

Secuencias

La secuencia es la estructura de datos más sencilla que existe.

Una estructura de datos es una forma de organizarlos para que un algoritmo complejo se convierta en trivial

Ponemos los datos uno detrás del otro y con eso hacemos que el *visitar* a cada uno de ellos, sin olvidarse de ninguno y en un cierto orden, sea algo trivial.

Por eso en el supermercado, piden que forme una fila para pagar: así es trivial atender a todos los clientes, en un cierto orden (el que haya llegado primero en este caso) y sin olvidarse de ninguno.

Por eso no solemos tener los libros así:



sino en filas, uno detrás del otro. ¿Donde será más fácil encontrar uno?



Secuencias en Python

Una *secuencia* es cualquier estructura de datos que nos permite tenerlos uno detrás de otro. La más sencilla es la **lista** (hay otras).

- Las secuencias tienen una longitud (`len`)
- Las secuencias se pueden recorrer
- En la secuencias se puede buscar un elemento

Listas

Las listas son un tipo de Secuencia.

Las listas en Python se representan como una secuencia de datos separados por comas y entre corchetes:

- `[1, 8, 93]`
- `[True]`
- `[[5,6], ["hola", "mundo"]]`
- `[]`
- `[True, 45, 'hola']`

Las listas pueden ser de cualquier cosa, incluso otras listas o estar vacías.

Cada lista es un valor de Python: se evalúan a sí mismos.

El tipo lista

Las listas son un tipo de Python:

- Tienen infinitas instancias
- Tienen operaciones en común:
 - tienen longitud (`len`)
 - se pueden recorrer
 - se pueden buscar
 - se puede seleccionar un elemento

Prueba a evaluar esto:

```
type([3, "hol"])
```

Ejercicio

- ¿Las cadenas son listas?
- ¿Las cadenas son secuencias?

Evalúa lo siguiente:

- `l1 = [[5,6], ["hola", "mundo"]]`
- `type(l1)`
- `len(l1)`
- `len([])`

Operaciones básicas sobre una secuencia

Habíamos dicho que eran 4:

- obtener su longitud
- obtener un elemento
- buscar en ellas
- recorrerlas

Longitud

Se obtiene con la función `len` de Python

In []:

```
l0 = len([[], [45, [56]]])
l1 = len("hola mundo")
```

Obtener un elemento

Se usa el operador `[]`, sabiendo que los **índices empiezan en 0** y llegan hasta su **longitud - 1**.

La cadena "hola" tiene longitud 4, su primer índice es el 0 y el último es el 3.

h	o	l	a
<hr/>			
0	1	2	3

Si intentas acceder más allá del último índices, te da un error.

In []:

```
"hola"[0]
```

Buscar en una secuencia

Para buscar en una secuencia, se utiliza el operador `in` que nos devuelve un booleano.

```
vowels = ["a", "e", "i", "o", "u"]  
"a" in vowels  
"w" in vowels
```

Recorrer una secuencia

Es muy común tener que recorrer una secuencia, pasando por todos los elementos. Hay varias formas de hacerlo y lo veremos más adelante.