



Arquivos

FULLSTACK - CAMPUS VILA OLÍMPIA
TURMA 03 - JULHO/2018

Aula passada...

- Embed
- Sintaxe curta
 - if: ... endif;
 - for: ... endfor;
- Métodos
 - GET, POST
- Validação de Formulários
- Redirect
 - header('location:teste.php');
- Debug



JSON

JSON, acrônimo de JavaScript Object Notation, é um formato de texto leve para a troca de dados, baseado em objeto de JavaScript.



JSON - Exemplo

```
{  
  "employees" : [  
    { "firstName" : "John", "lastName" : "Doe" },  
    { "firstName" : "Anna", "lastName" : "Smith"},  
    { "firstName" : "Peter", "lastName" : "Jones" }  
  ]  
}
```

JSON

O PHP oferece 2 funções para operar com variáveis com conteúdo correspondente a um JSON.

- `json_encode`
- `json_decode`



json_encode

Retorna a representação JSON do valor dado

```
<?php
$carro = [
    "Marca" => "Ford",
    "Cor" => "Preto"
];
json_encode($carro);
?>
```

json_encode

Retorna a representação JSON do valor dado

```
<?php
$carro = [
    "Marca" => "Ford",
    "Cor" => "Preto"
];
json_encode($carro);
?>
```

JSON

```
{
    "Marca": "Ford",
    "Cor": "Preto"
}
```

json_decode

Decodifica uma string de JSON

```
<?php
    $carro = [
        "Marca"=> "Ford",
        "Cor"=> "Preto"
    ];
    $json = json_encode($carro);

    json_decode($json);
?>
```

Retorna um objeto!!!

```
stdClass Object (
    [Marca] => "Ford"
    [Cor] => "Preto"
)
```


json_decode

Decodifica uma string de JSON

```
<?php
    $carro = [
        "Marca"=> "Ford",
        "Cor"=> "Preto"
    ];
    $json = json_encode($carro);

    json_decode($json, true);
?>
```

Retorna um array associativo

```
array(2) {
    [Marca] => "Ford"
    [Cor] => "Preto"
}
```

Arquivos

O PHP permite gerenciar arquivos como você quiser, tanto para processamento quanto para suporte. Nosso foco serão os arquivos JSON.

Gerenciamento de arquivos

```
<?php
```

```
$fp = fopen('dados.txt', 'w');  
    fwrite($fp, '1');  
    fwrite($fp, '23');  
    fclose($fp);
```

```
?>
```

O conteúdo de 'dados.txt' agora é 123

fopen

Abre um arquivo ou uma URL e **retorna um recurso** (resource).

```
fopen(string $filename, string $mode);
```

Alguns modos:

- **r**: Abre **somente para leitura**; coloca o ponteiro do arquivo no começo do arquivo.
- **w**: Abre **somente para escrita**; coloca o ponteiro do arquivo no começo do arquivo e reduz o comprimento do arquivo para zero. Se o arquivo não existir, tenta criá-lo.
- **a**: Abre **somente para escrita**; coloca o ponteiro do arquivo no final do arquivo. Se o arquivo não existir, tenta criá-lo.
- **x**: Cria e abre o arquivo **somente para escrita**; coloca o ponteiro no começo do arquivo. Se o arquivo já existir, a chamada a `fopen()` falhará, retornando FALSE e gerando um erro de nível E_WARNING. Se o arquivo não existir, tenta criá-lo.

fopen

Abre um arquivo ou uma URL e **retorna um recurso** (resource).

```
fopen(string $filename, string $mode);
```

Alguns modos:

- **r+:** Abre para **leitura e escrita**; coloca o ponteiro do arquivo no começo do arquivo.
- **w+:** Abre para **leitura e escrita**; coloca o ponteiro do arquivo no começo do arquivo e reduz o comprimento do arquivo para zero. Se o arquivo não existir, tenta criá-lo.
- **a+:** Abre para **leitura e escrita**; coloca o ponteiro do arquivo no final do arquivo. Se o arquivo não existir, tenta criá-lo.
- **x+:** Cria e abre o arquivo para **leitura e escrita**; coloca o ponteiro no começo do arquivo. Se o arquivo já existir, a chamada a fopen() falhará, retornando FALSE e gerando um erro de nível E_WARNING. Se o arquivo não existir, tenta criá-lo.

fwrite

Escreve um arquivo.

```
fwrite(resource $recurso, string $texto);
```

fclose

Fecha um arquivo aberto.

```
fclose(resource $recurso)
```

fread

Leitura de um arquivo.

fread(resource **\$recurso**, int **\$length**)

```
<?php
```

```
    $local_arquivo = "/usr/local/algo.txt";  
    $arquivo = fopen($local_arquivo, "r");  
    $tamanho = filesize($local_arquivo);  
    $conteudo = fread($arquivo, $tamanho);  
    fclose($arquivo);
```

```
?>
```


file_get_contents

Obtém um arquivo em uma cadeia de texto

```
<?php
    $paginaInicio = file_get_contents("dados.txt");
    echo $paginaInicio;
?>
```

É uma espécie de atalho para usar
fopen -> fread -> fclose

file_put_contents

Salva uma cadeia de texto em um arquivo

```
<?php
    $local_arquivo = "dados.txt";
    $arquivo = file_get_contents($local_arquivo);
    $arquivo .= "João Silva\n";
    file_put_contents($local_arquivo, $arquivo);
?>
```

É uma espécie de atalho para usar
fopen -> fwrite -> fclose

Cuidado, se usado assim sobrepõe todo o conteúdo do arquivo!!

Como é possível ler linha por linha?

- Ler arquivos de texto **muito grandes** pode provocar **problemas de memória**.
- Podemos utilizar **fread** ou **file_get_contents** e explodir o conteúdo com o caractere `\n`.
- Não parece a melhor ideia... E então?

fgets

Retorna uma linha de um arquivo aberto

```
<?php
    $gestor = fopen("dados.txt", "r");
    if ($gestor) {
        while (($linha = fgets($gestor)) !== false) {
            echo $linha;
        }
    }
    fclose($gestor);
?>
```

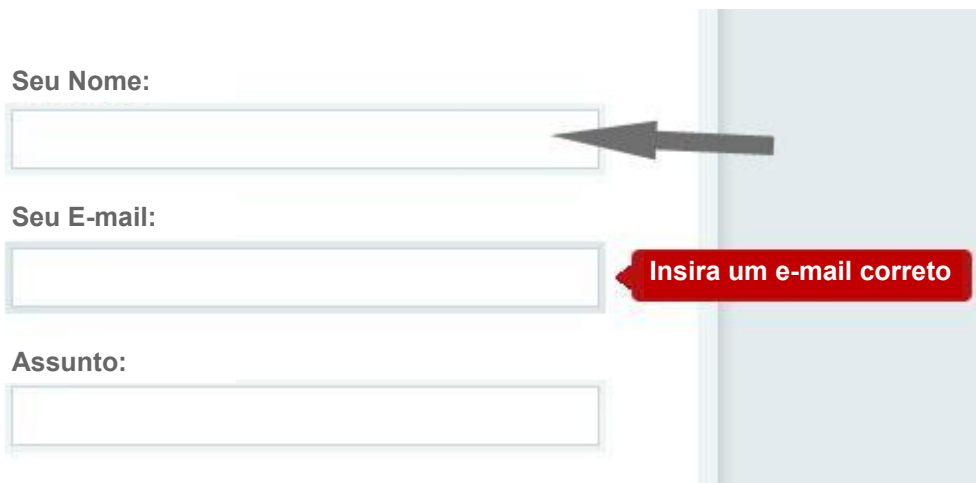
file_exists

Verifica se um arquivo existe

```
<?php
    $arquivo = 'dados.txt';
    if (file_exists($arquivo)) {
        echo "O arquivo $arquivo existe";
    } else {
        echo "O arquivo $arquivo NÃO existe";
    }
?>
```

Persistência de dados

O que acontece quando enviamos um formulário e ele retorna um erro de validação?



Seu Nome:

Seu E-mail:

Assunto:

Insira um e-mail correto

Persistência de dados

Podemos utilizar \$_GET ou \$_POST e assim modificar o atributo value do input:

```
<input type="text" name="firstname" value="<?php echo $_GET['firstname'];?>">
```

```
<input type="text" name="firstname" value="<?php echo $_POST['firstname'];?>">
```

Vamos praticar!



Exercícios

DAILY SCRUM

maximo 15min

- o que fiz ontem?
- o que estou fazendo?
- tem algum impedimento?

dica: use um cronometro!

OBRIGADO!

