Em Python, a manipulação de arquivos é bastante direta e poderosa, permitindo ler, escrever, e atualizar arquivos de diferentes tipos, como texto (.txt), binário (.bin), e muitos outros. Aqui estão os fundamentos:

**Abrindo Arquivos**

Para abrir um arquivo, utilizamos a função open(), que retorna um objeto de arquivo. Ela recebe o nome do arquivo e o modo de abertura:



### Modos de Abertura

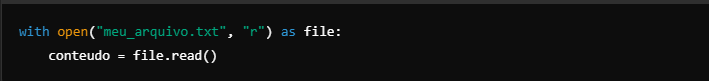
* **"r"**: leitura (read), gera um erro se o arquivo não existir.
* **"w"**: escrita (write), cria um novo arquivo se ele não existir, mas sobrescreve caso exista.
* **"a"**: acrescentar (append), abre o arquivo para adicionar conteúdo ao final dele.
* **"r+"**: leitura e escrita.

### Fechando Arquivos

Para liberar recursos, sempre feche o arquivo após usá-lo:



Ou use o bloco with, que fecha automaticamente o arquivo ao final:

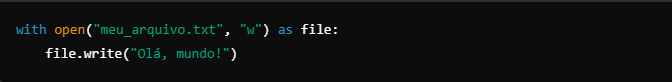


**Lendo Arquivos**

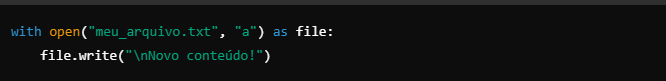
1. **file.read()**: lê todo o conteúdo do arquivo como uma única string.
2. **file.readline()**: lê linha por linha, útil para arquivos grandes.
3. **file.readlines()**: lê todas as linhas e as armazena em uma lista.

### Escrevendo e Acrescentando Dados

* **Escrita (write)**: substitui o conteúdo do arquivo.

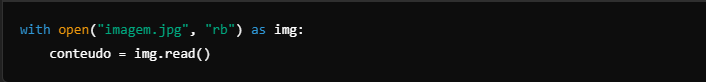


**Acrescentar (append)**: adiciona ao final sem apagar o conteúdo existente.

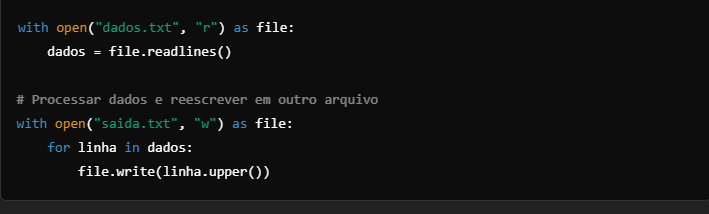


### Manipulando Arquivos Binários

Para arquivos binários (imagens, por exemplo), use modos como "rb", "wb", "ab".



Exemplo de Leitura e Escrita



### Dicas de Boas Práticas

1. **Utilize o bloco with** para garantir o fechamento do arquivo.
2. **Trate exceções** com try-except para lidar com possíveis erros de IO.
3. **Escolha modos de abertura apropriados** para evitar a perda de dados.