



Somos un **ecosistema** de desarrolladores de software



Listas []
Tuplas ()
Diccionarios {}

Listas []

```
lista = [7, 'hola', True, 3.5]  
#Almacena array, Objetos
```

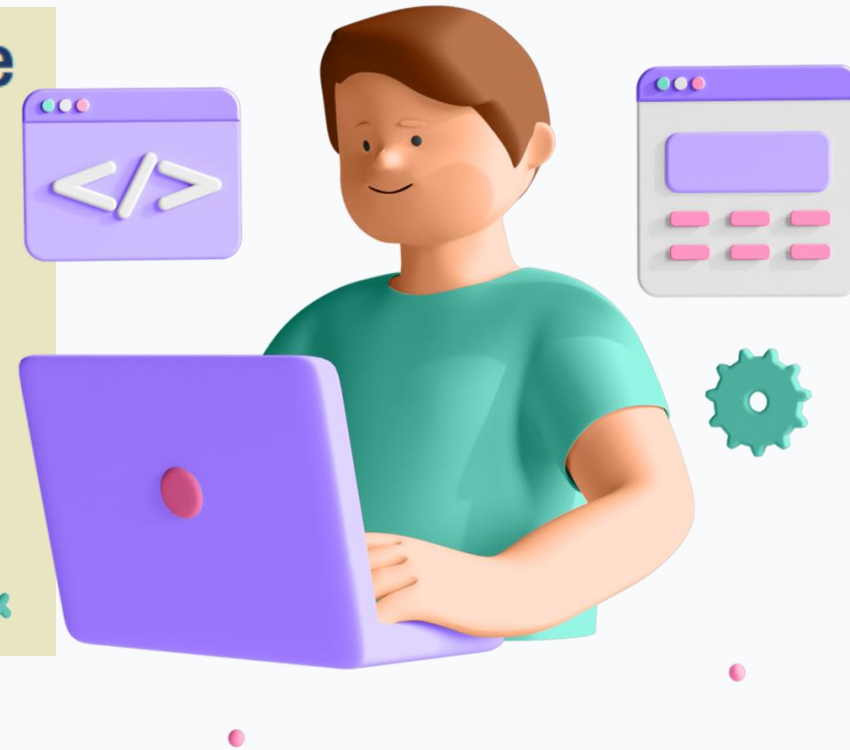
Valor	7	'Hola'	Ture	3.5
Índice	0	1	2	3

Listas []

Son conjuntos dinámicos de cualquier tipo de dato que se acceden desde un solo nombre a través de un índice.

Índice	0	1	2	3	4	5	6	7	8	9
Dato	Hola	5	45	Z	99	16	A	2	10	4

Imprimir dato en el índice 6



Listas []

```
lista = [7, 'hola', True, 3.5]  
print(lista)  
print(lista[2])
```

```
[7, 'hola', True, 3.5]  
True
```

Listas []

Las listas pueden ser modificadas agregando datos.

Índice	0	1	2	3	4	5	6	7	8	9
Dato	Hola	5	45	Z	99	16	A	2	10	4

Agregar dato en el índice 5:

Índice	0	1	2	3	4	5	6	7	8	9	10
Dato	Hola	5	45	Z	99	P	16	A	2	10	4

La lista crece.

las posiciones siguientes se desplazan hacia la derecha

Listas []

```
lista = [7, 'hola', True, 3.5]  
print(lista)  
lista.insert(2, 'manzana')  
print(lista)
```



```
[7, 'hola', True, 3.5]  
[7, 'hola', 'manzana', True, 3.5]
```

Listas []

También esta `.append` que agrega al final de la lista

```
lista = ["casa", "zapato", "manzana", "maleta"]  
print(lista)  
lista.append("celular")  
print(lista)
```

```
['casa', 'zapato', 'manzana', 'maleta']  
['casa', 'zapato', 'manzana', 'maleta', 'celular']
```


Listas []

También esta `.extend` amplía la lista agregando todos los elementos

```
lista = ["casa", "zapato", "manzana", "maleta"]  
lista2 = ["celular", "billetera"]  
lista.extend(lista2)  
print(lista)
```

```
['casa', 'zapato', 'manzana', 'maleta', 'celular', 'billetera']
```

Listas []

Las listas también pueden ser modificadas eliminando datos.

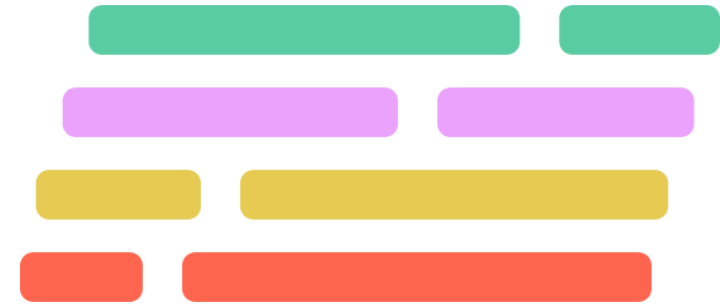
Índice	0	1	2	3	4	5	6	7	8	9
Dato	Hola	5	45	Z	99	16	A	2	10	4

Eliminar dato en el índice 5:

Índice	0	1	2	3	4	5	6	7	8
Dato	Hola	5	45	Z	99	A	2	10	4

La lista se reduce.
las posiciones posteriores se desplaza a la izquierda

Listas []



```
lista = [7, 'hola', True, 3.5]
print(lista)
del lista[1]
print(lista)
```

```
[7, 'hola', True, 3.5]
[7, True, 3.5]
```

Listas []

Eliminar el primer elemento de la lista cuyo valor sea igual a x

```
lista = ["casa", "zapato", "manzana", "maleta"]  
lista.remove("manzana")  
print(lista)
```

```
['casa', 'zapato', 'maleta']
```

Listas []

clear() lo utilizamos para borrar todos los elementos de la lista

```
lista = ["casa", "zapato", "manzana"]  
print(lista)  
lista.clear()  
print(lista)
```

```
['casa', 'zapato', 'manzana']  
[]
```

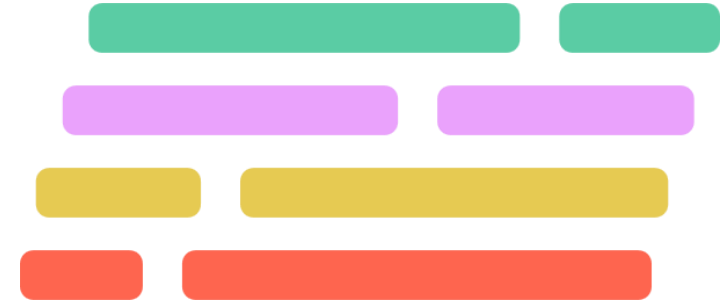
Listas []

Quitar un objeto en una posición dada dentro de la lista y lo retorna. Si no se especifica ningún índice, `a.pop()` elimina y devuelve el último elemento de la lista.



Listas []

```
lista = ["casa", "zapato", "manzana"]  
objt_eliminado = lista.pop(1)  
#si esta vacio el indice, eliminara el ultimo  
print(lista)  
print(objt_eliminado)
```



```
['casa', 'manzana']  
zapato
```

Listas []

Una de las formas de saber la cantidad de elementos que posee una lista es utilizando la función “len()”.

Índice	0	1	2	3	4	5	6	7	8	9
Dato	Hola	5	45	Z	99	16	A	2	10	4

El tamaño del arreglo es de 10 elementos

Listas []

```
lista = [7, 'hola', True, 3.5]  
print(len(lista))
```



4

Listas []

Las listas pueden ser tomadas en segmentos más pequeños de la misma.

Índice									
Dato	Hola	5	45	Z	99	16	A	2	10
	0	1	2	3	4	5	6	7	8

Imprimir datos desde el 4 hasta el 7 sin incluir el 7

Listas []

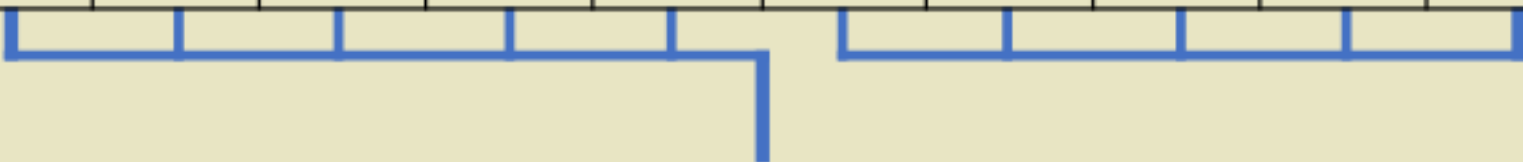
```
lista = [7, 'hola', True, 3.5, "casa",8]
print(lista)
print(lista[1:3])
```

```
[7, 'hola', True, 3.5, 'casa', 8]
['hola', True]
```

Listas []

Una manera de acceder a todos los datos de una lista uno por uno es utilizando el ciclo repetitivo “for”.

Índice	0	1	2	3	4	5	6	7	8	9
Dato	Hola	5	45	Z	99	16	A	2	10	4



Imprimir todos los datos de la lista uno por uno

Listas []

```
lista = [7, 'hola', True, 3.5, "casa", 8]
for objetos in lista:
    print(objetos)
```



```
7
hola
True
3.5
casa
8
```

Listas []

Las listas con datos numéricos (enteros o decimales), pueden ser ordenadas de manera ascendente.

Índice	0	1	2	3	4	5	6	7	8	9
Dato	2.5	3	4.5	5	1	1.5	2	3.5	5.5	4

Índice	0	1	2	3	4	5	6	7	8	9
Dato	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5

Listas []

```
lista = [2.3, 3, 4.5, 5, 1, 1.5, 2, 3.5, 5.5, 4]  
print(lista)  
lista.sort()  
print(lista)
```

```
[2.3, 3, 4.5, 5, 1, 1.5, 2, 3.5, 5.5, 4]  
[1, 1.5, 2, 2.3, 3, 3.5, 4, 4.5, 5, 5.5]
```

Las listas con caracteres y cadenas de caracteres también pueden ser ordenadas de manera ascendente por orden alfabético.

Índice	0	1	2	3	4
Dato	'Zapato'	'Sandalia'	'Armario'	'B'	'Espejo'

Índice	0	1	2	3	4
Dato	'Armario'	'B'	'Espejo'	'Sandalia'	'Zapato'

Listas []


```
lista = ["casa", "zapato", "manzana", "maleta"]  
print(lista)  
lista.sort()  
print(lista)
```

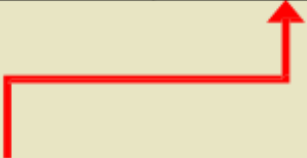
Listas []

```
['casa', 'zapato', 'manzana', 'maleta']  
['casa', 'maleta', 'manzana', 'zapato']
```

Listas []

Cuando se quiere ordenar una lista, es importante tener en cuenta que no se pueden mezclar caracteres y cadenas de caracteres con números, ya que Python no tiene establecido un criterio de comparación para ordenarlos.

Índice	0	1	2	3	4
Dato	'Zapato'	'Sandalia'	'Armario'	5	'Espejo'



El número 5 no puede compararse con las demás cadenas de caracteres:

</Be a
coder>