

# ***SPRINT 7***

---

**Edoardo Brega**

# ★ NIVELL 1

## → exercici 1

- Calculadora de l'índex de massa corporal:

**Escriu una funció que calculi l'IMC ingressat per l'usuari/ària, és a dir, qui ho executi haurà d'ingressar aquestes dades.**

```
# Importo la biblioteca `keyboard` para utilizar el boton 'esc'
import keyboard

# Creo una funcion que acepta dos valores de entradas y los reconoce como 'peso' y 'altura'
def calc_imc(peso, altura):

    # Calculo el IMC
    imc = peso / (altura ** 2)

    # Clasifico el IMC calculado en categorías
    if imc < 18.5: cat = "Bajo Peso"
    elif 18.5 <= imc < 25: cat = "Peso Normal"
    elif 25 <= imc < 30: cat = "Sobrepeso"
    else: cat = "Obesidad"

    # Devuelvo el IMC calculado la categoría clasificada
    return imc, cat

# Mensaje de inicio del programa
print("Programa para el cálculo del Índice de Masa Corporal")

# Empiezo el bucle para recibir el input correcto del peso
while True:

    # Si el usuario presiona 'esc' el programa acaba
    if keyboard.is_pressed('esc'):
        print("Salida del programa")
        break

    # Pido al usuario que ingrese el peso y cambio la coma con el punto si hace falta para poderlo aceptar como numero
    peso = input("Ingresa el peso en 'Kg' o presiona ESC para salir").replace(",", ".")

    # Intento convertir el valor ingresado a un numero decimal (float)
    try:
        peso = float(peso)

        # Si el valor es un numero decimal, controlo que este en el rango de certificacion del IMC
        if 29.9 <= peso <= 250.1:

            # Si el peso esta en este rango, salgo del bucle con el break para ir a pedir la altura
            break

        # Si el peso no esta en el rango, informo el usuario y vuelvo al principio del while para que el usuario ingrese otra vez el peso
    except:
        print(f"El valor ingresado es {peso} kg\nLa masa corporal se certifica solo para valores de peso entre 30 kg y 250 kg\nIngresa otra vez el peso")
```

```

# Si el valor ingresado no es un numero y no es 'esc' informo el usuario y vuelvo al principio del while para que el usuario ingrese otra vez el peso
except ValueError:
    if not keyboard.is_pressed('esc'):
        print(f"El valor ingresado no es válido: {peso}\nIngresa otra vez el peso")

# Si el usuario no ha presionado 'esc' procedo a pedir la altura con un nuevo bucle
if not keyboard.is_pressed('esc'):
    while True:

        # Si el usuario presiona 'esc' el programa acaba
        if keyboard.is_pressed('esc'):
            print("Salida del programa")
            break

        # Pido al usuario que ingrese la altura y cambio la coma con el punto si hace falta para poderlo aceptar como numero
        altura = input("Ingresa la altura en 'm' o presiona ESC para salir").replace(",", ".")

        # Intento convertir el valor ingresado a un numero decimal (float)
        try:
            altura = float(altura)

            # Si el valor es un numero decimal, controlo que este en el rango de certificacion del IMC
            if 0.9 <= altura <= 2.6:

                # Si la altura esta en este rango, salgo del bucle con el break para proceder al calculo
                break

            # Si la altura no esta en el rango, informo el usuario y vuelvo al principio del while para que el usuario ingrese otra vez la altura
            else:
                print(f"El valor ingresado es {altura} m\nLa masa corporal se certifica solo para valores de altura entre 1 m y 2.5 m\nIngresa otra vez la altura")

        # Si el valor ingresado no es un numero y no es 'esc' informo el usuario y vuelvo al principio del while para que el usuario ingrese otra vez la altura
        except ValueError:
            if not keyboard.is_pressed('esc'):
                print(f"El valor ingresado no es válido: {altura}\nIngresa otra vez la altura")

# Si el usuario ha ingresado bien peso y altura y no ha presionado 'esc' procedo al calculo a traves de la funcion
if not keyboard.is_pressed('esc'):

    # Defino dos variables que seran los resultados de la funcion
    res_imc, res_cat = calc_imc(peso, altura)

    # Muestro a video los resultados
    print(f"Datos de peso y altura recibidos: {peso} Kg, {altura} m\nIMC = {res_imc:.2f}\nCategoría = {res_cat}")

```

## → exercici 2

- **Convertidor de temperatures:**

Existeixen diverses unitats de temperatura utilitzades en diferents contextos i regions. Les més comunes són Celsius (°C), Fahrenheit (°F) i Kelvin (K). També existeixen altres unitats com Rankine (°Ra) i Réaumur (°Re). Selecciona almenys 2 conversors, de tal manera que en introduir una temperatura retorni, com a mínim, dues conversions.

```
# Importo la biblioteca `keyboard` para utilizar el boton 'esc'
import keyboard
```

```
# Creo las funciones para convertir las temperaturas
```

```
def celsius_2_fahrenheit(celsius): return (celsius * 9/5) + 32
def celsius_2_kelvin(celsius): return celsius + 273.15
def fahrenheit_2_celsius(fahrenheit): return (fahrenheit - 32) * 5/9
def fahrenheit_2_kelvin(fahrenheit): return (fahrenheit - 32) * 5 / 9 + 273.15
def kelvin_2_celsius(kelvin): return kelvin - 273.15
def kelvin_2_fahrenheit(kelvin): return (kelvin - 273.15) * 9 / 5 + 32
```

```
# Mensaje de inicio del programa
```

```
print("Programa para la conversión de temperaturas entre Celsius (c), Fahrenheit (f) e Kelvin (k)")
```

```
# Empiezo un bucle para pedir la temperatura
```

```
while True:
```

```
    # Si el usuario presiona 'esc' el programa acaba
```

```
    if keyboard.is_pressed('esc'):
        print("Salida del programa")
        break
```

```
    # Pido al usuario que ingrese la temperatura recordando de ingresar la unidad de medida tambien, quito el espacio, la coma o el simbolo ° si hace falta.
```

```
    temp_in = input("Ingresa la temperatura que quieres convertir con su unidad de medida\n(ejemplos: 25C, 77F, 300K): ").strip().replace(' ', '').replace('°', '')
```

```
    # Este control me ayuda a no tener mensajes duplicados o errores en el caso el usuario presione 'esc' o no ingrese ningun valor
```

```
    if not temp_in:
        continue
```

```
    # Empiezo el reconocimiento del valor ingresado
```

```
    try:
```

```
        # Si se reconoce el valor como celsius, procedo a las conversiones
```

```
        if temp_in[-1].upper() == 'C':
```

```
            # Quito el ultimo caracter que deberia ser la unidad de medida y intento la conversion a numero decimal
```

```
            celsius = float(temp_in[:-1])
```

```
            # Verifico que la temperatura este dentro del los valores permitidos por la unidad de medida
```

```
            if -273.15 <= celsius <= 1000:
```

```
                # Muestro a video las conversiones
```

```
                print(f"{celsius}°C = {celsius_2_fahrenheit(celsius):.2f}°F = {celsius_2_kelvin(celsius):.2f}K")
                break
            else:
```

```
                # Si el valor esta fuera del rango, informo el usuario para que vuelva a ingresarla bien
```

```
                print("Las temperaturas en Celsius tienen que ser entre -273.15°C y 1000°C")
```

```
        # Si se reconoce el valor como fahrenheit, procedo a las conversiones
```

```

elif temp_in[-1].upper() == 'F':
    fahrenheit = float(temp_in[-1])

    # Verifico que la temperatura este dentro del los valores permitidos por la unidad de medida
    if -459.67 <= fahrenheit <= 1832:

        # Muestro a video las conversiones
        print(f"{fahrenheit}°F = {fahrenheit_2_celsius(fahrenheit):.2f}°C = {fahrenheit_2_kelvin(fahrenheit):.2f}K")
        break
    else:

        # Si el valor esta fuera del rango, informo el usuario para que vuelva a ingresarla bien
        print("Las temperaturas en Fahrenheit tienen que ser entre -459.67°F y 1832°F")

    # Si se reconoce el valor como kelvin, procedo a las conversiones
elif temp_in[-1].upper() == 'K':
    kelvin = float(temp_in[-1])

    # Verifico que la temperatura este dentro del los valores permitidos por la unidad de medida
    if 0 <= kelvin <= 1273.15:

        # Muestro a video las conversiones
        print(f"{kelvin}K = {kelvin_2_celsius(kelvin):.2f}°C = {kelvin_2_fahrenheit(kelvin):.2f}°F")
        break
    else:

        # Si el valor esta fuera del rango, informo el usuario para que vuelva a ingresarla
        print("Las temperaturas en Kelvin tienen que ser entre 0K y 1273.15K")

else:

    # Si el ultimo caracter no es uno de los 3 requeridos, informo el usuario para que vuelva a ingresar la temperatura
    print("Unidad de medida no reconocida. Recuerda añadir C, F o K después del valor de la temperatura")

    # En caso que el valor no sea un numero quitado el ultimo caracter, informo el usuario para que vuelva a ingresar la temperatura
except ValueError:
    print("Ingresa un valor numérico válido para la temperatura")

```

## → exercici 3

- Comptador de paraules d'un text:

**Escriu una funció que donat un text, mostri les vegades que apareix cada paraula.**

```
# Importo la biblioteca `keyboard` para utilizar el boton 'esc'
import keyboard

# Creo la funcion para contar las palabras
def comptador_paraules(texto):

    # Convierto el texto en minusculas
    texto = texto.lower()

    # Convierto el texto en una lista de palabras usando el espacio como separador
    palabras = texto.split()

    # Inicializo un diccionario para almacenar y contar las palabras
    contador = {}

    # Empiezo un ciclo for para contar las palabras
    for word in palabras:

        # Verifico si la palabra ya esta en el diccionario y aumento de uno el contador de la palabra
        if word in contador:
            contador[word] += 1

        # Si la palabra no esta en el diccionario, la añado y inicializo el valor de la llave a 1
        else:
            contador[word] = 1

    # Devuelvo el diccionario
    return contador

# Mensaje de inicio del programa
print("Programa para contar las veces que aparece cada palabra o cada número en un texto")

# Empiezo un bucle para que el usuario ingrese el texto
while True:

    # Si el usuario presiona 'esc' el programa acaba
    if keyboard.is_pressed('esc'):
        print("Salida del programa")
        break

    # Pido al usuario que ingrese el texto
    texto_in = input("Escribe el texto: \n")

    # Llamo la funcion para el recuento de las palabras ingresadas
    resultado = comptador_paraules(texto_in)

    # Si el usuario no ha presionado 'esc' muestro a video el resultado
    if not keyboard.is_pressed('esc'):
        print("\nResultado:")

    # Utilizo sorted para ordenar alfabeticamente el diccionario
    for palabra in sorted(resultado.keys()):

        # Muestra cada palabra con su recuento y acaba el programa
        print(f"'{palabra}' = {resultado[palabra]} ")
    break
```

## → exercici 4

- **Diccionari invers:**

**Resulta que el client té una enquesta molt antiga que s'emmagatzema en un diccionari i els resultats els necessita al revés, és a dir, intercanviats les claus i els valors. Els valors i claus en el diccionari original són únics; si aquest no és el cas, la funció hauria d'imprimir un missatge d'avertiment.**

**# Creo una funcion para invertir un diccionario**

```
def diccionario_inverso(diccionario_original):
```

```
    # Inicializo un diccionario que almanecera las llaves y los valores invertidos
```

```
    diccionario_inv = {}
```

```
    # Inicializo un diccionario que almanecera los duplicados
```

```
    duplicates = {}
```

```
    # Empiezo un ciclo for para recorrer los elementos del diccionario original
```

```
    for key, value in diccionario_original.items():
```

```
        # Controlo si el valor que quiero sea la nueva llave ya existe en el diccionario invertido
```

```
        if value in diccionario_inv:
```

```
            # Si el elemento no esta como llave en el diccionario de los duplicados lo añado
```

```
            if value not in duplicates:
```

```
                duplicates[value] = [diccionario_inv[value]]
```

```
            # Si el elemento ya esta como llave en el diccionario de los duplicados añado la llave actual a la lista de claves asociadas a ese elemento duplicado
```

```
            duplicates[value].append(key)
```

```
        # Si el valor no es duplicado lo agrego al diccionario invertido con la clave como valor
```

```
        else:
```

```
            diccionario_inv[value] = key
```

```
    # La funcion devuelve el diccionario invertido y el diccionario de duplicados
```

```
    return diccionario_inv, duplicates
```

**# Diccionario de prueba 1 con llaves únicas**

```
diccionario_original1 = {
```

```
    'a': 1,
```

```
    'b': 2,
```

```
    'c': 3,
```

```
}
```

**# Diccionario de prueba 2 con llaves duplicadas**

```
diccionario_original2 = {
```

```
    'x': 'apple',
```

```
    'y': 'banana',
```

```
    'z': 'banana',
```

```
}
```

**# Muestro a video el primer diccionario original**

```
print(f"***\nDiccionario original 1:\n {diccionario_original1}")
```

**# Llamo a la funcion**

```
dic_inv1, duplicados1 = diccionario_inverso(diccionario_original1)
```

**# Muestro el diccionario invertido**

```
print(f"\nDiccionario invertito 1: \n{dic_inv1}")
```

**# Si el diccionario de duplicado existe muestro tambien los duplicados**

```
if duplicados1:
```

```
print(f"\nError: No se ha invertido todo el diccionario\nSe han encontrado estos duplicados: \n{duplicados1}\n")

# Si el diccionario de duplicados no existe muestro el mensaje a video para informar el usuario
else:
    print("\nEl diccionario ha sido invertido correctamente\n")

# Muestro a video el segundo diccionario original
print(f"\n***\nDiccionario original 2:\n {diccionario_original2}")

# Llamo a la funcion
dic_inv2, duplicados2 = diccionario_inverso(diccionario_original2)

# Muestro el diccionario invertido
print(f"\nDiccionario invertito 2: \n{dic_inv2}")

# Si el diccionario de duplicado existe muestro tambien los duplicados
if duplicados2:
    print(f"\nError: No se ha invertido todo el diccionario\nSe han encontrado estos duplicados: \n{duplicados2}\n")

# Si el diccionario de duplicados no existe muestro el mensaje a video para informar el usuario
else:
    print("\nEl diccionario ha sido invertido correctamente\n")
```



## ★ NIVELL 2

### → exercici 1

- **Diccionari invers amb duplicats:**

Continuant amb l'exercici 4 del nivell 1: al client es va oblidar de comentar un detall i resulta que els valors en el diccionari original poden duplicar-se i més, per la qual cosa les claus intercanviades poden tenir duplicats. En aquest cas, en l'exercici anterior imprimies un missatge d'avertiment, ara, els valors del diccionari resultant hauran d'emmagatzemar-se com una llista. Tingues en compte que si és un valor únic no ha de ser una llista.

```
# Creo una funcion para invertir un diccionario
```

```
def diccionario_inverso(diccionario_original):
```

```
    # Inicializo un diccionario que almanecera las llaves y los valores invertidos
```

```
    diccionario_inv = {}
```

```
    # Inicializo un diccionario que almanecera los duplicados
```

```
    duplicates = {}
```

```
    # Empiezo un ciclo for para recorrer los elementos del diccionario original
```

```
    for key, value in diccionario_original.items():
```

```
        # Controlo si el valor que quiero sea la nueva llave ya existe en el diccionario invertido
```

```
        if value in diccionario_inv:
```

```
            # Si el elemento ya existe pero aun no es una lista, lo convierto en una lista para almacenar todos los duplicados de la misma llave
```

```
            if not isinstance(diccionario_inv[value], list):
```

```
                diccionario_inv[value] = [diccionario_inv[value]]
```

```
            # Añado la llave a la lista de duplicados
```

```
            diccionario_inv[value].append(key)
```

```
            # Añado esta lista de duplicados en el diccionario de duplicados
```

```
            duplicates[value] = diccionario_inv[value]
```

```
        # Si el valor no es un duplicado lo añado al diccionario invertido
```

```
        else:
```

```
            diccionario_inv[value] = key
```

```
    # La funcion devuelve el diccionario invertido y el diccionario de duplicados
```

```
    return diccionario_inv, duplicates
```

```
# Diccionario de prueba 1 con llaves únicas
```

```
diccionario_original1 = {'a': 1, 'b': 2, 'c': 3}
```

```
# Diccionario de prueba 2 con llaves duplicadas
```

```
diccionario_original2 = {'x': 'apple', 'y': 'banana', 'z': 'banana'}
```

```
# Muestro a video el primer diccionario original
```

```
print(f"\n*****\nDiccionario original 1:\n{diccionario_original1}")
```

```
# Llamo a la funcion
```

```
dic_inv1, duplicados1 = diccionario_inverso(diccionario_original1)
```

```
# Si el diccionario de duplicado existe muestro el diccionario invertido y el diccionario de duplicados
```

```
if duplicados1:
```

```
print(f"\nDiccionario con duplicados transformado en lista:\n{dic_inv1}\n\nValores duplicados encontrados: {duplicados1}\n")
```

**# Si el diccionario de duplicados no existe muestro el diccionario invertido y informo el usuario**

else:

```
print(f"\nDiccionario invertido, no habian duplicados:\n{dic_inv1}\n")
```

**# Muestro a video el segundo diccionario original**

```
print(f"\n*****\nDiccionario original 2:\n{diccionario_original2}")
```

**# Llamo a la funcion**

```
dic_inv2, duplicados2 = diccionario_inverso(diccionario_original2)
```

**# Si el diccionario de duplicado existe muestro el diccionario invertido y el diccionario de duplicados**  
if duplicados2:

```
print(f"\nDiccionario con duplicados transformado en lista:\n{dic_inv2}\n\nValores duplicados encontrados: {duplicados2}\n")
```

**# Si el diccionario de duplicados no existe muestro el diccionario invertido y informo el usuario**

else:

```
print(f"\nDiccionario invertido, no habian duplicados:\n{dic_inv2}\n")
```

## → exercici 2

- La gerència està interessada a analitzar més a fons les vendes en relació amb el mes. Per tant, et demanen que facis els ajustos necessaris per a mostrar la informació d'aquesta manera.

```
# Creo una funcion para separar los numeros de los elementos que no son numeros
def separa_numeros(lista):

    # Inicializo una lista para los elementos convertibles
    lista_de_convertibles = []

    # Inicializo una lista para los elementos no convertibles
    lista_de_no_convertibles = []

    # Empiezo un ciclo for para controlar los elementos de la lista recibida
    for item in lista:

        # Controlo si el elemento es una tupla o una lista
        if isinstance(item, (tuple, list)):

            # Si el elemento es una tupla o una lista empiezo un ciclo for para controlar los elementos que lleva dentro
            for sub_item in item:

                # Intento convertir el elemento dentro la tupla o la lista en un numero decimal float y añadirlo a la lista de convertibles
                try:
                    lista_de_convertibles.append(float(sub_item))

                # Si no se puede convertir lo añado a la lista de no convertibles
                except (ValueError, TypeError):
                    lista_de_no_convertibles.append(sub_item)

            # Si el elemento no es una tupla o lista intento convertirlo en un numero decimal float y añadirlo a la lista de convertibles
            else:
                try:
                    lista_de_convertibles.append(float(item))

                # Si no se puede convertir lo añado a la lista de no convertibles
                except (ValueError, TypeError):
                    lista_de_no_convertibles.append(item)

    # La funcion devuelve las dos listas
    return lista_de_convertibles, lista_de_no_convertibles

# Lista de prueba
lista_cliente = ['1.3', 'one', '1e10', 'seven', '3-1/2', ('2', 1, 1.4, 'not-a-number'), [1, 2, '3', '3.4']]

# Llamo la funcion
lista_convertibles, lista_no_convertibles = separa_numeros(lista_cliente)

# Muestro a video las dos listas
print(f"Elementos convertibles: \n{lista_convertibles}\nElementos no convertibles: \n{lista_no_convertibles}\n")
```

# ★ NIVELL 3

## → exercici 1

- Comptador i endreçador de paraules d'un text:

El client va quedar content amb el comptador de paraules, però ara vol llegir arxius TXT i que calculi la freqüència de cada paraula ordenades dins de les entrades habituals del diccionari segons la lletra amb la qual comencen, és a dir, les claus han d'anar de la A a la Z i dins de la A hem d'anar de la A la Z. Per exemple, per a l'arxiu "tu\_me\_quieres\_blanca.txt" la sortida esperada seria:

```
# Creo una funcion para contar la frecuencia de las palabras desde un file
def cuenta_palabras(file):

    # Inicializo un diccionario para contar la frecuencia de cada palabra
    frecuencia = {}

    # Intento abrir el file en modo lectura con codificación UTF-8
    try:
        with open(file, 'r', encoding='utf-8') as file:

            # Leo el file y lo convierto en minúsculas y elimino la puntuación encontrada
            palabras = file.read().lower().translate(str.maketrans(" ", " ", '.,!?:;(){}'))

            # Convierto el texto en una lista con los elementos divididos por el espacio
            palabras = palabras.split()

            # Empiezo un ciclo for para controlar cada palabra de la lista palabras
            for word in palabras:

                # Si la palabra ya esta en el diccionario incremento su contador
                if word in frecuencia:
                    frecuencia[word] += 1

                # Si la palabra no esta ya en el diccionario, la añade como llave y incremento su contador
                else:
                    frecuencia[word] = 1

            # Inicializo un diccionario para agrupar y ordenar la palabras alfabeticamente
            diccionario_ordenado = {}

            # Empiezo un ciclo for para controlar las palabras ordeadas alfabeticamente con sorted
            for word in sorted(frecuencia.keys()):

                # Inicializo una variable con la primera letra de la palabra
                primeraletra = word[0]

                # Controlo si la primera letra de la palabra no esta en el diccionario ordenado y la agrego
                if primeraletra not in diccionario_ordenado:
                    diccionario_ordenado[primeraletra] = {}

                # Agrego la palabra y su frecuencia al diccionario ordenado por la letra correspondiente
                diccionario_ordenado[primeraletra][word] = frecuencia[word]

            # Muestro el resultado con dos ciclos for, primero para la primera letra y luego para las palabras correspondientes
            for letra, palabras in diccionario_ordenado.items():
                print(f"\nPalabras con '{letra}':")
                for palabra, count in palabras.items():
```

```
print(f'{palabra}: {count}')
```

```
# Si el file no se encuentra muestro un mensaje de error
```

```
except FileNotFoundError:
```

```
    print("File no encontrado")
```

```
# Elijo el file por leer
```

```
file = 'tu_me_quieres_blanca.txt'
```

```
# Llamo la funcion
```

```
cuenta_palabras(file)
```