

Listas



Concatenação

- ✓ Operador +
 - cria uma nova lista a partir de listas existentes
- ✓ Exemplo

```
l1 = [17, 10, 69]  
l2 = [2, 12, 98]  
l3 = l1 + l2  
print (l3)
```

[17, 10, 69, 2, 12, 98]

Replicação

✓ Operador *

- cria uma nova lista com cópias da lista especificada

✓ Exemplos

```
l1 = [17, 10, 69]
```

```
l4 = l1 * 3
```

```
print (l4)
```

[17, 10, 69, 17, 10, 69, 17, 10, 69]

```
l5 = [0] * 5
```

```
print (l5)
```

[0, 0, 0, 0, 0]

3

Pertinência

✓ Operador `in`

- verifica se um valor está contido em uma lista
- retorna `True` ou `False`

✓ Exemplos

```
l1 = [17, 10, 69]
```

```
l2 = [2, 12, [69, 98]]
```

```
print (69 in l1) → True
```

```
print (69 in l2) → False
```

```
print (10 not in l1) → False
```

```
print ([69, 98] in l2) → True
```

4

Fatiamento

- ✓ Cria uma nova lista com a fatia selecionada
 - **lista[a:b:n]**
 - seleciona uma fatia da posição **a** (inclusive) até a posição **b** (exclusive) de **n** em **n** elementos
 - se **a** não definido → considerado como zero
 - se **b** não definido → considerado como tamanho da lista
 - se **n** não definido → considerado como 1

5

Fatiamento

- ✓ Exemplos
- `l = [17, 10, 69, 2, 12, 98]`
- `print (l[:5])` → `[17, 10, 69, 2, 12]`
- `print (l[5:2])` → `[17, 69, 12]`
- `print (l[-1::-1])` → `[98, 12, 2, 69, 10, 17]`

6

Fatiamento e atribuição

- ✓ Exemplos

```
l = [17, 10, 69, 2, 12, 98]
```

```
l[0:2] = [13, 4]
```

```
print(l)
```

```
l[0:2] = ['oi', -9, 8752]
```

```
print(l)
```

```
l[0:4] = [-1]
```

```
print(l)
```

```
l[0:2] = []
```

```
print(l)
```

```
[13, 4, 69, 2, 12, 98]
```

```
['oi', -9, 8752, 69, 2, 12, 98]
```

```
[-1, 2, 12, 98]
```

```
12, 98]
```

7

Inserção de elementos

- ✓ Para inserir um ou mais elementos ao final de uma lista

```
lista = lista + [elementos]
```

- ✓ Exemplos

```
l = list()
```

```
l = l + [-27]
```

```
l = l + [6, 64]
```

```
print(l)
```

```
[-27, 6, 64]
```

```
l = l + [[6, 64]]
```

```
print(l)
```

```
[-27, 6, 64, [6, 64]]
```

8

Inserção de elementos

- ✓ Para inserir um elemento ao final de uma lista

```
lista.append (elemento)
```

- ✓ Exemplos

```
l = list()
l.append (-27)
l.append (6)
l.append (64)
print (l) → [-27, 6, 64]
```

```
l.append ([6, 64])
print (l) → [-27, 6, 64, [6, 64]]
```

9

Inserção de elementos

- ✓ Para inserir um elemento em determinada posição de uma lista

```
lista.insert(índice, elemento)
```

- ✓ Exemplos

```
l = ['b', 'e']
l.insert (0, 'a') → ['a', 'b', 'e']
l.insert (2, 'c') → ['a', 'b', 'c', 'e']
l.insert (-1, 'd') → ['a', 'b', 'c', 'd', 'e']
l.insert (len(l), 'f') → ['a', 'b', 'c', 'd', 'e', 'f']
l.insert (20, 'g') → ['a', 'b', 'c', 'd', 'e', 'f', 'g']
l.insert (1, ['w', 'z']) → ['a', ['w', 'z'], 'b', 'c', 'd', 'e', 'f', 'g']
```

10

Inserção de elementos

- ✓ Para inserir elementos da sequência ao final de uma lista

`lista.extend (sequência)`

- ✓ Exemplos

`l1 = ['a', 'b']`

`l2 = ['e', 'f']`

`l1.extend (l2)`

`l1.extend ('Oi')`

`l1.extend (range(5))`

[*a*, *b*, *e*, *f*] → [*a*, *b*, *e*, *f*, *O*, *i*] → [*a*, *b*, *e*, *f*, *O*, *i*, 0, 1, 2, 3, 4]

11

Remoção de elementos

- ✓ Para remover a 1^ª ocorrência de um elemento de uma lista

`lista.remove (elemento)`

- ✓ Exemplos

`l = ['a', 'b', 'c', ['b', 'c'], 'b', 'e']`

`l.remove ('b')`

`l.remove ('b')`

`l.remove ('b')`

→ [*a*, *c*, [*b*, *c*], *b*, *e*]

→ [*a*, *c*, [*b*, *c*], *e*]

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

12

Remoção de elementos

- ✓ Para remover o último elemento de uma lista

```
elemento = lista.pop ()
```

- ✓ Exemplos

```
l1 = ['a', 'c', ['b', 'c'], 'b', 'e']
```

```
l2 = []
```

```
valor1 = l1.pop ()
```

```
print (valor1)
```

```
valor2 = l2.pop ()
```

['a', 'c', ['b', 'c'], 'b']

'e'

IndexError: pop
from empty list

13

Remoção de elementos

- ✓ Para remover um elemento de uma lista

```
elemento = lista.pop (índice)
```

- ✓ Exemplos

```
l = ['a', 'c', ['b', 'c'], 'b', 'e']
```

```
valor = l.pop (2)
```

```
print (valor)
```

```
valor = l.pop (7)
```

['a', 'c', 'b', 'e']

['b', 'c']

IndexError: pop
index out of range

14

Remoção de elementos

- ✓ Para remover um elemento de uma lista

`del lista[índice]`

- ✓ Exemplos

`l = [0, 1, 2, 4, 3, 4, 5, 6, 7]`

`del l[3]` → [0, 1, 2, 3, 4, 5, 6, 7]

`del l[-1]` → [0, 1, 2, 3, 4, 5, 6]

`del l[8]` → IndexError: list assignment
index out of range

15

Remoção de elementos

- ✓ Para remover elementos de uma lista

`del lista[a:b:n]`

- ✓ Exemplos

`l = [0, 1, 2, 3, 4, 5, 6, 7]`

`del l[5:]` → [0, 1, 2, 3, 4]

`del l[0::2]` → [1, 3]

`del l[:]` → []

`del l` → Remove a variável l

16