

# Tasca S7.01. Estructures de dades i de control

Natalia Durán Gadea

16 de març de 2025

- **Descripció:**

Resoldràs alguns problemes de la vida quotidiana aplicant les estructures de dades i control en Python.

Un client de l'empresa en la qual treballes demana una llista de programes molt senzills, però que li facilitarien molts processos. No obstant això, el departament de TU està molt complicat amb el temps, per la qual cosa et demanen que facis la programació.

- **Objectius:**

- Comprendre i familiaritzar-se amb les estructures bàsiques de dades i control en Python.

- **Durada:** 3 dies.

- **Lliurament:** Emmagatzema en un repositori del teu GitHub una carpeta amb els documents d'aquest exercici:

- L'arxiu Jupyter notebook (.ipynb) que contingui cada enunciat acompanyat pels codis de les respostes i la seva respectiva execució.

## 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.
- La funció ha de classificar el resultat en les seves respectives categories

```
[47]: def calcular_imc():
    peso = float(input("Introduce tu peso en kg: ").replace(",","."))
    altura = float(input("Introduce tu altura en cm: ").replace(",",".")) / 100 # Convertir cm a m

    if peso <= 0 or altura <= 0:
        print("Error: Peso y altura deben ser valores positivos.")
        return

    imc = round(peso / (altura ** 2), 2)
    categoria = ("PESO BAJO" if imc < 18.5 else
                "PESO NORMAL" if imc < 25 else
                "SOBREPESO" if imc < 30 else
                "OBESIDAD")

    print(f"Tu IMC es {imc} kg/m² y estás en la categoría: {categoria}")

calcular_imc()

Introduce tu peso en kg: 55,2
Introduce tu altura en cm: 162,3
Tu IMC es 20.96 kg/m² y estás en la categoría: PESO NORMAL
```

Figura 1: IMC

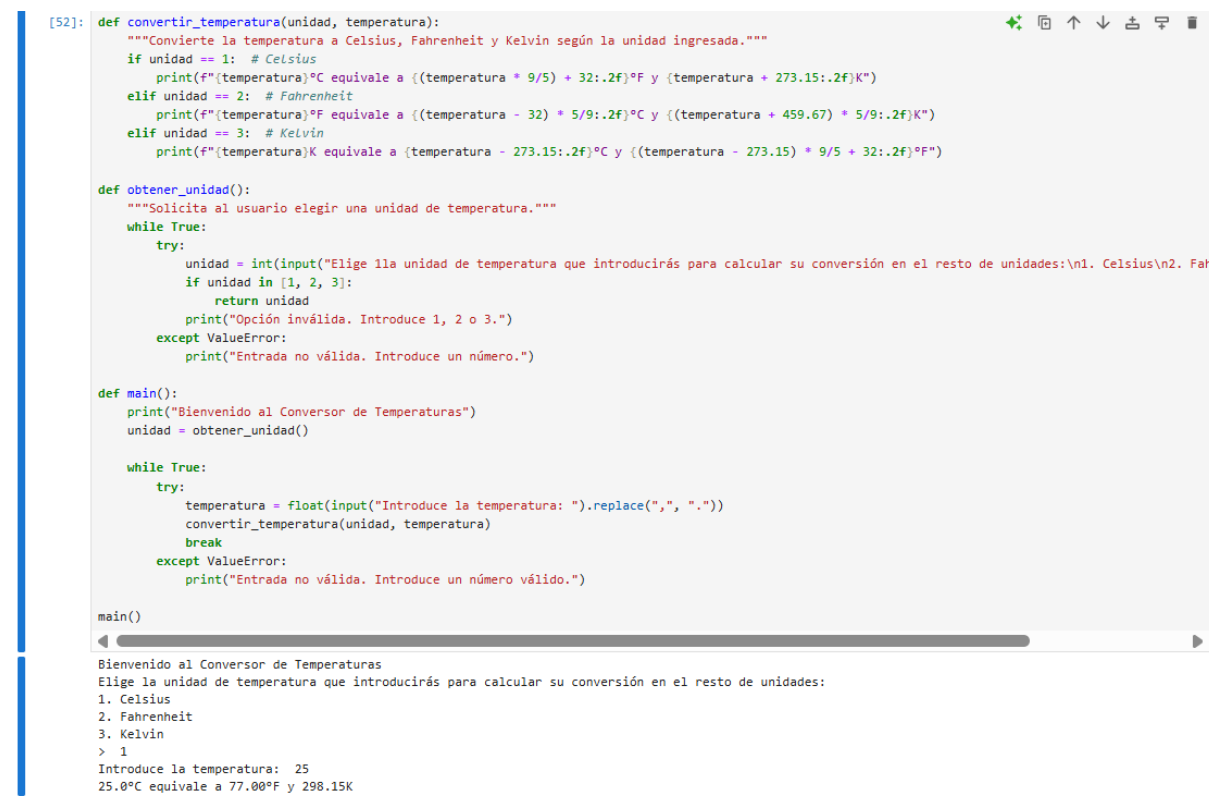
- Aquest codi calcula l'IMC de l'usuari a partir del pes i l'altura ingressats i classifica el resultat en diferents categories.
- **Funcionament del codi:**

- Es demana a l'usuari que introdueixi el seu pes (kg) i altura (m).
- Es calcula l'IMC amb la fórmula:

$$IMC = \frac{pes}{altura^2}$$

- Es classifica el resultat en una de les següents categories:
  - \* Pes insuficient si  $IMC < 18.5$
  - \* Pes normal si  $18.5 \leq IMC < 25$
  - \* Sobrepès si  $25 \leq IMC < 30$
  - \* Obesitat si  $IMC \geq 30$
- Es mostra el resultat per pantalla.

- **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.



```
[52]: def convertir_temperatura(unidad, temperatura):
    """Convierte la temperatura a Celsius, Fahrenheit y Kelvin según la unidad ingresada."""
    if unidad == 1: # Celsius
        print(f"{temperatura}°C equivale a {(temperatura * 9/5) + 32:.2f}°F y {(temperatura + 273.15:.2f)K}")
    elif unidad == 2: # Fahrenheit
        print(f"{temperatura}°F equivale a {(temperatura - 32) * 5/9:.2f}°C y {(temperatura + 459.67) * 5/9:.2f}K")
    elif unidad == 3: # Kelvin
        print(f"{temperatura}K equivale a {(temperatura - 273.15:.2f)°C y {(temperatura - 273.15) * 9/5 + 32:.2f}°F")

    def obtener_unidad():
        """Solicita al usuario elegir una unidad de temperatura."""
        while True:
            try:
                unidad = int(input("Elige 1 la unidad de temperatura que introducirás para calcular su conversión en el resto de unidades:\n1. Celsius\n2. Fahrenheit\n3. Kelvin\n"))
                if unidad in [1, 2, 3]:
                    return unidad
            except ValueError:
                print("Opción inválida. Introduce 1, 2 o 3.")
            except KeyboardInterrupt:
                print("Entrada no válida. Introduce un número.")

    def main():
        print("Bienvenido al Conversor de Temperaturas")
        unidad = obtener_unidad()

        while True:
            try:
                temperatura = float(input("Introduce la temperatura: ").replace(",","."))
                convertir_temperatura(unidad, temperatura)
                break
            except ValueError:
                print("Entrada no válida. Introduce un número válido.")

    main()

Bienvenido al Conversor de Temperaturas
Elige la unidad de temperatura que introducirás para calcular su conversión en el resto de unidades:
1. Celsius
2. Fahrenheit
3. Kelvin
> 1
Introduce la temperatura: 25
25.0°C equivale a 77.00°F y 298.15K
```

Figura 2: Convertidor de temperatures

- L'usuari escull en quina unitat vol introduir la temperatura (Celsius, Fahrenheit o Kelvin).
- Depenent de l'elecció, el programa converteix la temperatura a dues altres unitats.
- Utilitza les fórmules de conversió adequades per cada unitat.

- **Exercici 3:** Comptador de paraules d'un text. Escriu una funció que donat un text, mostri les vegades que apareix cada paraula.

```
[54]: import string

def remove_punctuation(text):
    """Elimina signos de puntuación del texto."""
    return text.translate(str.maketrans('', '', string.punctuation + "¡;"))

def count_words(text):
    """Cuenta la frecuencia de cada palabra en un texto."""
    words = text.lower().split()
    word_count = {}
    for word in words:
        word_count[word] = word_count.get(word, 0) + 1
    return word_count

def main():
    text = input("Introduce un texto: ")
    clean_text = remove_punctuation(text)
    word_counts = count_words(clean_text)

    print("Frecuencia de palabras:")
    for word, count in word_counts.items():
        print(f"{word}: {count}")

main()

Introduce un texto: Buenos días Martha
Frecuencia de palabras:
buenos: 1
días: 1
martha: 1
```

Figura 3: Comptador de paraules

- L'usuari introdueix un text.
- Es treuen signes de puntuació per evitar comptar paraules amb símbols diferents.
- Es converteix el text a minúscules per evitar distincions entre majúscules i minúscules.
- Es divideix el text en paraules i es fa servir un diccionari per comptar la freqüència de cada paraula.
- Es mostra el recompte de paraules.

- **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.

```
reverse_dictionary({'a': 1, 'b': 2, 'c': 3})

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

```
reverse_dictionary({'x': 'apple', 'y': 'banana', 'z': 'banana'})

Error: multiple keys for one value
```

Figura 4: Diccionari invers

```
[59]: original_dict = {
    'nombre': 'Paz',
    'apellido': 'Moreno',
    'apellido2': 'Moreno',
    'edad': 76,
    'país': 'España'
}

def invert_dict(d):
    """Invierte un diccionario intercambiando claves y valores, evitando duplicados."""
    inverted = {}
    for k, v in d.items():
        if v in inverted:
            print(f"⚠️ ATENCIÓN: El valor '{v}' ya existe. No se añadirá nuevamente.")
        else:
            inverted[v] = k
    return inverted

# Mostrar los resultados
inverse_dict = invert_dict(original_dict)
print("Diccionario original:", original_dict)
print("Diccionario invertido:", inverse_dict)

⚠️ ATENCIÓN: El valor 'Moreno' ya existe. No se añadirá nuevamente.
Diccionario original: {'nombre': 'Paz', 'apellido': 'Moreno', 'apellido2': 'Moreno', 'edad': 76, 'país': 'España'}
Diccionario invertido: {'Paz': 'nombre', 'Moreno': 'apellido', 76: 'edad', 'España': 'país'}
```

Figura 5: Diccionari invers

- Es recorre cada parell (clau, valor) del diccionari original.
- Si cada valor és únic, es guarda invertit.
- Si hi ha valors duplicats, el programa mostra un missatge d'avertència.

## 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 imprimíem 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.

```
reverse_dictionary_plus({'x': 'apple', 'y': 'banana', 'z': 'banana'})

{'apple': 'x', 'banana': ['y', 'z']}
```

Figura 6: Diccionari

```
[67]: original_dict = {
    'nombre': 'Paco',
    'apellido': 'Gutierrez',
    'apellido2': 'Gutierrez',
    'apellido 3': 'Gutierrez',
    'edad': 52,
    'país': 'España'
}

def invert_dict(d):
    """Invierte un diccionario, agrupando claves duplicadas en una lista."""
    inverted = {}

    for k, v in d.items():
        if v in inverted:
            # Si ya existe, convierte en lista o agrega a la existente
            inverted[v] = [inverted[v]] if not isinstance(inverted[v], list) else inverted[v]
            inverted[v].append(k)
        else:
            inverted[v] = k # Primera aparición

    return inverted

# Mostrar resultados
inverse_dict = invert_dict(original_dict)
print("Diccionario original:", original_dict)
print("Diccionario invertido:", inverse_dict)

Diccionario original: {'nombre': 'Paco', 'apellido': 'Gutierrez', 'apellido2': 'Gutierrez', 'apellido 3': 'Gutierrez', 'edad': 52, 'país': 'España'}
Diccionario invertido: {'Paco': 'nombre', 'Gutierrez': ['apellido', 'apellido2', 'apellido 3'], 52: 'edad', 'España': 'país'}
```

Figura 7: Diccionari invers amb duplicats

- Si un valor es repeteix, en lloc de mostrar una advertència, es guarda com una llista de claus associades.

- **Exercici 2:** Conversió del tipus de dades. El client rep una llista de dades i necessita generar dues llistes, la primera on estaran tots els elements que es van poder convertir en flotants i l'altra on estan els elements que no es van poder convertir. Exemple de la llista que rep el client: [ '1.3', 'one' , '1e10' , 'seven', '3-1/2', ('2',1,1.4,'not-a-number'), [1,2,'3','3.4']]

```
conversion([ '1.3', 'one' , '1e10' , 'seven', '3-1/2', ('2',1,1.4,'not-a-number'), [1,2,'3','3.4']])

([1.3, 10000000000.0, 2.0, 1.0, 1.4, 1.0, 2.0, 3.0, 3.4],
 ['one', 'seven', '3-1/2', 'not-a-number'])
```

Figura 8: Conversió

```
[70]: original_list = ['1.3', 'one', '1e10', 'seven', '3-1/2', ('2', 1, 1.4, 'not-a-number'), [1, 2, '3', '3.4']]

def filter_floats(lst):
    """Clasifica elementos en flotantes y no flotantes."""
    potential_floats, non_floats = [], []

    for item in lst:
        elements = item if isinstance(item, (list, tuple)) else [item]
        for e in elements:
            try:
                float(e)
                potential_floats.append(e)
            except ValueError:
                non_floats.append(e)

    return potential_floats, non_floats

pot_floats, non_floats = filter_floats(original_list)

print("Lista original:", original_list)
print("Potential floats:", pot_floats)
print("Non-floats:", non_floats)

Lista original: ['1.3', 'one', '1e10', 'seven', '3-1/2', ('2', 1, 1.4, 'not-a-number'), [1, 2, '3', '3.4']]
Potential floats: ['1.3', '1e10', '2', 1, 1.4, 1, 2, '3', '3.4']
Non-floats: ['one', 'seven', '3-1/2', 'not-a-number']
```

Figura 9: Conversor del tipus de dades

- Es recorre una llista d'elements.
- Es comprova si cada element es pot convertir a float.
- Els elements que poden ser floats s'afegeixen a una llista.
- Els altres s'afegeixen a una llista separada.