



El futuro digital
es de todos

MinTIC

Diccionarios

+ Rogerio Orlando Beltrán Castro



Ejercicio01

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.:

Solución Ejercicio01

```
monedas = {'Euro':'€', 'Dollar':'$', 'Yen':'¥'}  
moneda = input("Introduce una divisa: ")  
if moneda.title() in monedas:  
    print(monedas[moneda.title()])  
else:  
    print("La divisa no está.")
```

Ejercicio02

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>.

Solución Ejercicio02

```
nombre = input('¿Cómo te llamas? ')
edad = input('¿Cuántos años tienes? ')
direccion = input('¿Cuál es tu dirección? ')
telefono = input('¿Cuál es tu número de teléfono? ')
persona = {'nombre': nombre, 'edad': edad, 'direccion': direccion,
'telefono': telefono}
print(persona['nombre'], 'tiene', persona['edad'], 'años, vive en',
persona['direccion'], 'y su número de teléfono es', persona['telefono'])
```


Ejercicio03

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.

Fruta	Precio
Plátano:	1.35
Manzana:	0.80
Pera:	0.85
Naranja:	0.70

Solución Ejercicio03

```
frutas = {'Plátano':1.35, 'Manzana':0.8, 'Pera':0.85, 'Naranja':0.7}
```

```
fruta = input('¿Qué fruta quieres? ').title()
```

```
kg = float(input('¿Cuántos kilos? '))
```

```
if fruta in frutas:
```

```
    print(kg, 'kilos de', fruta, 'valen', frutas[fruta]*kg, '€')
```

```
else:
```

```
    print("Lo siento, la fruta", fruta, "no está disponible.")
```

Ejercicio04

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 <asignatura> tiene <créditos> créditos, donde <asignatura> es cada una de las asignaturas del curso, y <créditos> son sus créditos. Al final debe mostrar también el número total de créditos del curso.

Solución Ejercicio04

```
curso = {'Matemáticas': 6, 'Física': 4, 'Química': 5}
total_creditos = 0
for asignatura, creditos in curso.items():
    print(asignatura, 'tiene', creditos, 'créditos')
    total_creditos += creditos
print('Número total de créditos del curso: ', total_creditos)
```

Ejercicio05

Escribir un programa que gestione las facturas pendientes de cobro de una empresa. Las facturas se almacenarán en un diccionario donde la clave de cada factura será el número de factura y el valor el coste de la factura. El programa debe preguntar al usuario si quiere añadir una nueva factura, pagar una existente o terminar. Si desea añadir una nueva factura se preguntará por el número de factura y su coste y se añadirá al diccionario. Si se desea pagar una factura se preguntará por el número de factura y se eliminará del diccionario. Después de cada operación el programa debe mostrar por pantalla la cantidad cobrada hasta el momento y la cantidad pendiente de cobro.

Solución Ejercicio05

```
facturas = {}
cobrado = 0
pendiente = 0
more = ''
while more != 'T':
    if more == 'A':
        clave = input('Introduce el número de la factura: ')
        coste = float(input('Introduce el coste de la factura: '))
        facturas[clave] = coste
        pendiente += coste
    if more == 'P':
        clave = input('Introduce el número de la factura a pagar: ')
        coste = facturas.pop(clave, 0)
        cobrado += coste
        pendiente -= coste
print('Recaudado:', cobrado)
print('Pendiente de cobro: ', pendiente)
more = input('¿Quieres añadir una nueva factura (A), pagarla (P) o terminar (T)?')
')
```

Ejercicio06

Escribir un programa que permita gestionar la base de datos de clientes de una empresa. Los clientes se guardarán en un diccionario en el que la clave de cada cliente será su NIF, y el valor será otro diccionario con los datos del cliente (nombre, dirección, teléfono, correo, preferente), donde preferente tendrá el valor True si se trata de un cliente preferente. El programa debe preguntar al usuario por una opción del siguiente menú: (1) Añadir cliente, (2) Eliminar cliente, (3) Mostrar cliente, (4) Listar todos los clientes, (5) Listar clientes preferentes, (6) Terminar. En función de la opción elegida el programa tendrá que hacer lo siguiente:

Preguntar los datos del cliente, crear un diccionario con los datos y añadirlo a la base de datos.

Preguntar por el NIF del cliente y eliminar sus datos de la base de datos.

Preguntar por el NIF del cliente y mostrar sus datos.

Mostrar lista de todos los clientes de la base datos con su NIF y nombre.

Mostrar la lista de clientes preferentes de la base de datos con su NIF y nombre.

Terminar el programa.

Solución Ejercicio06

```
clientes = {}
opcion = ""
while opcion != '6':
    if opcion == '1':
        nif = input('Introduce NIF del cliente: ')
        nombre = input('Introduce el nombre del cliente: ')
        direccion = input('Introduce la dirección del cliente: ')
        telefono = input('Introduce el teléfono del cliente: ')
        email = input('Introduce el correo electrónico del cliente: ')
        vip = input('¿Es un cliente preferente (S/N)? ')
        cliente = {'nombre':nombre, 'dirección':direccion, 'teléfono':telefono, 'email':email, 'preferente':vip=='S'}
        clientes[nif] = cliente
    if opcion == '2':
        nif = input('Introduce NIF del cliente: ')
        if nif in clientes:
            del clientes[nif]
        else:
            print('No existe el cliente con el nif, nif)
    if opcion == '3':
        nif = input('Introduce NIF del cliente: ')
        if nif in clientes:
            print('NIF:', nif)
            for clave, valor in clientes[nif].items():
                print(clave.title() + ': ', valor)
        else:
            print('No existe el cliente con el nif, nif)
    if opcion == '4':
        print('Lista de clientes')
        for clave, valor in clientes.items():
            print(clave, valor['nombre'])
    if opcion == '5':
        print('Lista de clientes preferentes')
        for clave, valor in clientes.items():
            if valor['preferente']:
                print(clave, valor['nombre'])
    opcion = input('Menú de opciones\n(1) Añadir cliente\n(2) Eliminar cliente\n(3) Mostrar cliente\n(4) Listar\nclientes\n(5) Listar clientes preferentes\n(6) Terminar\nElige una opción:')
```


Ejercicio07

El directorio de los clientes de una empresa está organizado en una cadena de texto como la de más abajo, donde cada línea contiene la información del nombre, email, teléfono, nif, y el descuento que se le aplica. Las líneas se separan con el carácter de cambio de línea `\n` y la primera línea contiene los nombres de los campos con la información contenida en el directorio.

Solución Ejercicio07

```
# Cadena con los datos de los clientes de la empresa
datos_clientes = "nif;nombre;email;teléfono;descuento\n01234567L;Luis González;luigonzalez@mail.com;656343576;12.5\n71476342J;Macarena Ramírez;macaren
a@mail.com;692839321;8\n63823376M;Juan José Martínez;juanjo@mail.com;664888233;5.2\n98376547F;Carmen Sánchez;carmen@mail.com;667677855;15.7"
# Dividimos la cadena por el caracter de cambio de línea \n y creamos una lista con las subcadenas
lista_clientes = datos_clientes.split('\n')
# Inicializamos el diccionario que va a contener el directorio de clientes a vacío.
directorio = {}
# Dividimos la cadena del primer elemento de la lista de clientes (que contienen los
# nombres de los campos) por el caracter ; y creamos una lista con los campos.
lista_campos = lista_clientes[0].split(';')
# Bucle iterativo para recorrer los elementos de la lista lista_clientes.
# la variable cliente recorre desde el segundo elemento hasta el último elemento de la lista
# (el primer elemento contiene los nombres de campo así que no corresponde a un cliente)
for i in lista_clientes[1:]:
    # Inicializamos el diccionario que va a contener los datos del cliente actual a vacío.
    cliente = {}
    # Dividimos la cadena i por el caracter ; y creamos una lista con las subcadenas con la
    # información del cliente
    lista_info = i.split(';')
    # Bucle iterativo para recorrer los campos y añadir los pares al diccionario del cliente.
    # j toma valores de 1 al número de campos menos 1. El primer elemento (posición 0) corresponde
    # al nif y no se añade al diccionario porque se utilizará después como clave en el diccionario
    # principal
    for j in range(1, len(lista_campos)):
        # Condicional. Si el campo actual es descuento convertimos su valor en real
        if lista_campos[j] == 'descuento':
            lista_info[j] = float(lista_info[j])
            cliente[lista_campos[j]] = lista_info[j]
    # Añadimos un par al diccionario del directorio con la clave el nif del cliente y valor
    # el diccionario que acabamos de crear con el resto de sus datos.
    directorio[lista_info[0]] = cliente
# Mostramos el diccionario por pantalla
print(directorio)
```