

Comandos Essenciais de Listas, Tuplas e Dicionários em Python

LISTAS (mutáveis)

append(x): Adiciona um elemento no final da lista.

```
lista = [1, 2] lista.append(3) # [1, 2, 3]
```

insert(i, x): Insere o elemento x na posição i.

```
lista.insert(1, 'a') # [1, 'a', 2, 3]
```

remove(x): Remove o primeiro elemento igual a x.

```
lista.remove(2) # [1, 'a', 3]
```

pop([i]): Remove e retorna o elemento da posição i.

```
lista.pop(1) # Remove o 'a'
```

clear(): Remove todos os elementos da lista.

```
lista.clear() # []
```

index(x): Retorna o índice da primeira ocorrência de x.

```
lista = [1, 2, 3, 2] lista.index(2) # 1
```

count(x): Conta quantas vezes x aparece.

```
lista.count(2) # 2
```

sort(): Ordena a lista em ordem crescente.

```
lista.sort()
```

sort(reverse=True): Ordena em ordem decrescente.

```
lista.sort(reverse=True)
```

reverse(): Inverte a ordem dos elementos.

```
lista.reverse()
```

copy(): Cria uma cópia da lista.

```
nova = lista.copy()
```

extend(iterável): Adiciona vários elementos de outro iterável.

```
lista.extend([4, 5])
```

len(lista): Retorna o tamanho da lista.

```
len(lista)
```

max(lista): Retorna o maior valor.

```
max(lista)
```

min(lista): Retorna o menor valor.

```
min(lista)
```

sum(lista): Soma os valores numéricos.

```
sum([1, 2, 3]) # 6
```

list(range(início, fim, passo)): Cria lista com intervalo numérico.

```
list(range(1, 5)) # [1, 2, 3, 4]
```

del lista[i]: Deleta o elemento do índice i.

```
del lista[0]
```

in / not in: Verifica se o valor está ou não na lista.

```
'a' in lista
```

TUPLAS (imutáveis)

count(x): Conta quantas vezes x aparece.

```
tupla = (1, 2, 2, 3) tupla.count(2) # 2
```

index(x): Retorna o índice da primeira ocorrência.

```
tupla.index(3) # 3
```

len(tupla): Retorna o tamanho.

```
len(tupla)
```

max(tupla): Retorna o maior valor.

```
max(tupla)
```

min(tupla): Retorna o menor valor.

```
min(tupla)
```

sum(tupla): Soma valores numéricos.

```
sum(tupla)
```

tuple(lista): Converte lista em tupla.

```
tuple([1, 2, 3])
```

in / not in: Verifica se o elemento existe.

```
3 in tupla
```

sorted(tupla): Retorna uma lista ordenada com os elementos.

```
sorted(tupla)
```

+: Concatena tuplas.

```
(1, 2) + (3, 4) # (1, 2, 3, 4)
```

***:** Repete os elementos da tupla.

```
(1, 2) * 2 # (1, 2, 1, 2)
```

DICIONÁRIOS (chave: valor)

dict(): Cria um dicionário vazio.

```
dados = dict()
```

get('chave', padrão): Retorna valor ou padrão se não existir.

```
dados.get('nome', 'Sem nome')
```

keys(): Retorna todas as chaves.

```
dados.keys()
```

values(): Retorna todos os valores.

```
dados.values()
```

items(): Retorna pares chave-valor.

```
dados.items()
```

update({'chave': valor}): Atualiza ou adiciona um par.

```
dados.update({'idade': 20})
```

pop('chave'): Remove e retorna o valor da chave.

```
dados.pop('idade')
```

popitem(): Remove o último item adicionado.

```
dados.popitem()
```

clear(): Remove todos os itens.

```
dados.clear()
```

'chave' in dicionario: Verifica se a chave existe.

```
'nome' in dados
```

len(dicionario): Retorna quantidade de pares.

```
len(dados)
```

copy(): Cria uma cópia rasa do dicionário.

```
copia = dados.copy()
```

del dicionario['chave']: Remove uma chave específica.

```
del dados['nome']
```

for k, v in dicionario.items(): Percorre chaves e valores.

```
for k, v in dados.items(): print(k, v)
```

fromkeys(iterável, valor): Cria um dicionário novo com as mesmas chaves.

```
dict.fromkeys(['a', 'b'], 0)
```

setdefault('chave', valor): Adiciona chave se não existir.

```
dados.setdefault('idade', 18)
```

dict comprehension: Cria rapidamente um dicionário.

```
{x: x**2 for x in range(3)}
```