



Tarea S7.01. Estructuras de datos y de control

Descripción

Resolverás algunos problemas de la vida cotidiana aplicando las estructuras de datos y control en Python.

Un cliente de la empresa en la que trabajas pide una lista de programas muy sencillos, pero que le facilitarían muchos procesos. Sin embargo, el departamento está muy complicado con el tiempo, por lo que te piden que hagas la programación.

Nivel 1

- Ejercicio 1

Calculadora del índice de masa corporal

- Escribe una función que calcule el IMC ingresado por el usuario/a, es decir, quien lo ejecute deberá ingresar estos datos. Puedes obtener más información de su cálculo en:
-> [Índice de masa corporal IMC que es y cómo se calcula.](#)

- La función debe clasificar el resultado en sus respectivas categorías
-

El cálculo para saber el índice de masa corporal es:

El **peso** dividido la **altura** al cuadrado.

Dependiendo de tu índice de masa corporal indica **bajo peso**, **peso normal**, **sobrepeso** o **obesidad**.

```
#Bajo peso: menos de 18,5.  
#Peso normal: 18,5-24,9.  
#Sobrepeso: 25-29,9.  
#Obesidad: más de 30.
```

Explicaré el paso a paso de la función que he creado, que es interactiva con el usuario, que le pide e indica cuales son los valores correctos para ingresar y cuales no, que también tiene una forma de salir si así lo desea. Y que luego en base a lo que ingrese le dará el resultado respectivo a su índice de masa corporal.

PARTE 1: Definir la función, información e interacción con el usuario.

Primero definimos la función que se llamará **calculadora_imc**

```
def calculadora_imc():
```

Una explicacion de que va a ir esta función:

```
'''  
    Calculadora el Indice de Grasa Corporal  
    Para calcular el IMC es necesario dividir el peso en kg de un paciente  
    entre el cuadrado de su altura en metros (peso/(altura**))  
    '''
```

Aquí inicia la interacción con el usuario:

Un saludo con **print()**

```
print("Hola, bienvenido/a a la caaaalculadora de Indice de masa corporal.  
Cual es tu nombre?")
```

Salto de línea en el output con **print("\n")**

```
print("\n")
```

Creando una **variable** pidiéndole el **nombre** para referirse al mismo **input()**.

```
nombre = input("Ingresa tu nombre")
```

Luego procederá a saludar:

```
print(f"Hola!!! {nombre}")
```

Y también explicar una serie de reglas que el usuario tiene que tener en cuenta:

```
print("Recuerda de ingresar la altura en metros.")
print("por ejemplo: forma correcta: 1.75. forma incorrecta: 175")
print("\n")
print("Si no lo has hecho así, vuélvelo a intentar")
```

Si desea salir de la calculadora, como puede hacerlo.

```
print("Si deseas salir, escribe 'salir' en cualquier momento.")
print("\n")
```

PARTE 2: Bucle while, peso y altura.

Como quiero que el usuario ingrese el peso y la altura de determinada manera, porque de lo contrario dará error, o algo que no tenga sentido. Necesite hacer un bucle while que en el caso de que ingresara mal la información pedida, se lo vuelva a pedir. Y también, agregarle la posibilidad de salir al escribir 'salir', tanto cuando se le pide la altura como el peso.

Cuando esto se cumpla:

```
while True:
```

Cree una **variable altura** en la que le va a pedir al usuario que diga su altura. **input()**:

```
altura = input("Bienvenido/a, indica tu altura en metros.")
altura = str(altura.replace(',','.'))
```

La variable **altura** le agrego **.replace(',','.')'** por si la altura que ingresó en vez de que la haya ingresado con punto(.) la ingresa con coma (,) se convierte a punto.

Luego empecé con las condiciones

La primera va a ser la posibilidad de salir escribiendo salir

```
if altura.lower() == "salir":
```

Si la altura(.lower()) es para que por más que el usuario escriba con mayúsculas algunas letras o todas, lo convierte a minúsculas para que coincida con la condición para salir.

Luego un **print()** que le indique lo que ha sucedido
return para que efectivamente salga de la función.

```
print("Has decidido salir de la calculadora.")  
    return
```

También he pensado en las alturas que no “serían validas” como menos de 1 metro o mayores de 3 metros.

```
elif float(altura) > 3.00 or float(altura) < 1.00:  
    print("Esa altura no es valida")
```

para eso tengo que pasar la variable a float, para indicarle ese valor.

Después de haber pensado y haber tomado medidas para los errores.
Cuando el usuario ingrese bien su altura:

```
else:  
    try:  
        altura = float(altura)  
        print(f"Perfecto!! tu altura es {altura}")  
        break  
    except ValueError:  
        print("Debes ingresar la altura en metros. Forma correcta  
1.75. Forma incorrecta = 175")
```

Luego que esa **altura se convierte** aquí en **float()** porque como antes quería que el usuario pueda escribir un **string** para poder salir.

Realizó un print **concatenando** la **altura** que el usuario haya ingresado.

Aquí termina el bucle **while** que se encarga de pedirle la **altura** al usuario

Luego se repite el proceso con el **peso**, en realidad es similar, pero teniendo en cuenta los **errores** que puede tener el usuario al escribir su **peso**, son distintos que al escribir la **altura**.

```
while True:  
    peso = input("Ahora indica tu Peso.")  
    if peso.lower() == "salir":
```

```
print("Has decidido salir de la calculadora.")  
return
```

En este caso se repite **la posibilidad al usuario de salir escribiendo 'salir'**.

Luego he puesto una condición que también puede ser subjetiva, pero más que nada por si el usuario escribe un peso ilógico, para un ser humano, que el programa no lo permita.

```
elif int(peso) > 400:  
    print("Es un peso muy elevado, ¿estas seguro? Intentalo de nuevo")
```

Se repite la conversión del peso a número flotante para que antes el usuario tenga la posibilidad de salir y que si ha ingresado bien la información termine el bucle con un mensaje **print()** que le indique que lo ha hecho bien concatenando su peso.

Y que si se ha equivocado, ingrese de nuevo o pueda salir.

```
else:  
    try:  
        peso = peso.replace(',','.')  
        peso = float(peso)  
        print(f"Genial!! tu peso es {peso}")  
        break  
    except ValueError:  
        print("Introduzca un numero valido o escribe 'salir'")
```

Concluyendo la **PARTE 2**

```
print("\n")  
print(f"Bueno entonces pesas {peso} y mides {altura}")  
print("\n")
```

Se sigue interactuando con el usuario y con saltos para que tanto texto muy junto no sea abrumador para leer.

PARTE 3: El cálculo del Índice de masa corporal, y las condiciones dependiendo a que corresponda ese índice.

Se crea la **variable** que definirá el **cálculo** del índice de masa corporal y se le indica cuanto es al usuario con **print()**.

```
imc = peso/(altura ** 2)  
print(f"Tu IMC es {imc:.2f}")
```

Luego las condiciones:

Si tu índice de masa corporal es menor de 18.5 tienes bajo peso, y se le informa al usuario.

```
if imc < 18.5: #Bajo peso: menos de 18,5.  
    print("Segun mis calculos tu indice de masa corporal indica que  
tienes bajo peso")
```

Luego si tu índice es mayor igual a 18.5 y menor a 24.9 tu peso es normal

```
elif 18.5 <= imc < 24.9: #Peso normal: 18,5-24,9  
    print("Segun mis calculos tu indice de masa corporal indica que tu  
peso es normal")
```

Se repite hasta cumplir con los índices

```
elif 25 <= imc < 29.9: #Sobrepeso: 25-29,9.  
    print("Segun mis calculos tu indice de masa corporal indica que  
tienes Sobrepeso")  
    elif 30 <= imc < 39.9: #Obesidad: más de 30.  
        print("Segun mis calculos tu indice de masa corporal indica que  
puedes padecer de Obesidad")  
    else:  
        print("Segun mis calculos tu indice de masa corporal indica que  
estas por de mas de lo indicado por obesidad")
```

Solamente resta llamar a la función

```
calculadora_imc()
```

Mi resultado ha sido:

```
Hola, bienvenido/a a la caaaalculadora de Indice de masa corporal. Cual es tu nombre?
```

```
Hola!!! Luciano
```

```
Recuerda de ingresar la altura en metros.
```

```
por ejemplo: forma correcta: 1.75. forma incorrecta: 175
```

```
Si no lo has hecho asi, vuelvelo a intentar
```

```
Si deseas salir, escribe 'salir' en cualquier momento.
```

```
Perfecto!! tu altura es 1.75
```

```
Genial!! tu peso es 75.0
```

```
Tu IMC es 24.49
```

```
Segun mis calculos tu indice de masa corporal indica que tu peso es normal
```

- Ejercicio 2

Conversor de temperaturas.

Existen diversas unidades de temperatura utilizadas en distintos contextos y regiones.

Las más comunes son Celsius (°C), Fahrenheit (°F) y Kelvin (K).

También existen otras unidades como Rankine (°Ra) y Réaumur (°Re).

Selecciona al menos 2 conversores, de modo que al introducir una temperatura devuelva, como mínimo, dos conversiones.

Primero he realizado una investigación y luego de aprender sobre las unidades de temperatura he entendido las conversiones y de cómo se calcula para pasar de una a otra.

Aquí también he realizado como un programa interactivo con el usuario, para que elija alguna unidad de temperatura e ingrese una temperatura y vea lo que ha ingresado como sería en las otras unidades de temperatura.

PARTE 1: Definir función, explicación, bienvenida.

Primero defino la función como **conversor_temperaturas**

```
def conversor_temperaturas():
```

Una explicación de qué va la función.

```
'''
    Conversor de temperaturas
    Puedes escoger alguno de los tipos de unidades de temperatura
    Introducir el valor, y te dira las conversiones de las demas.
'''
```

Luego se **interactúa** con el usuario con una serie de **print()** explicativos para que entienda que es lo que tiene y puede hacer, como en el ejercicio anterior ofreciéndole la posibilidad de poder **salir**.

```
print("Bienvenido al conversor de temperaturas!")
print("Puedes convertir entre Celcius (C), Fahrenheit (F) y Kelvin (K)")
print("Deberas elegir alguno de ellos")
print("Si deseas salir, solo escribe 'salir'")
print("\n")
```

PARTE 2: Bucle while, pedir unidad de temperatura y valor en grados, errores.

Como necesito que el usuario solamente elija una **unidad de temperatura** y de la manera en la que yo quiero que la elija, y como también quiero que pueda salir si así lo desea voy a hacer un bucle **while**. En la que se determinen las siguientes condiciones. Es parecido a la estructura del ejercicio 1 porque también requiere que el usuario ingrese 2 valores de una determinada manera.

En el primer bucle se le pide la **unidad de temperatura**.

Mientras esto sea **TRUE**:

```
while True:
    elegir_unidad = input("¿Cual es la unidad que deseas convertir?
    Ingresa 'C' para Celsius, 'F' para Fahrenheit o 'K' para
    Kelvin").strip().lower()
    if elegir_unidad.lower() == "salir":
        print("Has decidido salir del conversor de temperatura.")
        return
```

Creo la variable **elegir_unidad** en la que le pido al usuario a través de **input()** la unidad de temperatura que desea convertir.

A su vez en ese **input()** le indico **.strip()** por si el usuario escribe con algún espacio al inicio o al final, esta función se encargará de eliminar esos espacios, y **lower()** para

que lo que haya introducido lo ponga en **minúsculas** porque en el futuro las condiciones elegí que sean así.

Y luego la condición de que si el **usuario** en ese **input()** introdujo **salir** que se salga (**return**) y con un **print()** que le avise que ha salido.

Luego esta es la condición con lo que le permite al usuario pasar a elegir los grados de temperatura. Escribiendo las unidades de temperatura de la siguiente manera Cree una lista para ingresar cada uno de ellos.

```
if elegir_unidad in ['c','celsius','f','fahrenheit','k','kelvin']:
    break
```

Y que si lo ha ingresado bien termine (**break**) el loop, ya obtuvimos lo que queríamos y cómo lo queríamos así que no le va a pedir más. Y en el caso de que quiera salir tiene que escribirlo.

Si no realiza las anteriores condiciones, este mensaje recibirá el usuario para que lo vuelva a intentar.

```
else:
    print("No has ingresado correctamente")
```

Luego pasamos al segundo bucle **while** que en este le pediremos al **usuario** que ingrese el valor, el número de grados para realizar **la conversión**.

Creo una variable que se llamará **ingresar_numero** en la que a través de un **input()** le pediré al usuario que ingrese el valor a la temperatura correspondiente.

Como en el anterior a través de una **condición** le ofrecemos al usuario la posibilidad de **salir**. **Return** para que se termine el programa.

```
while True:
    ingresar_numero = input("Ingresa un numero, el valor de la
temperatura que deseas convertir: ")
    if ingresar_numero.lower() == 'salir':
        print("Has decidido salir del conversor de temperatura.")
        return
```

Luego convertir la variable **ingresar_numero** en un **float()** porque como antes necesitaba que pueda escribir **salir** no podía determinar que esa variable sea un número con decimal. Y **concatenar** la **variable** mostrándole al **usuario** cuál es el número que ha ingresado.

Y que si lo ha hecho bien, **break**, que termine de seguir buscando inputs.

```
try:
    ingresar_numero = float(ingresar_numero)
    print(f"El valor ingresado es:{ingresar_numero}")
```

```
break
```

Si no opera dentro de los parametros asignados, le dara error, se lo indico con un **print()** y lo tendra que volver a intentar o salir.

```
except ValueError:
    print("Ingresaste un numero incorrecto, por favor vuelvelo a ingresar")
    print("\n")
```

PARTE 3: Dependiendo la unidad de temperatura que elija, se convertirá en a las otras con los cálculos que sean necesarios.

De **Celsius** a **Fahrenheit** = $x * (9/5) + 32$

De **Celsius** a **Kelvin** = $x + 273.15$

De **Fahrenheit** a **Celsius** = $(x - 32) * (5/9)$

De **Fahrenheit** a **Kelvin** = $(x - 32) * (5/9) + 273.15$

De **Kelvin** a **Celsius** = $x - 273.15$

De **Kelvin** a **Fahrenheit** = $(x - 273.15) * (9/5) + 32$

Luego de tener en claro cómo se calculan las conversiones según la unidad de temperatura escogida, haré los condicionales.

Si la **variable**, que contiene el **input()**, en el que se solicito al usuario que escoja una unidad de temperatura. Y en este caso si lo que escogió está en esta lista donde se encuentra ['c', 'celsius].

Se le informa al usuario lo que ha escogido con un **print()**

Se hacen las conversiones:

La variable **fahrenheit** va a ser igual a cuando lo pedí al usuario que escoja la temperatura después de que haya escogido la unidad $* (9/5) + 32$, que sería la forma de pasar de **Celsius** a **Fahrenheit**

La variable **Kelvin** es igual excepto que la conversión distinta, en este caso el número que ha ingresado el usuario solamente se le sumará 273.15.

Luego se concateno un **print()** en el que le muestro el numero que ingreso, equivalen a tales grados **Fahrenheit(2.f 2 decimales)** y **Kelvin**.

```
if elegir_unidad in ['c', 'celsius']:
```

```

print("Escogiste Celsius")
fahrenheit = ingresar_numero * (9/5) + 32
kelvin = ingresar_numero + 273.15
print(f"Entonces {ingresar_numero} grados Celsius equivale a
{fahrenheit:.2f} grados Fahrenheit y a {kelvin:.2f} Kelvin")

```

Todo esto sucederá en el caso de que haya escogido la unidad de temperatura Celsius.

Luego se repetirá el mismo proceso utilizando **elif** con sus debidos cálculos con el resto de los grados que el usuario puede elegir.

```

elif elegir_unidad in ['f','fahrenheit']:
    print("Escogiste Fahrenheit")
    celsius = (ingresar_numero - 32) * (5/9)
    kelvin = (ingresar_numero - 32) * (5/9) + 273.15
    print(f"Entonces {ingresar_numero} grados Fahrenheit equivale a
{celsius:.2f} grados Celsius y a {kelvin:.2f} Kelvin")
    elif elegir_unidad in ['k','kelvin']:
        print("Escogiste Kelvin")
        celsius = ingresar_numero - 273.15
        fahrenheit = (ingresar_numero - 273.15) * (9/5) + 32
        print(f"Entonces {ingresar_numero} Kelvin equivale a {celsius:.2f}
grados Celsius y a {fahrenheit:.2f} grados Fahrenheit")

```

Una vez terminado llamamos a la funcion

```

conversor_temperaturas()

```

Esta sería el resultado de una interaccion con la funcion

```

Bienvenido al conversor de temperaturas!
Puedes convertir entre Celcius (C), Fahrenheit (F) y Kelvin (K)
Deberas elegir alguno de ellos
Si deseas salir, solo escribe 'salir'

```

```

El valor ingresado es: 25.0
Escogiste Celsius
Entonces 25.0 grados Celsius equivale a 77.00 grados Fahrenheit y a 298.15
Kelvin

```

- Ejercicio 3

Contador de palabras de un texto.

Escribe una función que, dado un texto, muestre las veces que aparece cada palabra.

Para este ejercicio se me ha ocurrido que los textos podrían ser letras de canciones, para poder identificar cuantas veces se repiten algunas palabras. Es común que haya palabras como el, la, yo, se repitan.

Para eso voy a crear la función **contador_palabras**:

```
def contador_palabras(texto):
```

Utilizó la **variable** texto dentro de la función para indicar que allí se introduce una variable que contenga texto, o texto mismo dentro de allí

Luego una breve explicación de lo que va esta función:

```
'''
    En este contador de palabras vamos a analizar
    cuantas palabras se repiten en
    las siguientes canciones o textos.
'''
```

Dentro de la **variable texto** que (el texto que se vaya a introducir) se convierte a minúsculas para evitar errores.

```
texto = texto.lower()
```

Creo una **variable** que se llame **palabras** donde se voy a alojar las palabras utilizando un bucle **for** porque voy a ejecutar una condición de que vaya contando letra por letra teniendo en cuenta los espacios como espacios (**.isspace()**) y también si la letra tiene signos de puntuación (**.isalpha()**) que se van a ir sumando una x una a la variable **palabras**.

```
for letra in texto:
    if letra.isalpha() or letra.isspace():
        palabras += letra
```

Luego como la variable **palabras** ya recolectó todo en forma de texto para realizar el conteo hay que convertir ese texto en **lista** creo una **variable** llamada **lista_palabras** donde introducir **palabras.split()** utilizando la función **split()** para convertirlo en **lista**.

```
lista_palabras = palabras.split()
```

Después creó un **diccionario** que me permite introducir cada **palabra** como **clave** y a su **valor** como **cantidad de veces que aparece**.

```
conteo = {}
```

Cree un bucle **for** donde por cada **palabra** de la lista **palabras** que se **(if)** encuentre en el **diccionario** se incremente el valor +1. **(else)** Y que por cada **palabra** que no se encuentra se crea **el primer valor**.

```
conteo = {}
for palabra in lista_palabras:
    if palabra in conteo:
        conteo[palabra] += 1
    else:
        conteo[palabra] = 1
```

Hago un **print()** informando lo que sucederá. Y también a través del bucle **for** le indico que por cada palabra me lo ordene alfabéticamente. Utilizando **.sorted()**. También un **print()** concatenando **palabra(del bucle)** me indique el **diccionario de conteo, el valor**.

```
print("Conteo de palabras:")
for palabra in sorted(conteo):
    print(f"{palabra}: {conteo[palabra]}")
```

Luego introduje 2 canciones en variables

Ejemplo 1:

```
# Canciones

crimen_cerati = """La espera me agotó
No sé nada de vos
Dejaste tanto en mí
En llamas me acosté
Y en un lento degradé
```

```
Supe que te perdí"""
```

Llamé a la función con la variable donde puse parte de la letra de una canción.

```
contador_palabras(crimen_cerati)
```

Output:

Conteo de palabras:

acosté: 1

agotó: 1

de: 1

degradé: 1

dejaste: 1

en: 3

espera: 1

la: 1

lento: 1

llamas: 1

me: 2

- Ejercicio 4

Diccionario inverso.

Resulta que el cliente tiene una encuesta muy antigua que se almacena en un diccionario y los resultados los necesita a la inversa, es decir, intercambiados las claves y valores. Los valores y claves en el diccionario original son únicos; si éste no es el caso, la función debería imprimir un mensaje de advertencia.

Para este ejercicio llame la función **invertir_diccionario()**:

```
def invertir_diccionario(diccionario):
```

Donde dentro de la función tendrá que introducir el diccionario.

Con una breve explicación de lo que se trata:

```
"""
    Inverte un diccionario, intercambiando claves y valores.
    Si hay claves o valores repetidos, imprime un mensaje de advertencia.
    """
```

Luego creare las diferentes variables de distintos tipos para almacenar

1. el **diccionario** invertido como un diccionario.
2. una variable donde se alojarán las **claves únicas** del diccionario y cómo son únicos será en un **set()** y otra variable donde suceda lo mismo con los **valores**. Como se precisa que sea, que no existan duplicados.

```
dic_invertido = {}  
dic_claves = set()  
dic_valores = set()
```

Se crea un bucle **for** donde por cada clave y cada valor en el **diccionario.items()** (el que introducimos). **.items()** es para que recorra tanto por su **clave** como por su **valor**. Como no tiene que haber duplicados de ningún tipo es necesario.

Primero para evitar el error, la condición será, si alguna **clave** existe dentro de **dic_claves** o si algún **valor** existe en **dic_valores** que me imprima **el mensaje de advertencia que el diccionario que estamos introduciendo tiene duplicados. Diciéndole cuál es la clave o el valor duplicado.**

```
for clave, valor in diccionario.items():  
    if clave in dic_claves or valor in dic_valores:  
        print(f"Advertencia: clave '{clave}' o valor '{valor}' ya  
existe en el diccionario original.")
```

Luego si no hay duplicados, hacemos que se invierta el **diccionario** agregando las **claves** a sus **sets** utilizando el **.add()** que es para añadir elementos al **set** e **invertimos el diccionario.**

```
else:  
    dic_claves.add(clave)  
    dic_valores.add(valor)  
    dic_invertido[valor] = clave
```

Después de tener ya el diccionario invertido se termina el bucle y **return** resultado de **dic_invertido**

```
return dic_invertido
```

Luego introduciremos los **diccionarios** para hacer la comprobación.

Los introduje en variables para que lo enseñe en el output

```
diccionarios = ({'a': 1, 'b': 2, 'c': 3})
diccionario2 = ({'x': "apple", 'y': "banana", 'z': "banana"})

resultado = invertir_diccionario(diccionarios)
resultado2 = invertir_diccionario(diccionario2)
print(resultado2)
print("\n")
```

Output:

Advertencia: clave 'z' o valor 'banana' ya existe en el diccionario original.

```
{'apple': 'x', 'banana': 'y'}
```

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