



Na prática

A organização eficiente de informações é essencial em diversas áreas, inclusive no universo das bibliotecas. No contexto atual, com o crescimento das bibliotecas virtuais, a necessidade de estruturar e gerenciar grandes quantidades de dados torna-se ainda mais relevante.

Neste Na Prática, você irá entender como as estruturas de listas e tuplas em Python podem ser aplicadas para organizar uma biblioteca virtual.



BIBLIOTECA VIRTUAL

Uma livraria está precisando de um sistema de gerenciamento para a biblioteca virtual, que tem um acervo diversificado. A equipe precisa de uma **forma eficiente de organizar e acessar** essas informações.

Para a criação desse sistema, a livraria contratou a analista Erika, que realizou uma **análise de requisitos** do sistema de gerenciamento virtual da biblioteca.



Desenvolvimento

Para o desenvolvimento da biblioteca virtual, Erika optou por utilizar a linguagem de programação Python, com os recursos de **listas e tuplas**.

Ela começou com a criação de uma lista chamada `livros` para armazenar os títulos dos livros disponíveis na biblioteca. Cada título é representado por uma *string* dentro da lista.

Por exemplo:

```
1 livros = ["Dom Casmurro", "O Pequeno Príncipe", "1984", "O Senhor dos Anéis"]
```

Depois, Erika criou uma tupla chamada `autores` para armazenar as informações sobre os autores de cada livro. Cada tupla contém o nome do autor e o ano de nascimento.

Por exemplo:

```
2 autores = (("Machado de Assis", 1839), ("Antoine de Saint-Exupéry", 1900), ("George Orwell", 1903), ("J.R.R. Tolkien", 1892))
```

Com essas estruturas implantadas, ela pode facilmente adicionar novos livros ao acervo, inserindo títulos na lista `livros` e informações sobre os autores na tupla `autores`.

Por exemplo:

```
1 livros.append("Harry Potter")
2 autores.append(("J.K. Rowling", 1965))
```

Além disso, é possível buscar informações específicas, como o título de um livro ou o autor de uma obra em particular. Para isso, ela criou uma função de busca que percorre a lista de livros e retorna as informações do livro correspondente.

Por exemplo:

```
1 def buscar_livro(titulo):
2     for i in range(len(livros)):
3         if livros[i] == titulo:
4             return autores[i]
5     return "Livro não encontrado"
6
7 # Exemplo de utilização da função
8 titulo_busca = "1984"
9 informacoes_livro = buscar_livro(titulo_busca)
10 print(f"As informações do livro '{titulo_busca}' são: {informacoes_livro}")
```

Combinação



Uma **lista** é utilizada para **armazenar os títulos dos livros** devido à flexibilidade e capacidade de adicionar, remover e modificar elementos. A ordem dos títulos é preservada e é possível acessá-los por índice.

Uma **tupla** é usada para **armazenar as informações dos autores** devido à sua imutabilidade e eficiência em termos de espaço e processamento. As informações dos autores tendem a ser fixas e a imutabilidade da tupla protege os dados.





Anterior

Próximo



Organizando essas estruturas de dados, é possível criar soluções eficientes de gerenciamento e busca de informações. Com uma função simples, é possível **buscar informações específicas** de um livro.

