



## ✓ Diccionarios

### Generalidades

Los diccionarios son un tipo de colección donde cada elemento almacenado contiene una estructura de tipo clave-valor.

## ✓ Creacion de diccionarios

La definición al igual que los conjuntos es por medio de llaves {} o utilizando el constructor "dict()"

```
diccionario = {1:'uno', 2:'dos', 3:'tres'}  
print(type(diccionario))  
#dic2= dict()  
#print(type(dic2))
```

```
<class 'dict'>
```

En los diccionarios las claves deben ser únicas, pues es a través de ellas que podemos acceder a los valores.

```
diccionario = {"tres":"three", "dos":"two", "cuatro":"four", "cinco":"five"}  
print (type (diccionario))  
print (diccionario)
```

```
<class 'dict'>  
{'tres': 'three', 'dos': 'two', 'cuatro': 'four', 'cinco': 'five'}
```

Cada elemento en un diccionario está asociado con una clave única que actúa como un identificador para acceder al valor correspondiente.

```
diccionario = {"tres":"three","dos":"two","cuatro":"four","cinco":"five"}
print (diccionario["tres"])

three
```

Los diccionarios pueden estar conformados por diferentes tipos de datos como cadenas de texto, numeros enteros, decimales y estructuras de datos como listas, y otros diccionarios.

```
dic = {"llave1":"a",2:[2,34,5],4:"c",5:8,("fila","columna"): [23,21], "tupla":(22,"a")}
#print(type(dic))
#print (dic)
print(dic["llave1"])
print(dic[2])

a
[2, 34, 5]
```

## ✓ Acceso a un diccionario

Para acceder a un valor determinado del diccionario ya no lo hacemos por su posición sino por su clave. Si intentamos acceder a un elemento del diccionario a través de una clave que no existe obtendremos un error.

```
capitales = {"Caldas": "Manizales", "Risaralda": "Pereira", "Valle": "Cali"}
print (f"La capital de Caldas es {capitales['Caldas']}")#acceso por clave
```

```
La capital de Caldas es Manizales
```

```
x = {"3":2,"4":[1,"s",3],"5":2}
print (type(x))
print(x["4"][1])
```

```
<class 'dict'>
s
```

Podemos acceder a un elemento utilizando el método get() partir de su clave y si no lo encuentra devuelve un valor por defecto que podemos especificar:

```
x = {"3":2,"4":[1,"s",3],"5":2}
x.get('5','Lo que buscas no esta en el diccionario')
x.get(10,'Lo que buscas no esta en el diccionario')
```

'Lo que buscas no esta en el diccionario'

## ✓ RETO 1

Supongamos que tenemos un diccionario que almacena los precios de algunos productos en una tienda. Queremos que, dado el nombre de un producto, obtengamos su precio. Si el producto no está en el diccionario, queremos mostrar un mensaje predeterminado "El producto no esta disponible."

```
productos = {  
    "manzanas": 2.5,  
    "bananas": 1.8,  
    "naranjas": 3.0,  
    "peras": 2.0  
}  
pregunta = input("Ingrese el producto: ")  
print(productos.get(pregunta,"El producto no esta disponible"))  
  
Ingrese el producto: uvas  
El producto no esta disponible
```

Para acceder a las claves de un diccionario podemos utilizar el método "keys()"

```
x = {"3":2,"4":[1,"s",3],"5":2}  
print(x.keys())
```

```
dict_keys(['3', '4', '5'])
```

```
x = {"3":2,"4":[1,"s",3],"5":2}  
claves= list(x.keys())  
print(claves[1])
```

```
4
```

Para acceder a los valores de un diccionario podemos utilizar el método "values()"

```
x = {"3":2,"4":[1,"s",3],"5":2}  
x.values()
```

```
dict_values([2, [1, 's', 3], 2])
```

Podemos acceder a la clave-valor de un diccionario utilizando el método "items()"

Creando una lista de tuplas:

```
x = {"3":2,"4":[1,"s",3],"5":2}
#x.items()
print(x)

{'3': 2, '4': [1, 's', 3], '5': 2}
```

Creando dos variables para el almacenamiento de la clave y del valor

```
x = {"3":2,"4":[1,"s",3],"5":2}
for clave, valor in x.items():
    print(clave, valor)

3 2
4 [1, 's', 3]
5 2
```

O creando una tupla con los elementos clave y valor

```
edades = {'Hector':27,'Juan':45,'Maria':34}
#print (edades)
for item in edades.items():
    print(item) #obteniendo clave-valor en la misma variable
    print(item[0], "tiene", item[1], "años")

('Hector', 27)
Hector tiene 27 años
('Juan', 45)
Juan tiene 45 años
('Maria', 34)
Maria tiene 34 años
```

## ✓ RETO 2

Tenemos un diccionario con las materias y notas respectivamente de un estudiante, utilizando los métodos vistos, accede a las notas del estudiante y realiza el promedio

```

estudiante1 = {
    "matematicas": 8,
    "historia": 7,
    "ciencias": 9,
    "ingles": 8,
    "arte": 6
}
promedio = 0

for nota in estudiante1.values():
    promedio = promedio + nota

promedio = promedio / len(estudiante1)

print(promedio)

```

7.6

```

estudiante1 = {
    "matematicas": 8,
    "historia": 7,
    "ciencias": 9,
    "ingles": 8,
    "arte": 6
}
promedio = sum(estudiante1.values())/len(estudiante1)
print(f"el promedio de los estudiantes es {promedio}")

```

el promedio de los estudiantes es 7.6

Los diccionarios son mutables, por lo tanto podemos cambiar los valores de sus elementos

```

capitales = {"Caldas": "Manizales", "Risaralda": "Pereira", "Valle": "Cali"}
print(capitales)
capitales["Valle"] = "Santiago de Cali"
print (capitales)

{'Caldas': 'Manizales', 'Risaralda': 'Pereira', 'Valle': 'Cali'}
{'Caldas': 'Manizales', 'Risaralda': 'Pereira', 'Valle': 'Santiago de Cali'}

```

## ✓ Adición de elementos.

Se puede agregar datos a un diccionario por asignacion directa

```
producto = {"nombre": "Laptop", "precio": 1200, "stock": 50}
```

```
producto["marca"] = "Acer"  
print(producto)
```

```
{'nombre': 'Laptop', 'precio': 1200, 'stock': 50, 'marca': 'Acer'}
```

También lo podemos hacer mediante el método "update()", el cual nos permite agregar uno o varios elementos simultáneos al diccionario

```
producto = {"nombre": "Laptop", "precio": 1200, "stock": 50, "marca": "Acer"}
```

```
producto.update({"serial": "sn12en12", "color": "plateado"})  
print(producto)
```

```
{'nombre': 'Laptop', 'precio': 1200, 'stock': 50, 'marca': 'Acer', 'serial': 'sn12en12',
```



## ✓ Eliminación

Usamos la función **del()** para borrar un elemento del diccionario

```
capitales = {"Caldas": "Manizales", "Risaralda": "Pereira", "Valle": "Cali"}  
del(capitales ["Valle"]) #se elimina el elemento con la clave dada  
print (capitales)
```

```
{'Caldas': 'Manizales', 'Risaralda': 'Pereira'}
```

También podemos utilizar el método "pop"

```
capitales = {"Caldas": "Manizales", "Risaralda": "Pereira", "Valle": "Cali"}  
print(capitales.pop('Caldas'))  
print (capitales)
```

```
Manizales  
{'Risaralda': 'Pereira', 'Valle': 'Cali'}
```

## ✓ RETO 3

Se ha organizado una rifa con 5 participantes. En los primeros 2 intentos se elimina al azar un participante y se muestra quién fue el eliminado de la lista para que no pueda ser seleccionado

nuevamente. El 3 seleccionado se anuncia quién como el ganador de la rifa.

```
import random
participantes = {
    'Juan': '123456',
    'María': '234567',
    'Pedro': '345678',
    'Laura': '456789',
    'Carlos': '567890'
}

contador = 3

while contador > 1:
    eliminar = random.choice(list(participantes.keys()))
    del participantes[eliminar]
    print (f"Se eliminó {eliminar}")
    contador -= 1

ganador = random.choice(list(participantes.keys()))
print (f"El ganador es {ganador}")

#Lastrañomano

    Se eliminó Juan
    Se eliminó María
    El ganador es Laura
```

## ✓ Recorrido de diccionarios

Los diccionarios se pueden recorrer utilizando bucles for para acceder a todas las claves, valores o pares clave-valor.

```
edades = {'Hector':27,'Juan':45,'Maria':34}
print (edades)

for clave in edades:
    print(f"Acceso a la clave {clave}")

    {'Hector': 27, 'Juan': 45, 'Maria': 34}
    Acceso a la clave Hector
    Acceso a la clave Juan
    Acceso a la clave Maria

edades = {'Hector':27,'Juan':45,'Maria':34}
print (edades)

for clave in edades:
```

```
print(f"Acceso a los valores {edades[clave]}")
```

```
{'Hector': 27, 'Juan': 45, 'Maria': 34}  
Acceso a los valores 27  
Acceso a los valores 45  
Acceso a los valores 34
```

```
edades = {'Hector':27,'Juan':45,'Maria':34}  
print (edades)  
for clave in edades:  
    print(f"acceso a claves y los valore {clave, edades[clave]}")  
  
{'Hector': 27, 'Juan': 45, 'Maria': 34}  
acceso a claves y los valore ('Hector', 27)  
acceso a claves y los valore ('Juan', 45)  
acceso a claves y los valore ('Maria', 34)
```

## ✓ Conversiones de colecciones

### Diccionarios a Listas o Tuplas

Puedes convertir un diccionario en una lista de sus claves, una lista de sus valores o una lista de tuplas (clave, valor).

```
mi_diccionario = {'a': 1, 'b': 2, 'c': 3}  
  
lista_claves = list(mi_diccionario.keys())  
print("Lista de claves:", lista_claves)  
  
lista_valores = list(mi_diccionario.values())  
print("Lista de valores:", lista_valores)  
  
lista_tuplas = list(mi_diccionario.items())  
print("Lista de tuplas:", lista_tuplas)
```

### Listas o Tuplas a Diccionarios

Puedes convertir una lista de tuplas (clave, valor) en un diccionario.

```
lista_tuplas = [('a', 1), ('b', 2), ('c', 3)]  
  
nuevo_diccionario = dict(lista_tuplas)  
print("Nuevo diccionario:", nuevo_diccionario)
```

### Conjuntos a Listas o Tuplas



Puedes convertir un conjunto en una lista o una tupla.

```
mi_conjunto = {1, 2, 3, 4, 5}

lista_desde_conjunto = list(mi_conjunto)
print("Lista desde conjunto:", lista_desde_conjunto)

tupla_desde_conjunto = tuple(mi_conjunto)
print("Tupla desde conjunto:", tupla_desde_conjunto)
```

## Listas o Tuplas a Conjuntos

Puedes convertir una lista o una tupla en un conjunto, eliminando así los duplicados.

```
mi_lista = [1, 2, 3, 4, 5, 3, 4]
conjunto_desde_lista = set(mi_lista)
print("Conjunto desde lista:", conjunto_desde_lista)

mi_tupla = (1, 2, 3, 4, 5, 3, 4)
conjunto_desde_tupla = set(mi_tupla)
print("Conjunto desde tupla:", conjunto_desde_tupla)
```

## ✓ Apropiación

1. Crea un diccionario llamado `empleados` donde las claves sean los números de identificación y los valores sean diccionarios con el nombre y el cargo del empleado. Agrega al menos 5 empleados con su respectiva información. Permite al usuario buscar empleados por su número de identificación y mostrar toda su información. Permite al usuario eliminar un empleado por su número de identificación.
2. Crea un diccionario `contactos_avanzado` donde las claves sean los nombres de las personas y los valores sean diccionarios con información detallada como teléfono, correo electrónico y dirección.
  - a. Agrega al menos 3 contactos con su respectiva información detallada.
  - b. Muestra al usuario la información detallada de un contacto específico.
  - b. Permite al usuario actualizar la información de contacto de una persona existente.

3. Se te proporciona una lista de ventas mensuales de diferentes productos. Debes eliminar los duplicados, luego convertirlos en un diccionario donde las claves sean los nombres de los productos y los valores sean las ventas mensuales.

```
ventas_mensuales = [("Producto A", 150), ("Producto B", 100), ("Producto A", 150), ("Producto C", 180), ("Producto B", 100), ("Producto A", 150), ("Producto D", 200), ("Producto C", 150), ("Producto D", 200), ("Producto E", 400), ("Producto F", 200)]
```

Se requiere presentar un informe con la siguiente información:

- a. El total de las ventas
  - b. El promedio de ventas mensual
  - c. cual fue la maxima venta y de que producto
  - d. cual fue la menor venta y de que producto
4. Crea un diccionario `puntajes_juegos` donde las claves sean los nombres de los juegos y los valores sean listas de puntajes obtenidos por los jugadores. Agrega al menos 3 juegos con sus respectivos puntajes. Muestra al usuario el promedio de puntajes de un juego específico. Permite al usuario agregar un nuevo puntaje a un juego existente.
5. Crea un diccionario llamado `inventario` donde las claves sean los nombres de los productos y los valores sean diccionarios con la cantidad y el precio del producto. Agrega al menos 10 productos con su respectiva información.
- a. El programa debe mostrar el precio total de todo el inventario
  - b. Permite al usuario buscar un producto por su nombre y mostrar toda su información y el precio total del inventario de ese producto.
  - c. Permite al usuario actualizar el precio de un producto existente.
6. Crea un diccionario `inventario_libros` donde las claves sean los codigos de los libros y los valores sean tuplas con el títulos, autor, el número de copias disponibles y valor del libro. Agrega al menos 5 libros con sus respectivos autores y copias disponibles.
- a. Muestra al usuario los libros disponibles de un autor específico.
  - b. Permite al usuario comprar de un libro la cantidad si esta disponible y actualizar el número de copias disponibles de ese libro. Se debe mostrar al cliente el valor a pagar.

