

# Upload de arquivos com o método POST

- O PHP é capaz de receber o upload de qualquer browser que siga as normas da W3C.
- Isto permite que se faça o upload de arquivos de texto e binários.
- Com as funções de autenticação e manipulação de arquivos do PHP, você tem o controle completo de quem pode fazer o upload de arquivo e o que fazer com o arquivo após seu upload.

## Ex. #1 Formulário para Upload de Arquivo

- Uma tela para upload de arquivo pode ser criada com um formulário especial parecido com este:

```
<form enctype="multipart/form-data" action="upload.php" method="post">  
  <input type="hidden" name="MAX_FILE_SIZE" value="30000" />  
  Enviar este arquivo: <input name="userfile" type="file" />  
  <input type="submit" value="Enviar Arquivo" />  
</form>
```

O campo escondido *MAX\_FILE\_SIZE* (medido em bytes) deve preceder o campo de input do arquivo, e seu valor é o tamanho limite aceito para o arquivo.

Também tenha certeza que seu formulário de upload de arquivo tenha *enctype="multipart/form-data"* em outro caso o upload do arquivo não irá funcionar.

## Aviso sobre a variável \$MAX\_FILE\_SIZE

- O valor de *MAX\_FILE\_SIZE* é um aviso para o browser. É fácil contornar este limite. Então não conte que o browser irá obedecer a sua vontade.
- O que foi estabelecido para maximum-size no PHP não pode ser enganado. Mas você deve adicionar *MAX\_FILE\_SIZE* em qualquer caso, já que salva os usuários do problema de esperar por um grande arquivo ser transferido somente para descobrir depois de tudo que ele é muito grande.

# Variável \$\_FILES

- \$\_FILES é uma array autoglobal do PHP. Ou seja é uma variável interna que recebe informações do arquivo movido por upload.
- Os conteúdos de \$\_FILES do nosso script de exemplo é como segue. Note que isso assume que o nome do upload do arquivo é *userfile*, como o usado no exemplo acima. Pode ser qualquer nome.
  - `$_FILES['userfile']['name']`
    - O nome original do arquivo no computador do usuário.
  - `$_FILES['userfile']['type']`
    - O tipo mime do arquivo, se o browser deu esta informação. Um exemplo pode ser *"image/gif"*.
  - `$_FILES['userfile']['size']`
    - O tamanho, em bytes, do arquivo.
  - `$_FILES['userfile']['tmp_name']`
    - O nome temporário do arquivo, como foi guardado no servidor.
  - `$_FILES['userfile']['error']`
    - O código de erro associado a este upload de arquivo.

# A FUNÇÃO `move_uploaded_file`

- O script PHP que irá receber o arquivo do upload deve ser como este:

```
bool move_uploaded_file ( string $filename , string $destination )
```

Esta função verifica para ter certeza de que o arquivo designado por *filename* é um arquivo de upload válido (que tenha sido enviado pelo mecanismo PHP de envio por POST HTTP). Se o arquivo for válido, ele será movido para o nome de arquivo dado por *destination* .

Este tipo de verificação é especialmente importante para garantir que qualquer coisa feita com os arquivos enviados possa revelar seu conteúdo ao usuário.

## Parâmetros

*filename*

O nome do arquivo enviado.

*destination*

O destino do arquivo movido

## Valor Retornado

- Se *filename* não for um arquivo enviado válido, então nenhuma ação ocorrerá e **move\_uploaded\_file()** retornará **FALSE**.
- Se *filename* for um arquivo enviado válido, mas não puder ser movido por alguma razão, nenhuma ação ocorrerá e **move\_uploaded\_file()** retornará **FALSE**. Adicionalmente, um alerta será emitido.
- Se o arquivo de destino já existir, ele será sobrescrito.

## Ex. #2 Validando o upload de arquivos

- O script PHP que irá receber o arquivo do upload deve ser como este:

```
<?php
$uploadaddir = 'arquivos/';
$uploadfile = $uploadaddir . $_FILES['userfile']['name'];
if (move_uploaded_file($_FILES['userfile']['tmp_name'], $uploadfile)) {
    echo "O arquivo foi carregado com sucesso!";
} else {
    echo "Erro ao transferir arquivo!";

}

?>
```

- Nesse script você pode implementar qualquer lógica que for necessária para determinar o que deve ser feito com o arquivo do upload, usando os valores da variável `$_FILES`.
- Se nenhum arquivo for selecionado em seu formulário, o PHP irá retornar `$_FILES['userfile']['size']` como 0, e `$_FILES['userfile']['tmp_name']` como none.
- O arquivo será excluído do diretório temporário ao fim do script se não tiver sido movido ou renomeado.