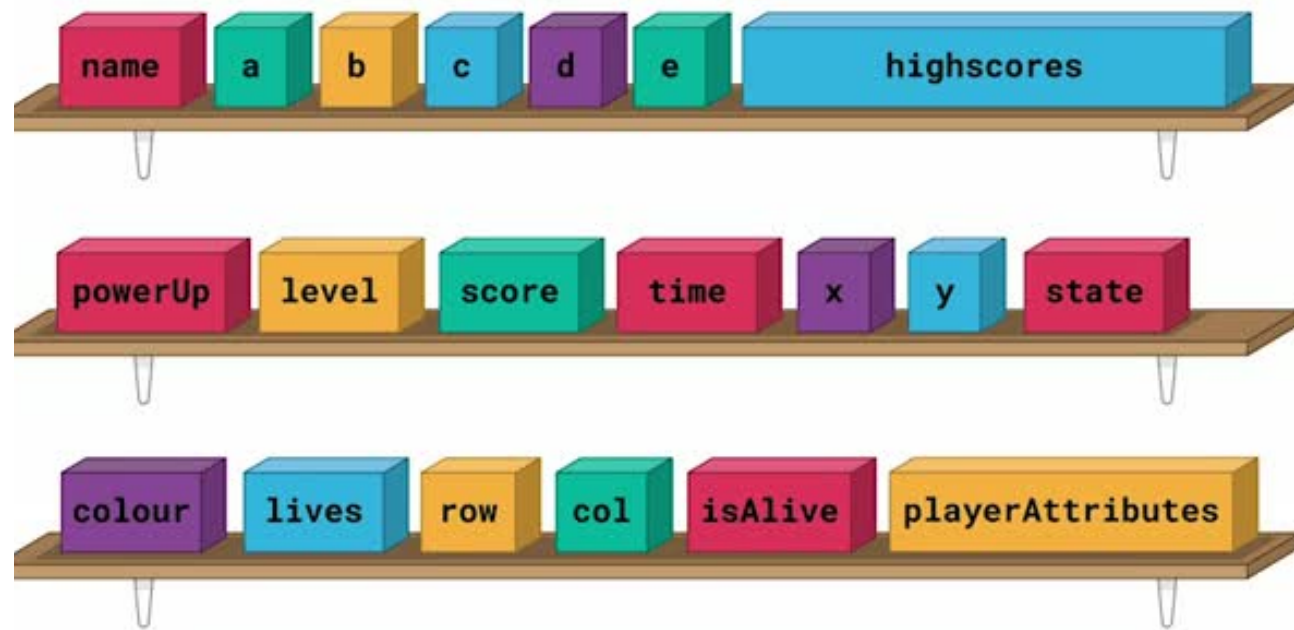


Listák és tuplek



Listák

- probléma: hogy tároljak nagyobb mennyiségű adatot? Az összes telefonszámot, összes könyvcímet?
- egy egyszerű változóban csak 1 dolog fér el
- megoldás: adatstruktúrák - több egymással összefüggő adatot tároljunk
- egyike a **lista**

Mire képes egy lista?

- egy listában képesek vagyunk tárolni adatokat, tudva a sorrendjüket, nem kell hogy azonos típusú legyen. Iterálható. FONTOS : tudjuk őket folyamatosan változtatni.

```
myList = ["tim", 43, 7.0, True]
fruits = ["apple", "pear", "orange"]
numbers = [1, 4, 6, 77, -8]
```

elemekhez való hozzáférés - indexelés

- az elemekhez [] - el férünk hozzá, beleírjuk az elem indexét, ami a helye 0-tól számolva. pl.: a myList-ben a 7.0 2-es indexű tehát `myList[2] = 7.0`

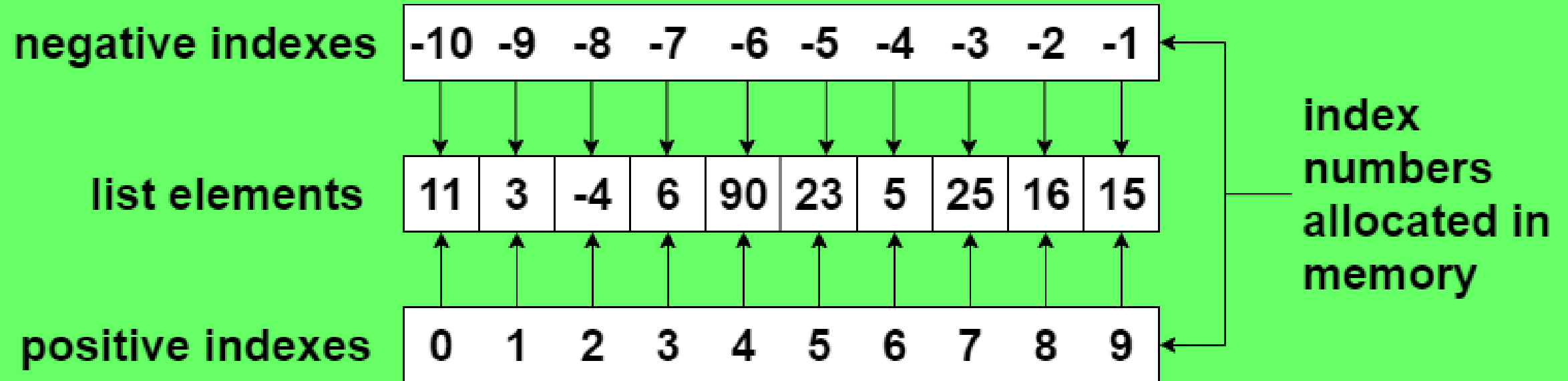
Python List

elements



index 0 1 2 3 4 5

Indexing in lists



adatok hozzáadása és törlése

- Ezek a metódusok mindig a lista után ponttal vannak hozzáírva

1. append - `list.append(elem)` - a lista végére teszi az elemet

2. insert - `list.insert(index, elem)` - az adott index elé teszi az elemet

3. [] - `list[index] = elem` - a lista index - 1 ik elemét átírja az elemre

```
fruits = ["apple", "pear", "orange"]

fruits[1] = "banana"

print(fruits)  # prints ["apple", "banana", "orange"]

fruits = ["apple", "pear", "orange"]

fruits.append("banana")

print(fruits)  # prints ["apple", "pear", "orange", "banana"]

fruits.insert(0, "mango")

print(fruits)  # prints ["mango", "apple", "pear", "orange", "banana"]
```

4. `remove(elem)` - kitörli az adott elemet a listából

5. `pop(index)` - kitörli az adott indexű elemet

```
# Using .remove()
fruits = ["apple", "pear", "orange"]

fruits.remove("apple")

print(fruits) # prints ["pear", "orange"]


# Using del
fruits = ["apple", "pear", "orange"]

fruits.pop(0)

print(fruits) # prints ["pear", "orange"]
```


Tuplek

- hasonlóak mint a lista, csak ha egyszer létrehozod, akkor nem tudod megváltoztatni őket. Iterálható.

```
nums = (1,4,5,6)
print(nums[0]) # prints 1

# NOT ALLOWED
nums[1] == 4
nums.append(2)
nums.remove(4)
nums.pop(3)
# These will cause errors
```

Iterálás

- ha egy adatstruktúra iterálható, akkor végig tudunk menni az elemein
- pl egy lista iterálható, mivel sorba tudunk menni egy elemein, egy string is, viszont egy integer nem
- ha egy ilyen struktúrát akarunk iterálni erre for loopot használunk

```
x = [4, "sas", 6, "Bob", 3.1415926]
```

```
for elem in x:  
    print(elem)
```

```
# 4  
# sas  
# 6  
# Bob  
# 3.1415926
```

- string iterálása

```
s = "infokör"
```

```
for c in s:  
    print(c)
```

```
# i  
# n  
# f  
# o  
# k  
# ö  
# r
```