

Webプログラミング演習 II

Pure PHP

前回の復習

1. カナ変換（半角カナから全角カナへ）
2. 数値変換（全角数字から半角数値へ）

テキストP.151

復習

5-5



フリガナを半角から 全角に変換する

`mb_convert_kana`関数

半角カタカナで入力された文字列を全角カタカナに変換するスクリプトを作成しましょう。例えばフリガナを入力するときに、入力自体は半角と全角のいずれでもできますが、結果は全角に統一することができます。

▼ここで行うこと

PHP お名前フリガナを入力してください。

マツウラ ケンイチロウ 確定

PHP



PHP フリガナは「マツウラ ケンイチロウ」です。

半角カタカナを全角カタカナへ置き換える

半角で入力された文字列を全角に変換して処理に利用できるようにしましょう。

mb_convert_kana関数 P.152

復習



mb_convert_kana関数

`mb_convert_kana`関数は、PHPのマルチバイト文字列関数の一種です。マルチバイト文字列関数とは、日本語の文字など、コンピュータの内部で複数のバイトを使って表現する文字列に対して、いろいろな機能を提供する関数です。マルチバイト文字列関数は、関数名の先頭に「mb」が付いています。

`mb_convert_kana`関数を次のように呼び出すと、文字列に含まれる半角カタカナを、全角カタカナに変換します。

半角カナから全角カナへ変換



```
mb_convert_kana(文字列)
```

- 入力された文字列を取得し、半角カタカナの場合は全角カタカナに変換する

```
mb_convert_kana($_POST['furigana'])
```

5-6

テキストP.154

復習

数値を全角から半角へ
変換する

mb_convert_kana関数



全角数字を半角数字に変換するスクリプトを作成しましょう。例えば購入個数などの数値を入力するときには、半角と全角のいずれでも入力できますが、半角に統一して利用することができます。郵便番号や住所の入力にも応用できます。

▼ここで行うこと

PHP 購入個数を入力してください。

123 確定

PHP

数値を全角から半角へ
置き換える

PHP 123個を購入します。

全角で入力された数値を半角に変換して
利用できるようにしましょう。



数値を全角から半角へ変換

```
mb_convert_kana(文字列,オプション)
```

- 入力された文字列を取得し、オプションで指定された形に変換する

```
$count=mb_convert_kana($_POST['count'], mode: 'n');
```

オプション

オプションの指定により、様々な変換を行うことができます。

Table mb_convert_kana関数のオプション

オプション	意味
r	全角英字を半角に変換
R	半角英字を全角に変換
n	全角数字を半角に変換
N	半角数字を全角に変換
a	全角英数字を半角に変換
A	半角英数字を全角に変換
s	全角スペースを半角に変換
S	半角スペースを全角に変換

k	全角カタカナを半角カタカナに変換
K	半角カタカナを全角カタカナに変換
h	全角ひらがなを半角カタカナに変換
H	半角カタカナを全角ひらがなに変換
c	全角カタカナを全角ひらがなに変換
C	全角ひらがなを全角カタカナに変換
V	濁点付きの文字を1文字に変換。KやHと併用する

本日の目標

1. ファイルのアップロード
2. 課題18

今回はテキスト5-8に沿って
学習していくよ！



少し難しいからゆっくり進めようね。



難しいって聞くと
ドキドキする。。。



5-8



サーバにファイルを アップロードする

ファイルのアップロード



サーバにファイルをアップロードするスクリプトを作成しましょう。SNSなどでプロフィール用の写真画像などをアップロードする機能を実現することができます。

▼ここで作ること

PHP

アップロードするファイルを指定してください。

ファイルを選択 item0.png

アップロード

PHP

選択したファイルをサーバにアップロードできるようにしましょう。

動作を確認しよう！

①upload-input.phpを開いた画面

アップロードするファイルを指定してください。

参照... ファイルが選択されていません。

アップロード

②PC内の画像ファイルを選択した状態

アップロードするファイルを指定してください。

参照... IMG_0012.JPG

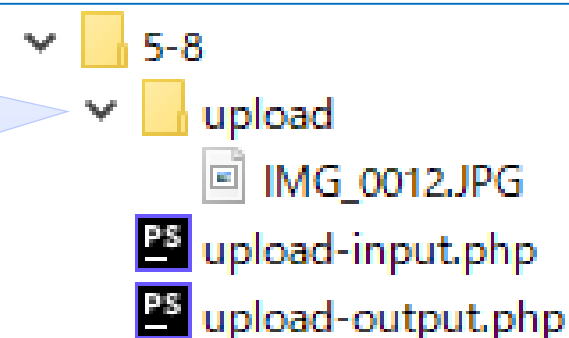
アップロード

③アップロードボタン押下後の upload-output.phpの画面

upload/IMG_0012.JPGのアップロードに成功しました。

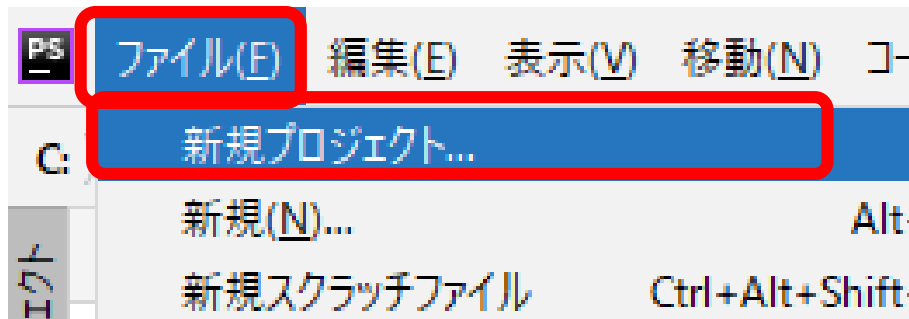


ロリポップサーバを見ると
自動でuploadフォルダが
作成されてファイルが入っ
ています！



実習準備

①PHPStorm—ファイル—新規プロジェクト



②untitledを消して upload-input で作成

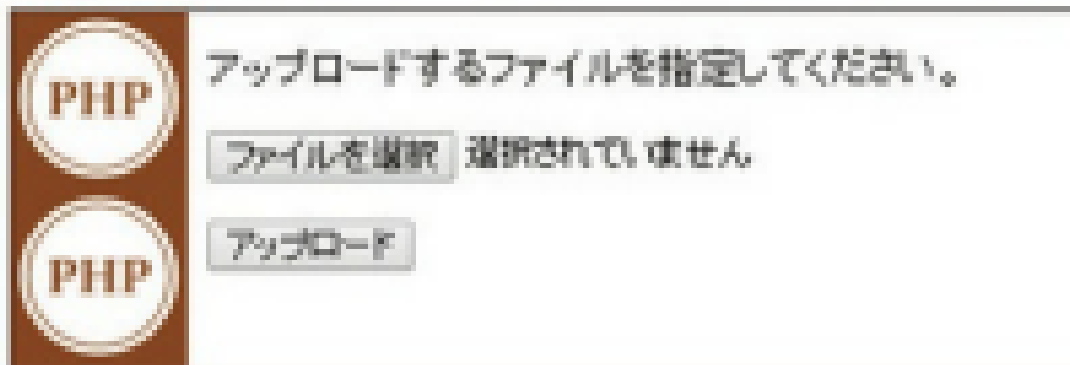
新規プロジェクト

場所(L): C:\Users\imamura\PhpstormProjects\untitled

upload-input.php の作成

```
<?php require '../header.php'; ?>
<p>アップロードするファイルを指定してください。</p>
<form action="upload-output.php" method="post" enctype="multipart/form-data">
<p><input type="file" name="file"></p>
<p><input type="submit" value="アップロード"></p>
</form>
<?php require '../footer.php'; ?>
```

Fig ファイルの選択画面



アップロードするファイルを指定してください。

ファイルを選択 選択されていません

アップロード

upload-output.php の作成

```
<?php require '../header.php'; ?>
<?php
if (is_uploaded_file($_FILES['file']['tmp_name'])) {
    if (!file_exists( filename: 'upload')) {
        mkdir( directory: 'upload');
    }
    $file='upload/'.basename($_FILES['file']['name']);
    if (move_uploaded_file($_FILES['file']['tmp_name'], $file)) {
        echo $file, 'のアップロードに成功しました。';
        echo '<p></p>';
    } else {
        echo 'アップロードに失敗しました。';
    }
} else {
    echo 'ファイルを選択してください。';
}
?>
<?php require '../footer.php'; ?>
```

◆ ロリポップサーバのアップロード場所

ルートディレクトリ/php/kadai/5-8/upload-input.php

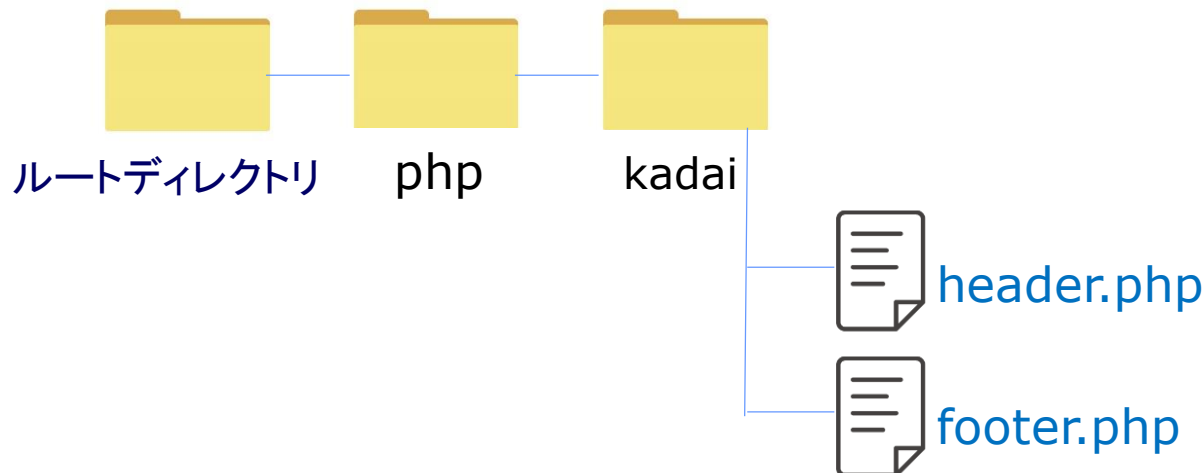
ルートディレクトリ/php/kadai/5-8/upload-output.php

5-8フォルダを作成

※スライド13の授業で次の2ファイルは格納済

ルートディレクトリ/php/kadai/header.php

ルートディレクトリ/php/kadai/footer.php



ファイルアップロードform P.166

```
<form action="" method="post" enctype="multipart/form-data">  
  <input type="file" name="file">  
  <input type="submit" value="アップロード">  
</form>
```

`enctype="multipart/form-data"`

HTMLフォームでファイルアップロードする際に使われる。
テキストファイル、画像ファイルなど様々なファイルをアップ
ロードするマルチ指定。

ファイルアップロードform P.166

enctypeとmultipart/form-dataの意味



enctypeは、MIMEタイプを指定するための属性です。MIME (Multipurpose Internet Mail Extension) とは、データの種類を表現する方法や、データを変換する方法を定めた規格です。もともとはメールでいろいろなデータを扱うための規格ですが、Webでも使用されています。

multipart/form-dataは、HTTPでファイルをアップロードする際に用いるMIMEタイプです。multipartは複数のファイルをまとめるための形式で、メールでは本文と添付ファイルをまとめるために使います。HTTPではフォームへの入力内容とアップロードするファイルをまとめるために使います。

HTTPヘッダの話などあります。
こちらについては、JavaEE (Servlet) の時に勉強します。

[PHP8 マニュアル](#)

<https://www.yoheim.net/blog.php?q=20171201>

upload-output.php の作成

```
<?php require '../header.php'; ?>
<?php
if (is_uploaded_file($_FILES['file']['tmp_name'])) {
    if (!file_exists( filename: 'upload')) {
        mkdir( directory: 'upload');
    }
    $file='upload/'.basename($_FILES['file']['name']);
    if (move_uploaded_file($_FILES['file']['tmp_name'], $file)) {
        echo $file, 'のアップロードに成功しました。';
        echo '<p>
```

tmp_name : Webサーバ上の一時的な領域に保存された時の名前を取得

ファイルの確認 P.169

`is_uploaded_file(ファイル名)`

アップロードされたファイルの時、trueを返す。
通常は、ifステートメントと併せて利用する。

```
if (is_uploaded_file($_FILES['file']['tmp_name'])) {
```

アップロードして一時領域に入っているファイルは、
本当にアップロードされたものかチェックしている。



is_uploaded_file関数の意味

is_uploaded_file関数は、指定したファイルがアップロードされたファイルかどうかを確認します。この確認はセキュリティのためです。例えば、スクリプトが重要なファイルを操作するように仕向ける攻撃を防止する効果があります。



フォルダ確認 P.169

file_exists(フォルダ名)

アップロードされたファイルを保存するためにサーバ上にフォルダを作成するが、指定したフォルダ名が存在しないかを調べるために利用する。

指定したフォルダ名が存在する場合はtrueを返す。

通常は、ifステートメントと併せて利用する。

```
if (!file_exists( filename: 'upload' )) {
```

trueでない(指定したフォルダが無い)

フォルダ作成 P.170

mkdir(フォルダ名)

アップロードされたファイルを保存するためにサーバ上にフォルダを作成する

```
mkdir( directory: 'upload');
```

uploadというフォルダを作成

ここまでのまとめ

```
if (is_uploaded_file($_FILES['file']['tmp_name'])) {  
    if (!file_exists( filename: 'upload')) {  
        mkdir( directory: 'upload');  
    }  
}
```

本当にアップロードされたファイルか調べる

↓ (true 本当にアップロードされたファイル)

uploadというフォルダはあるか調べる

↓ (false 同じフォルダ名はない)

uploadというフォルダを作成する

upload-output.php の作成

```
<?php require '../header.php'; ?>
<?php
if (is_uploaded_file($_FILES['file']['tmp_name'])) {
    if (!file_exists( filename: 'upload')) {
        mkdir( directory: 'upload');
    }
    $file='upload/' . basename($_FILES['file']['name']);
    if (move_uploaded_file($_FILES['file']['tmp_name'], $file)) {
        echo $file, 'のアップロードに成功しました。';
        echo '<p></p>';
    } else {
        echo 'アップロードに失敗しました。';
    }
} else {
    echo 'ファイルを選択してください。';
}
?>
<?php require '../footer.php'; ?>
```

アップロードしたファイルを保存 P.170

```
$_FILES['file']['name']
```

file : 入力画面で付けた名前

```
<input type="file" name="file">
```

name : アップロードしたファイルの名前(自分のPC内にあった名前)

パスからファイル名だけを取得 P.171

basename(パス)

ファイルあるいはディレクトリへのパスを受け取って、最後にある名前の部分を返す。セキュリティ上の問題で必ず入れるようにします。

```
$file='upload/' . basename($_FILES['file']['name']);
```

uploadというフォルダの後にファイル名を追加したパスを作っています。`.`は文字列連結です。
下記のようにupload/ファイル名となる。

upload/IMG_0012.JPGのアップロードに成功しました。

実際にサーバー上へ保存 P.171

`move_uploaded_file(一時的なファイル,保存先のファイル)`

サーバの一時的な場所から正式な場所へアップロード（保存）する。
正常に保存できたらtrueを返す。
通常はifステートメントと併せて利用する。

```
if (move_uploaded_file($_FILES['file']['tmp_name'], $file)) {  
    echo $file, 'のアップロードに成功しました。';  
    echo '<p><img src="' . $file . '></p>';  
} else {  
    echo 'アップロードに失敗しました。';  
}
```


ここまでのまとめ

```
$file='upload/'.basename($_FILES['file']['name']);  
if (move_uploaded_file($_FILES['file']['tmp_name'], $file)) {  
    echo $file, 'のアップロードに成功しました。';  
    echo '<p></p>';  
} else {  
    echo 'アップロードに失敗しました。';  
}
```

アップロードしたファイル名を取り、uploadフォルダからのパスを作成

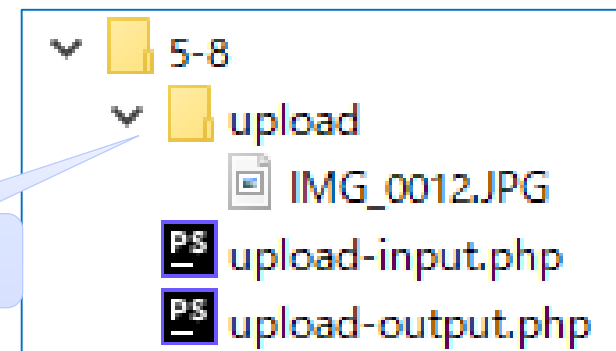


一時領域から正式な場所へアップロードする

↓ (true アップロード成功)

成功しましたメッセージを表示

正式な場所



課題18

◆ 本日授業で作成したプログラムを提出

入力画面のリンク、ソースコード2つを提出

ルートディレクトリ/php/kadai/5-8/[upload-input.php](#)

ルートディレクトリ/php/kadai/5-8/[upload-output.php](#)

課題19

◆ 3つの画像を一度にアップロード

入力画面のリンク、ソースコード2つを提出

ルートディレクトリ/php/kadai/5-8/**kadai19-in.php**

ルートディレクトリ/php/kadai/5-8/**kadai19-out.php**

お気に入りの写真を紹介してください。

参照... IMG_0090.JPG

参照... IMG_0078.JPG

参照... IMG_0012.JPG

アップロード

1つでも画像が選ばれていなければエラー
細かな仕様はお任せします。

アップロードに成功しました。

