**Ejercicio Nro. 1:**

**Escribir un programa que guarde en una variable el** **diccionario {'Euro':'€', 'Dollar':'$', 'Yen':'¥'}, pregunte al usuario por una divisa y muestre su símbolo o un mensaje de aviso si la divisa no está en el diccionario.**

Para comenzar con este programa, se crea una variable diccionario con las divisas establecidas en la consigna “diccionariodivisas {'Euro':'€', 'Dollar':'$', 'Yen':'¥'}”. Luego se ingresa mediante la función input() el nombre de la divisa y se formatea, utilizando la función [[1]](#footnote-1)“capitalize()”, la cadena ingresada con la primera letra mayúscula y el resto en minúsculas para que coincidan con las claves del diccionario. Para terminar, se verifica que la divisa exista en el diccionario usando la condición “if divisa in diccionarioDivisas”, guardando en una variable string el mensaje a mostrar por pantalla dependiendo si es verdadera o falsa dicha condición.

Tipo De Dato: String

Dato: “dollar”

Resultado obtenido: “El símbolo de la divisa Dollar es: $”

Tipo De Dato: String

Dato: “peso”

Resultado obtenido: “La divisa no se encuentra en el diccionario”

**Ejercicio Nro. 2:**

**Escribir un programa que pregunte al usuario su nombre, edad, dirección y teléfono y lo guarde en un diccionario. Después debe mostrar por pantalla el mensaje <nombre> tiene <edad> años, vive en <dirección> y su número de teléfono es <teléfono>.**

En este programa se declara un diccionario con las claves ‘nombre’, ’edad’, ’dirección’ y ‘teléfono’ con valores a definir por el usuario   
“persona = {"nombre”: "“, "edad”: "“, "dirección”: "“, "teléfono”: ""}”  
Luego, se hace uso de un bucle “for” para cargar los datos ingresados por el usuario. Vale aclarar que, se utiliza una variable string “entrada” en la cual se van a ir guardando los valores para cada clave, es decir, por cada iteración las claves van a ir cambiando. Permitiendo ahorrar en declarar varias variables.  
for campo in persona:

entrada = input(f"Ingrese su {campo}: ")

persona[campo] = entrada

Finalmente, se muestra por pantalla según el formato establecido en la consigna los datos ingresados por el usuario “**el mensaje <nombre> tiene <edad> años, vive en <dirección> y su número de teléfono es <teléfono>.”**

TIPO DE DATOS: STRING

DATOS: Javier; 40; Schmidl 6625; 1133465734

RESULTADO OBTENIDO: “Javier tiene 40 años, vive en Schmidl 6625 y su número de teléfono es 1133465734.”

**Ejercicio Nro. 3**

**Escribir un programa que guarde en un diccionario los precios de las frutas de la tabla, pregunte al usuario por una fruta, un número de kilos y muestre por pantalla el precio de ese número de kilos de fruta. Si la fruta no está en el diccionario debe mostrar un mensaje informando de ello.**

En este programa como primer paso, se crea el diccionario donde se van a alojar las frutas y sus precios dados en la tabla, clave y valor respectivamente. Como segundo paso, se pide al usuario mediante la función input() la fruta que quiere comprar, se formatea la cadena ingresada con la función capitalize() para que coincida con las claves del diccionario. Como ultimo paso, se verifica si la fruta se encuentra en el diccionario “if fruta in preciosFrutas”. De ser verdadera la condición, se pide al usuario que ingrese cuantos kilos va a comprar para poder hacer el calculo final del precio a pagar. De ser falsa la condición, se le notifica por pantalla al usuario que la fruta no está en el diccionario.

TIPOS DE DATOS: STRING, INT

DATOS: Manzana; 3

RESULTADO OBTENIDO: “El precio de Manzana por la cantidad de 3.0 Kg es: 2.4000000000000004 pesos”

TIPO DE DATO: STRING

DATO: uva

RESULTADO OBTENIDO: “La fruta Uva no se encuentra en el diccionario”

**Ejercicio Nro. 4**

**Escribir un programa que pregunte una fecha en formato dd/mm/aaaa y muestre por pantalla la misma fecha en formato dd de mm de aaaa donde mm es el nombre del mes.**

En primer lugar, se declara una tupla “meses” con los nombres de los meses del calendario y un diccionario (“diccionarioFecha”) con las claves ‘dd’, ’mm’, ’aaaa’ y sus valores en cero. En segundo lugar, se pide al usuario que ingrese por pantalla, con la función input(), una fecha con el formato establecido dd/mm/aaaa (longitud de la cadena = 10) y se verifica que cumpla con las siguientes condiciones para tener un formato valido:  
1. Tener caracteres ‘/’

2. Longitud estricta de la cadena, 10 caracteres

Luego, con la función Split() se genera una lista a partir de la cadena ingresada. Donde cada valor de la lista se obtiene gracias a que se agrega como parámetro el carácter ‘/’ a dicha función para que haga la división cuando se encuentra con este mismo carácter en la cadena. O sea, 12/02/1984 🡺 Split(‘/’) 🡺[12 , 02 , 1984]. Una vez realizado esto, se verifican los criterios para que los días, meses y años sean válidos, por si hubo una falla en la primera condición del formato valido. En tercer lugar, una vez corroboradas todas las verificaciones, se declara una variable “i” para iterar en los índices de la nueva lista “fecha”. Se crea un bucle “for” para recorrer el diccionario elemento por elemento y de esta forma guardar en cada clave el valor correspondiente.   
for elemento in diccionarioFecha:

diccionarioFecha[elemento] = fecha[i]

i+=1

En último lugar, se imprime por pantalla según el formato establecido en la consigna la fecha ingresada. Vale aclarar que, se utiliza como índice de la tupla el valor de la clave (se castea a entero) ‘mm’ menos 1 para que en pantalla aparezca el nombre del mes.  
meses[int(diccionarioFecha['mm'])-1] 🡺’mm’ : 02 ; meses[1] = “Febrero” 🡺 meses[int(02)-1] 🡺 meses[1]

TIPO DE DATO: STRING

DATO: “12/04/1984”

RESULTADO OBTENIDO: “12 de Abril de 1984.”

TIPO DE DATO: STRING

DATO: 12041984

RESULTADO OBTENIDO: “Formato de fecha invalido”

TIPO DE DATO: STRING

DATO: 12/04/19844

RESULTADO OBTENIDO: “Formato de fecha invalido”

TIPO DE DATO: STRING

DATO: 12004/1984

RESULTADO OBTENIDO: “No existe el dia 12004 en el calendario”

TIPO DE DATO: STRING

DATO: 32/04/1984

RESULTADO OBTENIDO: “No existe el dia 32 en el calendario”

TIPO DE DATO: STRING

DATO: 00/04/1984

RESULTADO OBTENIDO: “No existe el dia 00 en el calendario”

TIPO DE DATO: STRING

DATO: 12/13/1984

RESULTADO OBTENIDO: “No existe mes N° 13 en el calendario”

TIPO DE DATO: STRING

DATO: 12/00/1984

RESULTADO OBTENIDO: “No existe mes N° 00 en el calendario”

**Ejercicio Nro. 5**

**Escribir un programa que almacene el diccionario con los créditos de las asignaturas de un curso {'Matemáticas': 6, 'Física': 4, 'Química': 5} y después muestre por pantalla los créditos de cada asignatura en el formato tiene créditos, donde es cada una de las asignaturas del curso, y son sus créditos. Al final debe mostrar también el número total de créditos del curso.**

En este programa se declara un diccionario con los nombres de las materias y sus respectivos créditos y un acumulador para acumular la suma de los valores de los créditos de las asignaturas. Se usa el bucle for para iterar en el diccionario y mostrar por pantalla la clave y valor con el formato estipulado en la consigna: "{asignaturas} tiene {creditosAsignaturas[asignaturas]} créditos". Donde asignaturas es la clave y creditosAsignaturas[asignaturas] el valor.   
A su vez, se van a cumulando las sumas de los valores de cada asignatura en el acumulador:  
creditosTotalDelCurso += creditosAsignaturas[asignaturas]. Donde creditosTotalDelCurso es el acumulador y creditosAsignaturas[asignaturas] los valores que se suman. Por último, se imprime por pantalla el total de los créditos en el curso.

**Ejercicio Nro. 6**

**Escribir un programa que cree un diccionario vacío y lo vaya llenado con información sobre una persona (por ejemplo nombre, edad, sexo, teléfono, correo electrónico, etc.) que se le pida al usuario. Cada vez que se añada un nuevo dato debe imprimirse el contenido del diccionario.**

En este programa en principio se crea un diccionario persona vacío y una variable booleana bandera en True, para poder inicializar el bucle while. Luego, se usa el bucle while con la condición que mientras que bandera sea True va a seguir repitiendo el bloque. Dentro de este bloque while, se ingresan en dos variables (clave – valor), de tipo string, las claves y los valores del diccionario persona a través de la función input(). Una vez ingresado esos datos se cargan al diccionario y se muestran por pantalla. Luego, se le pregunta al usuario si desea seguir cargando datos al diccionario utilizando una variable de control “seguir” la cual verifica: si el usuario ingresa “si”, bandera sigue siendo True. En cambio si el usuario ingresa “no”, bandera pasa a ser False y por consiguiente sale del bucle while finalizando el programa.

TIPOS DE DATOS: STRING

DATOS: nombre , Javier ; si ; edad , 40 ; no

RESULTADO OBTENIDO:   
“Ingrese el nombre del campo: nombre

Ingrese el valor del campo: Javier

Contenido del diccionario  
 {'nombre': 'Javier'}

¿Desea seguir? (Si / No): si

Ingrese el nombre del campo: edad

Ingrese el valor del campo: 40

Contenido del diccionario  
 {'nombre': 'Javier', 'edad': '40'}

¿Desea seguir? (Si / No): no”

Ejercicio Nro. 7

**Escribir un programa que cree un diccionario simulando una cesta de la compra. El programa debe preguntar el artículo y su precio y añadir el par al diccionario, hasta que el usuario decida terminar. Después se debe mostrar por pantalla la lista de la compra y el coste total, con el siguiente formato.**

Este programa tiene un funcionamiento similar al ejercicio N°6, con las siguientes diferencias:

1. El ingreso en la variable “valor” se castea a “int” para poder hacer la operación matemática correspondiente.
2. Para mostrar por pantalla la lista de compras con los artículos y sus precios, uno debajo del otro y calcular el total de la compra, se utiliza un bucle for que va a iterar en el diccionario listaCompras (aquí se guardan las claves y valores ingresados en las variables “clave” y “valor”).

TIPOS DE DATOS: STRING

DATOS: SHAMPOO , 2000 ; SI ; LECHE , 1100 ; SI ; FIDEOS , 850 ; NO

RESULTADO OBTENIDO:

“LISTA DE LA COMPRA  
---------------------------------------

Articulo Precio  
---------------------------------------

SHAMPOO $ 2000

LECHE $ 1100

FIDEOS $ 850

---------------------------------------

TOTAL $ 3950”

1. Función buscada en chatGPT [↑](#footnote-ref-1)