

# transforme ■ se



# Estrutura de dados

## Objetivos:

- 1- Introduzir os conceitos de listas, tuplas e dicionários em Python
- 2- Explicar as características e diferença entre os três tipos de dados
- 3- Demonstrar as aplicações e benefícios de cada tipo de dado em diferentes situações

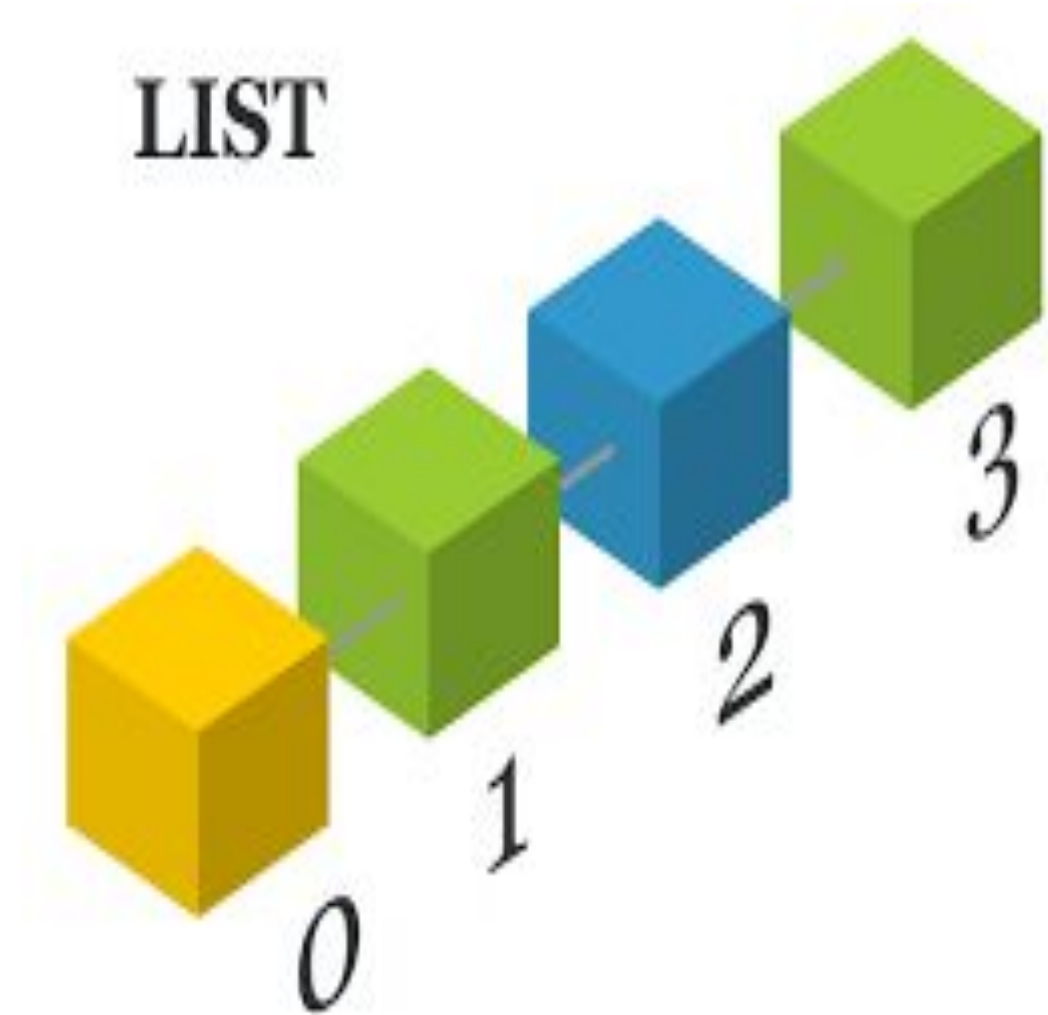
Lista

# Estrutura de dados

Uma lista é uma estrutura de dados que permite armazenar e organizar uma coleção de elementos.

Esses elementos podem ser de qualquer tipo, como números, strings, objetos ou até mesmo outras listas.

As listas são mutáveis, o que significa que você pode adicionar, remover e modificar elementos após sua criação.



# Exemplos práticos

# Tuplas

# Estrutura de dados

Uma tupla é uma sequência imutável de elementos

Esses elementos podem conter diferentes tipos de dados, como números, strings, booleanos e até mesmo outras tuplas

É semelhante a uma lista, mas a principal diferença é que uma tupla é imutável, o que significa que seus elementos não podem ser alterados depois de criados.



# Exemplos práticos



# Dicionários

# Estrutura de dados

Um dicionário é uma estrutura de dados poderosa e flexível que permite armazenar pares de chave-valor

É semelhante a um dicionário no mundo real, onde você pode procurar uma palavra (chave) e encontrar o seu significado (valor) associado.

Ao contrário das listas ou tuplas, que são indexadas numericamente, os dicionários são indexados por chaves únicas e imutáveis.



# Exemplos práticos

# Exercícios

# Estrutura de dados

1 - Crie uma lista chamada "frutas" contendo as seguintes frutas: maçã, banana, laranja, abacaxi e melancia. Em seguida, exiba a lista completa na tela.

2- Adicione a fruta "uva" à lista "frutas" utilizando o método `append()`. Exiba a lista atualizada na tela.

3- remova a fruta "banana" da lista "frutas" utilizando o método `remove()`. Exiba a lista atualizada na tela.

# Estrutura de dados

4- Acesse o segundo elemento da lista "frutas" e armazene-o em uma variável chamada "fruta\_selecionada". Em seguida, exiba o valor armazenado na variável.

5- Crie uma tupla chamada "cores" contendo as seguintes cores: vermelho, azul, verde, amarelo e roxo. Em seguida, exiba a tupla completa na tela

6- Acesse o terceiro elemento da tupla "cores" e armazene-o em uma variável chamada "cor\_selecionada". Em seguida, exiba o valor armazenado na variável.

# Estrutura de dados

7- Tente adicionar a cor "laranja" à tupla "cores" e observe o resultado. Explique o motivo pelo qual ocorreu um erro fazendo um comentário no código.

8- Crie um dicionário chamado "aluno" contendo as seguintes informações: nome, idade e cidade. Utilize as chaves "nome", "idade" e "cidade" para representar cada informação. Em seguida, exiba o dicionário completo na tela.

9- Acesse a idade do aluno no dicionário "aluno" e armazene-o em uma variável chamada "idade\_aluno". Em seguida, exiba o valor armazenado na variável

# Estrutura de dados

10- Adicione a informação do gênero do aluno ao dicionário "aluno" utilizando a chave "gênero" e o valor correspondente. Exiba o dicionário atualizado na tela.

11- Remova a informação da cidade do dicionário "aluno" utilizando o método `pop()` e a chave correspondente. Exiba o dicionário atualizado na tela.



# Conclusão

# Estrutura de dados

As listas são mutáveis, o que significa que você pode adicionar, remover e modificar elementos após sua criação.

As tuplas são úteis quando você precisa armazenar um conjunto de elementos que não devem ser modificados.

Os dicionários são amplamente utilizados em Python devido à sua capacidade de associar informações de forma flexível e eficiente

# transforme ■ se

O conhecimento é o poder  
de transformar o seu futuro.