

第5回 ファイルアップロード

5-1. ファイルのアップロード

ファイルをアップロードするには、**form タグに enctype プロパティの設定を行う必要があります。**
この設定がないと、アップロードしたファイル名のみが文字列として扱われることとなります。

<form>開始タグ

■enctype プロパティ

設定値	意味
application/x-www-form-urlencoded	URL エンコードを指定（初期値）
multipart/form-data	マルチパートを指定。 フォーム内でファイルを送信する場合は、この形式を指定 しておく必要がある

■SampleCode（HTML）

※input タグの設定にも注意。「type="text"」ではうまくいきません。【HTML の復習】

テキストボックスは「**name="upfile"**」にて設定。

```

<form action="sample01_6.php" method="POST" enctype="multipart/form-data">
  <div id="frame" class="p-5 border-info rounded" style="border:1px dashed;">
    <!-- ファイル大きさ制限数の保持(hidden) -->
    <input type="hidden" name="MAX_FILE_SIZE" value="2000000">
    <!-- ファイル入力箇所 -->
    <div id="file" class="input-group">
      <input type="file" id="inputFile"
        class="form-control form-control-lg" name="upfile">
    </div>
  </div>
  <div class="p-5 d-grid gap-2 d-md-flex justify-content-md-end">
    <button type="submit" class="btn btn-danger btn-lg">アップロード</button>
  </div>
</form>

```

■Sample（入力画面） ファイル選択

※[ファイル選択]ボタン押下にて、ファイルを選択すると、選択したファイル名が右側に表示される。



5 – 2. FILE データの受け取り

■FILE データを取得できるスーパーグローバル変数

\$_FILES	HTTP ファイルアップロード変数 POST で送信された ファイルの情報が連想配列で取得 できます。
----------	---------------------------------------------------------------

■SampleCode（入力画面の HTML テキストボックスは「**name="upfile"**」にて設定。）

```
<?php
// 値チェックなどを行うので、変数に格納して使用が一般的
$upFile = $_FILES["upfile"];
?>
```

■\$_FILES[userfile]の内容

※userfile には<input type="file">の name 属性名が入る。

SampleCode では name="upfile" なので、`$_FILES["upfile"]` で取得可能。

※表は次のページにも続きます。

キー	内容
<code>\$_FILES[userfile]["name"]</code>	クライアント PC 上のファイル名
<code>\$_FILES[userfile]["full_path"]</code>	ファイルのフルパス。 この値は実際のディレクトリ構造を反映しているとは必ずしも言えないため、信用できません。 PHP 8.1.0 以降で利用可能です。
<code>\$_FILES[userfile]["type"]</code>	ファイルの MIME タイプ
<code>\$_FILES[userfile]["tmp_name"]</code>	仮ディレクトリ内のテンポラリファイルの名前

\$ _FILES[userfile][“error”]	エラー種別を表す数値		
	定数	値	内容
	UPLOAD_ERR_OK	0	成功
	UPLOAD_ERR_INI_SIZE	1	ファイルは、 php.ini の upload_max_filesize ディレクティブの値を超えて います。
	UPLOAD_ERR_FORM_SIZE	2	ファイルは、HTML フォーム で 指 定 さ れ た MAX_FILE_SIZE を超えてい ます。
	UPLOAD_ERR_PARTIAL	3	ファイルは一部のみしかアッ プロードされていません。
	UPLOAD_ERR_NO_FILE	4	ファイルはアップロードされ ませんでした。
	UPLOAD_ERR_NO_TMP_DIR	6	テンポラリフォルダがありま せん。
	UPLOAD_ERR_CANT_WRITE	7	ディスクへの書き込みに失敗 しました。
UPLOAD_ERR_EXTENSION	8	PHP の拡張モジュールがファ イルのアップロードを中止し ました。 どの拡張モジュール がファイルアップロードを中 止させたのかを突き止めるこ とはできません。	
\$ _FILES[userfile][“size”]	ファイルのサイズ（バイト単位）		

SampleCode で `$_FILES["upfile"]` の内容を表示してみます。

■Sample 画面（ `$_FILES["upfile"]` の内容を表示）

Array
(
[name] => kadai02-product-2-lg.jpeg
[full_path] => kadai02-product-2-lg.jpeg
[type] => image/jpeg
[tmp_name] => C:\Windows\Temp\phpECB9.tmp
[error] => 0
[size] => 766748
)

サーバーサイドスクリプト
画像のアップロード結果
アップロード結果

結果画面に配列内容を表示
`print_r($_FILES["upfile"]);`

ファイルアップロード（`move_uploaded_file` 関数）

■`move_uploaded_file`関数

`move_uploaded_file(from, to)`

戻り値: 成功→true、失敗→false

■引数

from	移動元ファイルパス ※ <code>\$_FILES[userfile]["tmp_name"]</code> に保存されている仮ディレクトリ
to	移動先ファイルパス

```
<?php
$upFile = $_FILES["upfile"];

// ↓ 結果格納用の連想配列
$result = [
    "status" => true,    // 結果フラグ用
    "message" => null,   // 結果メッセージ用
    "result" => null,    // 結果のファイルパス用
];

// ↓ 移動先のディレクトリ参照（「_DIR_」は自分自身のディレクトリを示す）
$moveFilePath = __DIR__ . "/asset/storage/{$upFile["name"]}";

if (move_uploaded_file($upFile["tmp_name"], $moveFilePath)) {
    $result["message"] = "ファイルのアップロードに成功しました";
    // ↓ HTML の src プロパティに設定するパス用なので、フォルダパスではなく、URL で指定
    $result["result"] =
        "http://localhost/php1_2023_配布用/asset/storage/{$upFile["name"]}";
} else {
    $result["status"] = false;
    $result["message"] = "ファイルのアップロードに失敗しました";
}
```

■SampleCode (HTML 一部抜粋)

```
<!-- ファイル結果表示箇所 -->
<div class="text-center">
  <?php if ($result["status"]) : ?>
    <figure class="d-inline-block me-1 mt-1 mb-5">
      ">
    </figure>
  <?php else : ?>
    <p class="text-danger"><?= $result["message"] ?></p>
  <?php endif ?>
</div>
```

■Sample (入力画面) ファイル選択

・入力 (ファイル選択後)

サーバーサイドスクリプト演習 1

サンプル

画像のアップロード

ファイルを選択

kadai02-product-3-lg.jpeg

アップロード

・結果

サーバーサイドスクリプト演習 1

画像のアップロード結果

アップロード結果



戻る

5 - 3. その他 関数など

ファイル情報取得とファイルアップロードは5 - 2までで完了しました。ただし、入力画面から送信されたデータ形式や値の確認はまだしていません。

ここからは、データのチェックに必要な関数の一例を学びましょう。

POST でデータが送信されたか確認

■Web クライアントの情報を取得できるスーパーグローバル変数

<code>\$_SERVER</code>	サーバー情報および実行時の環境情報 Web クライアントの情報が連想配列で取得 できます。
------------------------	-------------------------------------------------------------

■`$_SERVER` 変数の主なキー

※一部抜粋

キー	内容
<code>PHP_SELF</code>	URL ドキュメントルート以下
<code>SERVER_ADDR</code>	Web サーバーの IP アドレス
<code>SERVER_NAME</code>	Web サーバーのホスト名
<code>REQUEST_METHOD</code>	ページにアクセスする際に使用されたリクエストのメソッド名 'GET', 'HEAD', 'POST', 'PUT' など。
<code>HTTP_USER_AGENT</code>	クライアントのブラウザ種別を表す User-Agent 文字列
<code>HTTPS</code>	http://接続時は空欄、https://接続時は何らかの文字列
<code>SCRIPT_FILENAME</code>	アクセスされている PHP ファイルの絶対パス
<code>SCRIPT_NAME</code>	アクセスされている PHP の URL のパス

■SampleCode（結果画面 sample01_6.php）

※POST 形式のリクエストがなければ、入力画面 sample01_5.php に遷移する。

```
<?php
// POST 形式でなければ
if ($_SERVER["REQUEST_METHOD"] !== "POST") {
    header("Location: sample01_5.php");    //sample01_5.php へ画面遷移
    exit;    // 処理を終了させる
}

$upFile = $_FILES["upfile"];
?>
```

リダイレクト処理（header 関数）

```
header( "Location: /* 遷移先の URL */" );
```

※他に引数を指定できますが、今回は概要のみ。

※「 Location: 」と使用することで、指定した URL へリダイレクト（自動的に転送）します。

※header 関数の前に echo や print、HTMLなどを指定しないこと。画面遷移できなくなります。

処理終了（exit）

```
exit;
```

※以降の処理は行われません。

続いて、ファイルアップロードを行うより前に、データのチェックを行います。

値チェック (isset 関数)

■isset関数

isset(var)

戻り値: 変数が定義されており、かつ値がnullでない→true
それ以外→false

■引数

var	調べたい変数
-----	--------

■SampleCode (結果画面 sample01_6.php)

※\$_FILES["upfile"]が定義されていない、または null である場合、結果用変数にエラーとして値をセット。

```
<?php

// ～(省略)～
// 定義されていない、または null である場合
if (!isset($_FILES["upfile"])) {
    $result["status"] = false; // エラーのとき false とする
    $result["message"] = "ファイルのアップロードに失敗しました"; // 画面表示用
}

$upFile = $_FILES["upfile"];

// ～(省略)～

?>
```

■Sample 画面

・入力（ファイル選択なし）
クに該当）

・結果（isset のエラーチェッ

サーバーサイドスクリプト演習 1

サンプル

画像のアップロード

ファイルを選択 選択されていません

アップロード

サーバーサイドスクリプト演習 1

画像のアップロード結果

アップロード結果

ファイルがありません

戻る

続いて、ファイルの種別が画像であるかを調べます。

パターンチェック（preg_match 関数）

■preg_match関数

`preg_match(pattern, subject, matches, flgs, offset)`

戻り値: pattern が指定した subject にマッチした場合 → 1
マッチしない場合 → 0

■引数

pattern	string 型。検索するパターンを表す文字列。正規表現で記載。
subject	string 型。入力文字列
matches	array 型。省略可。初期値 null
flags	int 型。省略可。初期値 0
offset	int 型。省略可。初期値 0

■SampleCode（ファイルが画像であるかパターンチェック）

※ファイル種別は \$upFile["type"] に格納される。

画像の場合は、「image/種別」となるので、「image」の文字列が入っているかを確認。

```
<?php

// ～(省略)～

$upFile = $_FILES["upfile"];

//文字列 image を含まないとき＝画像ではないとき、エラーとして連想配列に値を格納
//※文字列のパターンは「'/image/'」( / スラッシュで囲む)
//※条件の前の「!」は否定(NOT) 初回の復習
if (!preg_match('/image/', $upFile["type"])) {
    $result["status"] = false;
    $result["message"] = "画像ファイル以外はアップロードできません";
}

// ～(省略)～

?>
```

■Sample 画面

・入力（word ファイルを選択）
扱い）

・結果（パターンチェックでエラー

サーバーサイドスクリプト演習 1

サンプル

画像のアップロード

ファイルを選択 01_課題01.docx

アップロード

サーバーサイドスクリプト演習 1

画像のアップロード結果

アップロード結果

画像ファイル以外はアップロードできません

戻る

文字列を文字列により分割する（explode 関数）

■explode関数

explode(separator, string, limit)

戻り値:string の内容を separator で分割した文字列の配列

■引数

separator	区切り文字列。
string	入力文字列。
limit	int 型。省略可。最大要素数。

■SampleCode 1

※ファイル種別は \$upFile[“name”] に格納される。

たとえば、ファイルが「kadai02-product-8-lg.jpeg」であるとき、

ファイル名を「.」で分割してみます。

```

<?php
// ～(省略)～

$upFile = $_FILES["upfile"];

// ～(省略※エラーチェックなど)～

//拡張子付きの画像ファイル名を「.」前後で分割。ファイル名と拡張子に分ける
$text = explode(".", $upFile["name"]);

//確認用に画面に連想配列の値を表示
echo "<pre>";
print_r($text);
echo "</pre>";

// ～(省略)～
?>

```

■ Sample 画面

・ 入力（画像ファイルを選択）

・ 結果

サーバーサイドスクリプト演習 1

サンプル

画像のアップロード

ファイルを選択 kadai02-product-8-lg.jpeg

アップロード



■ SampleCode 2

※たとえば、ファイル名が「kadai02-product-8-lg.jpeg」であり、
ファイル名を「日時.拡張子」に変更したい場合、
名前は日時に上書きすればいいが、拡張子はアップロードしたファイルによって変わるため、
拡張子のみ別に保存しておき、新たなファイル名の末尾に拡張子を追加する。
そうすれば拡張子が強制的に変更されてファイルが破損するなどのミスを防げる。

```

<?php
// ～(省略)～

$upFile = $_FILES["upfile"];

// ～(省略※エラーチェックなど)～

//拡張子付きの画像ファイル名を「.」前後で分割。ファイル名と拡張子に分ける
$reFileName = date("YmdHis");// date 関数で日時を取得
$ext = explode(".", $upFile["name"]); //ファイル名分割
$ext = $ext[count($ext) - 1]; // 連想配列の末尾＝拡張子のみ取得

//確認用に画面に連想配列の値を表示
echo "<pre>";
print_r($reFileName.".". $ext); //日付のファイル名と拡張子を文字列連結して表示
echo "</pre>";

// ～(省略)～

?>

```

■ Sample 画面

- ・ 入力（画像ファイルを選択）
- ・ 結果

サーバーサイドスクリプト演習 1

サンプル

画像のアップロード

ファイルを選択

kadai02-product-8-lg.jpeg

アップロード

20230922180905.jpeg

サーバーサイドスクリプト演習 1

画像のアップロード結果

アップロード結果

