

ARRAYS

By: Guillermo Andres De Mendoza Corrales

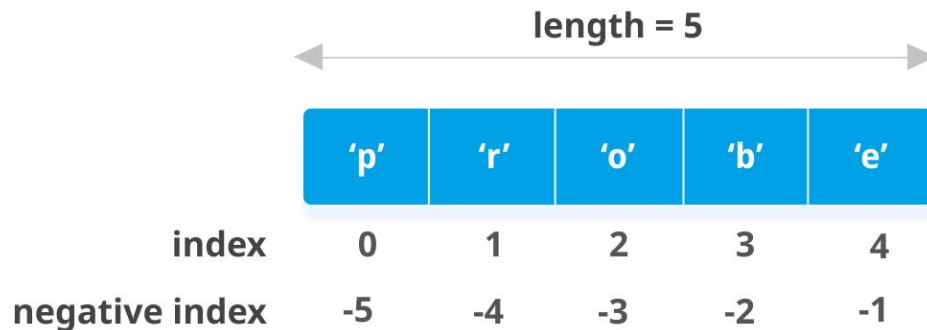


Theory

A **List** is a built-in, versatile data structure in used to store a collection of items in a specific sequence. It is defined as an **ordered**, **mutable**, and **heterogeneous** container.

Key Characteristics

- **Ordered:** Lists maintain the order of elements based on their index. The first item is at index `[0]`, the second at `[1]`, and so on.
- **Mutable:** Unlike tuples or strings, lists can be modified after they are created. You can add, remove, or change elements in place.
- **Heterogeneous:** A single list can contain elements of different data types, such as integers, strings, and even other lists simultaneously.
- **Dynamic:** Python lists are implemented as **dynamic arrays**. This means they automatically manage memory by over-allocating space (capacity) to ensure that adding new elements (appending) is efficient.



Instance a new list

empty

```
personas = []
```

starting elements

```
personas = ["A", "B", "C"]
```



Append

Add elements to the last position of the list

```
personas = []
```

```
personas.append("Mario")
```

```
personas.append("Luisa")
```

```
personas.append("Pedro")
```

```
personas.append("Miguel")
```

```
personas.append("Pablo")
```

```
print(personas)
```

```
['Mario', 'Luisa', 'Pedro', 'Miguel', 'Pablo']
```



Get list data

Access to elements by positive or negative index

```
personas = []
```

```
personas.append("Mario")  
personas.append("Luisa")  
personas.append("Pedro")  
personas.append("Miguel")  
personas.append("Pablo")
```

```
print(personas)  
print(personas[3])
```

```
['Mario', 'Luisa', 'Pedro', 'Miguel', 'Pablo']  
Miguel
```

```
personas = []
```

```
personas.append("Mario")  
personas.append("Luisa")  
personas.append("Pedro")  
personas.append("Miguel")  
personas.append("Pablo")
```

```
print(personas)  
print(personas[-1])
```

```
['Mario', 'Luisa', 'Pedro', 'Miguel', 'Pablo']  
Pablo
```



Insert

Add elements to an index position of the list

```
personas = []

personas.append("Mario")
personas.append("Luisa")
personas.append("Pedro")
personas.append("Miguel")
personas.append("Pablo")

print(personas)

personas.insert(2,"Guillermo")

print(personas)

['Mario', 'Luisa', 'Pedro', 'Miguel', 'Pablo']
['Mario', 'Luisa', 'Guillermo', 'Pedro', 'Miguel', 'Pablo']
```



Size

getting the number of elements in the list

```
personas = ["A","B","C"]  
  
print(len(personas))  
  
personas.append("C")  
  
print(len(personas))
```

3

4



Delete by element

Existing element in list

```
personas = ["Mario", "Luisa", "Pedro", "Miguel", "Pablo"]
```

```
personas.remove("Miguel")
```

```
print(personas)
```

```
['Mario', 'Luisa', 'Pedro', 'Pablo']
```

Non existing element in list

```
personas = ["Mario", "Luisa", "Pedro", "Miguel", "Pablo"]
```

```
personas.remove("Paloma")
```

```
print(personas)
```

ValueError

Traceback (most recent call last)

Cell **In[11]**, line 3

```
1 personas = ["Mario", "Luisa", "Pedro", "Miguel", "Pablo"]
```

```
----> 3 personas.remove("Paloma")
```

```
4 print(personas)
```

ValueError: list.remove(x): x not in list

[Fix Code](#)



Delete by index

```
personas = ["Mario", "Luisa", "Pedro", "Miguel", "Pablo"]
```

```
personas.pop(3)
```

```
print(personas)
```

```
['Mario', 'Luisa', 'Pedro', 'Pablo']
```



Ask for element in list

```
personas = ["Mario", "Luisa", "Pedro", "Miguel", "Pablo"]  
  
print("Mario" in personas)  
print("Paloma" in personas)
```

True

False

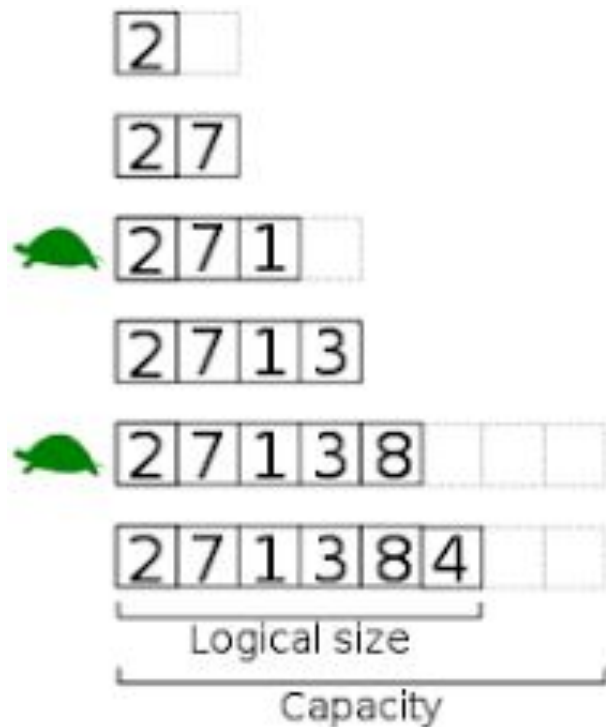


list resizing

List resizing involves dynamically adjusting a list's storage (capacity) as elements are added or removed, often automatically by creating a new, larger underlying array and copying data

Increase in python by $\sim 1.125x$ plus a small extra constant value:

0, 4, 8, 16, 24, 32, 40, 52, 64...



list resizing

```
import sys

lista = []
base_list_size = sys.getsizeof([])
pointer_size = 8 # bytes

print(f"{'Elementos':<10} | {'Memoria (Bytes)':<15} | {'Capacidad':<10}")
print("-" * 40)

for i in range(30):
    tamaño_lista = sys.getsizeof(lista)
    capacidad = (tamaño_lista - base_list_size) // pointer_size
    print(f"{i:<10} | {tamaño_lista - base_list_size:<15} | {capacidad:<10}")
    lista.append(i)
```

Elementos	Memoria (Bytes)	Capacidad
0	0	0
1	32	4
2	32	4
3	32	4
4	32	4
5	64	8
6	64	8
7	64	8
8	64	8
9	128	16
10	128	16
11	128	16
12	128	16
13	128	16
14	128	16
15	128	16
16	128	16
17	192	24
18	192	24
19	192	24
20	192	24
21	192	24
22	192	24
23	192	24
24	192	24
25	256	32
26	256	32
27	256	32
28	256	32
29	256	32



instance a list without resizing effects

```
[17]: n = 1000
      lista = [None] * n
```

```
[19]: lista
```

```
[19]: [None,
```

