

Listas en Python

Listas

¿Qué son las listas en Python?

Una lista en Python nos **permite almacenar gran cantidad de valores** (equivalente a los array en otros lenguajes de programación) en una única estructura, lo que nos permite tener los datos más agrupados, localizables y más manejables.

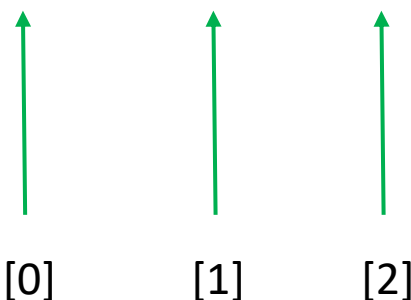
Normalmente en una lista se almacenan datos que tienen alguna relación entre sí, aunque no es imprescindible que así sea.

En Python, **las listas pueden guardar diferente tipo de valores** a diferencia de los array en otros lenguajes (para los que estén familiarizados con estas estructuras) donde los datos almacenados deben ser siempre del mismo tipo. En Python esa limitación no existe con las listas. Podemos guardar en una misma lista datos de tipo texto, numérico, objetos (se verán más adelante) booleanos etc.

Las listas en Python cuentan con numerosas ventajas respecto a otras estructuras de almacenamiento más restrictivas, como por ejemplo que **se pueden expandir dinámicamente** añadiendo nuevos elementos (otra novedad respecto a los arrays en otros lenguajes)

Sintaxis de las listas

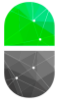
nombreLista=[elem1, elem2, elem3.....]



Los elementos almacenados en una lista tienen todos una posición o índice representado entre []. El índice comienza siempre en [0]. Es frecuente para el principiante confundir índice con nº de elemento así que cuidado con este detalle (sobre todo cuando abordemos en el futuro las estructuras de control de flujo).

Ejemplos de listas:

```
trabajadores=["Ana","Antonio", "María", "Pedro", "Laura"]
```



```
datosTrabajador=["Ana", "Gómez",1200, True]
```

Con el operador `+` puedes unir 2 listas entre sí:

```
trabajadoresEdificio1=["Ana","Antonio", "María", "Pedro", "Laura"]
trabajadoresEdificio2=["Jorge","Francisco", "Patricia", "Óscar",]
print (trabajadoresEdificio1+trabajadoresEdificio2)
```

La función **`append()`** permite agregar elementos a una lista:

```
trabajadores=["Ana", "Antonio", "María", "Pedro", "Laura"]
trabajadores.append("Juan")    # Juan será agregado
print(trabajadores)
```

La instrucción **`del`** permite borrar un elemento de la lista indicando el índice del elemento:

```
trabajadores=["Ana", "Antonio", "María", "Pedro", "Laura"]
del trabajadores[1]    # Antonio será eliminado
print(trabajadores)
```

La función **`remove()`** elimina un elemento indicando su valor:

```
trabajadores=["Ana", "Antonio", "María", "Pedro", "Laura"]
trabajadores.remove("Pedro")    #Pedro será eliminado
print(trabajadores)
```



La función **index()**, permite averiguar el índice de un elemento, algo muy útil con listas con muchos elementos:

```
trabajadores=["Ana", "Antonio", "María", "Pedro", "Laura"]  
  
print(trabajadores.index("Pedro"))  # imprime "3" en consola
```

La función **count()** nos permite saber cuántas veces aparece un mismo elemento en una lista:

```
trabajadores=["Ana", "Antonio", "María", "Pedro", "Laura", "Antonio"]  
  
print(trabajadores.count("Antonio"))  # imprime "2" en consola  
print (trabajadores.count("María"))   # imprime "1" en consola
```

Estas son quizás las instrucciones y métodos más frecuentes a la hora de trabajar con listas, pero hay más. Puedes consultar la documentación de Python para ver el resto.