

Taller de Física Computacional

Estructuras de datos

Cristián G. Sánchez y Carlos J. Ruestes

2020

Estructuras de datos

Las estructuras de datos son la forma en que vemos y manipulamos esto:

```
55 fb 89 2c b8 b7 84 26 #00000000
b8 e0 15 04 31 45 e2 bb #00000001
44 a0 04 5a e6 d7 85 d2 #00000002
e1 6b 5d d3 a1 b9 f8 e6 #00000003
e3 6b 1d ff 39 06 c3 54 #00000004
2e 3d 4e 83 cf a4 3b 1d #00000005
54 94 45 42 81 6a b6 1d #00000006
3c 2b 75 bb 8c e9 2b d4 #00000007
```

de una forma más “amigable”.

Estructuras de datos

Datos en RAM

En la RAM los datos se representan en forma binaria, ordenados de acuerdo a una secuencia de *direcciones* que permiten al processador leerlos o escribirlos.

Estructuras de datos

Las estructuras de datos son formas de organización manejo y almacenamiento de los datos que permiten acceso y modificación *eficientes*. Son abstracciones dentro del lenguaje que se representan de alguna manera en la RAM que **en principio** no nos interesa.



Los tipos de datos (u objetos en general) en python pueden ser *mutables* o *inmutables*:

Objetos Inmutables

Son los objetos cuyo valor tienen un valor fijo, que no se puede cambiar una vez creados. Son inmutables, por ejemplo, los literales.

Objetos Mutables

Son los objetos que pueden cambiar su valor pero mantienen su identificación (`id()`)



La inmutabilidad es un requisito para que un objeto sea *hasheable*.

Función *Hash*

Una función *hash* es un mapa (en principio unívoco) entre datos de tamaño arbitrario a valores de tamaño fijo. El valor puede ser utilizado como índice en una tabla (*hashtables*).

- La palabra *hash* tiene muchas traducciones literales que dependen del contexto. La de contexto culinario (picar y mezclar) es la que probablemente da origen a su uso. No existe una palabra castellana exactamente equivalente y por tanto se utiliza sin traducir, algunos prefieren la palabra “resumen”.

Estructuras de datos en Python

Aparte de las estructuras de datos simples que vimos hasta ahora (enteros, punto flotante, complejos, cadenas, etc.) Python implementa (entre otras) las siguientes estructuras de datos de forma nativa:

Secuencias

- **tuple:** Secuencia inmutable usualmente utilizada para almacenar colecciones de elementos heterogéneos.
- **list:** Secuencia mutable usualmente utilizada para almacenar colecciones de elementos homogéneos.
- **range:** Secuencia inmutable de enteros ordenados, comunmente utilizada en bucles.



Conjuntos

- **set:** Colección no ordenada de objetos que deben ser *hasheables*, es mutable.
- **frozenset:** Un **set** inmutable.

Mapas

- **dict** Una *hashtable* que mapea un objeto *hasheable* a otro objeto arbitrario. Los diccionarios son mutables.

Hay otras estructuras de datos implementadas en la biblioteca estándar, en particular en el módulo *collections*, para más detalles ver la sección Data Types del manual de la biblioteca estándar.

- [Función Hash en Wikipedia](#)
- [Introduction to Algorithms, Cormen *et al.*](#)
- [The Art of Computer Programming](#)
- [Manual de referencia de Python](#)
- [Manual de la Librería estándar de Python](#)