

```

# EJERCICIOS:

# Remueve los caracteres a la izquierda de nuestro texto principal:
# Utiliza el metodo lstrip(). Imprime el resultado en pantalla.
# Busca en la documentacion acerca del funcionamiento del metodo solicitado para saber como actua.
# Puedes utilizar variables intermedias si las necesitas.

# Este metodo elimina caracteres especificos que deberemos indicarle en parentesis solo del lado izquierdo
# (o principio) de la cadena, hasta # que se encuentra un caracter que no esta en la lista de caracteres a eliminar.
x = ",:;#/,.,,:;:##Python_ _Total,.,.,,:;#"
y = x.lstrip(",:;#")
print(y)

# Añade el elemento "naranja" como el cuarto elemento de la siguiente
# lista "frutas", utilizando el metodo insert():
# Busca en la documentacion acerca del funcionamiento del metodo solicitado
# para saber como actua y como es su funcionamiento.

# METODO INSERT: Permite insertar elementos en una lista en una posicion exacta.
frutas = ["mango", "banana", "cereza", "ciruela", "pomelo"]
frutas.insert(3,"naranja")

# Verifica si los sets a continuacion forman conjuntos aislados (es decir, que no tienen elementos en comun), utilizando el
# metodo isdisjoint(). Almacena dicho resultado en la variable conjuntos_aislados
marcas_smartphones = {"Samsung", "Xiaomi", "Apple", "Huawei", "LG"}
marcas_tv = {"Sony", "Philips", "Samsung", "LG"}
conjuntos_aislados = marcas_smartphones.isdisjoint(marcas_tv)

# FUNCIONES

# PARA CREAR UNA FUNCION:
# def + nombre funcion(variable):
#   accion a realizar
def mi_funcion(nombre):
    # Esta funcion sirve para saludar
    print("Hola " + nombre)

mi_funcion("Stefan")

# Declara una funcion llamada saludar, que cada vez que sea llamada imprima en pantalla "¡Hola mundo!"
# Solo debes definir la funcion, no debes invocarla luego.
def saludar():
    print ("¡Hola mundo!")

# Declara una funcion llamada bienvenida, que tome como argumento el nombre de una persona, y que cada vez que sea
# llamada imprima en pantalla "¡Bienvenido {nombre_persona}!"
# Crea la variable nombre_persona, y almacena dentro de la misma el nombre que prefieras.
def bienvenida(nombre_persona):
    print(f"¡Bienvenido {nombre_persona}!")

nombre_persona = 'Stefan'

# Declara una funcion llamada cuadrado, que tome como argumento un numero cualquiera, y que cada vez que sea llamada, imprima
# en pantalla el cuadrado de dicho numero (es decir, la potencia 2 del valor).
# El nombre del argumento que debe tomar dicha funcion es un_numero. Crea dicha variable y asigne un numero cualquiera.
def cuadrado(un_numero):
    print(un_numero**2)

un_numero = 2

# RETURN: sirve para almacenar el resultado de una funcion y poder crear variables a partir de ellos.
def sumar(num1,num2):
    total = num1+num2
    return total # ALMACENA EL RESULTADO

resultado = sumar(10,20)
print(resultado)

# EJERCICIOS:

# Crea una funcion llamada potencia que tome dos valores numericos como argumentos. Debera devolver el numero que resulte
# de resolver una potencia, utilizando el primer numero como base, y el segundo como exponente
def potencia(num1, num2):
    return num1 ** num2

# Crea una funcion llamada usd_a_eur que tome como unico parametro un valor numerico (un monto en dolares estadounidenses),
# y devuelva como resultado el monto equivalente en euros. A fines de este ejemplo, tomaremos la conversion 1 USD = 0.90 EUR.
# Crea una variable llamada dolares y almacena en ella un monto cualquiera para entregarselo a tu funcion y evaluar su resultado.
def usd_a_eur(dolares):
    return dolares*0.9

dolares = 2
print(usd_a_eur(dolares))

# Crea una funcion llamada invertir_palabra que tome los caracteres de una palabra dada como argumento, invierta el orden de sus
# caracteres y los devuelva de ese modo y en mayusculas. Por ejemplo, si le proporcionamos la palabra "Python", debera devolver: "NOHTYP"
# Tambien, deberas crear una variable llamada palabra, que contenga el string que tu prefieras, para sumisitrarle como argumento a la funcion creada.
def invertir_palabra(palabra):
    return palabra[::-1].upper()

palabra = "Curso"

print(invertir_palabra(palabra))

# FUNCIONES DINAMICAS:

# COMPROBAR SI EL NUMERO QUE INTRODUCIMOS TIENE 3 CIFRAS
def comprobar_3_cifras(num1):
    return num1 in range(100,1000)

numero = int(input("Introduce un numero: "))
resultado = comprobar_3_cifras(numero)
print(resultado)

# COMPROBAR SI ALGUN NUMERO DE LA LISTA TIENE 3 CIFRAS
def comprobar_3_cifras(lista):
    for n in lista:
        if n in range(100,1000):
            return True

```

```

        else:
            pass
        return False

resultado = comprobar_3_cifras([97,527,600])
print(resultado)

# COMPROBAR SI ALGUN NUMERO DE LA LISTA TIENE 3 CIFRAS E IMPRIMIRLAS
def comprobar_3_cifras(lista):

    lista_3_cifras = []

    for n in lista:
        if n in range(100,1000):
            lista_3_cifras.append(n)
        else:
            pass

    return lista_3_cifras

resultado = comprobar_3_cifras([55,994,60000])
print(resultado)

# EJERCICIOS

# Crea una funcion (todos_positivos) que reciba una lista de numeros como parametro, y devuelva True si todos los valores
# de una lista son positivos, y False si al menos uno de los valores es negativo. Crea una lista llamada lista_numeros con valores positivos y negativos.
def todos_positivos(lista_numeros):
    for n in lista_numeros:
        if n > 0:
            pass
        else:
            return False
    return True

lista_numeros = [1, -50, 502, -5000, 755, 600, 33, 6]
resultado = todos_positivos(lista_numeros)
print(resultado)

# Crea una funcion (suma_menores) que sume los numeros de una lista (almacenada en la variable lista_numeros) siempre y
# cuando sean mayores a 0 y menores a 1000, y devuelva el resultado de dicha suma.
def suma_menores(lista_numeros):
    suma = 0
    for n in lista_numeros:
        if n in range(0,1000):
            suma += n
    return suma

lista_numeros = [1,5,8,7,65,8,32,5,2,26,34,8,5,9,8,3,5,42,2,5,16,4]
resultado = suma_menores(lista_numeros)
print(resultado)

# Crea una funcion (cantidad_pares) que cuente la cantidad de numeros pares que existen en una lista (lista_numeros)
# y devuelva el resultado de dicha cuenta.

lista_numeros = [1, 5, 8, 7, 65, 8, 32, 5, 2, 26, 34, 8, 5, 9, 8, 3, 5, 42, 2, 5, 16, 4]

def cantidad_pares(lista_numeros):
    cantidad = 0

    for n in lista_numeros:
        if n % 2 == 0:
            cantidad += 1
    return cantidad

resultado = cantidad_pares(lista_numeros)
print(resultado)

# EJEMPLO PRACTICO DE FUNCIONES

# LISTAR CAFES
precios_cafe = [("Cortado", 1.5), ("Solo", 2), ("Americano", 1.25)]

for elemento in precios_cafe:
    print(elemento)

# CUAL ES EL CAFE MAS CARO?
precios_cafe = [("Cortado", 1.5), ("Solo", 2), ("Americano", 1.25)]

def caro(lista_precios):
    precio_mayor = 0
    cafe_mas_caro = ''

    for cafe,precio in lista_precios:
        if precio > precio_mayor:
            precio_mayor = precio
            cafe_mas_caro = cafe
        else:
            pass

    return cafe_mas_caro, precio_mayor

print(caro(precios_cafe))
cafe,precio = caro(precios_cafe)
print(f"El cafe mas caro es el {cafe} con un precio de {precio}€")

# INTERACCIONES ENTRE FUNCIONES.

# VAMOS A HACER UN JUEGO DE "ADIVINA EL PALO". VARIOS PALOS CON UN TAMAÑO DIFERENTE, EL USUARIO DEBERA ELEGIR UNO DE ELLOS.
# SI LE TOCA EL PALO MAS CORTO PIERDE. LOS PASOS A SEGUIR PARA REALIZAR ESTO SON LOS SIGUIENTES

from random import shuffle

# LISTA INICIAL
palos = ['-','-','---','----']

# MEZCLAR PALOS
def mezclar(lista):
    shuffle(lista)
    return lista

# PEDIRLE INTENTO

```

```

def intento():
    intento = ''

    while intento not in ['1','2','3','4']:
        print("El numero debe ser entre el 1 y el 4, ambos incluidos.")
        intento = input("Elige un numero del 1 al 4: ")

    return int(intento)

# COMPROBACION NUMERO DE INTENTO
def comprobar_intento(lista_palos,intento):
    if lista_palos[intento - 1] == '-':
        print("Has perdido panoli")
    else:
        print("Has ganado")
    print(f"El palo que te ha tocado es {lista_palos[intento-1]}")

palos_mezclados = mezclar(palos)
seleccion = intento()
comprobar_intento(palos_mezclados,seleccion)

# EJERCICIOS

# Crea una funcion (lanzar_dados) que arroje dos dados al azar y devuelva sus resultados:
# La funcion debe retornar dos valores resultado, que se encuentren entre 1 y 6).
# Dicha funcion no debe requerir argumentos para funcionar, sino que debe generar internamente los valores aleatorios.
# Proporciona el resultado de estos dos dados a una funcion que se llame evaluar_jugada (es decir, esta segunda funcion debe
# recibir dos argumentos) y que retorne -sin imprimirlo- un mensaje segun la suma de estos valores:
# Si la suma es menor o igual a 6:
# "La suma de tus dados es {suma_dados}. Lamentable"
# Si la suma es mayor a 6 y menor a 10:
# "La suma de tus dados es {suma_dados}. Tienes buenas chances"
# Si la suma es mayor o igual a 10:
# "La suma de tus dados es {suma_dados}. Parece una jugada ganadora"

from random import randint

def lanzar_dados():
    resultado1 = randint(0, 6)
    resultado2 = randint(0, 6)

    return resultado1, resultado2

uno, dos = lanzar_dados()
# print(uno, dos)

def evaluar_jugada(num1, num2):
    suma_dados = num1 + num2

    if suma_dados <= 6:
        # print(f"La suma de tus dados es {suma_dados}. Lamentable")
        return (f"La suma de tus dados es {suma_dados}. Lamentable")

    elif (suma_dados > 6) and (suma_dados < 10):
        # print(f"La suma de tus dados es {suma_dados}. Tienes buenas chances")
        return (f"La suma de tus dados es {suma_dados}. Tienes buenas chances")

    else:
        # print(f"La suma de tus dados es {suma_dados}. Parece una jugada ganadora")
        return (f"La suma de tus dados es {suma_dados}. Parece una jugada ganadora")

evaluar_jugada(uno, dos)

# Crea una funcion llamada reducir_lista() que tome una lista como argumento (crea tambien la variable lista_numeros), y
# devuelva la misma lista, pero eliminando duplicados (dejando uno solo de los numeros si hay repetidos) y eliminando el valor mas alto. El orden de los elementos puede modificarse.
# Por ejemplo, si se le proporciona la lista [1,2,15,7,2] debe devolver [1,2,7].
# Crea una funcion llamada promedio() que pueda recibir como argumento la lista devuelta por la anterior funcion, y que
# calcule el promedio de los valores de la misma. Debe devolver el resultado, sin imprimirlo.

lista_numeros = [1, 2, 15, 7, 2]

def reducir_lista(lista):
    duplicados = []

    for n in lista:
        if n not in duplicados:
            duplicados.append(n)
        else:
            pass
    duplicados.sort()
    duplicados.pop()

    return duplicados

def promedio(resultado):
    div=len(resultado)
    suma=0

    for n in resultado:
        suma = suma+n
    media = round(suma/div,1)

    return media

# Crea una funcion (llamada lanzar_moneda) que devuelva el resultado de lanzar una moneda (al azar). Dicha funcion debe
# poder devolver los resultados "Cara" o "Cruz", y no debe recibir argumentos para funcionar.
# Crea una segunda funcion (llamada probar_suerte), que tome dos argumentos: el primero, debe ser el resultado del lanzamiento
# de la moneda. El segundo argumento, sera una lista de numeros cualquiera (debes crear una lista con valores y llamarla lista_numeros).
# Si se le proporciona una "Cara", debe mostrar el mensaje al usuario: "La lista se autodestruira", y eliminarla
# (devolverla como lista vacia []). Si se le proporciona una "Cruz", debe imprimir en pantalla: "La lista fue salvada" y devolver la lista intacta.
# Pistas: utiliza el metodo choice de la biblioteca random para elegir un elemento al azar de una secuencia.

from random import choice

lista_numeros = [1, 2, 3, 4]

def lanzar_moneda():
    moneda = choice(['Cara', 'Cruz'])
    return moneda

def probar_suerte(moneda, lista):
    if moneda == 'Cara':
        lista.clear()
        print(f"La lista se autodestruira:{lista}")
        return lista
    else:
        print(f"La lista fue salvada {lista}")

```

```

        return lista

lanzar = lanzar_moneda()
print(probar_suerte(lanzar, lista_numeros))

# ARGUMENTOS INDEFINIDOS: *args y **kwargs: se pueden definir funciones las cuales los numeros de los argumentos es variable.

# *args

# SUMA TODOS LOS NUMEROS, SIN DEFINIR LA CANTIDAD A PASAR:
def suma(*args):
    total = 0

    for n in args:
        total += n
    return total

print(suma(5,5,6,342,4,56))

# LO MISMO
def suma(*args):

    return sum(args)

print(suma(5,5,6,342,4,56))

# EJERCICIOS

# Crea una funcion llamada suma_cuadrados que tome una cantidad indeterminada de argumentos numericos, y que retorne la
# suma de sus valores al cuadrado.

def suma_cuadrados(*args):
    total = 0

    for n in args:
        n = n**2
        total += n
    return total

print(suma_cuadrados(2,3,5,-7))

# Crea una funcion llamada suma_absolutos, que tome un conjunto de argumentos de cualquier extension, y retorne la suma de sus valores absolutos (es decir,
# que tome los valores sin signo y los sume, o lo que es lo mismo, los considere a todos -negativos y positivos- como positivos).
def suma_absolutos(*args):
    total = 0

    for n in args:
        total += abs(n)
    return total

print(suma_absolutos(2,3,5,-7))

# Crea una funcion llamada numeros_persona que reciba, como primer argumento, un nombre, y a continuacion, una cantidad indefinida de numeros.
# La funcion debe devolver el siguiente mensaje:
# "(nombre), la suma de tus numeros es {suma_numeros}"
def numeros_persona(nombre, *args):
    suma_numeros = 0

    for n in args:
        suma_numeros += n

    return(f"{nombre}, la suma de tus numeros es {suma_numeros}")

print(numeros_persona("Stefan", 2, 3, 5, -7))

# **kwargs
def suma(**kwargs):
    for clave,valor in kwargs.items():
        print(f"{clave} = {valor}")

suma(x=3,y=5,z=3)

def suma(**kwargs):

    total = 0

    for clave,valor in kwargs.items():
        print(f"{clave} = {valor}")
        total += valor

    return total

print(suma(x=3,y=5,z=3))

def prueba(num1, num2, *args, **kwargs):

    print(f"El primer valor es {num1}")
    print(f"El segundo valor es {num2}")

    for arg in args:
        print(f"arg = {arg}")

    for clave,valor in kwargs.items():
        print(f"{clave} = {valor}")

prueba(15,50,256,55512,55632,54412,x="uno", y="dos", z="tres")

# EJERCICIOS

# Crea una funcion llamada cantidad_atributos que cuente la cantidad de paremetros que se entregan,
# y devuelva esa cantidad como resultado.

def cantidad_atributos(**kwargs):
    cantidad = 0

    for clave, valor in kwargs.items():
        cantidad += 1

    return cantidad

print(cantidad_atributos(x=324, z=23, u=5213, y=45213))

```

```
# Crea una funcion llamada lista_atributos que devuelva en forma de lista los valores de los atributos entregados en forma
# de palabras clave (keywords). La funcion debe preveer recibir cualquier cantidad de argumentos de este tipo.
def lista_atributos(**kwargs):
    lista = []
    for valor in kwargs.values():
        lista.append(valor)
    print(lista)
    return lista
```

```
lista_atributos(nombre='Stefan', apellido='Vlad', nac='01/01/2000')
```

```
# Crea una funcion llamada describir_persona, que tome como parametros su nombre y luego una cantidad
# indeterminada de argumentos. Esta funcion debera mostrar en pantalla
```

```
def describir_persona(nombre, **kwargs):
    print(f"Caracteristicas de {nombre}:")

    for clave, valor in kwargs.items():
        print(f"{clave}: {valor}")
```

```
# Crea una funcion llamada devolver_distintos() que reciba 3
# integers como parametros.
# Si la suma de los 3 numeros es mayor a 15, va a devolver el
# numero mayor.
# Si la suma de los 3 numeros es menor a 10, va a devolver el
# numero menor.
# Si la suma de los 3 numeros es un valor entre 10 y 15
# (incluidos) va a devolver el numero de valor intermedio.
```

```
def devolver_distintos(a,b,c):
    suma = a+b+c
    lista = [a,b,c]

    if suma > 15:
        return max(lista)
    elif suma < 10:
        return min(lista)
    else:
        lista.sort()
        return lista[1]
```

```
devolver_distintos(1,3,6)
```

```
# Escribe una funcion (puedes ponerle cualquier nombre que
# quieras) que reciba cualquier palabra como parametro, y que
# devuelva todas sus letras unicas (sin repetir) pero en orden
# alfabetico.
# Por ejemplo si al invocar esta funcion pasamos la palabra
# "entretenido", deberia devolver ['d', 'e', 'i', 'n', 'o', 'r', 't']
```

```
def letras_unicas(palabra):
    mi_set = set()

    for letra in palabra:
        mi_set.add(letra)

    mi_lista = list(mi_set)
    mi_lista.sort()

    return mi_lista
```

```
print(letras_unicas("imaginacion"))
```

```
# Escribe una funcion que requiera una cantidad indefinida de
# argumentos. Lo que hara esta funcion es devolver True si en
# algun momento se ha ingresado al numero cero repetido dos
# veces consecutivas.
```

```
def ceros_vecinos(*args):
    contador = 0

    for num in args:
        if contador + 1 == len(args):
            return False
        elif args[contador] == 0 and args[contador + 1] == 0:
            return True
        else:
            contador += 1

    return False
```

```
print(ceros_vecinos(3,5,0,0,3,54))
```

```
# Escribe una funcion llamada contar_primos() que requiera un
# solo argumento numerico.
# Esta funcion va a mostrar en pantalla
# todos los numeros
# primos existentes en el rango que va desde cero hasta ese
# numero incluido, y va a devolver la cantidad de numeros
# primos que encontro.
# Aclaracion, por convencion el 0 y el 1 no se consideran primos.
```

```
def contar_primos(numero):
    primos = [2]
    iteracion = 3

    if numero < 2:
        return 0

    while iteracion <= numero:
        for n in range(3,iteracion,2):
            if iteracion % n == 0:
                iteracion += 2
                break
            else:
                primos.append(iteracion)
                iteracion += 2
        print(primos)
```

```
    return len(primos)
print(contar_primos(1))
```