

PROGRAMAÇÃO DE COMPUTADORES



Técnico Integrado em
INFORMÁTICA

Professor Jefferson Chaves
jefferson.chaves@ifc-araquari.edu.br

- Manipulação de arquivos;
 - Criar, ler, alterar, e apagar um arquivo e seu conteúdo.
- Manipulação de pastas;
 - Criar, ler, alterar, e apagar um arquivo e seu conteúdo.
- Implementar um gerenciador simples de arquivos;

A faint, dark green silhouette of an elephant is visible in the background on the left side of the slide.

**PROGRAMAÇÃO DE
COMPUTADORES**

**MANIPULAÇÃO
DE ARQUIVOS**

PROGRAMAÇÃO DE COMPUTADORES

MANIPULAÇÃO DE ARQUIVOS



IMPORTANTE!

Manipulação de dados é uma parte importante de um aplicação web. Você pode precisar abrir e processar arquivos para inúmeras tarefas.

Mas tome cuidado! Se alguma coisa sair errado isso pode ser perigoso. Erros comuns são: editar o arquivo errado, encher o disco com dados desnecessários, ou mesmo deletar um ou mais arquivos por acidente!

<http://curiosamente.diariodepernambuco.com.br/project/por-conta-de-um-codigo-errado-homem-deleta-sua-empresa-e-de-clientes-da-internet/>

PROGRAMAÇÃO DE COMPUTADORES

MANIPULAÇÃO DE ARQUIVOS

FUNÇÃO	O QUE FAZ?
<u>file_get_contents()</u>	Pega todo conteúdo do arquivo aberto como uma string.
<u>fopen()</u>	'Abre' o arquivo para que possa ser manipulado.
<u>fgets()</u>	Pega linha a linha do arquivo e posiciona o cursor no início da próxima linha.
<u>feof()</u>	Verifica se a leitura chegou ao final do arquivo.
<u>ftruncate()</u>	Reduz o tamanho do arquivo. Usado para apagar seu conteúdo.
<u>fwrite()</u>	Escreve no arquivo.

A faint, dark green silhouette of an elephant is visible in the background on the left side of the slide.

**PROGRAMAÇÃO DE
COMPUTADORES**

**LEITURA DE UM
ARQUIVO**

- Existem algumas formas de se ler um arquivo em PHP. Veremos duas mais uteis:

- Função `file_get_contents()`;
 - Sua chamada retorna o conteúdo de um arquivo para quem a chamou. É possível passar um arquivo disponível em uma URL.
 - Essa função apenas para leitura de um arquivo, não permitindo sua manipulação.


```
1 $conteudo = file_get_contents("arquivo.txt");  
2  
3 echo $conteudo;  
4  
5  
6  
7  
8  
9  
10
```



**PROGRAMAÇÃO DE
COMPUTADORES**

**ABRINDO E LENDO
UM ARQUIVO**

- Antes de abrirmos um arquivos, devemos saber quais os modos de operação disponíveis:

PROGRAMAÇÃO DE COMPUTADORES

ABRINDO E LENDO UM ARQUIVO

MODO	DESCRIÇÃO
'w'	Abre o arquivo para apenas escrita; coloca o ponteiro do arquivo no começo do arquivo e diminui (trunca) o tamanho do arquivo para zero. Se o arquivo não existe, tenta criá-lo.
'w+'	Abre o arquivo para leitura e escrita; coloca o ponteiro do arquivo no começo e diminui (trunca) o tamanho do arquivo para zero. Se o arquivo não existe, tenta criá-lo.
'r'	Abre o arquivo somente para leitura; coloca o ponteiro de escrita no começo do arquivo. [Retorna um erro caso o arquivo não exista e não tenta cria-lo.]
'r+'	Abre para leitura e escrita; coloca o ponteiro de escrita no começo do arquivo. [Retorna um erro caso o arquivo não exista e não tenta cria-lo.]
'a'	Abre para somente para escrita somente; coloca o ponteiro do arquivo no final. Se o arquivo não existir, tenta criá-lo.
'a+'	Abre o arquivo para leitura e escrita; coloca o ponteiro do arquivo no final. Se o arquivo não existir, tenta criá-lo.

- Função `fopen(nomedoarquivo, modo)`
 - Possui dois argumentos: O nome do arquivo e o modo que esse arquivo será criado.
 - **nomedoarquivo** - O nome do arquivo pode conter o caminho absoluto onde você deseja criar o seu arquivo. Exemplo, 'www/programa/arquivo.txt'.
 - **modo** - O modo indica as permissões de acesso para o arquivo que foi criado. Por exemplo, ele pode ser criado apenas para leitura (consulta de dados) ou para escrita e leitura.

```
1 $arquivo = fopen("arquivo.txt", "w");  
2  
3 if ($arquivo == false) {  
4     die('Não foi possível criar o arquivo.');
```

5 }

```
6  
7 fclose($arquivo);  
8  
9  
10
```

- O código anterior cria um novo arquivo (para escrita somente) no diretório corrente do arquivo PHP executado.
- **Se o arquivo já existir, ele deletará todo o conteúdo.**
- Se a função funcionar ele retorna o ponteiro do arquivo. Mas se a função falhar por qualquer motivo, a função retorna false.

- Observe também que utilizamos a função `fclose()` para fechar o arquivo que está aberto.

- Há casos em que não podemos deletar o conteúdo, mas sim acrescentar mais informações quando o arquivo já existir.
- Para esses casos devemos abrir o arquivo em modo 'r'

- O modo 'r' abre o arquivo e coloca o ponteiro no início do arquivo, lendo todo seu conteúdo até o final.
- Use o modo 'r+' para ativar também a escrita junto com a leitura.
- Como o ponteiro é posicionado no início do arquivo, o novo conteúdo ficará no início e não no final.
- Caso o arquivo não exista, ele retorna um erro e não tenta criar o arquivo.
- Caso você queira criar o arquivo caso ele não exista, você pode usar o modo 'a'.

A faint, dark green silhouette of an elephant is visible in the background on the left side of the slide.

**PROGRAMAÇÃO DE
COMPUTADORES**

**LENDO
ARQUIVOS**

- Para lermos um arquivo, temos duas funções principais definidas:
 - Função `fread()` #lê o arquivo todo;
 - Função `fgets()` #lê um arquivo linha a linha;
 - Função `file()` #retorna cada linha como índice de um array

- A função `fread()` é usada para leitura de um arquivo inteiro. Os argumentos são o ponteiro e o tamanho em bytes.
 - **ponteiro** - O ponteiro aqui utilizado, está armazenado na variável que você usou para abrir ou criar o arquivo.
 - **tamanho** - Como já foi dito, é o tamanho em número de bytes do arquivo.

```
1 $arquivo = fopen("arquivo.txt", "r");
2
3 if ($arquivo == false) {
4     die('Não foi possível criar o arquivo.');
```

5 }

```
6
7 echo fread($arquivo, filesize($arquivo));
8
9 fclose($arquivo);
10
```

- A **função fgets()** é usada para leitura de um arquivo inteiro. O argumento obrigatório é o ponteiro para o arquivo.
 - **ponteiro** - O ponteiro aqui utilizado, está armazenado na variável que você usou para abrir ou criar o arquivo.

```
1 $arquivo = fopen("arquivo.txt", "r");
2
3 if ($arquivo == false) {
4     die('Não foi possível criar o arquivo.');
```

5 }

```
6
7 echo fgets($arquivo);
8
9 fclose($arquivo);
10
```


- No código anterior abrimos um arquivo com o **modo 'r'** que posiciona o ponteiro no início do arquivo.
- Depois efetuamos o comando `fgets()` para ler a primeira linha do arquivo e posicionar o cursor na próxima linha.

- Podemos percorrer o arquivo todo, linha a linha para obter dados;
- Para tanto podemos usar um laço de repetição junto com a função `feof()` – end-of-file - que indica o fim de um arquivo.

```
1 $arquivo = fopen("arquivo.txt", "r");
2
3 if ($arquivo == false) {
4     die('Não foi possível criar o arquivo.');
```

5 }

```
6
7 while( !feof( $arquivo ) ) {
8     echo fgets($arquivo);
9 }
10 fclose($arquivo);
```

- **Observação:** Sistemas operacionais, ou ambientes diferentes tem convenções de delimitação de linhas diferentes.

A faint, dark green silhouette of an elephant is visible in the background on the left side of the slide.

**PROGRAMAÇÃO DE
COMPUTADORES**

**ESCREVENDO
EM ARQUIVOS**

- Para escrever em arquivos é necessário ter certeza que seu arquivo está aberto e que tenha permissão para escrita.
- Depois basta utilizar a função `fwrite()` e seus parâmetros. Segue a estrutura da função:

- Os argumentos são o arquivo e o texto a ser escrito.
 - **arquivo** - O ponteiro aqui utilizado, está armazenado na variável que você usou para abrir ou criar o arquivo.
 - **texto** - O conteúdo textual que você deseja escrever em seu arquivo.

- **escrevendo uma simples linha em um arquivo**
 - No código a seguir abriremos o arquivo indicado no modo 'w+' permitindo sua escrita.
 - Caso o arquivo não exista, ele será criado.
- É importante observar que o modo 'w' cria arquivos mas também **trunca (deleta)** o conteúdo do arquivo caso ele exista.


```
1 $arquivo = fopen("arquivo.txt", "w+");
2
3 if ($arquivo == false) {
4     die('Não foi possível criar o arquivo.');
```

5 }

```
6
7 $txt = " Araquari, cidade moderna.";
8 fwrite($arquivo, $txt);
9
10 fclose( $arquivo );
```



**PROGRAMAÇÃO DE
COMPUTADORES**

FUNÇÕES UTEIS

- A função **file()** converte um arquivo em um array;
- Cada posição do array irá conter uma linha do arquivo;

```
1 <?php
```

```
2
```

```
3     $lista = file("teste.txt");
```

```
4
```

```
5     print_r($lista);
```

```
6
```

```
7
```

```
8
```

```
9
```

```
10
```

- Para apagar um arquivo, devemos usar a função **unlink()**;
- Cuidado, pois essa operação não poderá ser desfeita;

- A função **file_exists()** verifica se um arquivo ou pasta existem em uma pasta;
- **Retorna** TRUE se existirem, caso contrário, retorna falso;

A black and white photograph of a bodybuilder with a mustache and curly hair, wearing a dark singlet and white socks with stripes. He is in a squat position, holding a barbell with large weights overhead with both arms. The background shows a gym setting with a bench and some equipment.

EXERCÍCIOS

**MANIPULAÇÃO
DE ARQUIVOS**

Jefferson de Oliveira Chaves

1. Crie um formulário que permita o cadastro de clientes com:
Nome ; Data de nascimento; Cidade-Estado.
2. Cada registro deve ser cadastrado em um arquivo chamado **clientes.txt**;
3. Cada cliente deve estar em uma linha e seus dados devem estar separados por ponto e vírgula.

4. Mostre a lista desses cliente cadastrados na tela do seu navegador;