**SPRINT 9**

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.

Se’t demanarà que programis varies funcions. Per a fer l’entrega, hauràs d’entregar tant el codi de la funció:

*def funcioexemple(variable1, variable2):*

*resultat = variable1 + variable2 # sumem les dues variables*

*return resultat*

Com la seva execució per demostrar que funciona correctament, ensenyant-ne l’output:

*funcioexemple(3, 8)*

Intenta que tot el codi que generis sigui el més reproduïble possible.

**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.

Pots obtenir més informació del seu càlcul en: 🔗 Índice de masa corporal IMC que es y como se calcula.

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

Consell: Intenta validar les dades prèviament, perquè enviï un missatge d’advertència si les dades introduïdes per l’usuari/a no

estan en el format adequat o no pren valors raonables.

Es defineix la funció **calcular\_imc** amb els valors que l’usuari ha d’introduir, pes (peso) i alçada (altura) que són necessaris per calcular l’Index de Massa Corporal (IMC).

Per tal d’evitar error quan a l’hora d’introduir ambdós valors, s’utilitza ***try*** i ***except*** que permeten controlar si el valor és un número. Si no ho és, retorna un missatge indicant que cal introduir un número, però sense que el programa s’aturi . Aquest control d’errors es fa amb la funció ***ValueError*** ja inclosa a Python.

Desprès, també es comprova que s’introdueixen valors dins d’un rang raonable (el pes entre 35 i 200 kg i l’alçada entre 1.20 i 2.20 m) i en cas que no ho estiguin es retorna un missatge per tal que l’usuari ho pugui corregir.

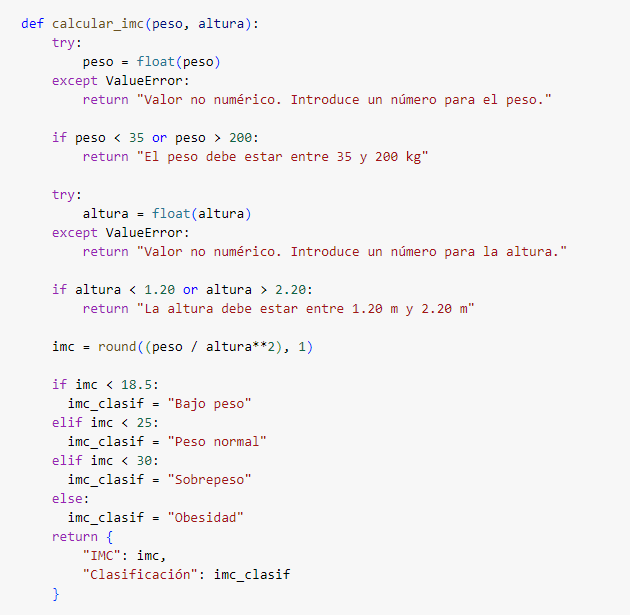
En els casos descrits es fa servir ***return*** per tal que aparegui el missatge d’advertència.

Un cop assegurat que els valors de pes i alçada són vàlids, es calcula l’IMC mitjançant la fórmula i es fa un arrodoniment a un sol decimal amb ***round***.

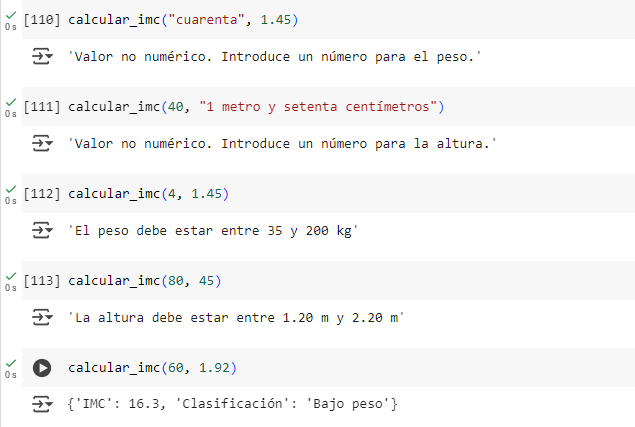
Finalment es fa la classificació segons el valor de l’IMC fent servir ***if***, ***elif*** i ***else*** per veure en quin rang es troba:

* Si és inferior a 18.5 es considera que el pes és baix.
* Entre 18.5 i 25 es considera que el pes és normal.
* Entre 25 i 30 es considera que es té sobrepès.
* A partir de 30 es classifica com a obesitat.

Tot això es retorna en un **diccionari**, que mostra el valor de l’IMC i la classificació corresponent.



A continuació es mostra l’execució de la funció amb diversos valors per tal de mostrar els diferents missatges i resultats que poden aparèixer:



***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, de manera que es puguin guardar (recorda que un print() no es pot guardar mai).

Consell: Intenta validar les dades prèviament, perquè enviï un missatge d’advertència si les dades introduïdes per l’usuari/a no estan en el format adequat.

(EXTRA): Pensa una manera d’emmagatzemar totes les possibles conversions en un sol objecte (Llista? Diccionari? DataFrame?) en comptes d’escriure molts if else en funció de la temperatura d’origen i la temperatura de destí.

Es defineix la funció **convertir\_temperatura** que rep dues entrades, la unitat de temperatura que vol convertir (unitat\_temp) i el valor numèric de la temperatura (valor\_temp).

Primerament, es crea un petit diccionari (limites) que conté les temperatures mínimes que pot assolir cada escala segons la ciència i que corresponen al zero absolut, Cap temperatura pot estar per sota d’aquests valors:

* Celsius: -273.15
* Fahrenheit: -459.67
* Kelvin: 0

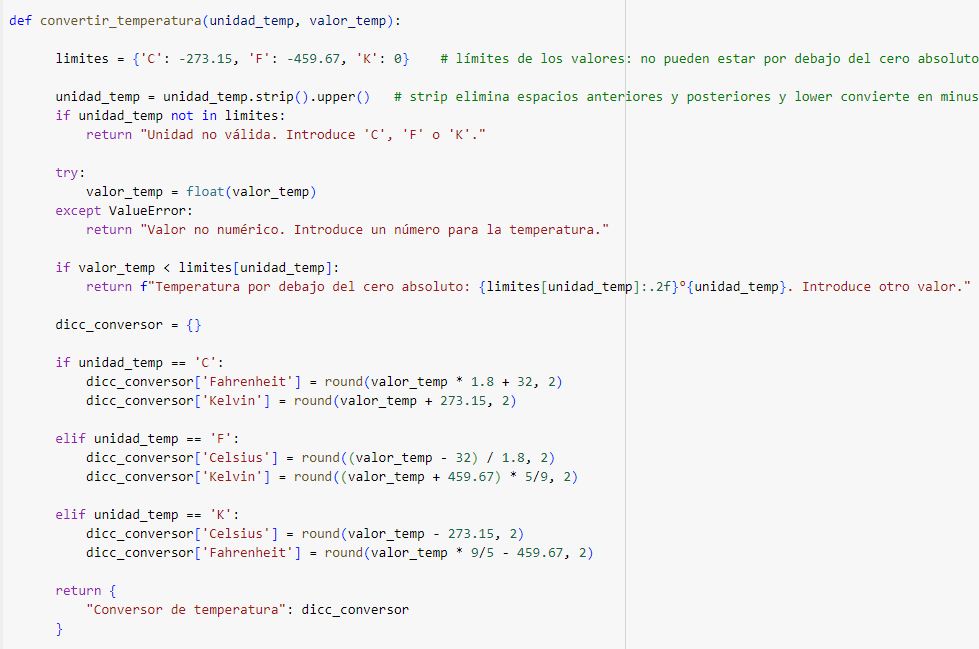
A continuació es recull la unitat (unidad\_temp). Amb la funció ***strip*** s’eliminen els espais sobrants i amb la funció ***upper*** es converteix a majúscula per tal d’assegurar que el programa reconeix la unitat correctament. Desprès es comprova que la unitat sigui vàlida (‘C’ per Celsius, ‘F’ per Fahrenheit i ‘K’ per Kelvin). En cas de que no ho sigui es retorna un missatge d’advertència per tal que l’usuari ho corregeixi.

Desprès es revisa que la temperatura sigui un número, utilitzant ***try*** i ***except*** ***ValueError***. Si no ho és, el programa ho detecta i retorna un missatge d’error sense que per això s’aturi. En aquest punt es comprova que el valor no estigui per sota del zero absolut per la seva escala. Si ho està, es retorna un missatge indicant quin és el valor mínim possible.

Quant la unitat i la temperatura tenen valors vàlids, es crea un diccionari (dicc\_conversor) on es posaran els resultats de les conversions arrodonides a dues xifres decimals amb ***round***:

* Si la unitat és Celsius, es calculen Fahrenheit i Kelvin
* Si és Fahrenheit, es calculen Celsius i Kelvin
* Si és Kelvin, es calculen Celsius i Fahrenheit

Finalment, la funció retorna un **diccionari amb les conversions** que mostrarà les escales i els valors corresponents.



A continuació es mostra l’execució de la funció amb diversos valors per tal de mostrar els diferents missatges i resultats que poden aparèixer:



***Exercici 3. Comptador de paraules d'un text***

Escriu una funció que donat un text, mostri les vegades que apareix cada paraula. Intenta que es gestionin totes les casuístiques possibles que facin que el programa no funcioni correctament.

(EXTRA): Quina és la llargada mitjana de les paraules del text que has escrit? “Hola com va?” hauria de retornar (4+3+2) / 3 = 3

Es defineix la funció **contar\_palabras\_y\_calculo\_media** que rep com a entrada un text qualsevol.

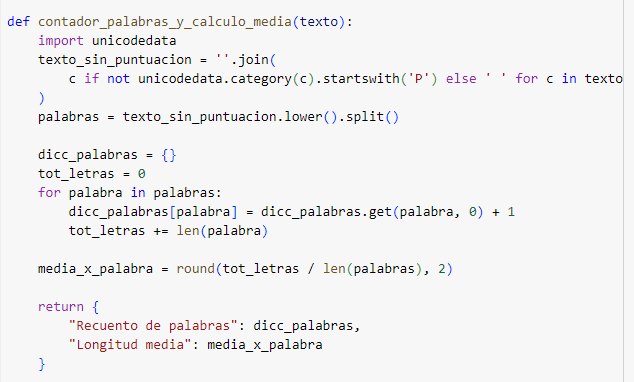
Primer de tot, s’importa el mòdul unicodedata que serveix per treballar amb les propietat dels caràcters, concretament per saber si un caràcter és un signe de puntuació.

Amb ***for c in texto***, que és un bucle, es recorre tot el text. Amb la funció ***startswith*** es comprova si el caràcter pertany a la categoria puntuació i, amb la funció ***category***, cada signe de puntuació es substitueix per un espai, si no ho és, es manté el caràcter original. D’aquesta manera les paraules queden separades i diferenciades unes de les altres. Finalment, amb la funció ***join*** tots els caràcters s’ajunten en una sola cadena sense cap signe de puntuació (texto\_sin\_puntuacion).

Ara, amb ***split*** s’elimina qualsevol espai sobrant i amb ***lower*** es posen totes les lletres en minúscules per evitar que el programa faci distincions amb la mateixa paraula.

A continuació, es crea un diccionari (dicc\_palabras) i amb el bucle ***for*** es recorren totes les paraules i, amb la funció ***get*** la pròpia paraula (palabra) es fa servir com a clau. Si ja existia, se li suma 1 al comptador i, si no, comença des de 1. Al mateix temps, es va sumant la longitud de cada paraula i que servirà posteriorment per calcular la mitjana que es presentarà arrodonida amb dues xifres decimals amb la funció ***round*** per facilitar la lectura.

Finalment, la funció retorna un diccionari que en primer lloc mostra un recompte de paraules (cada paraula amb el nombre de vegades que apareixen), i en segon lloc, mostra la mitjana de lletres per paraula del text.



A continuació es mostra l’execució de la funció amb diversos valors per tal de mostrar els diferents resultats que poden aparèixer:



***Exercici 4. Diccionari invers (amb possibilitat de duplicats)***

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, intercanviant 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, juntament amb una llista amb els valors associats a la clau repetida.

Es defineix la funció **invertir\_diccionario** que rep com a entrada un diccionari qualsevol. Primerament es crea un diccionari (dicc\_invertido) que servirà per intercanviar les claus i els valors, és a dir, els valors del diccionari original passaran a ser les noves claus, i les claus originals passaran a ser els seus valors.

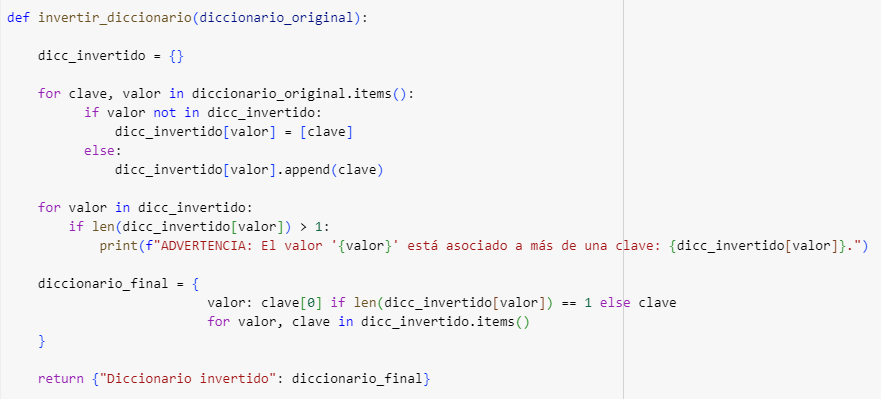
Per fer això, es fa servir un bucle ***for*** que recorre cada parella clau-valor del diccionari original amb la funció ***items***. Si no hi és, es crea una nova entrada amb aquest valor com a clau a dicc\_invertido i s’hi assigna una llista que conté la clau original. Si ja hi és, vol dir que el valor està repetit i s’afegeix la nova clau a la llista ja existent amb la funció ***append***.

Desprès, es comprova amb un altra bucle ***for*** si hi ha algun cas on una mateixa clau ara té més d’un valor associat. Si és així, s’imprimeix un missatge d’advertència per avisar l’usuari.

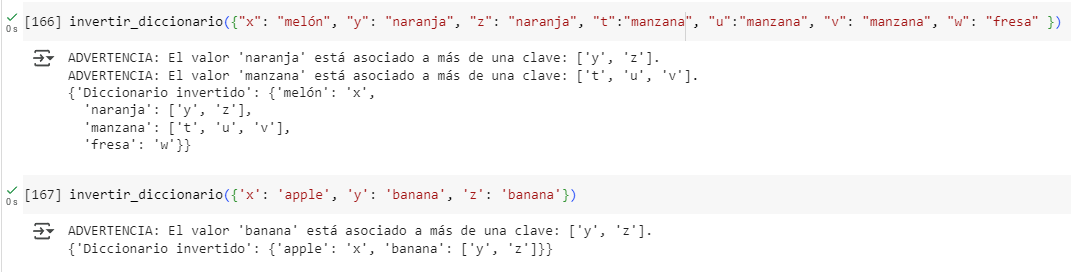
Desprès, es prepara un diccionari final (diccionario\_final) que mostra els valors que es volen mostrar quan s’executi la funció:

* Si una clau del diccionari invertit té un sol element a la llista, és a dir, només hi havia una clau original amb aquell valor, es mostra directament la clau.
* Si té més d’un element a la llista, es manté la llista per mostrar tots els elements.

Finalment, es retorna aquest nou diccionari com a resultat de la funció, dins d’un altre diccionari amb la clau "Diccionario invertido".



A continuació es mostra l’execució de la funció amb l’exemple de dos diccionaris diferents:



**NIVELL 2**

***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 a la Z.