Listas

- sintaxe: nome = [..., ...]
  - ↳ podem conter elementos de vários tipos
- referência por índice

```
friends = ["Kevin", "Karen", "Jim"]
print(friends[0] + "\n" + friends[1] + "\n" + friends[2])
```

- valores de índice negativos contam a partir do fim da lista

```
print(friends[-1]) → imprime o último elemento
```

- múltiplos índices

```
print(friends[1:]) → imprime do elemento de índice 1 (inclusivé), até ao último elemento da lista

print(friends[1:3]) → imprime do elemento de índice 1 (inclusivé), até ao elemento de índice 3 (exclusivé)
```

- modificar elementos

```
friends[1] = "Mike" → nova atribuição
```

List Functions

- extend: aumenta uma lista concatenando-lhe novos elementos

```
friends = ["Kevin", "Karen", "Jim", "Oscar", "Toby"]
lucky_numbers = [4, 8, 15, 16, 23, 42]
friends.extend(lucky_numbers)
print(friends)
["Kevin", "Karen", "Jim", "Oscar", "Toby", 4, 8, 15, 16, 23, 42]
```

- append: concatena um novo elemento a uma lista

```
friends.append("Creed")
friends.append(lucky_numbers[1])
```

- insert: insere um novo elemento, num determinado índice

```
friends.insert(1, "Kelly")
friends.insert(0, lucky_numbers[3])
```

- remove: remove um elemento da lista

```
friends.remove("Jim") → por nome
friends.remove(friends[0]) → por índice
```

- clear: esvazia uma lista

```
lucky_numbers.clear()
[]
```

- pop: remove último elemento da lista

```
friends.pop()
```

- index: se o seu argumento pertencer à lista, devolve o seu índice, caso contrário gera um erro

```
print(friends.index("Kevin"))
0 → é o primeiro elemento da lista
```

- count: conta o número de ocorrências do seu argumento na lista

```
print(friends.count("Jim")
```

- sort: ordena alfanumericamente a lista no sentido ascendente. Não consegue comparar strings a valores numéricos

```
friends.sort()
print(friends)
```
} print(friends.sort()) não funciona

- reverse: ordena alfanumericamente no sentido descendente. Não consegue comparar strings a valores numéricos

```
friends.reverse()
print(friends)
```
} tal como o sort, print(friends.reverse()) não funciona

- copy: torna uma lista numa cópia de outra. Declarar que uma lista é igual à outra parece ter o mesmo resultado

```
friends2 = friends.copy()
```
parece ter o mesmo resultado que
```
friends2 = friends
```