
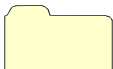





Manipulação de arquivos 

- Leitura e gravação de dados em arquivos
- Configurações (.ini), Textos (.txt)
- Posicionamento de ponteiro 
- Permissões de acesso
 - Acesso r/w nos diretórios em questão
 - Acesso r/w nos arquivos em questão
 - Grande parte dos problemas é falta de permissão

Abrindo arquivos 

- fopen

Sintaxe

```
(resource) fopen(caminho, modo);
```

Exemplo

```
$arquivo = fopen("teste.txt", "w+");
// Resultado: a referência do arquivo teste.txt será armazenada em $arquivo
```

Modos

r	Somente para leitura, colocando o ponteiro no começo do arquivo.
r+	Leitura e escrita, colocando o ponteiro no começo do arquivo.
w	Somente para escrita, colocando o ponteiro no começo do arquivo, criando o arquivo caso ele não exista.
w+	Leitura e escrita, colocando o ponteiro no começo do arquivo, criando o arquivo caso ele não exista.
a	Somente para escrita, colocando o ponteiro no final deste, criando o arquivo caso ele não exista.
a+	Leitura e escrita, colocando o ponteiro no final deste, criando o arquivo caso o mesmo não exista.
x	Cria e abre o arquivo somente para escrita, com o ponteiro no início, gerando um erro caso o arquivo já exista.
x+	Cria e abre o arquivo para leitura e escrita, com o ponteiro no início, gerando um erro caso o arquivo já exista.

Tamanho de arquivos

- **filesize**

Sintaxe

```
(int) filesize(caminho);
```

Exemplo

```
echo filesize("C:\teste.txt");  
// Resultado: tamanho em bytes do arquivo em questão
```

Importante

O caminho do arquivo deve ser relativo ao servidor onde o servidor HTTP com o PHP está sendo executado.

Lendo dados em arquivos

- **fread**

Sintaxe

```
(string) fread(arquivo, bytesASeremLidos);
```

Exemplo

```
$arquivo = fopen("teste.txt", "w+");  
$buffer = fread($arquivo, 20);  
echo $buffer;  
// Resultado: imprime na tela as 20 primeiras posições do arquivo  
  
$arquivo = fopen("teste.txt", "w+");  
$buffer = fread($arquivo, filesize("teste.txt"));  
echo $buffer;  
// Resultado: imprime na tela todo o conteúdo do arquivo
```

Importante

Leitura completa em um único comando para arquivos grandes pode impactar na performance do servidor

Lendo arquivos como arrays

- **file**

Sintaxe

```
(array) file(caminho);
```

Exemplo

```
$arquivo = file("teste.txt");  
  
for($i=0; $i<count($arquivo); $i++) {  
    echo "Linha ".$i.". ". $arquivo[$i];  
}  
// Resultado: imprime na tela o arquivo linha por linha
```

Gravando dados em arquivos



- **fwrite**

Sintaxe

```
(int) fwrite(arquivo, string[, bytesASeremGravados]);
```

Exemplo

```
$arquivo = fopen("teste.txt", "w+");  
fwrite($arquivo, "Texto gravado dentro do arquivo");  
// Resultado: grava no arquivo a string em questão  
  
fwrite($arquivo, "Mais informações");  
// Resultado: continua gravando no arquivo as demais informações
```

Fechando arquivos



- **fclose**

Sintaxe

```
(bool) fclose(arquivo);
```

Exemplo

```
$arquivo = fopen("teste.txt", "w+");  
fwrite($arquivo, "Texto gravado dentro do arquivo");  
fclose($arquivo);  
// Resultado: fecha o arquivo após utilizá-lo
```

Manipulação de diretórios



- **mkdir** (criação de diretórios)
- **rmdir** (exclusão de diretórios)

Sintaxes

```
(bool) mkdir(caminho[, modo]);  
(bool) rmdir(caminho);
```

Exemplo

```
mkdir("testeDir", 0777);  
rmdir("testeDir");  
// Resultado:  
// Cria o diretório "testeDir" com todas as permissões de acesso, e depois o exclui
```

Modos

<http://pt.wikipedia.org/wiki/Chmod>

Identificado arquivo ou diretório



- `is_file`

Sintaxe

```
(bool) is_file(caminho);
```

Exemplo

```
echo is_file("teste.txt"); // Resultado: 1 (TRUE) se "teste.txt" for um arquivo
echo is_file("testeDir"); // Resultado: (FALSE) se "testeDir" não for um arquivo
```

Navegando em um diretório



- `opendir` (abertura de diretórios)
- `readdir` (leitura de diretórios)
- `closedir` (fechamento de diretórios)

Sintaxe

```
(resource) opendir(caminho);
(resource) readdir(diretório);
(resource) closedir(diretório);
```

Exemplo

```
$dir = opendir("C:/"); // Abre o diretório
echo readdir($dir); // Retorna o nome do primeiro arquivo do diretório
echo readdir($dir); // Retorna o nome do segundo arquivo do diretório
echo readdir($dir); // Retorna o nome do terceiro arquivo do diretório
closedir($dir); // Fecha o diretório
```

Aulas práticas e manuais on-line



Assista agora as aulas práticas, que apresentam o uso dos comandos abordados nesta aula teórica.

[Clique aqui](#) para visualizar as aulas práticas disponíveis