

U3 -

Diccionarios

Indice

Diccionarios en Python

La clase dict de Python es un tipo mapa que asocia claves a valores. A diferencia de los tipos secuenciales list, tuple, range o str, que son indexados por un índice numérico, los diccionarios son indexados por claves. Estas claves siempre deben ser de un tipo inmutable, concretamente un tipo hashable.

Nota sobre Hashable

En principio, los objetos que son instancias de clases definidas por el usuario son hashables. También lo son la mayoría de tipos inmutables definidos por Python (int, float o str).

Pares: Clave - Valor

Piensa siempre en un diccionario como un contenedor de pares clave: valor, en el que la clave puede ser de cualquier tipo hashable y es única en el diccionario que la contiene. Generalmente, se suelen usar como claves los tipos `int` y `str` aunque, como se ha comentado, cualquier tipo hashable puede ser una clave.

Operaciones Principales

Las principales operaciones que se suelen realizar con diccionarios son almacenar un valor asociado a una clave y recuperar un valor a partir de una clave. Esta es la esencia de los diccionarios y es aquí donde son realmente importantes. En un diccionario, el acceso a un elemento a partir de una clave es una operación realmente rápida, eficaz y que consume pocos recursos si lo comparamos con cómo lo haríamos con otros tipos de datos.

Características Adicionales

- Es un tipo mutable, es decir, su contenido se puede modificar después de haber sido creado.
- Es un tipo ordenado. Preserva el orden en que se insertan los pares clave: valor.
- Es un tipo eficiente. El acceso a un elemento a partir de una clave es muy rápido.

Diccionarios en Python

Cómo crear un diccionario

En Python hay varias formas de crear un diccionario.

Las veremos todas a continuación.

La más simple es encerrar una secuencia de pares clave:
valor separados por comas entre llaves:

Diccionarios en Python

Cómo crear un diccionario (cont.)

Para crear un diccionario vacío, simplemente asigna a una variable el valor {}.

Diccionarios en Python

Cómo acceder a los
elementos de un diccionario
en Python

El método `get()`

La clase `dict` también ofrece el método `get (clave [, valor por defecto])`. Este método devuelve el valor correspondiente a la clave `clave`. En caso de que la clave no exista no lanza ningún error, sino que devuelve el segundo argumento `valor por defecto`. Si no se proporciona este argumento, se devuelve el valor `None`.

```
d = {'uno': 1, 'dos': 2, 'tres': 3}
print(d.get('uno')) # Output: 1
```

Diccionarios en Python

Cómo recorrer un diccionario – For dict Python

Hay varias formas de recorrer los elementos de un diccionario: recorrer solo las claves, solo los valores o

Diccionarios en Python

Añadir elementos a un diccionario en Python

Como hemos comentado, la clase dict es mutable, por lo que se pueden añadir, modificar y/o eliminar

Añadir elementos mediante el método `setdefault()`

También existe el método `setdefault(clave[, valor])`. Este método devuelve el valor de la clave si ya existe y, en caso contrario, le asigna el valor que se pasa como segundo argumento. Si no se especifica este segundo argumento, por defecto es `None`.

```
d = {'uno': 1, 'dos': 2}
print(d.setdefault('uno', 1.0)) # Output: 1
```

Diccionarios en Python

Modificar elementos de un diccionario

En el apartado anterior hemos visto que para actualizar el valor asociado a una clave, simplemente se asigna un

Diccionarios en Python

Eliminar un elemento de un diccionario en Python

En Python existen diversos modos de eliminar un elemento de un diccionario. Son los siguientes:

Diccionarios en Python

Número de elementos (`len`)
de un diccionario en Python

Al igual que sucede con otros tipos contenedores, se puede usar la función de Python `len()` para obtener el

Diccionarios en Python

Comprobar si un elemento
está en un diccionario en
Python

Diccionarios en Python

Comparar si dos diccionarios son iguales

En Python se puede utilizar el operador de igualdad `==` para comparar si dos diccionarios son iguales. Dos

Diccionarios en Python

Diccionarios Anidados

Un diccionario puede contener un valor de cualquier tipo, entre ellos, otro diccionario. Esto se conoce como diccionarios anidados.

Para acceder al valor de una de las claves de un

Diccionarios en Python

Obtener una Lista de Claves

A veces, es necesario tener almacenado en una lista las claves de un diccionario. Para ello, simplemente pasa el diccionario como argumento del constructor `list()`.

```
d = {'uno': 1, 'dos': 2, 'tres': 3}
```

Diccionarios en Python

Objetos Vista de un Diccionario

La clase dict implementa tres métodos muy particulares, dado que devuelven un tipo de dato,

Diccionarios en Python

Listado de Métodos de la Clase dict

- `clear()`: Elimina todos los elementos del diccionario.

Diccionarios en Python

Depuración

Conforme trabajes con conjuntos de datos más grandes puede ser complicado depurar imprimiendo y revisando los datos a mano. Aquí hay algunas sugerencias para depurar grandes conjuntos de datos: