PHP - File Upload & Hashing

DigitalHouse >
Coding School

Revisão

NA AULA PASSADA...

- Manipulação de arquivos
  - file_exists
  - fopen, fread, fwrite, fclose, fgets
  - file_get_contents e file_put_contents
- JSON
  - json_encode
  - json_decode

FILE UPLOAD

É usado no formulário HTML para permitir o upload de arquivos para o servidor. Inicialmente os arquivos vão para um diretório temporário e depois realocado de acordo com o script PHP.

FORM ENCTYPE

- **application/x-www-form-urlencoded** - Padrão. Todos os caracteres são codificados para o formato de URL antes do envio (espaços são convertidos em "+", e caracteres especiais são convertidos em valores HEXADECIMAIS ASCII, ex: %5F).

- **multipart/form-data** - Nenhum caractere é codificado. Esse valor é necessário para formulários com upload de arquivos.

- **text/plain** - Espaços são convertidos em "+", mas nenhum caractere especial é codificado.

FORM ENCTYPE - Exemplo

Observe este exemplo de como alterar seu **form**, adicionando o **enctype** para efetuar upload de arquivos com sucesso.

```
<form action="" method="post" enctype="multipart/form-data">
    Arquivo: <input type="file" name="imgPerfil">
    <button type="submit">Enviar</button>
</form>
```

$_FILES

É um array associativo de elementos com upload através do método POST.

O formulário anterior terá uma chave chamada: **imgPerfil**. Exemplo:

```
<?php
    $_FILES[“imgPerfil”];
?>
```

$_FILES

Para cada arquivo, haverá um array associativo.

Neste exemplo, **$_FILES[“imgPerfil”]** é um array associativo que contém algumas chaves importantes, como **error, name** e **tmp_name**. Exemplo:

error - **$_FILES["imgPerfil"]["error"]**

name - **$_FILES["imgPerfil"]["name"]**

tmp_name - **$_FILES["imgPerfil"]["tmp_name"]**

$_FILES - Error

Indica se o upload da imagem foi concluído.

É comparada com a constante **UPLOAD_ERR_OK** que indica que não houve erros.

```
<?php

    if($_FILES["imgPerfil"]["error"] == UPLOAD_ERR_OK)

?>
```

$_FILES - Name

Name indica o nome do arquivo carregado.

```
<?php
    $nome = $_FILES["imgPerfil"]["name"];
?>
```

A variável **$nome** terá o nome original do arquivo.

$_FILES - Tmp_Name

É o endereço do arquivo **enviado** que foi gravado em uma **pasta temporária**.

```php
<?php
    $nome = $_FILES["imgPerfil"]["name"];
    $arquivo = $_FILES["imgPerfil"]["tmp_name"];
?>
```

$_FILES - Move_Uploaded_File

Para salvar o arquivo em uma pasta, na verdade movemos ele da **pasta temporária** para **o lugar que queremos** com **move_uploaded_file**.

```
<?php
    $nome = $_FILES["imgPerfil"]["name"];
    $arquivo = $_FILES["imgPerfil"]["tmp_name"];
    $movendo = move_uploaded_file($arquivo, $nome);
?>
```

$_FILES - Move_Uploaded_File

Caso queira gravar a imagem em uma pasta chamada “**uploads**”, no mesmo diretório do nosso projeto.:

```php
<?php
    $nome = $_FILES["imgPerfil"]["name"];
    $arquivo = $_FILES["imgPerfil"]["tmp_name"];
    $caminho = "uploads/".$nome;
    $movendo = move_uploaded_file($arquivo, $caminho);
?>
```

DIRNAME & __FILE__

→ Retorna o caminho completo do arquivo php que chamou a função:

```php
<?php
    echo __FILE__;
?>
```

→ Retorna o caminho completo da pasta em que o arquivo se encontra:

```php
<?php
    $diretorio = dirname(__FILE__);
?>
```

Efetuando upload de arquivo

Efetuando upload de arquivo utilizando **move_uploaded_file().**

```php
<?php
    $nome = $_FILES["imgPerfil"]["name"];
    $nomeTemp = $_FILES["imgPerfil"]["tmp_name"];
    $diretorio = dirname(__FILE__);
    $nomeDaPasta = "/uploads/";
    $caminhoCompleto = $diretorio.$nomeDaPasta.$nome;

    move_uploaded_file($nomeTemp, $caminhoCompleto);
?>
```

Obrigado até a próxima

DigitalHouse >
Coding School