ${saldo:.2f}\n')
26     elif opcion == 4:
27         print(f'Saliendo del cajero automático. Hasta pronto!')
28         salir = True
29     else:
30         print('Opción inválida. Selecciona otra opción\n')
31 else:
32     print('Terminando la aplicación de Cajero Automático!')
```


Calculadora

Crear la aplicación de calculadora con las opciones de:

- 1.Suma.
- 2.Resta.
- 3.Multiplicación
- 4.División

El programa debe mostrar un menú con cada opción, y debe solicitar los valores de operando1 y operando2 para realizar la operación seleccionada

Calculadora Parte 1

 calculadora.py ×

```
1 print('*** Calculadora en Python ***')
2
3 operando1 = operando2 = resultado = 0
4 salir = False
5 while not salir:
6     print(f'''Operaciones que puedes realizar:
7         1. Suma
8         2. Resta
9         3. Multiplicacion
10        4. Division
11        5. Salir''')
12    opcion = int(input('Escoje una opción: '))
```

Calculadora Parte 2

```
12 opcion = int(input('Escoje una opción: '))
13 # Vamos a solicitar el valor de los operandos
14 if 1 <= opcion <= 4:
15     operando1 = float(input('Dame el valor 1: '))
16     operando2 = float(input('Dame el valor 2: '))
17 # Revisamos el tipo de operación a realizar
18 if opcion == 1: # Suma
19     resultado = operando1 + operando2
20     print(f'El resultado de la suma es: {resultado:.2f}\n')
21 elif opcion == 2: # Resta
22     resultado = operando1 - operando2
23     print(f'El resultado de la resta es: {resultado:.2f}\n')
24 elif opcion == 3: # Multiplicacion
25     resultado = operando1 * operando2
26     print(f'El resultado de la multiplicación es: {resultado:.2f}\n')
27 elif opcion == 4:
28     resultado = operando1 / operando2
29     print(f'El resultado de la división es: {resultado:.2f}\n')
30 elif opcion == 5:
31     print(f'Saliendo del programa de Calculadora. Hasta pronto!')
32     salir = True
33 else:
34     print(f'Opción inválida, selecciona otra opción...\n')
```

Creación y validación de password


Crear un programa para solicitar la validación de una password o contraseña.

La contraseña debe tener al menos 6 caracteres.

En caso de cumplir con esta condición el programa debe volver a solicitar un nuevo valor hasta que cumpla con la condición.

Si el valor proporcionado es valido, se debe imprimir : “PASSWORD VÁLIDO” y debe terminar la ejecución del sistema.

Creación y validación de password

 validacion_password.py ×

```
1 print('Creación y Validación de un Password ***')
2
3 password = input('Ingresa un password (debe tener al menos 6 caracteres: ')
4
5 # Validar el password
6 while len(password) < 6:
7     print('El password no cumple con los requisitos. Debe tener al menos 6 caracteres')
8     password = input('Ingresa un nuevo valor de password: ')
9 else:
10     print('El valor de password es válido')
11
```

✓ 15 ^

Juego de adivinanzas

Crea un juego donde el jugador debe adivinar un numero secreto.

Puedes usar un ciclo while hasta que el jugador adivine correctamente.

El numero secreto se puede crear utilizando la función randint para generar un numero aleatorio entre 1 y 50.

Por cada intento fallido se debe incrementar una variable que lleve el conteo de intentos

Juego de adivinanzas Parte 1

juego_adivinanzas.py ×

```
1  from random import randint
2
3  print('*** Juego de Adivinanzas ***')
4
5  numero_secreto = randint(1, 50)
6  intentos = 0
7  adivinanza = None
8  INTENTOS_MAXIMOS = 5
9
10 while adivinanza != numero_secreto and intentos < INTENTOS_MAXIMOS:
11     adivinanza = int(input('Adivina el número secreto (1-50): '))
12     # Agregamos una ayuda para orientar al jugador
13     if adivinanza < numero_secreto:
14         print('El número secreto es mayor')
15     elif adivinanza > numero_secreto:
16         print('El número secreto es menor')
17     # Incrementamos la variable de intentos
18     intentos += 1
```

Juego de adivinanzas Parte 2

```
17     # Incrementamos la variable de intentos
18     intentos += 1
19 # Conclusion del juego
20 if adivinanza == numero_secreto:
21     print(f'Felicitades, adivinaste el número secreto en {intentos} intentos')
22 else:
23     print(f'Lo siento, has agotado tus intentos máximos: {INTENTOS_MAXIMOS}')
24     print(f'El número secreto era: {numero_secreto}')
```

Función rango en Python

La función rango es una función incorporada que genera una secuencia de números.

Es comúnmente utilizada para iterar sobre ciclos del tipo for.


```
# Sintaxis funcion range
# inicio - valor inicial (opcional)
# fin - valor final, sin incluirlo
# incremento - diferencia entre cada número (opcional)
range(inicio, fin, incremento)
```

Función range en Python

```
# Usar range para imprimir 0 al 4
for i in range(5):
    print(i)


# Usar range para imprimir del 0 al 9
# con incremento de 2
for i in range(0, 10, 2):
    print(i) # Imprime 0, 2, 4, 6, 8
```

Función range en Python

 funcion_range.py ×

```
1 print('*** Función range ***')
2
3 print('Secuencia del 0 al 4: ')
4 # inicio = 0 (opcional) default
5 # fin = 5 - 1 = 4
6 # incremento = 1 (opcional) default
7 for i in range(5): # fin = 5 - 1
8     print(i, end=' ')
9
10 print('\n\nSecuencia del 10 al 20:')
11 # incremento = 1 (default y es opcional)
12 for i in range(10, 20 + 1): # 20+1 pra icluir el 20
13     print(i, end=' ')      #o podriamos poner 21
14
15 print('\n\nSecuencia del 20 al 30 de 2 en 2: ')
16 for i in range(20, 30 + 1, 2):
17     print(i, end=' ')
```



Repetición de un mensaje

 impresion_mensaje.py ×

```
1  print('*** Repetición de un Mensaje ***')
2
3  mensaje = input('Proporciona un mensaje a repetir: ')
4  numero_de_repeticiones = int(input('Proporciona el número de repeticiones: '))
5
6  # Iterar sobre el rango de repeticiones
7  for i in range(numero_de_repeticiones):
8      #el inicio por default es cero
9      #el incremento por default es de +1
10     print(f'{i+1} - {mensaje}')
11
```

! 2 ✓

Triángulo simétrico

 triangulo_simetrico.py ×

```
1 print('*** Dibujar Triángulo Simétrico ***')
2
3 numero_filas = int(input('Proporciona el número de filas: '))
4
5 # Iterar sobre cada fila del triángulo
6 for fila in range(1, numero_filas + 1):
7     espacios_blanco = ' ' * (numero_filas - fila)
8     asteriscos = '*' * (2 * fila - 1)
9     print(f'{espacios_blanco}{asteriscos}')
10
```

Ciclos break y continue

 ciclos_break_continue.py ×

```
1 print('*** break y continue ***')
2
3 # Ejemplo con break
4 print('Palabra break:')
5 for numero in range(1, 10):
6     if numero % 2 == 0: # numero par
7         print(numero)
8         break # Salimos del ciclo inmediatamente
9
10 # Ejemplo con continue
11 print('\nPalabra continue: ')
12 for numero in range(1, 10):
13     if numero % 2 == 1: # numero impar
14         continue
15     print(numero) # numeros pares
```