TinyDB

TinyDB é uma biblioteca com o objetivo de ser uma solução simples para banco de dados em aplicações pequenas que automaticamente atribui uma indexação para cada elemento adicionado.

Instalação:



Classes que serão importadas:



Métodos da biblioteca utilizados:

**Insert** **–** Adicionar novos dados



**Get –** Retornar um dado específico



**Update –** Atualizar dados já existentes



**Remove –** Remover dados



**All –** Retornar todos os dados de um banco

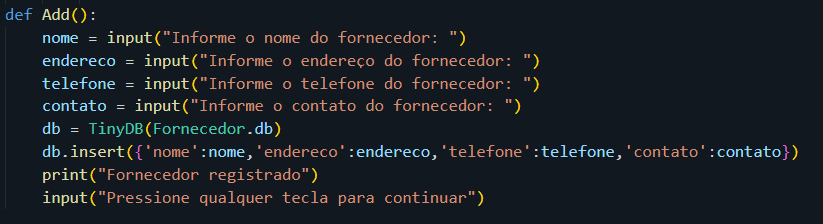


**Search –** Retornar múltiplos dados baseado em um filtro



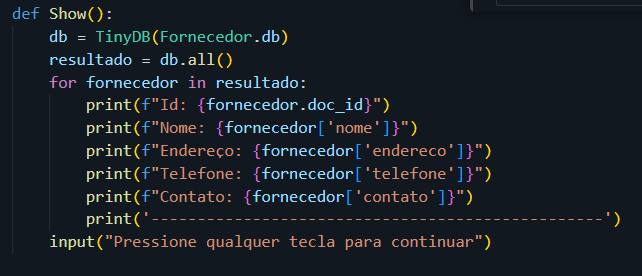
Métodos criados no projeto:

Adicionar novos fornecedores



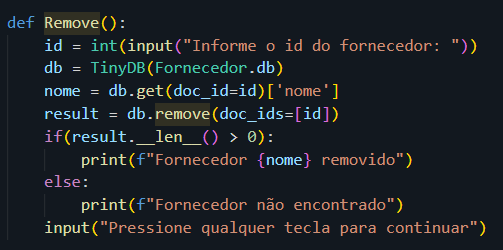
* Os dados são solicitados ao usuário
* Determinamos que *db* é igual ao nome do banco de dados que queremos inserir novos dados. Nesse caso *db = fornecedores.json*
* Usamos *db.insert* para adicionar os dados inseridos pelo usuário. Note que os dados do objeto devem ser passados no formato de dicionário.

Mostrar todos os fornecedores do banco de dados:



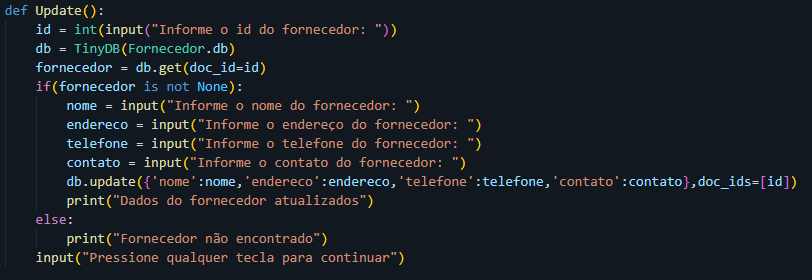
* Determinamos de qual banco de dados mostraremos as informações definindo a varável *db*
* Usamos o método *db.all()* para criar uma lista de dicionários e armazenar em *resultado*
* Fazemos um for loop dentro da variável *resultado* definindo *fornecedor* como cada item do loop
* Exibimos os dados de forma organizada no console. Note que as informações não retornam como objeto, retornam como um dicionário que deve ser chamado como *fornecedor[‘nome’]*

Remover um fornecedor do banco de dados:



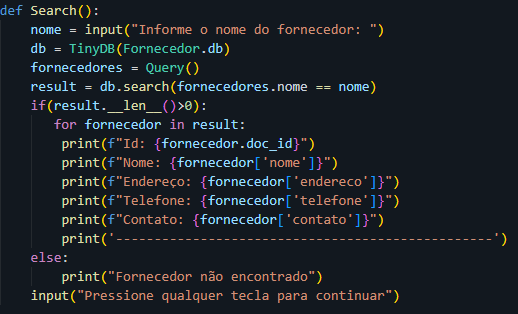
* Solicitamos o *id* do fornecedor que será removido
* Determinamos qual banco de dados será utilizado
* Usamos o *db.get(doc\_id=id)['nome']* para retornar o nome do fornecedor que será removido para confirmar que o removido será o selecionado
* Removemos pelo *id* usando *db.remove(doc\_ids=[id])*
* Confirmamos que algum fornecedor foi realmente removido usando *result.\_\_len\_\_() > 0*
* Caso funcione informamos qual fornecedor foi removido
* Se nenhum for removido informamos que o fornecedor não foi encontrado

Atualizar dados de um fornecedor já existente:



* Solicitamos o *id* do fornecedor
* Definimos o banco de dados em que será utilizado
* Buscamos pelo *id* e armazenamos as informações do fornecedor na variável *fornecedor*
* Verificamos se a variável *fornecedor* realmente possui alguma informação
* Caso *fornecedor* tenha informações, solicitamos as novas informações para o usuário
* Usamos *db.update()*, informando quais são os novos dados e qual o *id* do fornecedor a ser atualizado e confirmamos que o processo foi concluído
* Caso *fornecedor* não tenha nenhuma informação, informamos que o fornecedor não foi encontrado

Pesquisar fornecedores usando um filtro:



* Solicitamos o nome do fornecedor ao usuário
* Definimos o banco de dados em que será feita a pesquisa
* Atribuímos *Query()* a variável *fornecedores*, *Query* é a classe com a qual definimos qual filtro será utilizado na pesquisa
* Executamos a pesquisa no banco com *db.search(fornecedores.nome == nome)*. Nesse caso a pesquisa é feita pelo nome do fornecedor, mas poderia ser feita com qualquer outra informação, como endereço ou *id*
* Armazenamos o resultado na variável *result* que se torna uma lista de dicionários com as informações dos fornecedores que correspondem ao filtro
* Verificamos se *result* contém ao menos um fornecedor
* Caso *result* tenha ao menos um fornecedor, usamos um *for loop* para exibir as informações de todos os fornecedores em *result*
* Caso nenhum fornecedor tenha sido encontrado, o *for loop* não será executado e informamos que nenhum fornecedor foi encontrado