

```

# OPERACIONES DE COMPARACION

# IGUAL A ==
bool = 5==5
print(bool)

bool = 'blanco'=='blanco' # tambien se pueden comparar STRINGS
print(bool)

bool = 'blanco'=='Blanco' # es sensible a las mayusculas
print(bool)

bool = 'blanco'=='Blanco'.lower() # esto SI es igual
print(bool)

# EJERCICIOS

# Crea dos variables (num1 y num2) con los valores 36 y 17 respectivamente.
# Verifica si num1 es mayor o igual que num2 y almacena el resultado de dicha comparacion en una variable llamada mi_bool
num1 = 36
num2 = 17
mi_bool = num1 >= num2

# Crea dos variables (num1 y num2):
# Dentro de num1, almacena el resultado de la operacion raiz cuadrada de 25
# Dentro de num2, almacena el numero 5.
# Verifica si num1 es igual a num2 y almacena el resultado de dicha comparacion en una variable llamada mi_bool.
num1 = 25**0.5
num2 = 5
mi_bool = num1 == num2

#Crea dos variables (num1 y num2):
# Dentro de num1, almacena el resultado de la operacion 64 x 3
# Dentro de num2, almacena el resultado de la operacion 24 x 8
# Verifica si num1 es diferente a num2 y almacena el resultado de dicha comparacion en una variable llamada mi_bool.
num1 = 64*3
num2 = 24*8

mi_bool = num1 != num2

# OPERADORES LOGICOS: y, o, no -> AND, OR, NOT
bool = (4 < 5) and (5 > 6)
print(bool) # false

bool = (4 < 5) and (5 == 2+3) # se pueden realizar operaciones dentro
print(bool) # true

bool = (55 == 55) and ('perro' == 'perro')
print(bool) # true

bool = (1 == 10) or (23 == 23)
print(bool) # true

bool = (1 == 10) or (23 == 23)
print(bool) # true

texto = "frase breve"
mi_bool = "frase" in texto # INDICA SI LA PALABRA "FRASE" SE ENCUENTRA DENTRO DE LA CADENA DE TEXTO
print(mi_bool) # true

texto = "frase breve"
mi_bool = ("frase" in texto) and ("breve" in texto) # INDICA SI LA PALABRA "frase" Y LA PALABRA "breve" SE ENCUENTRAN DENTRO DE LA CADENA DE TEXTO
print(mi_bool) # true

mi_bool = not ('a' == 'a') # LA COMPARACION ES VERDADERA ENTONCES AL PONERLE EL ESTAS DICIENDO QUE ESTO NO ES IGUAL, ES FALSO PORQUE SI ES IGUAL
print(mi_bool) # false

# EJERCICIOS

# Crea tres variables (num1 , num2 y num3):
# Dentro de num1, almacena el valor 36
# Dentro de num2, almacena el resultado de la operacion 72/2
# Dentro de num3, almacena el valor 48
# Verifica si num1 es mayor que num2, y menor que num3. Almacena el resultado de dicha comparacion en una variable llamada mi_bool.
num1 = 36
num2 = 72/2
num3 = 48
mi_bool = (num1>num2) and (num1<num3)

# Crea tres variables (num1 , num2 y num3):
# Dentro de num1, almacena el valor 36
# Dentro de num2, almacena el resultado de la operacion 72/2
# Dentro de num3, almacena el valor 48
# Verifica si num1 es mayor que num2, o menor que num3. Almacena el resultado de dicha comparacion en una variable llamada mi_bool.
num1 = 36
num2 = 72/2
num3 = 48
mi_bool = (num1>num2) or (num1<num3)

# Verifica si las palabras almacenadas en las siguientes variables:
# palabra1 = "exito", y
# palabra2 = "tecnologia"
# no se encuentran en la frase a continuacion, y almacena el resultado de esta comprobacion (un booleano) en una variable llamada mi_bool
frase = "Cuando algo es lo suficientemente importante, lo haces incluso si las probabilidades de que salga bien no te acompañan"
palabra1 = "exito"
palabra2 = "tecnologia"
mi_bool = not (palabra1 in frase) and not (palabra2 in frase)

# CONTROL DE FLUJO: IF, ELIF Y ELSE
if 10 == 9:
    print("Es correcto")
else:
    print("No es correcto")

```

```

mascota = 'perro'

if mascota == 'gato':
    print("Tienes un gato")
else:
    print("No se que animal es")
#
#
mascota = 'perro'

if mascota == 'gato':
    print("Tienes un gato")
elif mascota == 'perro':
    print("Tienes un perro")
else:
    print("No se que animal es")

edad = 18
nota = 9
if edad < 18:
    print("Menor de edad")
    if nota >= 7:
        print("Aprobado")
    else:
        print("Suspendido")
else:
    print("Mayor de edad")

# EJERCICIOS
# Utilizando las variables num1 y num2, que se alimentan con el input del usuario (tal como en el código ya proporcionado):
# Crea una estructura de control de flujo que compare los valores de las variables, y arroje un resultado de acuerdo al caso:
#     "num1 es mayor que num2"
#     "num2 es mayor que num1"
#     "num1 y num2 son iguales"
# Debes mostrar en pantalla el valor de las variables ingresadas en lugar de num1 y num2.
num1 = int(input("Ingresa un numero:"))
num2 = int(input("Ingresa otro numero:"))

if num1 > num2:
    print(f"{num1} es mayor que {num2}")
elif num2 > num1:
    print(f"{num2} es mayor que {num1}")
else:
    print(f"{num1} y {num2} son iguales")

# Las leyes de un país establecen que un adulto puede conducir si tiene licencia para hacerlo, y para optar por una licencia para conducir, debe de tener 18 años o más.
# Crea una estructura condicional para verificar si una persona de 16 años sin licencia puede conducir, y muestra el resultado que corresponda en pantalla
edad = 16
tiene_licencia = False

if not tiene_licencia:
    print("No puedes conducir aun. Debes tener 18 años y contar con una licencia")
elif edad < 18:
    print("No puedes conducir. Necesitas contar con una licencia")
else:
    print("Puedes conducir")

# Para acceder a un determinado puesto de trabajo, el candidato debe ser capaz de programar en Python y tener conocimientos de inglés.
# Crea una estructura condicional para evaluar a un candidato dadas estas condiciones, y muestra el mensaje correspondiente en pantalla
habla_ingles = True
sabe_python = False

if not habla_ingles and sabe_python:
    print("Para postularte, necesitas saber programar en Python y tener conocimientos de inglés")

elif habla_ingles:
    print("Para postularte, necesitas saber programar en Python")
elif sabe_python:
    print("Para postularte, necesitas tener conocimientos de inglés")
else:
    print("Cumples con los requisitos para postularte")

# LOOPS: bucle
# loop for:
nombres = ['Juan', 'Stefan', 'Marcos', 'Manuel']

for elemento in nombres:
    print("Hola " + elemento)

lista = ['a', 'b', 'c']
for letra in lista:
    print(letra)

lista = ['a', 'b', 'c']
for letra in lista:
    numero_letra = lista.index(letra) # saca el índice de la letra de la lista
    print(f"La letra {letra} tiene el índice {numero_letra}")

# SI EL NOMBRE EMPIEZA POR L SI QUE LO MOSTRARA.
lista = ['Juan', 'Stefan', 'Marcos', 'Manuel']
for nombre in lista:
    if nombre.startswith('M'): # metodo .startswith comprueba si empieza por el caracter que le indiques
        print(nombre)

# SI DEJAMOS EL PRINT FUERA, IMPRIMIRA EN VALOR AL RESULTADO DEL FOR.
numeros = [1,2,3,4,5]
mi_valor = 0
for numero in numeros:
    mi_valor = mi_valor + numero
print(mi_valor)

# SI DEJAMOS EL PRINT DENTRO, IMPRIMIRA UNO POR CADA VEZ QUE RECORRA EL FOR
numeros = [1,2,3,4,5]
mi_valor = 0

```

```

for numero in numeros:
    mi_valor = mi_valor + numero
    print(mi_valor)

# IMPRIME LETRA POR LETRA EL CONTENIDO DE PALABRA
palabra = "python"
for letra in palabra:
    print(letra)

# EN ESTE CASO, DENTRO DEL FOR, A ES EL PRIMER PARAMETRO DENTRO DEL OBJETO DENTRO DEL OBJETO, OSEA 1,3,5 Y B EL SEGUNDO
objeto = [[1,2],[3,4],[5,6]]
for a,b in objeto:
    print(a)
    print(b)

# SOLAMENTE IMPRIME LA CLAVE DEL DICCIONARIO
dic = {'clave1':'a','clave2':'b','clave3':'c'}
for item in dic:
    print(item)

# IMPRIME CLAVE Y VALOR
dic = {'clave1':'a','clave2':'b','clave3':'c'}
for item in dic.items():
    print(item)

# IMPRIME SOLO VALORES
dic = {'clave1':'a','clave2':'b','clave3':'c'}
for item in dic.values():
    print(item)

# EJERCICIOS
# Utilizando loops For, saluda a todos los miembros de una clase, imprimiendo "Hola" + su nombre
alumnos_clase = ["Maria", "Jose", "Carlos", "Martina", "Isabel", "Tomas", "Daniela"]

for nombre in alumnos_clase:
    print("Hola " + nombre)

# Dada la siguiente lista de numeros, realiza la suma de todos los numeros utilizando
# loops For y almacena el resultado de la suma en una variable llamada suma_numeros
lista_numeros = [1,5,8,7,6,8,2,5,2,6,4,8,5,9,8,3,5,4,2,5,6,4]
suma_numeros = 0
for numero in lista_numeros:
    suma_numeros = suma_numeros + numero
print(suma_numeros)

# Dada la siguiente lista de numeros, realiza la suma de todos los numeros pares e impares*
# por separado en las variables suma_pares y suma_impares respectivamente
lista_numeros = [1,5,8,7,6,8,2,5,2,6,4,8,5,9,8,3,5,4,2,5,6,4]
suma_pares = 0
suma_impares = 0

for par in lista_numeros:
    if par%2 == 0:
        suma_pares = suma_pares + par
print(suma_pares)

for impar in lista_numeros:
    if impar%2 == 1:
        suma_impares = suma_impares + impar
print(suma_impares)

# LOOP WHILE: se repiten MIENTRAS (while) se cumpla una condicion

monedas = 5

while monedas > 0:
    print(f"Tengo {monedas} monedas")
    monedas -= 1
else:
    print("No tienes mas monedas")

respuesta = 's'

while respuesta == 's':
    respuesta = input("sigue? (s/n)")
else:
    print("Gracias")

# METODO PASS: dentro del bucle no puedes no ponerle una accion a realizar.
# Poniendo pass, le indicas que no haga nada de momento.

respuesta = 's'

while respuesta == 's':
    pass

print("Hola")

# METODO break: Interrumpe la interaccion actual del loop y sale de el

# muestra todas las letras hasta que llega a la R, ahí se interrumpe
nombre = "marcos"
for letra in nombre:
    if letra == 'r':
        break
    print(letra)

# METODO CONTINUE: omite la condicion y pasa a la siguiente
# muestra todas las letras hasta que llega a la R, ahí se la salta y sigue
nombre = "marcos"
for letra in nombre:
    if letra == 'r':
        continue

```

```

    print(letra)

# EJERCICIOS

# Crea un Loop While que se imprima en pantalla los numeros del 10 al 0, uno a la vez.
numero = 10
while numero >= 0:
    print(numero)
    numero -= 1

# Crea un Loop While que reste de uno en uno los numeros desde el 50 al 0
# (ambos numeros incluidos) con las siguientes condiciones adicionales:
# - Si el numero es divisible por 5, mostrar dicho numero en pantalla
# (;recuerda que aqui puedes utilizar la operacion modulo dividiendo por 5 y verificando el resto!)
# - Si el numero no es divisible por 5, continuar ejecutando el loop sin
# mostrar el valor en pantalla (no te olvides de seguir restando para que el programa no corra infinitamente).
numero = 50
while numero >= 0:
    print(numero)
    if numero%5 == 0:
        numero -= 5
    else:
        continue

# Crea un loop For a lo largo de la siguiente lista de numeros, imprimiendo en pantalla cada uno de sus elementos,
# e interrumpe el flujo en el momento que encuentres un valor negativo
lista_numeros = [4,5,8,7,6,9,8,2,4,5,7,1,9,5,6,-1,-5,6,-6,-4,-3]
for numero in lista_numeros:
    if numero < 0:
        break
    print(numero)

# FUNCION RANGO: te permite crear un rango de numeros sobre los que itenere tu bucle sin necesidad de una lista

# TE MUESTRA LOS NUMEROS DEL 0 AL 4
for numero in range(5):
    print(numero)

# TE MUESTRA LOS NUMEROS DEL 1 AL 5
for numero in range(1,6):
    print(numero)

# TE MUESTRA LOS NUMEROS DEL 10 AL 60 pero con saltos de 10 en 10
for numero in range(10,61,10):
    print(numero)

# TAMBIEN PUEDES CREAR LISTAS MAS LARGAS
lista = list(range(1,101))
print(lista)

# EJERCICIOS
# Crea una lista formada por todos los numeros desde el 2500 al 2585 (inclusive). Almacena dicha lista en la variable mi_lista.
mi_lista = list(range(2500,2586))

# Utilizando la funcion range(), crea en una unica linea de codigo una lista formada por todos
# los numeros multiplos de 3 desde el 3 al 300 (inclusive). Almacena dicha lista en la variable mi_lista.
mi_lista = list(range(3,301,3))

# Utiliza la funcion range() y un loop para sumar los cuadrados de todos los numeros del 1 al 15 (inclusive).
# Almacena el resultado en una variable llamada suma_cuadrados
numeros = list(range(1,16))
suma_cuadrados = 0
for numero in numeros:
    suma_cuadrados = suma_cuadrados + numero**2
print(suma_cuadrados)

# ENUMERADOR: para acceder a los indices de una coleccion

# PARA VER EL INDICE DE LAS POSICIONES DE UNA LISTA SIN USAR ENUMERADOR
lista = ['a','b','c']
indice = 0

for item in lista:
    print(indice,item)
    indice += 1

# USANDO ENUMERADOR
lista = ['a','b','c']

for indice,item in enumerate(lista):
    print(indice,item)

# EJERCICIOS

# Imprime en pantalla frases como la siguiente:
# '{nombre} se encuentra en el indice {indice}'
# Donde nombre debe ser cada uno de los nombres de la lista a continuacion, y el indice, obtenido mediante enumerate().
lista_nombres = ["Marcos", "Laura", "Monica", "Javier", "Celina", "Marta", "Dario", "Emiliano", "Melisa"]

for indice,nombre in enumerate(lista_nombres):
    print(f'{nombre} se encuentra en el indice {indice}')

# Crea una lista formada por las tuplas (indice, elemento), formadas a partir de obtener mediante enumerate() los indices
# de cada caracter del string "Python".
# Llama a la lista obtenida con el nombre de variable lista_indices .
lista_indices = list(enumerate("Python"))

# Imprime en pantalla unicamente los indices de aquellos nombres de la lista a continuacion, que empiecen con M
lista_nombres = ["Marcos", "Laura", "Monica", "Javier", "Celina", "Marta", "Dario", "Emiliano", "Melisa"]

for indice,nombre in enumerate(lista_nombres):
    if nombre.startswith('M'):
        print(indice)

# FUNCION ZIP: combinar dos o mas listas entrelazando sus elementos en forma de tabla

```

```

nombres = ["Marcos", "Laura", "Monica"]
num_personas = len(nombres)
edades = list(range(10, 10 + 3 * num_personas, 3))
ciudades = ["Zaragoza", "Barcelona", "Lugo"]

# SI LO EJECUTAS ASI SERA TIPO ZIP
combinados = zip(nombres, edades, ciudades)
print(combinados) # <zip object at 0x000002427B1838C0>

# PARA PODER VERLO HAY QUE METERLO EN UNA LISTA
final = list(combinados)
print(final)

for nombre, edad, ciudad in combinados:
    print(f"{nombre} tiene {edad} años y vive en {ciudad}")

# EJERCICIOS

# Muestra en pantalla frases como la del siguiente ejemplo:
# La capital de Alemania es Berlin
# Utiliza la funcion zip, loops, y las siguientes listas de paises y capitales para resolverlo rapida y eficientemente.
capitales = ["Berlin", "Tokio", "Paris", "Helsinki", "Ottawa", "Canberra"]
paises = ["Alemania", "Japon", "Francia", "Finlandia", "Canada", "Australia"]

combinacion = zip(capitales, paises)

for capitales, paises in combinacion:
    print(f"La capital de {paises} es {capitales}")

# Crea un objeto zip formado a partir de listas, de un conjunto de marcas y
# productos que tu prefieras, dentro de la variable mi_zip.
marcas = ["Adidas", "Nike", "Asos"]
productos = ["Sudadera", "Chaqueta", "Zapatillas"]
mi_zip = zip(marcas, productos)

# Crea el zip con las traducciones los numeros del 1 al 5 en español, portugues e ingles
# (en el mismo orden), y convierte el objeto generado en una lista almacenada en la variable numeros:
numeros = list(zip(
    ('uno', 'dos', 'tres', 'cuatro', 'cinco'),
    ('um', 'dois', 'três', 'quatro', 'cinco'),
    ('one', 'two', 'three', 'four', 'five')
))

print(numeros)

# FUNCIONES MIN Y MAX: sirven para detectar los valores mas bajos y mas altos

menor = min(67, 25, 12, 42, 94)
print(menor)

mayor = max(67, 25, 12, 42, 94)
print(mayor)

lista = [67, 25, 12, 42, 94]
print(max(lista))
print(f"El menor es {min(lista)} y el mayor es {max(lista)}")

# TAMBIEN SIRVE CON STRINGS
nombres = ['Juan', 'Stefan', 'Marcos', 'Manuel']
print(min(nombres)) # IMPRIME EN ORDEN ALFABETICO

nombre = "Stefanz"

# BUSCA PRIMERO MAYUSCULAS LUEGO MINUSCULAS
print(min(nombre)) # S
print(min(nombre.lower())) # a

dic = {'C1': 45, 'C2': 11}
print(min(dic)) # C1. Muestra el valor mas pequeño de la parte "clave" del diccionario

print(min(dic.values())) # 11. Muestra el valor mas pequeño de la parte "valor" del diccionario

# EJERCICIOS

# Obten el valor maximo entre los valores de la siguiente lista, y almacenalo en una variable llamada valor_maximo:
lista_numeros = [44542247/2, 21310/5, 2134747*33, 44556475, 121676, 6654067, 353254, 123134, 55**12, 611**5]
valor_maximo = max(lista_numeros)

# Calcula la diferencia entre el valor maximo y el minimo en la siguiente lista de numeros, y almacenalo en una variable llamada rango
lista_numeros = [44542247, 21310, 2134747, 44556475, 121676, 6654067, 353254, 123134, 552512, 611665]

rango = (max(lista_numeros)) - (min(lista_numeros))
print(rango)

# Utilizando max(), min() y metodos de diccionarios, obten el minimo valor a partir del siguiente diccionario.
# Almacena dicho valor en una variable llamada edad_minima.
# Tambien, obten el nombre que se ubica ultimo en orden alfabetico, y almacenalo en una variable llamada ultimo_nombre.
diccionario_edades = {"Carlos": 55, "Maria": 42, "Mabel": 78, "Jose": 44, "Lucas": 24, "Rocio": 35, "Sebastian": 19, "Catalina": 2, "Dario": 49}
edad_minima = min(diccionario_edades.values())
ultimo_nombre = max(diccionario_edades)

# FUNCION RANDOM: Generacion de numeros aleatorios.

# randint(): NUMERO INTEGER aleatorio dentro del rango que le indiquemos
from random import randint
aleatorio = randint(1, 50) # GENERA UN NUMERO RANDOM ENTRE EL 1 Y EL 50
print(aleatorio)

```

```

# uniform(): NUMERO DECIMAL aleatorio dentro del rango que le indiquemos
from random import uniform
aleatorio = uniform(1,5) # te devuelve con muchos decimales
aleatorio = round(uniform(1,5),2) # te devuelve solamente 2 decimales
print(aleatorio)

# random(): imprime un numero decimal aleatorio entre 0 y 1.
from random import random
aleatorio = random()
print(round(aleatorio),2)

# choice(): imprime un objeto random de la lista que le pasemos
from random import choice
colores = ['verde', 'amarillo', 'rojo', 'azul']
aleatorio = choice(colores)
print(aleatorio)

# shuffle(): mezcla los valores de la lista que le pasemos de manera aleatoria. sirve con numeros, letras y decimales
from random import shuffle

# CON NUMEROS
numeros = list(range(5,55,5))
shuffle(numeros)
print(numeros)

# CON LETRAS O PALABRAS
palabras = ['Primero', 'Segundo', 'Tercero']
shuffle(palabras)
print(palabras)

# EJERCICIOS:

# Implementa la funcion randint() de la libreria random que te permita obtener un numero entero del 1 al 10,
# y almacena dicho valor en una variable llamada aleatorio
from random import randint
aleatorio = randint(1,10)
print(aleatorio)

# Implementa la funcion random() de la libreria random que te permita obtener un numero decimal entre 0 y 1, y
# almacena dicho valor en una variable llamada aleatorio
from random import random
aleatorio = random()
print(aleatorio)

# Utiliza el metodo choice() de la libreria random para obtener un elemento al azar de la lista de nombres a continuacion,
# y almacena el nombre escogido en una variable llamada sorteo.
from random import choice
nombres = ["Carlos", "Julia", "Nicole", "Laura", "Mailen"]
sorteo = choice(nombres)

# COMPRENSION DE LISTAS: manera dinamica de hacer listas
palabra = 'python'
lista = [letra for letra in palabra]
print(lista)

# EJERCICIOS:

# Crea una lista valores_cuadrado formada por los numeros de la lista valores, elevados al cuadrado.
valores = [1, 2, 3, 4, 5, 6, 9.5]
valores_cuadrado = [num * num for num in valores]
print(valores_cuadrado)

# Crea una lista valores_pares formada por los numeros de la lista valores que (¡adivinaste!) sean pares.
valores = [1, 2, 3, 4, 5, 6, 9.5]
valores_pares = [valores for valores in valores if valores%2 == 0]
print(valores_pares)

# Para la siguiente lista de temperaturas en grados Fahrenheit, expresa estos mismos valores en una nueva lista de valores
# de temperatura en grados Celsius. La conversion entre tipo de unidades es la siguiente: °C = (°F - 32) * (5/9) o expresado de otro modo:
# (grados_fahrenheit-32)*(5/9). La lista de temperaturas es la siguiente: temperatura_fahrenheit = [32, 212, 275]
# Almacena esta nueva lista en una variable llamada grados_celsius
temperatura_fahrenheit = [32, 212, 275]
grados_celsius = [(num-32)*(5/9) for num in temperatura_fahrenheit]
print(grados_celsius)

# FUNCION MATCH: se utiliza para verificar si una cadena al principio coincide con una expresion regular.
cliente = {'nombre':'Stefan','edad':22, 'puesto':'informatico'}

pelicula = {'titulo':'matrix',
            'ficha_tecnica':
                {'protagonista':'Stefan Vlad', 'director': 'nacho'}}

elementos = [cliente, pelicula, 'libros']

for e in elementos:
    match e:
        case {'nombre': nombre,
              'edad': edad,
              'puesto': puesto}:
            print(cliente)

```