

Python Fundamentals



Construindo o Projeto

Anotações		

4LINUX Construindo o Projeto

Objetivos da Aula

- ✔ Criar o arquivo de conexão com o banco de dados.
- Criar script para interação de usuários.

Construindo	0	Proj	eto
-------------	---	------	-----

Nesta aula criaremos os arquivos responsáveis pela conexão com o MongoDB e a interação com o usuário.

Anotações			

Importando Módulos e criando conexão

Arquivo: banco.py

```
from pymongo import MongoClient, DESCENDING
   import time
2
   try:
       client = MongoClient()
       db = client['chat']
   except Exception as e:
6
       print('ERRO: {}'.format(e))
       exit()
```

Importando módulos e criando conexão com banco de dados

In [1]: do modulo pymongo, importamos mongoclient e descending.

In [2]: importamos o módulo time.

Anotações

In [3]: criando a conexão com o banco com um tratamento de exceção, caso a conexão falhe, será encerrada com script.

Função para cadastrar mensagens

Arquivo: banco.py

Função para cadastrar mensagens

In [8]: criamos a função de cadastro, esta, recebe dois parâmetros, nome e mensagem, armazenados em um dicionário, junto a hora de execução da função usando o modulo time. O dicionário é inserido no mongodb.

Anotações		

4LINUX Função para mostrar mensagens

Arquivo: banco.py

```
def select():
17
         ultimo = [x for x in db.chat.find().sort("_id", DESCENDING)]
18
         while True:
19
             date = [x for x in db.chat.find().sort("_id", DESCENDING)]
20
             if date != ultimo:
21
                 ultimo = date
22
23
                 print('[{}] {} : {} \n'.format(
                     date[0]['hora'], date[0]['nome'], date[0]['mensagem']))
24
             time.sleep(2)
```

Função para mostrar mensagens

In [17]: criamos a função select que busca o último registro do banco de dados, o armazena em uma variável, executa um loop infinito que refaz a busca no banco do último registro e o compara com a variável último, caso seja uma nova mensagem, retorna o valor de último, mostrando a mensagem, encerra com um sleep de 2 segundos.

Anotações			

4LINUX Criando o script

Arquivo: app.py

Script

In [1]: importamos o arquivo com funções do banco de dados.

In [2]: importamos threading para executar multiprocesso.

In [4]: verificamos o __name__ do arquivo garantindo que o script será executado somente por ele mesmo.

In [5]: atribuímos uma variável que recebe um usuário.

In [6]: criamos um tratamento de exceção que instancia uma thread para a função select, rodará em segundo plano em um loop infinito buscando últimas mensagens para mostrar na tela.

In [12]: criamos um segundo loop infinito que, enquanto a thread estiver rodando, recebe uma mensagem e cadastra no banco dados passando usuário e mensagem.

Δr	not:	ഹറ്	óes