

Jorge Calvo

Diccionarios y control de errores

Sígueme en - [LinkedIn](#) [Twitter](#) [Blog](#)



Este obra está bajo una [licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional](#).

Diccionarios

Los diccionarios son una estructura que almacena elementos a los que se les da un nombre o clave (en inglés key). De esta forma, en lugar de acceder al elemento por medio del índice que ocupa se accede a él por el nombre que tiene. Podemos definir un diccionario como un conjunto de parejas clave - valor (key - value) donde la clave es el nombre que se le da a un valor.

Los valores de un diccionario pueden tener cualquier forma: números, strings, listas...

```
asignaturas={"Luis":"Lengua", "Marta":"Matematicas"}
print(asignaturas)
```

El diccionario se crea usando llaves y dentro de ellas se sitúa cada pareja clave - valor con dos puntos entre la clave y el valor, utilizando una coma para separar cada pareja. Para acceder a cada valor se debe hacer referencia a la clave:

```
asignaturas={"Luis":"Lengua", "Marta":"Matematicas"}
print(asignaturas["Marta"])
```

Para añadir nuevas parejas clave - valor al diccionario sólo tenemos hacerlo como se ve en el ejemplo:

- Nombre_diccionario[clave]=valor asociado

```
asignaturas={"Luis":"Lengua","Marta":"Matematicas"} asignaturas["Jorge"]="Programación"
```

Así mismo, podemos eliminar cualquier clave - valor del diccionario con la siguiente instrucción:

- del Nombre_diccionario[clave]

Ordenar un diccionario es complicado y no tiene sentido hacerlo ya que se trata de una estructura de almacenamiento a la que se accede mediante una clave para que Python muestre el valor asociado. La posición que ocupa la clave no es muy relevante.

Los diccionarios también son OBJETOS, así que tienes a tu disposición un conjunto de funciones (métodos) que ya están programadas y sólo se tienen que "invocar" para obtener lo que se desea hacer.

Métodos de diccionario:

- clear() Elimina todos los elementos del diccionario
- copy() Devuelve una copia del diccionario
- fromkeys() Devuelve un diccionario con las claves y valores especificados
- get() Devuelve el valor de la clave especificada
- items() Devuelve una lista que contiene una tupla para cada par de valores clave
- keys() Devuelve una lista que contiene las claves del diccionario
- pop() Elimina el elemento con la clave especificada
- popitem() Elimina el último par clave-valor insertado
- update() Actualiza el diccionario con los pares clave-valor especificados
- values() Devuelve una lista de todos los valores en el diccionario

Hay que recordar que los métodos se llaman con el nombre del objeto particular que se ha creado, un punto, y el método en cuestión con los parámetros adecuados.

Control de errores

En Python a los errores se les llama **EXCEPCIONES**.

Antiguamente, cuando en un programa producía un error, el programa se interrumpía bruscamente. Los errores podían ser muy variados: se había producido una división entre cero, se había intentado acceder a un archivo que no existía, etc.

En muchos lenguajes modernos tenemos la posibilidad de contar con que pueden producirse esos errores y hacer que el programa responda inteligentemente a ellos, sin interrumpirse. Para manejar los errores se "envuelve" el código "peligroso" en un bloque precedido por **try** y controlar el error con el comando **except**

```
try:
    num=input("Dime un número: ")
    num=num+2
except:
    print("Error")

print("Fin")
```

Si no controlara este error la palabra **Fin** nunca se imprimiría debido al error de la línea 3

▼ P7: Los Destinos

El objetivo de este ejercicio es realizarlo de dos formas, por un lado usando listas tal y como se describe en el desarrollo de abajo y por otro lado hacer exactamente lo mismo pero en vez de usar las dos listas **usar un diccionario**

Posteriormente comprobar su efectividad usando el comando Timer() para medir ambos tiempos de ejecución

Ejercicio

Crear dos Listas:

- Que contenga nombres de ciudades (Destinos)
- Los códigos de esos destinos, único para cada ciudad. ¿Puedes hacerlos alfanuméricos?

Nota para aprobar la practica

- El programa te debe preguntar el destino y te debe dar el código único de esa ciudad
- Al principio el programa se puede realizar con una lista creada de forma manual con 5 destinos, posteriormente el programa debe ser realizado **y funcionar** con una lista de ciudades reales. (adjuntado en el ejercicio)
- La lista de códigos para los destinos reales **si debe crearse por vosotros (inventada)**
- La creación del diccionario para la segunda parte del ejercicio tambien debe ser creado por vosotros partiendo de las dos listas de los destinos reales.
- Controlar todos los posibles errores

Nivel 1

- Con un bucle que te vuelva a pedir el vuelo hasta que escribas la palabra "exit" donde el programa acabará.

Nivel 2

- Si el vuelo no es correcto o no existe, te debe indicar un mensaje de "**Vuelo incorrecto**"

Nivel 3

- Que el programa funcione igual en orden inverso, es decir poner el destino y te de el número de vuelo (en el mismo programa)

Nivel 4

- Comparar y valorar la eficacia de los dos algortimos (2 listas y diccionarios) y certificar cual puede ser más productivo en la realización de una busqueda.