Presentació, pantallazos Sprint 7

EXERCICI 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. Pots obtenir més informació del seu càlcul en:

In [6]:
```
# calculadora de IMC _ Index de Massa Corporal, Sprint 7 , exercici 1
def calcular_imc():
    """
    Funció que calcula l'Índex de Massa Corporal (IMC) basat en el pes i l'alçada ingressats per l'usuari/ària.
    Després, classifica l'IMC en les seves respectives categories.
    """
    print('Benvingut/da a la calculadora de l\'índex de massa corporal (IMC). Per calcular-ho, haurà d\'introduir el seu pes i a

    pes = input('Primer, introdueixi el seu pes en kg (ej. 60.5): ')
    alçada = input('Ara, introdueixi la seva alçada en m (ej. 1.80): ')

    try:
        pes = float(pes)
        alçada = float(alçada)

        if pes <= 0 or pes > 550:
            raise ValueError
        if alçada <= 0 or alçada > 3:
            raise ValueError

        print(f'Dades introduïdes: {pes} kg i {alçada} m.')

        # Calcular l'IMC utilitzant la fórmula: pes / alçada^2
        imc = pes / (alçada ** 2)

        # Classificar el resultat de l'IMC en les categories respectives
        if imc < 18.5:
            categoria = "Pesa per sota de la normalitat"
        elif imc < 25:
            categoria = "Pes adient"
        elif imc < 30:
            categoria = "Sobrepès"
        else:
            categoria = "Obès"

        # Mostrar el resultat de l'IMC i la categoria corresponent
        print(f'El seu IMC és: {imc:.1f} i pertany a la categoria de {categoria}.')
        print('Gràcies per utilitzar la nostra calculadora de IMC. Si desitja realitzar un altre càlcul, si us plau executi de n

    except ValueError:
        print(f'''Ha introduït {pes} kg i {alçada} m.
Aquestes dades no compleixen amb el format permès. Tingueu en compte que:
    · el pes i l'alçada han de ser valors numèrics
    · els decimals han d'escriure's amb . (ej. 60.5 kg)
    · el rang de pes acceptat és: entre 0 i 550 kg
    · el rang d'alçada acceptada és: entre 0 i 3 m
Si us plau, executi de nou el programa per realitzar el càlcul de l'IMC.''')

# Cridar la funció per executar el càlcul de l'IMC
calcular_imc()
```

```
Benvingut/da a la calculadora de l'índex de massa corporal (IMC). Per calcular-ho, haurà d'introduir el seu pes i alçada.
Primer, introdueixi el seu pes en kg (ej. 60.5): 63
Ara, introdueixi la seva alçada en m (ej. 1.80): 1.73
Dades introduïdes: 63.0 kg i 1.73 m.
El seu IMC és: 21.0 i pertany a la categoria de Pes adient.
Gràcies per utilitzar la nostra calculadora de IMC. Si desitja realitzar un altre càlcul, si us plau executi de nou el program
a.
```

Explicación del Código: Calculadora de IMC

A continuación, te presento una explicación detallada del código que he programado para calcular el Índice de Masa Corporal (IMC). Este ejercicio corresponde al Sprint 7, ejercicio 1. Te explicaré cada parte del código en primera persona, como si te lo estuviera presentando directamente.

Objetivo del Código

El objetivo de este código es calcular el Índice de Masa Corporal (IMC) de una persona basado en su peso y altura, y luego clasificar el resultado en una categoría específica (bajo peso, peso normal, sobrepeso, obesidad). El programa solicita al usuario que ingrese su peso y altura, realiza el cálculo del IMC y muestra la categoría correspondiente.

Estructura y Funcionamiento del Código

Definición de la Función calcular_imc

Comencé definiendo una función llamada calcular_imc. Esta función es la encargada de todo el proceso de cálculo del IMC y de la interacción con el usuario. Incluí una descripción breve dentro de la función (docstring) para explicar su propósito.

Mensajes de Bienvenida e Instrucciones

Inicié el programa con un mensaje de bienvenida que explica al usuario qué es la calculadora de IMC y qué datos necesitará proporcionar (peso y altura). Este mensaje es importante para orientar al usuario sobre cómo interactuar con el programa.

Solicitar Datos al Usuario

Utilicé la función input para solicitar al usuario que ingrese su peso en kilogramos y su altura en metros. Los datos ingresados se almacenan en las variables pes y alçada, respectivamente.

Conversión y Validación de Datos

Luego, intenté convertir los valores ingresados a números de punto flotante (float). Esta conversión es necesaria para poder realizar cálculos matemáticos con los datos. También implementé una validación para asegurarme de que los valores ingresados están dentro de un rango lógico (peso entre 0 y 550 kg, altura entre 0 y 3 m). Si los valores no cumplen con estos criterios, se lanza un ValueError.

Confirmación de Datos

Mostré un mensaje confirmando los datos ingresados para que el usuario pudiera verificar que los había ingresado correctamente. Esto ayuda a evitar errores en el cálculo debido a entradas incorrectas.

Cálculo del IMC

Realicé el cálculo del IMC utilizando la fórmula estándar: peso dividido por la altura al cuadrado. Este cálculo se almacena en una variable llamada imc.

Clasificación del IMC

Basado en el valor del IMC calculado, clasifiqué el resultado en una de las siguientes categorías: bajo peso, peso normal, sobrepeso, obesidad. Utilicé una serie de condiciones if-elif-else para determinar la categoría correspondiente.

Mostrar Resultados

Finalmente, mostré el resultado del IMC y la categoría correspondiente al usuario. También incluí un mensaje de agradecimiento por usar la calculadora y una invitación a ejecutar el programa nuevamente si deseaba realizar otro cálculo.

Manejo de Errores

Para manejar posibles errores en la entrada de datos, implementé un bloque try-except. Si los datos ingresados no son válidos (por ejemplo, no son numéricos o están fuera del rango permitido), el programa captura el ValueError y muestra un mensaje de error detallado. Este mensaje explica qué tipo de datos son aceptables y cómo deben ser ingresados.

Ejecución de la Función

Finalmente, llamé a la función calcular_imc para que el programa se ejecute cuando se ejecuta el script. Esto asegura que el cálculo del IMC se realice automáticamente cuando el usuario corre el programa.

Ejecución del Programa

Al ejecutar el programa, interactúo con el usuario pidiéndole que ingrese su peso y altura. Luego, el programa calcula el IMC, lo clasifica en la categoría correspondiente y muestra el resultado. Si se ingresan datos no válidos, se muestra un mensaje de error con instrucciones para corregirlo.

Por ejemplo, si el usuario ingresa un peso de 63 kg y una altura de 1.73 m, el programa calculará un IMC de 21.0 y lo clasificará en la categoría de "Pes adient".

Conclusión

Este programa es una herramienta simple pero efectiva para calcular y clasificar el IMC de un usuario basado en su peso y altura. A través de la validación de entradas y el manejo de errores, aseguro que los datos ingresados sean correctos y proporciono información útil y precisa sobre la salud del usuario.

EXERCICI 2

- 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 i construeix el convertidor.

In [10]:
```
# Conversor de temperatures, Sprint 7, exercici 2
def convertir_temperatura():
    """
    Funció que converteix temperatures entre Celsius i Fahrenheit.
    L'usuari/ària haurà d'introduir la temperatura i seleccionar la unitat inicial i la unitat desitjada.
    """
    print('Benvingut/da al convertidor de temperatures. Aquest programa permet convertir temperatures entre Celsius (°C) i Fahre

    # Sol·licitar a l'usuari/ària que introdueixi la temperatura a convertir
    temperatura = input('Introduïu la temperatura que voleu convertir (per exemple, 25): ')

    # Sol·licitar a l'usuari/ària que introdueixi la unitat inicial
    unitat_inicial = input('Introduïu la unitat de la temperatura que heu introduït (C per Celsius, F per Fahrenheit): ').upper(

    # Sol·licitar a l'usuari/ària que introdueixi la unitat desitjada
    unitat_final = input('Introduïu la unitat a la qual voleu convertir la temperatura (C per Celsius, F per Fahrenheit): ').upp

    try:
        temperatura = float(temperatura)

        if unitat_inicial not in ['C', 'F'] or unitat_final not in ['C', 'F']:
            raise ValueError("Les unitats han de ser 'C' per Celsius o 'F' per Fahrenheit.")

        if unitat_inicial == 'C' and unitat_final == 'F':
            temperatura_convertida = (temperatura * 9/5) + 32
        elif unitat_inicial == 'F' and unitat_final == 'C':
            temperatura_convertida = (temperatura - 32) * 5/9
        else:
            temperatura_convertida = temperatura

        print(f'La temperatura de {temperatura:.1f}°{unitat_inicial} és igual a {temperatura_convertida:.1f}°{unitat_final}.')

    except ValueError:
        print(f'Error: la temperatura introduïda o les unitats no són vàlides. Assegureu-vos d\'introduir una temperatura numèri

# Cridar la funció per executar el convertidor de temperatures
convertir_temperatura()
```

```
Benvingut/da al convertidor de temperatures. Aquest programa permet convertir temperatures entre Celsius (°C) i Fahrenheit (°
F).
Introduïu la temperatura que voleu convertir (per exemple, 25): 120
Introduïu la unitat de la temperatura que heu introduït (C per Celsius, F per Fahrenheit): F
Introduïu la unitat a la qual voleu convertir la temperatura (C per Celsius, F per Fahrenheit): C
La temperatura de 120.0°F és igual a 48.9°C.
```

Explicación del Código: Conversor de Temperaturas

A continuación, te presento una explicación detallada del código que he programado para convertir temperaturas entre Celsius y Fahrenheit. Este ejercicio corresponde al Sprint 7, ejercicio 2. Te explicaré cada parte del código en primera persona, como si te lo estuviera presentando directamente.

Objetivo del Código

El objetivo de este código es convertir temperaturas entre las unidades de Celsius (°C) y Fahrenheit (°F). El programa solicita al usuario que ingrese la temperatura, la unidad inicial y la unidad deseada, realiza la conversión correspondiente y muestra el resultado.

Estructura y Funcionamiento del Código

Definición de la Función convertir_temperatura

Comencé definiendo una función llamada convertir_temperatura. Esta función es la encargada de todo el proceso de conversión de temperaturas y de la interacción con el usuario. Incluí una descripción breve dentro de la función (docstring) para explicar su propósito.

Mensajes de Bienvenida e Instrucciones

Inicié el programa con un mensaje de bienvenida que explica al usuario qué hace el convertidor de temperaturas y qué datos necesitará proporcionar (temperatura, unidad inicial y unidad deseada). Este mensaje es importante para orientar al usuario sobre cómo interactuar con el programa.

Solicitar Datos al Usuario

Utilicé la función input para solicitar al usuario que ingrese la temperatura que desea convertir. Este dato se almacena en la variable temperatura.

Luego, pedí al usuario que ingrese la unidad de la temperatura ingresada (C para Celsius, F para Fahrenheit). La entrada del usuario se almacena en la variable unitat_inicial y se convierte a mayúsculas para facilitar la comparación.

Finalmente, solicité al usuario que ingrese la unidad a la que desea convertir la temperatura (C para Celsius, F para Fahrenheit). Este dato se almacena en la variable unitat_final, también convertido a mayúsculas.

Conversión y Validación de Datos

Luego, intenté convertir el valor ingresado para la temperatura a un número de punto flotante (float). Esta conversión es necesaria para poder realizar cálculos matemáticos con los datos.

También implementé una validación para asegurarme de que las unidades ingresadas sean válidas (es decir, 'C' o 'F'). Si las unidades no son correctas, se lanza un ValueError con un mensaje específico.

Realización de la Conversión

Dependiendo de las unidades ingresadas, el programa realiza la conversión correspondiente:

Si la unidad inicial es Celsius y la unidad final es Fahrenheit, utilizo la fórmula: (temperatura * 9/5) + 32.

Si la unidad inicial es Fahrenheit y la unidad final es Celsius, utilizo la fórmula: (temperatura - 32) * 5/9.

Si las unidades inicial y final son las mismas, la temperatura convertida es igual a la temperatura ingresada.

El resultado de la conversión se almacena en la variable temperatura_convertida.

Mostrar Resultados

Finalmente, mostré el resultado de la conversión al usuario, indicando la temperatura original y la temperatura convertida con sus respectivas unidades.

Manejo de Errores

Para manejar posibles errores en la entrada de datos, implementé un bloque try-except. Si los datos ingresados no son válidos (por ejemplo, no son numéricos o las unidades son incorrectas), el programa captura el ValueError y muestra un mensaje de error detallado. Este mensaje explica qué tipo de datos son aceptables y cómo deben ser ingresados.

Ejecución de la Función

Finalmente, llamé a la función convertir_temperatura para que el programa se ejecute cuando se ejecuta el script. Esto asegura que la conversión de temperaturas se realice automáticamente cuando el usuario corre el programa.

Ejecución del Programa

Al ejecutar el programa, interactúo con el usuario pidiéndole que ingrese la temperatura y las unidades de conversión. Luego, el programa realiza la conversión y muestra el resultado. Si se ingresan datos no válidos, se muestra un mensaje de error con instrucciones para corregirlo.

Por ejemplo, si el usuario ingresa una temperatura de 120°F y desea convertirla a Celsius, el programa calculará que 120°F es igual a 48.9°C y mostrará este resultado.

Conclusión

Este programa es una herramienta simple pero efectiva para convertir temperaturas entre Celsius y Fahrenheit. A través de la validación de entradas y el manejo de errores, aseguro que los datos ingresados sean correctos y proporciono información útil y precisa al usuario.

EXERCICI 3

- Exercici 3

Comptador de paraules d'un text.

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

In [22]:
```python
# Comptador de paraules, Sprint 7, exercici 3
def comptador_de_paraules(text):
    """
    Funció que, donat un text, mostra les vegades que apareix cada paraula.
    """
    # Convertir el text a minúscules per assegurar que el comptador no sigui sensible a majúscules/minúscules
    text = text.lower()

    # Eliminar la puntuació del text per comptar només paraules
    import string
    text = text.translate(str.maketrans('', '', string.punctuation))

    # Dividir el text en paraules
    paraules = text.split()

    # Utilitzar un diccionari per comptar les vegades que apareix cada paraula
    comptador = {}

    for paraula in paraules:
        if paraula in comptador:
            comptador[paraula] += 1
        else:
            comptador[paraula] = 1

    # Mostrar el resultat
    for paraula, vegades in comptador.items():
        print(f"'{paraula}' apareix {vegades} vegades")

# Exemple d'ús de la funció
text = input('Si us plau, introdueixi el texte: ')
#text = "Aquest és un exemple. It academy Barcelona."
comptador_de_paraules(text)
```

```
Si us plau, introdueixi el texte: Ferran i Lucia. Ferran és alumne i Lucia és profe
'ferran' apareix 2 vegades
'i' apareix 2 vegades
'lucia' apareix 2 vegades
'és' apareix 2 vegades
'alumne' apareix 1 vegades
'profe' apareix 1 vegades
```

Explicación del Código: Contador de Palabras

A continuación, te presento una explicación detallada del código que he programado para contar las palabras en un texto y mostrar cuántas veces aparece cada una. Este ejercicio corresponde al Sprint 7, ejercicio 3. Te explicaré cada parte del código en primera persona, como si te lo estuviera presentando directamente.

Objetivo del Código

El objetivo de este código es contar cuántas veces aparece cada palabra en un texto dado por el usuario. El programa convierte el texto a minúsculas, elimina la puntuación, divide el texto en palabras y utiliza un diccionario para contar las ocurrencias de cada palabra. Finalmente, muestra el resultado al usuario.

Estructura y Funcionamiento del Código

Definición de la Función comptador_de_paraules

Comencé definiendo una función llamada comptador_de_paraules. Esta función toma un texto como argumento y cuenta las veces que aparece cada palabra en el texto. Incluí una descripción breve dentro de la función (docstring) para explicar su propósito.

Convertir el Texto a Minúsculas

Para asegurarme de que el conteo de palabras no sea sensible a mayúsculas y minúsculas, convertí todo el texto a minúsculas utilizando el método lower(). Esto ayuda a tratar palabras como "Ferran" y "ferran" como la misma palabra.

Eliminar la Puntuación del Texto

Para contar solo palabras y no signos de puntuación, eliminé toda la puntuación del texto. Utilicé el módulo string y el método translate() para eliminar caracteres de puntuación como comas, puntos, signos de exclamación, etc.

Dividir el Texto en Palabras

Luego, dividí el texto en palabras utilizando el método split(). Esto crea una lista de palabras que se pueden procesar individualmente.

Utilizar un Diccionario para Contar las Palabras

Utilicé un diccionario para llevar el conteo de las palabras. Recorro cada palabra en la lista de palabras y actualizo el diccionario:

Si la palabra ya está en el diccionario, incremento su conteo en uno.

Si la palabra no está en el diccionario, la añado con un conteo inicial de uno.

Mostrar el Resultado

Finalmente, recorrí el diccionario y mostré el resultado al usuario. Para cada palabra y su conteo correspondiente, imprimí un mensaje indicando cuántas veces aparece la palabra en el texto.

Ejemplo de Uso de la Función

Para probar la función, pedí al usuario que ingrese un texto mediante la función input(). Luego, llamé a la función comptador_de_paraules con el texto ingresado.

Ejecución del Programa

Al ejecutar el programa, interactúo con el usuario pidiéndole que ingrese un texto. Luego, el programa cuenta las palabras en el texto y muestra cuántas veces aparece cada una. Por ejemplo, si el usuario ingresa el texto "Ferran i Lucia. Ferran és alumne i Lucia és profe", el programa mostrará que las palabras "Ferran", "i", "Lucia" y "és" aparecen dos veces cada una, mientras que "alumne" y "profe" aparecen una vez

.

Conclusión

Este programa es una herramienta simple pero efectiva para contar las palabras en un texto y mostrar cuántas veces aparece cada una. A través de la conversión a minúsculas y la eliminación de la puntuación, aseguro que el conteo de palabras sea preciso y uniforme.

EXERCICI 4

- 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'advertiment.

```
In [20]: # Capgirar un diccionari, Sprint 7, exercici 4
def reverse_dictionary(d):
    """
    Funció que inverteix les claus i els valors d'un diccionari.
    Si es troben valors duplicats, mostra un missatge d'advertiment.
    """
    reversed_dict = {}
    for key, value in d.items():
        if value in reversed_dict:
            print("Error: múltiples claus per un valor.")
            return None
        reversed_dict[value] = key
    return reversed_dict

# Exemple d'ús de la funció
print(reverse_dictionary({'a': 1, 'b': 2, 'c': 3}))  # {1: 'a', 2: 'b', 3: 'c'}
print(reverse_dictionary({'x': 'apple', 'y': 'banana', 'z': 'banana'}))  # Error: múltiples claus per un valor.

{1: 'a', 2: 'b', 3: 'c'}
Error: múltiples claus per un valor.
None
```

Explicación del Código: Invertir un Diccionario

A continuación, te presento una explicación detallada del código que he programado para invertir las claves y los valores de un diccionario. Este ejercicio corresponde al Sprint 7, ejercicio 4. Te explicaré cada parte del código en primera persona, como si te lo estuviera presentando directamente.

Objetivo del Código

El objetivo de este código es invertir las claves y los valores de un diccionario dado. Es decir, las claves se convierten en valores y los valores en claves. Si se encuentran valores duplicados en el diccionario original, el programa muestra un mensaje de advertencia y no realiza la inversión.

Estructura y Funcionamiento del Código

Definición de la Función reverse_dictionary

Comencé definiendo una función llamada reverse_dictionary. Esta función toma un diccionario como argumento y devuelve un nuevo diccionario con las claves y los valores invertidos. Incluí una descripción breve dentro de la función (docstring) para explicar su propósito.

Inicialización del Diccionario Invertido

Dentro de la función, inicialicé un nuevo diccionario vacío llamado reversed_dict. Este diccionario almacenará las claves y los valores invertidos.

Recorrido del Diccionario Original

Recorrí cada par clave-valor en el diccionario original utilizando un bucle for. Para cada par, verifiqué si el valor ya existe como clave en el diccionario invertido.

Verificación de Valores Duplicados

Si el valor ya existe como clave en el diccionario invertido, el programa muestra un mensaje de error indicando que hay múltiples claves para un mismo valor y retorna None. Esto asegura que cada valor en el diccionario original sea único, lo que es necesario para invertir correctamente el diccionario.

Inversión de Claves y Valores

Si no hay duplicados, añadí el par valor-clave al diccionario invertido. De esta manera, las claves originales se convierten en valores y los valores originales en claves.

Retorno del Diccionario Invertido

Una vez que se ha recorrido todo el diccionario original, la función retorna el diccionario invertido.

Ejemplo de Uso de la Función

Para demostrar el funcionamiento de la función, incluí ejemplos de uso en los que se llama a la función reverse_dictionary con diferentes diccionarios.

Ejecución del Programa

Al ejecutar el programa, la función reverse_dictionary procesa los diccionarios de ejemplo. En el primer ejemplo, el diccionario {'a': 1, 'b': 2, 'c': 3} se invierte correctamente, resultando en {1: 'a', 2: 'b', 3: 'c'}. En el segundo ejemplo, el diccionario {'x': 'apple', 'y': 'banana', 'z': 'banana'} tiene valores duplicados, por lo que el programa muestra un mensaje de error y retorna None.

Conclusión

Este programa es una herramienta simple pero efectiva para invertir las claves y los valores de un diccionario. A través de la verificación de duplicados, aseguro que el diccionario invertido sea válido y no tenga conflictos de claves.