Para abrir um arquivo para leitura ou escrita em modo binário, basta usar os mesmos códigos de abertura aprendidos anteriormente, acrescidos de uma letra b. Assim, open('dados.dat', 'rb') abre o arquivo dados.dat para leitura em modo binário, assim como open('dados.dat', 'wb') o abre para escrita em modo binário. De maneira simples, isso quer dizer que as chamadas ao método read retornam strings de bytes, ao invés de strings de caracteres, e que, analogamente, as chamadas ao método write devem receber strings de bytes como parâmetro. O quadro abaixo lista os modos de abertura binários na primeira coluna, detalhando-os na segunda.

|  |  |
| --- | --- |
| **Modo** | **Descrição** |
| 'rb' | Abre para leitura binária, falhando se o caminho não existir. |
| 'wb' | Abre para escrita binária, apagando se o caminho existir e criando se não existir. |
| 'ab' | Abre para escrita binária, começando do final se o caminho existir e criando se não existir. |
| 'rb+' | Abre para leitura/escrita binária, falhando se o caminho não existir. |
| 'wb+' | Abre para leitura/escrita binária, apagando se o caminho existir e criando se não existir. |
| 'ab+' | Abre para leitura/escrita binária, começando do final se o caminho existir e criando se não existir. |

Existem diversos outros módulos dedicados à leitura e à escrita de arquivos. Conheça alguns listados a seguir:

* csv - Lê e escreve arquivos com extensão .csv, um formato texto usado para trocar dados entre diferentes editores de planilhas, como o Excel;
* json - Lê e escreve arquivos com extensão .json, um formato texto extensivamente usado na Internet e para trocar dados entre diferentes linguagens de programação modernas;
* struct - Permite controle fino sobre a [codificação](https://leadfortaleza.com.br/ead/glossary/Codifica%C3%A7%C3%A3o) e a de[codificação](https://leadfortaleza.com.br/ead/glossary/Codifica%C3%A7%C3%A3o) de números e strings ao nível dos bytes individuais;
* pillow - Serve para ler/escrever arquivos de imagens.