



Programación Avanzada 2025

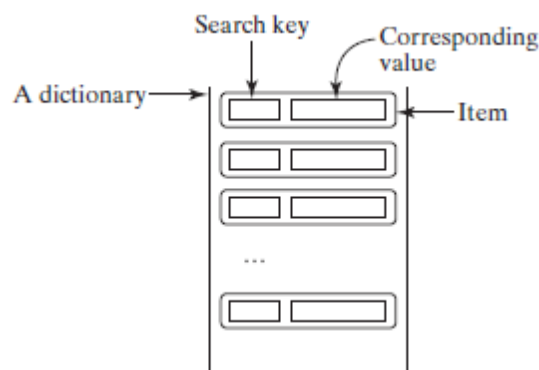
Lab. 3.2. Diccionarios en Python

Septiembre 05, 2025

Cree una carpeta (folder) en el disco D, nómbrela con su apellido paterno seguido de su código. Ejemplo: LOPEZ12345

Un diccionario es un objeto contenedor que almacena, sin un orden específico, una colección de pares *clave/valor* (key/value). Permite acceso rápido, eliminar y actualizar los valores usando sus claves.

Las claves se parecen a un operador índice. En una lista los índices son enteros. En un diccionario, la clave puede ser cualquier objeto que tenga un valor hash. Un diccionario no puede contener claves duplicadas. Cada clave corresponde (mapea) a un valor. Una clave y su correspondiente valor forman un *ítem* (entrada, elemento) almacenado en un diccionario, como se muestra en la siguiente figura:



Creando un diccionario

Un diccionario se crea encerrando los ítems dentro de un par de llaves (`{}`). Cada ítem consiste de una *clave* (llave), seguido por dos puntos, seguido por un *valor*. Los ítems son separados por comas. Por ejemplo, la siguiente instrucción:

```
students = {"111-34-3434": "John", "132-56-6290": "Peter"}
```

crea un diccionario con dos ítems. Un ítem tiene la forma **key:value**. La clave en el primer ítem es **111-34-3434**, y su correspondiente valor es **John**.

Mientras que los valores pueden ser de cualquier tipo y pueden repetir, las claves deben de ser inmutables (string, número, o tuplas con elementos inmutables) y deben de ser únicos.

Un **diccionario vacío** se crea de la siguiente manera:

```
students = {} # Crea un diccionario vacío  
students = dict() # Crea un diccionario vacío
```

Agregando y quitando elementos de un diccionario

Para **adicionar un ítem** al diccionario use la siguiente sintaxis:

```
dictionaryName[key] = value
```

por ejemplo:

```
students["234-56-9010"] = "Susan"
```

Si la clave existe en el diccionario, el valor correspondiente será reemplazado.

Otra forma de adicionar un ítem es usando el método `update()`.

```
students.update({"351-45-6030": "Steve"})
```

Para **eliminar un ítem** de un diccionario, use la siguiente sintaxis:

```
del dictionaryName[key]
```

For example:

```
del students["234-56-9010"]
```

Otras formas de eliminar un ítem es usando los métodos `pop()`, `popitem()` y `clear()`.

Para ejercicios del 1 al 8, verificar la salida con la ayuda del interpretador de Python

1. ¿Cómo se crea un diccionario vacío?

2. ¿Cuáles de los siguientes diccionarios están creados correctamente?

```
d = {1:[1, 2], 3:[3, 4]}
d = {[1, 2]:1, [3, 4]:3}
d = {(1, 2):1, (3, 4):3}
d = {1:"john", 3:"peter"}
d = {"john":1, "peter":3}
```

3. Cada ítem en un diccionario tiene dos partes, ¿cómo se llaman ellos?

4. Suponga que un diccionario llamado `students` es definido como `{"john":3, "peter":2}`.

¿Qué hacen los siguientes códigos?

```
(a) students["susan"] = 5
(b) students["peter"] = 5
(c) students["peter"] += 5
(d) del students["peter"]
```

5. Suponga que un diccionario llamado `students` es definido como `{"john":3, "peter":2}`.

¿Qué hacen los siguientes códigos?

```
(a) print(len(students))
(b) print(students.keys())
(c) print(students.values())
(d) print(students.items())
```

6. Muestre la salida del siguiente código:

```
def main():
    d = {"red":4, "blue":1, "green":14, "yellow":2}
    print(d["red"])
    print(list(d.keys()))
    print(list(d.values()))
    print("blue" in d)
    print("purple" in d)
    d["blue"] += 10
    print(d["blue"])
main() # Call the main function
```

7. Muestre la salida del siguiente código:

```
def main():
    d = {}
    d["susan"] = 50
    d["jim"] = 45
    d["joan"] = 54
    d["susan"] = 51
    d["john"] = 53
    print(len(d))
main() # Call the main function
```

8. Para un diccionario `d`, se puede usar `d[key]` o `d.get(key)` para retornar el valor de la clave. ¿Cuál es la diferencia entre ellos?

Para los siguientes ejercicios, elabore un programa en Python. Verifique los programas ejecutando y probando con distintas entradas.

1. Crea las siguientes bases de datos de una empresa usando diccionarios.
 - a) Base de datos de gerencias (departamentos) de la empresa. El diccionario debe tener como clave el **id** de una gerencia y como valor el nombre de la gerencia. Los códigos de las gerencias son:

```
101 RRHH,  
102 FINANZA,  
103 CONTABILIDAD,  
104 VENTAS,  
105 INGENIERIA,  
106 SOPORTE
```

- b) Base de datos de empleados. Use diccionarios anidados. Cada empleado es identificado por su **id**, y cada empleado, además, tiene asociado los siguientes datos, nombre, fecha de ingreso, y gerencia (departamento, dept) a la que pertenece.

Ejemplo de datos de un empleado:

Id	Nombre	Fing	Dept
1000	'Juan'	10-10-17	103

- c) Imprimir las bases de datos. Ejemplo de salidas:

Base de Datos de las Gerencias:

```
101 RRHH,  
102 FINANZA,  
103 CONTABILIDAD,  
104 VENTAS,  
105 INGENIERIA,  
106 SOPORTE
```

Base de Datos de los empleados:

```
Persona ID: 1000  
Nombre: Juan  
Fing: 10-10-17  
Dept: 103
```

```
Persona ID: 1001  
Nombre: Mary  
Fing: 01-11-88  
Dept: 101
```

- d) Mostrar los empleados cuyos **id** sean impares.
 - e) Agregue a la base de datos de empleados un empleado más con todos sus datos.
 - f) Modifique el nombre de uno de los empleados sin usar la función update()
 - g) Borre solamente el dato **dept** del empleado que agregó en el ejercicio (e)
 - h) Elimine al empleado que agregó en el apartad (e)
2. Usando la base de datos de empleados del ejercicio N° 1 realizar lo siguiente:
 - a) Mostrar en orden descendente la base de datos con respecto al nombre de los empleados.
 - b) Crea un nuevo diccionario con los empleados que pertenezcan a la gerencia (departamento) de finanzas.
 - c) Cambie el nombre de uno de los empleados usando la función update(), sus otros datos se mantienen sin cambios.

3. Guardar la base de datos de empleados del ejercicio N° 1 en archivos de los siguientes formatos:
- a) Formato .pickle
 - b) Formato .json
 - c) Formato .csv

En todos los casos, lea los datos desde los archivos y muéstrellos en pantalla.

4. En el archivo **alumnos.csv** se guarda la información de varios alumnos, cada fila corresponde a la información de un alumno.
- a) Implemente una función que reciba el nombre del archivo, luego lea el archivo, omitiendo el encabezado, convierta cada fila en un diccionario, agregue el diccionario a una lista de diccionarios, y por último devuelva la lista de diccionarios.
 - b) Mostrar la información de cada alumno, recorriendo la lista de diccionarios.
 - c) Determinar al alumno de mayor edad
 - d) Determinar el promedio de notas de los alumnos cuyas edades sean igual a 20 años.

OPCIONAL

5. Una persona tiene una lista con los precios de **n** artículos y dispone de una cierta cantidad de dinero. Escriba un programa para leer estos datos y almacenarlos en listas separadas, en una lista guarde los nombres de los artículos y en otra lista paralela guarde los precios correspondientes.

A partir de las dos listas anteriores genere un diccionario donde las llaves (claves) de los ítems deben ser los nombres de los productos y realizar lo siguiente:

- a) Muestre los artículos que puede comprar el usuario
- b) Para cada artículo cuyo precio es menor que la cantidad de dinero disponible, determine cuantas unidades puede comprar
- c) Determine cuántos artículos, de aquel de mayor precio, puede comprar el usuario, muestre también el nombre de aquel artículo.
- d) Mostrar los artículos que tengan los mismos precios.

Guarde todos vuestros programas en una carpeta con el nombre su **Apellido** paterno seguido de vuestro **DNI**, luego comprima esta carpeta. Envíe este archivo a Katherine Navarro katherine.navarro@upch.pe especificando como asunto **Lab3.2**.