

Tengo problemas técnicos comenzamos 5:08



Programas que involucran datos estructurados (Diccionarios)

Las **llaves {}** se utiliza para declarar un diccionario.

{ "Coffee": 10, "Tea": 5, "Cookie": 8, "Chips": 15 }

Key, Donde puede tener cualquier nombre (string).

Value, puede ser cualquier tipo (primitivo y estructurado)

Los **dos puntos :** se utiliza para asignar el value al key.

La **coma**, se utiliza para separar los elementos dentro de una lista.

Ejercicio 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.

```
divisa = {'Euro':'E', 'Dollar':'S', 'Yen':'Y'} # key: Value

divisa_usuario = raw_input("Escribe una divisa: ").lower()

for x in divisa:
    if x.lower() == divisa_usuario:
        print("El %s si existe y este es su moneda %s " %(x.lower(), divisa[x]))
```

Ejercicio 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>.

```
nombre = raw_input("Escribe tu nombre: ")
edad = int(raw_input("Escribe tu edad: "))
direccion = raw_input("Escribe tu direccion: ")
telefono = int(raw_input("Escribe tu telefono: "))

persona = {"nombre": nombre, "edad": edad, "direccion": direccion, "telefono": telefono }

print(persona)

print(persona["nombre"] + " tiene " + str(persona["edad"]) +
      " años, vive en " + persona["direccion"] +
      " y su número de telefono es " + str(persona["telefono"]))
```

Ejercicio 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.

Fruta	Precio
Plátano	1.35
Manzana	0.80
Pera	0.85
Naranja	0.70

```
fruta_user = input("Escribe la fruta: ").lower().replace("á", "a")
peso = int(input("Escribe el peso: "))

frutas = { "platano": 1.35, "manzana": .80, "pera":.85, "naranja":.70 }

if fruta_user in frutas:
    total = peso * frutas[fruta_user]
    print("El costo de %s es de %s" %(fruta_user, str(total)))
else:
    print("La fruta no existe.")
```

Ejercicio 4

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

```
user_mes = raw_input("Escribe la fecha en el formato dd-mm-aaaa: ").split("-")
mes = int(user_mes[1])
meses = ['1:'enero', 2:'febrero', 3:'marzo', 4:'abril', 5:'mayo', 6:'junio',
        7:'julio', 8:'agosto', 9:'septiembre', 10:'octubre', 11:'noviembre',
        12:'diciembre']

if mes > 12:
    print("Mes Invalida")
else:
    print("Dia " + user_mes[0] + " Mes " + meses[mes] + " Año " + user_mes[2])
```

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

```
materias = {'Matemáticas': 6, 'Física': 4, 'Química': 5}

# Solucion 1
texto = "David "
for materia in materias:
    texto += "tiene en " + materia + " " + str(materias[materia]) + ", "
print(texto)

# Solucion 2
texto = "David "
for i, j in materias.items():
    texto += "tiene en " + i + " " + str(j) + ", "
print(texto)
```

Ejercicio 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.

Ejercicio 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

Lista de la compra	
Artículo 1	Precio
Artículo 2	Precio
Artículo 3	Precio
...	...
Total	Coste

Ejercicio 8

Escribir un programa que cree un diccionario de traducción español-inglés. El usuario introducirá las palabras en español e inglés separadas por dos puntos, y cada par <palabra>:<traducción> separados por comas. El programa debe crear un diccionario con las palabras y sus traducciones. Después pedirá una frase en español y utilizará el diccionario para traducirla palabra a palabra. Si una palabra no está en el diccionario debe dejarla sin traducir.

Ejercicio 9

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.

Ejercicio 10

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:

1. Preguntar los datos del cliente, crear un diccionario con los datos y añadirlo a la base de datos.
2. Preguntar por el NIF del cliente y eliminar sus datos de la base de datos.
3. Preguntar por el NIF del cliente y mostrar sus datos.
4. Mostrar lista de todos los clientes de la base datos con su NIF y nombre.
5. Mostrar la lista de clientes preferentes de la base de datos con su NIF y nombre.
6. Terminar el programa.