

Resumo Freq 2 2526

November 4, 2025

1 Resumo da Frequênci a 2 - 2025/26

Funções e conceitos essenciais para a segunda frequência de Programação:

- Listas
- Dicionários
- Tuplos
- Conjuntos
- Lambda

O documento incluirá todas as funções built-in relevantes para a frequência.

1.0.1 Listas

Método	Descrição
<code>append()</code>	Adiciona um elemento no final da lista
<code>clear()</code>	Remove todos os elementos da lista
<code>copy()</code>	Retorna uma cópia da lista
<code>count()</code>	Retorna o número de elementos com o valor especificado
<code>extend()</code>	Adiciona os elementos de uma lista (ou qualquer iterável) ao final da lista atual
<code>index()</code>	Retorna o índice do primeiro elemento com o valor especificado
<code>insert()</code>	Adiciona um elemento na posição especificada
<code>pop()</code>	Remove o elemento na posição especificada
<code>remove()</code>	Remove o item com o valor especificado
<code>reverse()</code>	Inverte a ordem da lista
<code>sort()</code>	Ordena a lista

Compreensão de Listas A compreensão de listas oferece uma sintaxe mais curta quando queres criar uma nova lista com base nos valores de uma lista existente.

Syntax:

```
nova_lista = [expressão for item in iterável if condição == True]
```

1.0.2 Tuplos

Método	Descrição
<code>count()</code>	Retorna o número de vezes que um valor especificado ocorre numa tupla
<code>index()</code>	Procura um valor especificado na tupla e retorna a posição onde foi encontrado

1.0.3 Dicionários

Método	Descrição
<code>clear()</code>	Remove todos os elementos do dicionário
<code>copy()</code>	Retorna uma cópia do dicionário
<code>fromkeys()</code>	Retorna um dicionário com as chaves e valor especificados
<code>get()</code>	Retorna o valor da chave especificada
<code>items()</code>	Retorna uma lista contendo uma tupla para cada par chave-valor
<code>keys()</code>	Retorna uma lista com as chaves do dicionário
<code>pop()</code>	Remove o elemento com a chave especificada
<code>popitem()</code>	Remove o último par chave-valor inserido
<code>setdefault()</code>	Retorna o valor da chave especificada. Se a chave não existir, insere a chave com o valor especificado
<code>update()</code>	Atualiza o dicionário com os pares chave-valor especificados
<code>values()</code>	Retorna uma lista com todos os valores do dicionário

1.0.4 Conjuntos

Método	Descrição
<code>add()</code>	Adiciona um elemento ao conjunto
<code>clear()</code>	Remove todos os elementos do conjunto
<code>copy()</code>	Retorna uma cópia do conjunto
<code>difference()</code>	Retorna um conjunto com a diferença entre dois ou mais conjuntos
<code>difference_update()</code>	Remove os itens deste conjunto que também estão presentes noutra conjunto especificado
<code>discard()</code>	Remove o item especificado
<code>intersection()</code>	Retorna um conjunto que é a interseção de dois outros conjuntos
<code>intersection_update()</code>	Remove os itens deste conjunto que não estão presentes noutras conjuntos especificados
<code>isdisjoint()</code>	Retorna se dois conjuntos têm ou não interseção
<code>issubset()</code>	Retorna True se todos os itens deste conjunto estão presentes noutra conjunto
<code>issuperset()</code>	Retorna True se todos os itens de outro conjunto estão presentes neste conjunto
<code>pop()</code>	Remove um elemento do conjunto
<code>remove()</code>	Remove o elemento especificado
<code>symmetric_difference()</code>	Retorna um conjunto com a diferença simétrica de dois conjuntos
<code>symmetric_difference_update()</code>	Atua sobre as diferenças simétricas entre este conjunto e outro
<code>union()</code>	Retorna um conjunto com a união dos conjuntos
<code>update()</code>	Atualiza o conjunto com a união deste conjunto e de outros

1.0.5 Lambda

Uma função lambda é uma função anónima pequena.

Pode receber qualquer número de argumentos, mas apenas pode ter uma expressão.

Sintaxe:

```
lambda argumentos : expressão
```