

# Estructuras de datos

Las estructuras de datos sirven para almacenar información y manipularla luego.

En Python tenemos tres tipos:

- Listas
- Tuplas
- Diccionarios

## Listas

Una lista es un conjunto ordenado de objetos. Por objetos entendemos cualquiera de los tipos de dato ya mencionados, incluso otras listas.

Para crear una lista, especificamos sus elementos entre corchetes y separados por comas.

```
In [1]: #Indicamos los elementos entre corchetes  
marcas = ["Toyota", "Ford", "Fiat", "Chevrolet"]  
marcas
```

```
Out[1]: ['Toyota', 'Ford', 'Fiat', 'Chevrolet']
```

Una lista es una colección ordenada, se puede acceder a cada uno de sus elementos indicando su posición comenzando por 0.

```
In [2]: marcas[2]
```

```
Out[2]: 'Fiat'
```

y podemos acceder a una porción de la lista desde el índice de inicio hasta el índice de fin indicados, en caso de ser omitido, se toma desde el extremo.

	+-----+	+-----+	+-----+	+-----+	
	Toyota	Ford	Fiat	Chevrlet	
	+-----+	+-----+	+-----+	+-----+	
	0	1	2	3	4
	-4	-3	-2	-1	0

```
In [3]: marcas[1:3]
```

```
Out[3]: ['Ford', 'Fiat']
```

```
In [4]: marcas[1:]
```

```
Out[4]: ['Ford', 'Fiat', 'Chevrlet']
```

Podemos modificar un elemento indicando su índice y asignando un nuevo valor:

```
In [5]: marcas[3] = "Chevrolet"  
marcas
```

```
Out[5]: ['Toyota', 'Ford', 'Fiat', 'Chevrolet']
```

```
In [6]: #Eliminar un elemento  
del marcas[0]  
#Insertar un elemento, se indica la posición y el contenido  
marcas.insert(1, "Volkswagen")  
#Agregar un elemento al final  
marcas.append("Renault")  
marcas
```

```
Out[6]: ['Ford', 'Volkswagen', 'Fiat', 'Chevrolet', 'Renault']
```

## Tuplas

Una tupla es una colección de elementos, pero a diferencia de las listas, las tuplas son **inmutables**.

```
In [7]: # Indicamos los elementos entre paréntesis.
lenguajes = ("Español", "Inglés", "Alemán", "Portugués")
lenguajes
```

```
Out[7]: ('Español', 'Inglés', 'Alemán', 'Portugués')
```

```
In [8]: lenguajes.insert(1,"Chino")
```

```
-----
AttributeError                                Traceback (most recent
Input In [8], in <module>
----> 1 lenguajes.insert(1,"Chino")

AttributeError: 'tuple' object has no attribute 'insert'
```

## Diccionarios

Los diccionarios, a diferencia de las listas y las tuplas, son colecciones **no ordenadas** de objetos. Además, sus elementos tienen una particularidad: siempre conforman un par clave-valor.

Para crear un diccionario, indicamos los pares clave-valor separados por comas y estos, a su vez, separados por dos puntos.

```
In [9]: automoviles = {"Toyota":"Etios", "Ford":"Focus", "Fiat":"Chronos"}
automoviles
```

```
Out[9]: {'Toyota': 'Etios', 'Ford': 'Focus', 'Fiat': 'Chronos', 'Chevrolet': 'Onix'}
```

Para acceder a un elemento, se debe indicar su clave.

```
In [10]: automoviles["Toyota"]
```

```
Out[10]: 'Etios'
```

```
In [11]: # Añadir un par clave/valor.
automoviles["Volkswagen"] = "Bora"
# Modificar uno existente.
automoviles["Fiat"] = "Uno"
# Eliminar.
del automoviles["Toyota"]
print(automoviles)
```

```
{'Ford': 'Focus', 'Fiat': 'Uno', 'Chevrolet': 'Onix', 'Volkswagen': 'Bora'}
```

# Tabla Resumen

	Tupla	Lista	Diccionario
<b>Definicion</b>	mi_tupla = ('texto', 20, 1275.48)	mi_lista = ['texto', 20, 127.48]	mi_dict = {'clave uno':'texto', 'clave dos':20, 'clave tres':1275.48}
<b>Obtener valores</b>	print mi_tupla[1] #imprime 20	print mi_lista[1] #imprime 20	print mi_dict['clave dos'] #imprime 20
<b>Modificar un valor</b>	<b>**No se puede**</b>	mi_lista[0] = 'modificado'	mi_dict['clave dos'] = 34

## Operaciones comunes

Veremos algunas operaciones útiles para trabajar con colecciones.

```
In [12]: # len() retorna el número de elementos en una colección  
len(marcas)
```

```
Out[12]: 5
```

```
In [13]: # para saber si un elemento está dentro de una colección, emplea  
"Ferrari" in automoviles ##En diccionarios actua sobre claves, r
```

```
Out[13]: False
```

# Ejercicios

1. Cargar este Notebook en GitHub y luego ejecutarlo en la plataforma <https://mybinder.org/> o similar.
2. Escribir un programa que almacene las asignaturas de una carrera (por ejemplo Programación, Ética, Base de datos, etc) en una lista y muestre por pantalla el listado.
3. Modificar el programa anterior para que se almacenen las notas que obtuvo en cada asignatura. ¿Qué estructura de datos es más apropiada para esta tarea?.

Nota: deben ejecutar desde el principio para que las celdas guarden los valores correspondientes en las variables.

Autor: Ing. Lisandro Lanfranco

## Fuentes:

<https://es.wikipedia.org/wiki/Python>

<https://www.pythoncheatsheet.org/>

<https://www.eugeniabahit.com/>

<https://tutorial.recursopython.com/colecciones/>