

# OPTATIVO 1 – PYTHON I

---

**Prof. Ing. Aaron Zárate**



# Operadores en Python

Los operadores son símbolos especiales que están diseñados para realizar operaciones específicas. Tenemos varios tipos que son:


- Operadores aritméticos: Permiten realizar cálculos matemáticos básicos, como suma, resta, multiplicación y división.
- Operadores de asignación: Se utilizan para asignar valores a variables.
- Operadores de comparación: Se utilizan para comparar un valor con otro.
- Operadores lógicos: Se utilizan para combinar expresiones condicionales o lógicas.
- Operadores de identidad: Se utilizan para comparar si dos variables son el mismo objeto.
- Operadores de membresía: Se utilizan para probar si una secuencia (Ej., una subcadena) se presenta en el objeto.

# Operadores Aritméticos en Python

Los operadores aritméticos nos permiten realizar cálculos matemáticos básicos. Por ejemplo.

- Suma(+): Suma dos operadores.
- Resta(-): Resta dos operadores.
- Multiplicación(\*): Multiplica dos operandos
- División(/): Divide el primer operando con el segundo operando, resulta un valor flotante.
- División(//): (/): Divide el primer operando con el segundo operando, resulta un valor tipo entero.
- Modulo(%): Regresa el residuo de la división.
- Exponente(\*\*): Eleva el primer operando a la potencia del segundo operando.

# Operadores Aritméticos en Python

 operadores\_aritmeticos.py ×

```
1  #Operadores aritmticos
2  a = 10
3  b = 3
4  #Suma
5  suma = a+b
6  print(f"Suma= {suma}")
7  #Resta
8  resta=a-b
9  print(f"Resta= {resta}")
10 #multiplicacion
11 multiplicacion= a*b
12 print(f"Multiplicacion= {multiplicacion}")
```

# Operadores Aritméticos en Python

```
14     division= a/b
15     print(f"Division flotante= {division:.2f}")
16     #Division entera
17     division_entera = a//b
18     print(f"Division entera= {division_entera}")
19     #Operador Modulo
20     modulo = a%b
21     print(f"Residuo o modulo de la division: {modulo}")
22     #Exponente (potencia)
23     exponente= a**b # 10 elevando a la 3, 10*10*10 = 1000
24     print(f"Exponente= {exponente}")
25
```

# Operadores de Asignación en Python

Los operadores aritméticos nos permiten realizar cálculos matemáticos básicos. Por ejemplo.

```
# Sintaxis operador asignacion  
variable = valor
```


```
# Ejemplo Operador Asignación  
numero = 10  
texto = "Hola, mundo"
```

En Python también tenemos la asignación múltiple, lo que nos permite asignar valores a varias variables en una sola línea de código. El código es mas compacto y fácil de leer

```
# Sintaxis de Asignación Múltiple  
variable1, variable2 = valor1, valor2
```

```
# Ejemplo de Asignación Múltiple  
a, b, c = 10, 'Saludos', 14.5
```

# Operadores de Asignación en Python

 operadores\_asignacion.py ×

```
1 print("***Operadores de asignacion**")
2 numero = 5
3 print(f"Valor del numero= {numero}")
4 numero = 10
5 print(f"Nuevo valor del numero= {numero}")
6 cadena= "Saludos desde Python"
7 print(f"Valor de la cadena= {cadena}")
8 #Asiganacion multiple
9 x, y, z = 19, "Palmeiras", -9.25
10 print(f"Valor de x= {x}, El valor de y= {y}, El valor de z= {z}")
```

# Asignación encadenada

En Python también contamos con la asignación encadenada, esto permite asignar el mismo valor a múltiples variables.

```
# Sintaxis de Asignación Encadenada  
variable1 = variable2 = ... = valor
```

```
# Ejemplo. Inicializar contadores  
contador1 = contador2 = 0
```



# Asignación encadenada

```
#Asignacion encadenada
a=b=c=10
print(f"El valor de A={a}, El valor de B={b}, El valor de C={c}")
#Asignacion de valores en una cadena, sin utilizar variables temporales
x,y=5,10
print(f"Valores iniciales x= {x}, y = {y}")
#Aplicando concepto de asignacion multiple, intercambiamos valores
x,y= y,x
print(f"Invertir los valores de x={x} , y={y}")
#Recibir multiples valores de la entrada del usuario
nombre,apellido= input("Ingresa tu nombre y apellido separados por coma.").split(",")
print(f"Mi nombre es: {nombre.strip()} y mi apellido es : {apellido.strip()} ")
```




# Operadores de Asignación compuesto

Los operadores de asignación compuesto combinan una operación aritmética, haciendo las operaciones mas concisas. Los operadores pueden ser: += , -= , \*= , /= , etc.

```
# Sintaxis Operador Asignación Compuesto  
variable OPERADOR= valor
```

```
# Ejemplo Operador Asignación Compuesto  
contador = 0  
contador += 1 # contador = contador + 1
```

# Operadores de Asignación compuesto

 operadores\_asignacion\_compuestos.py ×

```
1 print("***Operadores ed asignacion compuestos***")
2 a,b = 10,15
3 print(f"Valore inicial de A: {a}, Valor inicial de B: {b}")
4 #Operador compuesto suma +=
5 a+=b #a=a+b
6 print(f"Operador a+=b: {a}")
7 # Operador compuesto de resta
8 a=10
9 a-=b
10 print(f"Operador a-=b: {a}")
11 #Operador compuesto de multiplicacion *=
12 a=10 #reiniciamos el valor de a
13 a*=b
14 print(f"Operador a*=b: {a}")
15 a=10
16 #Operador compuesto de division /=
17 a=10
18 a/=b # a= a/b
19 print(f"Operador a/=b: {a:.2f}")
```

# Operadores Condicionales

Los operadores condicionales se utilizan para comparar dos valores.

El resultado siempre es un valor booleano "True" o "False" dependiendo de si la condición se cumple o no.

- Operador igualdad (==): Compara si dos valores son iguales

```
# Sintaxis operador igualdad =  
a == b
```

```
# Ejemplo operador igualdad =  
print(5 == 5) # True  
print(5 == 6) # False
```

# Operadores Condicionales

Los operadores condicionales se utilizan para comparar dos valores.

El resultado siempre es un valor booleano “True” o “False” dependiendo de si la condición se cumple o no.

- Operador distinto (!=): Compara si dos valores son distintos

```
# Sintaxis operador distinto !=  
a != b
```

```
# Ejemplo operador distinto !=  
print(5 != 5) # False  
print(5 != 6) # True
```

# Operadores Condicionales

- Operador menor que (<): Compara si el valor de la izquierda es menor que el de la derecha, si es menor devuelve "True" si es mayor "False"

```
print(3 < 5)    # True  
print(5 < 3)    # False
```

- Operador menor o igual que (<=): Compara si el valor de la izquierda es menor o igual que el de la derecha, si es menor o igual devuelve "True" si es mayor "False"

```
print(3 <= 5)   # True  
print(5 <= 5)   # True  
print(6 <= 5)   # False
```

# Operadores Condicionales

- Operador mayor que (>): Compara si el valor de la izquierda es mayor que el de la derecha, si es mayor devuelve "True" si es menor "False"


```
print(5 > 3)    # True  
print(3 > 5)    # False
```

- Operador mayor o igual (>=): Compara si el valor de la izquierda es mayor o igual que el de la derecha, si es mayor o igual devuelve "True" si es menor "False"

```
print(5 >= 3)   # True  
print(5 >= 5)   # True  
print(3 >= 5)   # False
```



# Operadores Condicionales

 operadores\_comparacion.py ×

```
1 print("***Operadores de comparacion***")
2 a,b=7,5
3 print(f"El valor inicial de A: {a}, el valor inicial de B:{b}")
4 #operador de igualdad
5 resultado = a==b
6 print(f"Resultado a==b: {resultado}")
7 #Operdor diferente !=
8 resultado = a!=b
9 print(f"Resultado a!=b: {resultado}")
```



# Operadores Condicionales

```
10  #Operador mayor que >
11  resultado = a>b
12  print(f"Resultado a>b: {resultado}")
13  #Operador mayor o igual que >=
14  resultado= a>=b
15  print(f"Resultado a>=b: {resultado}")
16  #operador menor que <
17  resultado = a<b
18  print(f"Resultado a<b: {resultado}")
19  #operador menor que <=
20  resultado = a<=b
21  print(f"Resultado a<=b: {resultado}")
```

# Operadores Lógicos

---

Los operadores lógicos se utilizan para realizar operaciones lógicas con valores booleanos.

- Operador Lógico and (y): Devuelve True si ambos operandos son verdaderos.

```
exp1 = False
exp2 = True
print(exp1 and exp2) # False
```

# Operadores Lógicos

---

Operador Lógico or (o): Devuelve True si cualquiera de los operandos es verdadero.

```
exp1 = False  
exp2 = True  
print(exp1 or exp2) # True
```

# Operadores Lógicos

---

Operador Lógico not (o): Invierte el valor del operando. Es un operador unario.

```
exp1 = False  
print(not exp1) # True
```


# Operador and

El operador and regresa verdadero si ambos operandos son verdaderos.

Tabla de la verdad

a	b	a and b
False	False	False
False	True	False
True	False	False
True	True	True

# Operador and

 operador\_and.py ×

```
1 print(f"***Operador and***")
2 # Regresa verdadero sin ambos valores son verdaderos
3 condicion1 = False
4 condicion2 = False
5 resultado = condicion1 and condicion2
6 print(f"El resultado {condicion1} and {condicion2} es: {resultado}")
7
```

# Sistema Descuentos VIP


---

Una tienda de supermercado ofrece un descuento especial a clientes que compren 10 o mas artículos por día y además sen miembros de la tienda.

El sistema debe solicitar al cliente que indique cuantos artículos ha comprado en el día y preguntarle si cuenta con la membresía de la tienda.

En caso de haber comprado 10 o mas productos y ser miembro de la tienda, entonces tendrá acceso al descuento VIP.

# Sistema Descuentos VIP

 descuento\_vip.py ×

```
1 print("***Sistemas de descuentos VIP***")
2 NO_PRODUCTOS_DESCUENTO = 10
3 cantidad_productos = int(input(f"Cuantos productos compraste hoy?: "))
4 tiene_membresia = input("Tienes la membresia de la tienda (Si/No): ")
5
6 es_elegible_descuento = (cantidad_productos >= NO_PRODUCTOS_DESCUENTO
7                          and tiene_membresia.strip().lower() == "si")
8 print(f"Tienes acceso al descuento VIP? {es_elegible_descuento}")
9
```




# Operador or

El operador or regresa verdadero si cualquiera de los operandos es verdadero.

Tabla de la verdad

a	b	a or b
False	False	False
False	True	True
True	False	True
True	True	True

# Operador or

 operador\_or.py ×

```
1 print(f"***Operador or***)  
2 # Regresa verdadero sin ambos valores son verdaderos  
3 condicion1 = False  
4 condicion2 = True  
5 resultado = condicion1 or condicion2  
6 print(f"El resultado {condicion1} or {condicion2} es: {resultado}")  
7
```

# Sistema Préstamo de libro


---

Se pide crear un sistema para una biblioteca, la cual desea prestar libros si cumple con cualquiera de las siguientes condiciones:

- El usuario tiene credencial de estudiante.
- El usuario no viva a mas de 3km a la redonda.

Si cumple con cualquier de estas condiciones se le puede prestar el libro.

# Sistema Préstamo de libro

 prestamo\_libros.py ×



```
1 print("***Sistema de prestamo de libros**")
2 DISTANCIA_PREMITIDA_KM = 3
3 tiene_credencial = input("Cuentas con credencial de estudiante (Si/No: ")
4 distancia_biblioteca_km = int(input("A cuantos kilometros vives de la biblioteca?: "))
5 es_elegible_prestamo = (tiene_credencial.strip().lower()=="si"
6                        or distancia_biblioteca_km <= DISTANCIA_PREMITIDA_KM)
7 print(f"Eresa elegible para prestamos de libross?: {es_elegible_prestamo}")
~
```

! 2 ✓ 17 ^ v


# Operador not

El operador not invierte el valor del operando.

Tabla de la verdad

a	not a
False	True
True	False

# Operador not


■  operador\_not.py ✕

```
1 print(f"***Operador not***")
2
3 condicion1 = False
4 resultado =not condicion1
5 print(f"Operador not sobre {condicion1} es {resultado}")
6
7 #Revisar si es cadena vacia
8 nombre = ""
9 es_cadena_vacia = not nombre
10 print(f"\nLa variable no tiene ningun valor {es_cadena_vacia}")
11
12 #Revisar si una variable no tiene ningun valor asignado
13 variable = None
14 es_variable_sin_valor = not variable
15 print(f"\nLa variable no tiene ningun valor asignado? {es_variable_sin_valor}")
16
```

! 5 ✓ 16 ^

# Ejemplo fuera de rango not

Revisar si una variable se encuentra dentro del rango entre 1 y 10

 dentro\_rango\_not.py ×

```
1  #Revisar si esta dentro del rango entre 1 y 10
2  dato = int(input("Proporciona un dato entero: "))
3  #revisamos si esta dentro de rango
4  esta_dentro_rango = 1<=dato<=10
5  print(f"Variable esta dentro de rango (entre 1y 10)?: {esta_dentro_rango}")
6
7  #Revisamos la logica inversa, si el dato esta fuera der ango
8  #esta_fuera_rango = not(1<=dato<=10)
9  #print(f"Variable esta fuera de rango (entre 1 y 10)?: {esta_fuera_rango}")
10
```

# Generación Ticket de venta

---

Supongamos que compramos varios artículos en el supermercado y queremos obtener el ticket de venta total incluyendo impuestos.

El sistema solicitara el precio de cada producto a comprar y el usuario deberá indicar su precio (valor tipo con punto decimal).

El sistema debe realizar la suma de cada producto, calcular el impuesto y finalmente imprimir el total de la compra.



# Generación Ticket de venta

ticket\_venta.py

```
1 print("***Generacion ticket de venta**")
2
3 precio_leche = float(input("Precio de la leche: "))
4 precio_pan = float(input("Precio del pan:"))
5 precio_lechuga = float(input("Precio de la lechuga: "))
6 precio_banana = float(input("Precio de la banana"))
7 #Calculo del subtotal sin impuestos
8 subtotal = precio_banana + precio_pan + precio_leche + precio_lechuga
9 #Calculo con impuestos 19%
10 impuesto = subtotal * 0.19
11 #calculo total de la compra
12 costo_total = subtotal + impuesto
13 print(f"""
14 Subtotal: ${subtotal:.2f}
15 Impuestos (19%): ${impuesto:.2f}
16 Costo total de la compra: ${costo_total:.2f}""")
```

# Sistema de autenticación

---

Crea un programa para validar el usuario y password proporcionados por el usuario.

Crea dos constantes con valores correctos y posteriormente compara que el usuario y password proporcionados por el usuario son validos

Debe solicitar el usuario y el password al usuario y si son iguales que los valores correctos almacenados en las constantes debe imprimir True, de lo contrario debe imprimir False.

# Valor dentro de rango

---

Solicitar al usuario un valor dentro del rango 0 y 5 si el valor proporcionado esta dentro del rango.

Se deben definir dos constantes.  
VALOR\_MINIMO = 0 y VALOR\_MAXIMO = 5.

Y debemos comprobar si el valor proporcionado se encuentra en el rango entre 0 y 5.

Finalmente se debe imprimir :

Valor dentro del rango: True/False

# Calculo del área y perímetro de rectángulo

---

Se solicita calcular el área y perímetro de un rectángulo aplicando las siguientes formulas:



Área = base \* altura

Perímetro = 2\*(base + altura)

Finalmente imprimir estos valores en pantalla.