Usos do Sistema

1 Introdução

Este sistema permite a personalização de exibição de textos usando a biblioteca `rich` no Python. Através de uma interface de linha de comando, o usuário pode escolher se o texto será fornecido diretamente ou se será lido a partir de um arquivo, além de selecionar qual módulo será utilizado para formatar a saída (como `painel`, `layout`, entre outros).

2 Requisitos

- Python 3.12 ou superior
- Biblioteca `rich` instalada no ambiente virtual

3 Instruções de Instalação

3.1 Criação e Ativação do Ambiente Virtual

- Abra o terminal e navegue até o diretório do projeto.
- Para criar e ativar o ambiente virtual, use os seguintes comandos:

  Para MacOS e Linux:

  ```
  python3 -m venv meuambiente
  source meuambiente/bin/activate
  ```

  Para Windows:

  ```
  python -m venv meuambiente
  meuambiente\Scripts\activate
  ```

3.2 Instalando as Dependências

Após ativar o ambiente, instale a biblioteca `rich`:

```
pip install rich
```

4 Estrutura do Projeto

No diretório principal do projeto, sua estrutura de diretórios será semelhante a esta:

```
exercicios/

          main.py
          personalizador/
                __init__.py
                layout.py
                painel.py
                progresso.py
                estilo.py
          arquivos/
                mensagem1.txt
                mensagem2.txt
                painel_mensagem.txt
```

- **personalizador/**: Contém os módulos do projeto (`layout`, `painel`, `progresso`, `estilo`).
- **arquivos/**: Contém os arquivos de texto para testes.

5 Usando o Sistema

O script principal `main.py` utiliza a biblioteca `argparse` para fornecer uma interface de linha de comando. O usuário pode passar o texto diretamente ou o caminho de um arquivo para ser formatado e exibido usando os módulos disponíveis.

5.1 Exemplo de Uso com Arquivos

Para exibir o conteúdo de um arquivo usando o módulo `painel`, execute o comando:

```
python main.py "arquivos/painel_mensagem.txt" -a -m painel -f
    mostrar_painel
```

Explicação:

- `"arquivos/painel_mensagem.txt"`: Caminho do arquivo de texto.
- `-a`: Indica que estamos lidando com um arquivo.
- `-m painel`: Escolhe o módulo `painel`.
- `-f mostrar_painel`: Seleciona a função `mostrar_painel`.

5.2 Exemplo de Uso com Texto Direto

Você também pode passar o texto diretamente sem a flag `-a`. Exemplo com o módulo `layout`:

```
python main.py "Exibindo texto no layout." -m layout -f
    mostrar_layout
```

5.3 Usando Vários Arquivos

Para exibir vários arquivos consecutivamente, execute:

```
python main.py "arquivos/mensagem1.txt" -a -m painel -f
    mostrar_painel
python main.py "arquivos/mensagem2.txt" -a -m layout -f
    mostrar_layout
```

Explicação:

- O primeiro comando exibe o conteúdo do arquivo `mensagem1.txt` usando o módulo `painel`.
- O segundo comando exibe o conteúdo do arquivo `mensagem2.txt` usando o módulo `layout`.

6 Argumentos da Linha de Comando

A interface de linha de comando possui os seguintes argumentos:

- **Argumento obrigatório**: Texto ou caminho do arquivo para exibição.
- `-a, --arquivo`: Ativa a exibição quando o argumento é um arquivo.
- `-m, --modulo`: Define o módulo a ser utilizado (`painel`, `layout`, `progresso`, `estilo`).
- `-f, --funcao`: Define a função específica do módulo para exibir o conteúdo (`mostrar_painel`, `mostrar_layout`, etc.).

7 Conclusão

Este sistema permite a personalização de exibição de textos e arquivos em diversos formatos com a biblioteca `rich`, facilitando a apresentação de conteúdos com layouts, painéis, barras de progresso e estilos personalizados. Use os comandos descritos acima para formatar textos da maneira desejada.