

Python [Lists]

- 1.** Python Lists
- 2.** Accedir a valors
- 3.** Actualizar valors
- 4.** Inserir valors
- 5.** Eliminar valors
- 6.** Iterar valors
- 7.** Ordenar llistes
- 8.** Copiar llistes
- 9.** Concatenar llistes
- 10.** Lists methods

Python [Lists]

1. Python [Lists]



1. Python [Lists]

En Python té integrades **quatre tipus de col·leccions** de dades per a **emmagatzemar múltiples valors/objectes en una sola variable iterativa**: Llistes, Tuples, Sets i Diccionaris.

Les llistes **es comporten com un vector** dinàmic tal i com podem trobar en altres llenguatges.

1. Python [Lists]



1. Python [Lists]

- Les llistes es representen amb claudadors: `cars = ["Audi", "BMW", "Volkswagen"]`
- Les llistes son **indexades**.
- El primer element té **índex 0** i consecutivament.
- Les llistes son ordenades ja que accedim als valors usant un índex; però els seus valors es guardaran per ordre d'inserció.
- Els **valors** de les llistes poden ser actualitzats.
- Les llistes accepten **insercions, eliminacions i modificacions** dels seus ítems.
- Les llistes accepten **valors repetits**.
- Una llista **pot** contenir elements de diferent tipatge: `llista = ["abc", 34, True, 40, "male"]`

Accedir a valors

2. Python [Lists]: accedir a valors



2. Accedir a valors

- Donada una llista: **cars = ["Audi", "BMW", "Volkswagen"]**

Accedir	Python code
Un valor usant índex (error si 'out of bounds')	print(cars[2])
Un valor usant índex negatiu (últim element)	print(cars[-1])
Múltiples valors usant rang (tres options) *(inclou el primer índex del rang però no l'últim!) **(si obviem el final del rang, mostra fins al final)	print(cars[0:2]) / print(cars[:2]) / print(cars[1:])
Buscar un valor dins de la llista	if "BMW" in cars: print("Sí, hi ha un BMW!")

Actualizar valors

3. Python [Lists]: actualizar valors



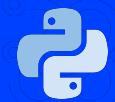
3. Actualitzar valors

- Donada una llista: **cars = ["Audi", "BMW", "Volkswagen"]**

Actualitzar	Python code
Un valor usant un índex Canviarà els elements cars[0] i cars[1] *(Recorda que l'índex final del rang no està inclòs)	<code>cars[1] = "Renault"</code>
Un rang d'elements <code>cars[0:2] = ["Renault", "Tesla"]</code>	
Canviar un element per dos *Canviem el valor "Audi" per "Citroën" i "Tesla" fent creixer la llista amb un element més. Es pot fer també l'acció inversa: canviar dos elements per uno.	<code>cars[0:1] = ["Citroën", "Tesla"]</code>

Inserir valors

4. Python [Lists]: inserir valors



4. Inserir valors

- Donada una llista: **cars = ["Audi", "BMW", "Volkswagen"]**

Inserir	Python code
Element(s) al final	<code>cars.append("Dacia")</code>
Element(s) a un índex	<code>cars.insert(2, "Peugeot")</code>
Una llista (o una otra estructura iterativa) al final	<code>cars.extend(["Lamborgini", "Lexus"])</code>

Eliminar valors

5. Python [Lists]: eliminar valors



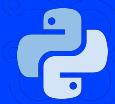
5. Eliminar valors

- Donada una llista: **cars = ["Audi", "BMW", "Volkswagen"]**

Eliminar	Python code
Un valor	<code>cars.remove("Audi")</code>
Un element per índex (dues options)	<code>cars.pop(2)</code> / <code>del cars[2]</code>
L'últim element	<code>cars.pop()</code>
Tots els valors	<code>cars.clear()</code>
Tots els valors i l'estructura de dades	<code>del cars</code>

Iterar valors

6. Python [Lists]: iterar valors



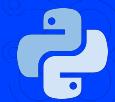
6. Iterar valors

Iterar	Python code
while	while i < len(cars): print(cars[i]) <i>i = i + 1</i>
for	for i in range(len(cars)): print(cars[i])
for-each	for c in cars: print(c)
for-enumerate	for i, c in enumerate(cars): print(i, "->", c)
List Comprehensive shorthand	[print(c) for c in cars]

<https://docs.python.org/2/tutorial/datastructures.html#looping-techniques>

Ordenar valors

7. Python [Lists]: ordenar valors



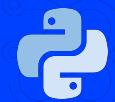
7. Ordenar valors

- Donada una llista: **cars = ["Audi", "BMW", "Volkswagen"]**

Ordenar	Python code
Numèrica o alfabèticament	<code>cars.sort()</code>
Descendentment (dues opcions)	<code>cars.sort(reverse = true)</code> / <code>cars.reverse()</code>
Alfabèticament case insensitive	<code>cars.sort(key = str.lower)</code>

Copiar llistes

8. Python [Lists]: copiar llistes



8. Copiar llistes

- Donada una llista: **cars = ["Audi", "BMW", "Volkswagen"]**
- A Python una llista no es pot copiar usant l'operador d'assignació '='. Això provoca una copia per referència ja que dues variables apuntaràn a la mateixa estructura de dades guardada en RAM, però no una copia de valors.
- Per crear una nova variable que contingui la mateixa estructura i valors que una llista donada, tenim dues opcions:

carsB = cars.copy()

carsC = list(cars)

Concatenar llistes

9. Python [Lists]: concatenar llistes



9. Concatenar llistes

- Donades dues llistes:
`cars = ["Audi", "BMW", "Volkswagen"]`
`motos = ["Vespa", "Honda", "Suzuki"]`
- Disposem de tres maneres de concatenar-les

`vehícules = cars + motos` "Crea una nova llista amb els valors de cars i motos "

`cars.extend(motos)` "Afegeix els valors de motos al final de cars "

`for c in motos:` "Afegeix els valors de motos al final de cars "
`cars.append(c)`

[Lists] methods

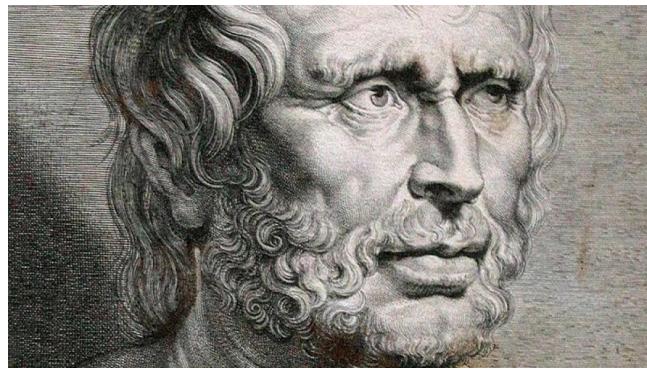
10. Python [Lists]: methods



10. [Lists] methods

Method	Description
<code>append()</code>	Adds an element at the end of the list
<code>clear()</code>	Removes all the elements from the list
<code>copy()</code>	Returns a copy of the list
<code>count()</code>	Returns the number of elements with the specified value
<code>extend()</code>	Add the elements of a list (or any iterable), to the end of the current list
<code>index()</code>	Returns the index of the first element with the specified value
<code>insert()</code>	Adds an element at the specified position
<code>pop()</code>	Removes the element at the specified position
<code>remove()</code>	Removes the item with the specified value
<code>reverse()</code>	Reverses the order of the list
<code>sort()</code>	Sorts the list

Font: https://www.w3schools.com/python/python_lists.asp



“Mediocre és aquell que exigeix molt als altres i poc
a sí mateix.”

Séneca (4 aC – 65 dC)

