

11. LISTAS

Es muy útil poder almacenar montones de información en un mismo lugar. Python utiliza un tipo de dato llamado "lista" para hacer esto.

Lo que aprenderás

- Qué son las listas
- Cómo se crean las listas
- Cómo se utilizan las listas mediante comandos

CREAR UNA LISTA SIMPLE

Una lista puede ser almacenada en una variable, acceder y modificar su contenido. Empieza por crear una lista de cosas que necesitamos comprar

Estas listas que contienen cadenas de caracteres se van a almacenar en una variable llamada **compras**. Una lista mantiene los artículos en orden.

```
compras = ['cerezas', 'queso', 'helado']  
print(compras)
```

La salida por pantalla mostraría:
cerezas
queso
helado

AGREGAR UN ELEMENTO AL FINAL DE LA LISTA

Podemos usar `append` para agregar un nuevo elemento al final de la lista ya existente de manera muy sencilla.

```
compras.append('banana')  
print(compras)
```

Python da a cada elemento de la lista un número de índice para identificarlo. Los números de índice empiezan siempre por cero e indican el orden de la palabra en la lista, lo que se conoce cómo **índice numerado**.

```
index: ['cerezas', 'queso', 'helado', 'banana']  
       0         1         2         3
```

TRABAJAR CON LISTAS

Puedes usar números de índice para cambiar, leer o eliminar un elemento de una lista. Para recuperar el artículo plátano de nuestra lista de compras deberás escribir el número del índice que marca este elemento en la lista.

Puedes usar un **bucle** para acceder a los elementos de la lista uno a uno y en orden.

```
print(compras[3])
```

El número 3 indica qué palabra de la lista se va a mostrar

```
for objeto in compras:  
    print(objeto)
```

Para mostrar todo los elementos de una lista, se puede hacer mediante el uso de un bucle

Para terminar con las listas vamos a ver otras posibilidades:

- Cambiar un ítem de una lista: con la siguiente instrucción cambiaremos lo que haya en la posición 3 de la lista por el término `mandarina`.

```
compras[3]='mandarina'
```

- Borrar un ítem: borraremos el elemento con posición 3 (recuerda que el primero siempre ocupa la posición 0).

```
del compras[3]
```

- Insertar un nuevo elemento en el orden que queramos: ponemos el elemento naranja en la posición 1.

```
compras.insert(1, 'oranges')
```