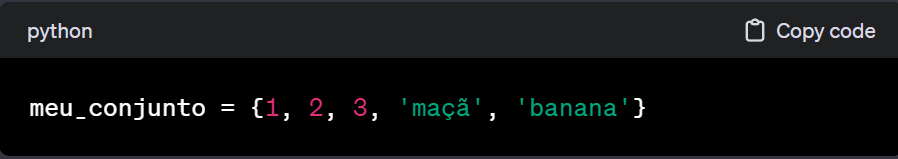
PYTHON sets

[Conjunto em Python](https://www.w3schools.com/python/python_sets.asp)

Um conjunto em Python é uma coleção não ordenada de elementos únicos. Ele é definido colocando os elementos entre chaves {}, separados por vírgulas.

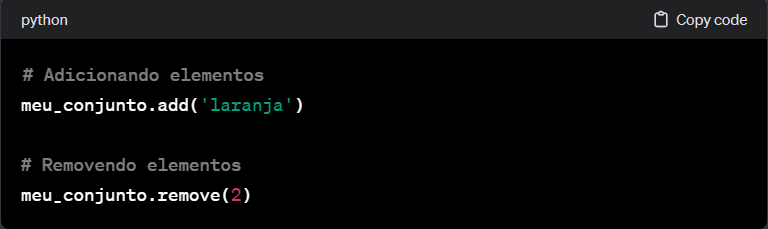
Conjuntos são semelhantes a listas e tuplas, mas não permitem elementos duplicados.

**CRIANDO CONJUNTO**

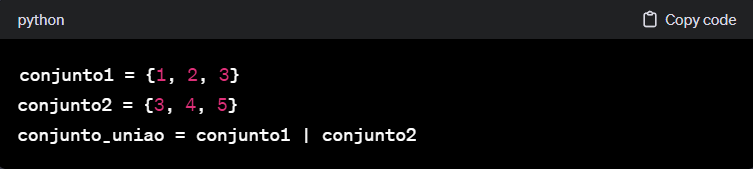
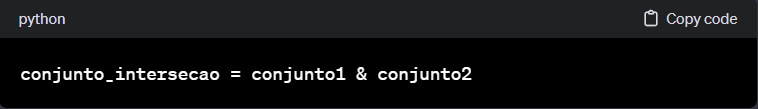
 Você pode criar um conjunto usando o construtor set( ) ou simplesmente colocando os elementos entre chaves { }

**ACESSANDO E MODIFICANDO**

Conjuntos são não ordenados, então eles não suportam indexação. No entanto, você pode iterar pelos elementos usando um loop. Você não pode modificar elementos individualmente, mas pode adicionar ou remover elementos.

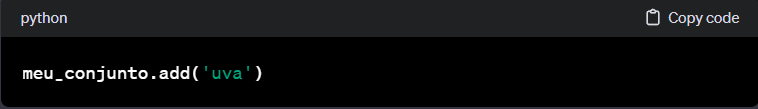


**OPERAÇÕES**

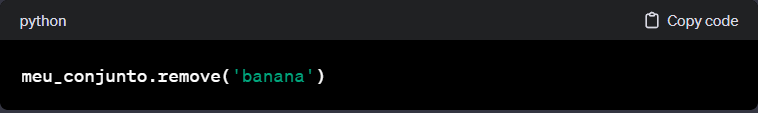
* **União (|):** Combina dois conjuntos, retornando um novo conjunto com todos os elementos únicos de ambos os conjuntos.
* **Interseção (&):** Retorna um novo conjunto com elementos comuns a ambos os conjuntos.
* **Diferença (-):** Retorna um novo conjunto com elementos do primeiro conjunto que não estão presentes no segundo conjunto.

[**MÉTODOS DE CONJUNTOS**](https://www.w3schools.com/python/python_sets_methods.asp)

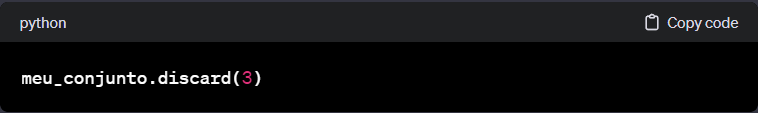
* **Add():** Adiciona um elemento ao conjunto.



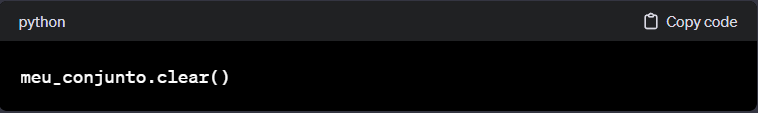
* **Remove():** Remove um elemento especificado do conjunto. Gera um erro se o elemento não estiver presente.



* **Discard():** Remove um elemento especificado do conjunto. Não gera um erro se o elemento não estiver presente.



* **Clear():** Remove todos os elementos do conjunto.



**QUANDO USAR CONJUNTOS**

* Quando você precisa armazenar uma coleção de elementos únicos;
* Quando a ordem não é importante;
* Quando você deseja realizar operações de conjuntos como união, interseção e diferença.

Conjuntos são particularmente úteis em cenários onde você precisa trabalhar com elementos únicos e realizar operações matemáticas de conjuntos de maneira eficiente.