

PHP 05



G's ACADEMY
FUKUOKA



本日の内容

講義 + 作業 : 2.5 h程度
演習 : 1.5 h程度

アジェンダ

- ファイルのアップロード練習
- タスクリストにファイル追加
- ajax
- 課題発表→チュータリング(演習)タイム

授業のルール

- 授業中は常にエディタを起動！
- 隣の人と相談するときは周りの迷惑にならない大きさで.
- 周りで困ってそうな人がいたらおしえてあげましょう！
- まずは**打ち間違い**を疑おう！

{ } ' " ; など

- 書いたら保存しよう！

command + s

ctrl + s

準備

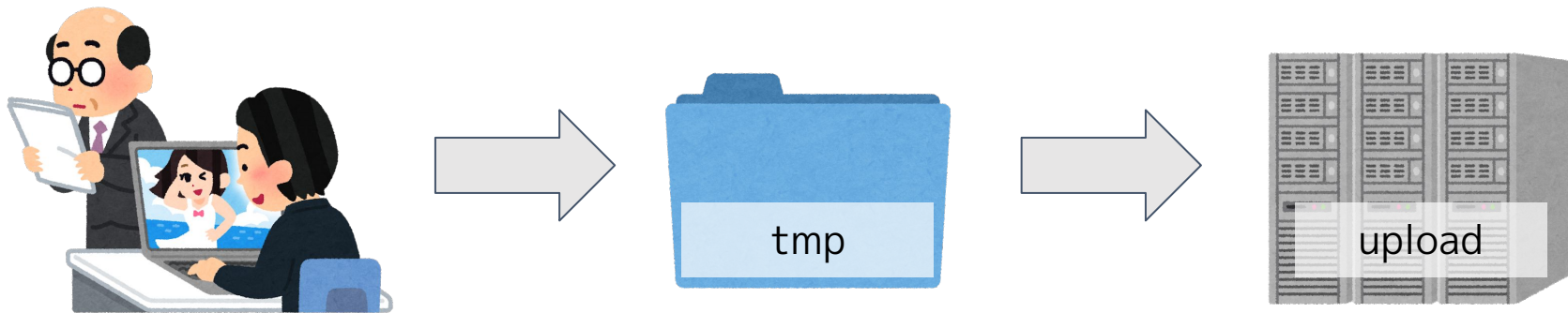
- MAMPの起動確認
- <http://localhost/>のアクセス確認
- サンプルフォルダを「htdocs」フォルダに入れる
- functions.phpのDB名とパスワードを設定

ファイルアップロード

phpでファイルをサーバに送信しよう！

■ファイルアップロードの流れ

- ①フォームからアップロード
- ②tmp領域（一時保存場所）に保存
- ③サーバの保存領域に移動（サンプルでは「upload」）
- ④（データベースに保存場所のパスを登録）



①フォームの準備

①フォームからアップロード

file_upform.html

- ・ <input type="file">を使用.
- ・ 使用時には「enctype="multipart/form-data"」が必須！！
- ・ methodはpostを使用！

※getだと容量不足の可能性が．．．！

■コード

```
<form method="post" action="hoge.php" enctype="multipart/form-data">
  <input type="file" name="upfile" accept="image/*" capture="camera">
  <input type="submit" name="submit" value="送信">
</form>
```

②-④ファイルの保存

ファイル保存の流れ

file_upload.php

準備：送信時にエラー等ないかどうか確認.

- ①送られてきたファイルの情報を取得（自動的にtmp領域に保管）
- ②ファイル名を準備（被らないように）
- ③サーバの保存領域に移動（サンプルでは「upload」）
（ファイル名に保存ディレクトリも含めている点に注意！）
- ④サンプルファイルではimgタグで表示

準備：送信時にエラー等ないかどうか確認

file_upload.php

- ・ ファイルが追加されていない or エラー発生の場合を分ける.
- ・ 送信されたファイルは\$_FILES['...'];で受け取る！

■コード

```
if (isset($_FILES['upfile']) && $_FILES['upfile']['error'] == 0) {  
    //送信が正常に行われたときの処理  
} else {  
    $img = '画像が送信されていません';  
}
```

①送信ファイルの情報を取得

file_upload.php

- ・ アップロードしたファイル名を取得.
- ・ 一時保管しているtmpフォルダの場所の取得.
- ・ アップロード先のパスの設定（サンプルではuploadフォルダ←作成！）

■コード

```
$file_name = $_FILES['upfile']['name'];      //ファイル名  
$tmp_path  = $_FILES['upfile']['tmp_name'];  //tmpフォルダ  
$file_dir_path = 'upload/';                 //アップロード先
```

②ファイル名の準備

file_upload.php

- ・ ファイルの拡張子の種類を取得.
- ・ ファイルごとにユニークな名前を作成. (最後に拡張子を追加)
- ・ ファイルの保存場所をファイル名に追加.

■コード

```
$extension = pathinfo($file_name, PATHINFO_EXTENSION);  
$uniq_name = date('YmdHis').md5(session_id()) . "." . $extension;  
$file_name = $file_dir_path.$uniq_name;
```

③④サーバの保存領域に移動→表示

file_upload.php

- ・ アップロード領域へファイルを移動.
- ・ 権限の変更.
- ・ で出力.

※権限 : <https://www.atmarkit.co.jp/ait/articles/1605/23/news020.html>

③④サーバの保存領域に移動→表示

file_upload.php

■コード

```
$img="";  
if (is_uploaded_file($tmp_path)) {  
    if (move_uploaded_file($tmp_path, $file_name)) {  
        chmod($file_name, 0644);  
        $img = '';  
    } else {  
        exit('Error:アップロードできませんでした. ');  
    }  
}
```


ファイルのアップロード

file_upload.php

■練習

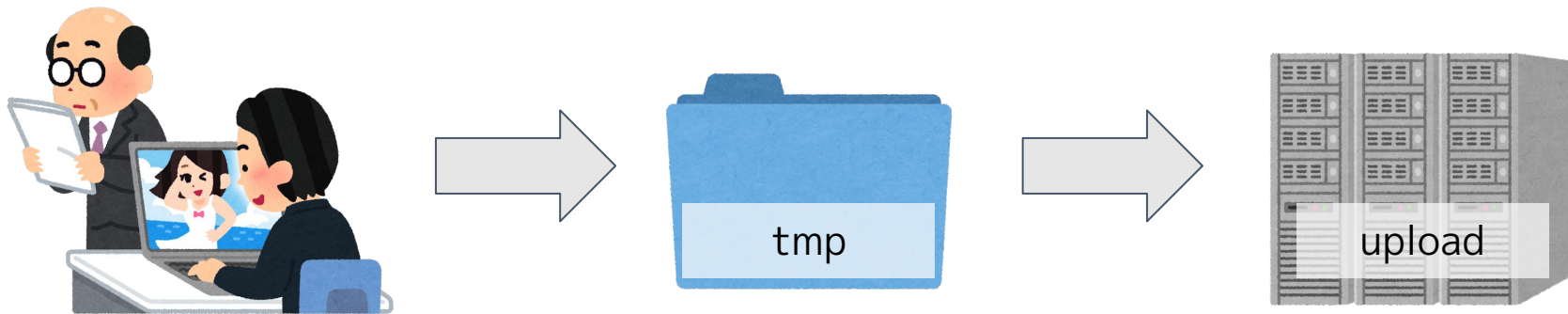
- ・ アップロード用のフォームを準備しよう！
- ・ アップロード処理を記述して画像をアップロードしよう！
- ・ アップロードしたファイルを表示しよう！
(file_upload.phpで\$imgを出力！)

タスクリストへ組み込み

phpでファイルをサーバに送信しよう！

■ファイルアップロードの流れ

- ①フォームからアップロード
- ②tmp領域（一時保存場所）に保存
- ③サーバの保存領域に移動（サンプルでは「upload」）
- ④データベースに保存場所のパスを登録



タスクリストへ画像追加

■準備①

- ・ php02_tableにカラムを追加する.
- ・ 「image」を追加！
- ・ 保存した画像のURLを登録する.

| | # | 名前 | データ型 | 照合順序 | 属性 | NULL | デフォルト値 | コメント | その他 | 操作 |
|--------------------------|---|----------|--------------|-----------------|----|------|--------|------|----------------|---|
| <input type="checkbox"/> | 1 | id 🗄️ | int(12) | | | いいえ | なし | | AUTO_INCREMENT |  変更  削除 ▼ その他 |
| <input type="checkbox"/> | 2 | task | varchar(64) | utf8_unicode_ci | | いいえ | なし | | |  変更  削除 ▼ その他 |
| <input type="checkbox"/> | 3 | deadline | date | | | いいえ | なし | | |  変更  削除 ▼ その他 |
| <input type="checkbox"/> | 4 | comment | text | utf8_unicode_ci | | はい | NULL | | |  変更  削除 ▼ その他 |
| <input type="checkbox"/> | 5 | image | varchar(128) | utf8_unicode_ci | | はい | NULL | | |  変更  削除 ▼ その他 |
| <input type="checkbox"/> | 6 | indate | datetime | | | いいえ | なし | | |  変更  削除 ▼ その他 |

■準備②

- ・ アップロードフォームの準備
- ・ input type="file"の追加, enctype属性の追加

■コード例

```
<form method="post" action="insert_file.php" enctype="multipart/form-data">
```

```
...
```

```
<input type="file" class="form-control-file" id="upfile" name="upfile" accept="image/*"  
capture="camera">
```

タスクリストへ画像追加

insert_file.php

■流れは先程と同じ！

準備：送信時にエラー等ないかどうか確認.

①送られてきたファイルの情報を取得（自動的にtmp領域に保管）

②ファイル名を準備（被らないように）

③サーバの保存領域に移動（サンプルでは「upload」）

（ファイル名に保存ディレクトリも含めている点に注意！）

---ここまで全く同じ---

④DBに情報を登録

⑤一覧画面に表示

準備：送信時にエラー等ないかどうか確認

insert_file.php

- ・ ファイルが追加されていない or エラー発生の場合を分ける.
- ・ 送信されたファイルは\$_FILE['...'];で受け取る！

全く同じ！

■コード例

```
if (isset($_FILES['upfile']) && $_FILES['upfile']['error'] == 0) {  
    //送信が正常に行われたときの処理  
} else {  
    $img = '画像が送信されていません';  
}
```

①送信ファイルの情報を取得

insert_file.php

- ・ アップロードしたファイル名を取得.
- ・ 一時保管しているtmpフォルダの場所の取得.
- ・ アップロード先のパスの設定（サンプルではupload/）

全く同じ！

■コード例

```
$file_name = $_FILES['upfile']['name'];           //ファイル名  
$tmp_path  = $_FILES['upfile']['tmp_name'];       //tmpフォルダ  
$file_dir_path = 'upload/';                      //アップロード先
```


②ファイル名の準備

insert_file.php

- ・ ファイルの拡張子の種類を取得.
- ・ ファイルごとにユニークな名前を作成. (最後に拡張子)
- ・ ファイルの保存場所をファイル名に追加.

全く同じ!

■コード例

```
$extension = pathinfo($file_name, PATHINFO_EXTENSION);  
$uniq_name = date('YmdHis').md5(session_id()) . "." . $extension;  
$file_name = $file_dir_path.$uniq_name;
```

③サーバの保存領域に移動

insert_file.php

- ・ アップロード領域へファイルを移動.
- ・ 権限の変更.
- ・ 今回は表示しない！

③サーバの保存領域に移動

insert_file.php

■コード

```
if (is_uploaded_file($tmp_path)) {  
    if (move_uploaded_file($tmp_path, $file_name)) {  
        chmod($file_name, 0644);  
    } else {  
        exit('Error:アップロードできませんでし  
    }  
}
```

今回は表示しない！

④DBへデータ登録

insert_file.php

- ・他のデータと一緒にDBへ登録！

■コード例

```
$sql = 'INSERT INTO php02_table(id, task, deadline, comment, image,  
indate) VALUES(NULL, :a1, :a2, :a3, :image, sysdate())';
```

...

```
$stmt->bindValue(':image', $file_name, PDO::PARAM_STR);
```

⑤一覧画面に表示

select.php

- ・一覧画面にimgタグで出力！

■コード例

...

```
$view .= '';
```

...

ファイルのアップロード

■練習

- ・ アップロード用のフォームを準備しよう！
- ・ アップロード処理を記述して画像をアップロードしよう！
- ・ アップロードしたファイルのURLをDBに保存しよう！
- ・ 一覧画面に画像を表示しよう！

ajax

ajaxとは. . .

PHPで実行するDBへの登録，表示などのhttp通信を

javascriptで

扱う手法！！

ajaxとは. . .

■メリット

- ・ データだけをやり取りするので速い！
- ・ 通信量が少ない！
- ・ ファイル数が少なくできる！

■デメリット

- ・ SEOに弱い（最近は大丈夫になってきている）
- ・ 構造が複雑になりがち.
- ・ ページを更新すると表示内容が初期状態に戻る.

今日やること

- ・ DBに登録されている情報を取得する.
- ・ フォームに入力した情報をDBに登録し，登録後の最新情報を取得する.

※javascriptとPHPが入り乱れるので都度ファイル名を確認！！！！

情報の取得

DBに登録されている情報を取得する

■必要なもの

- ・ javascriptのコード（**ajax_index.html**）

PHPファイルに対してリクエストを送る処理.

APIへのリクエストと同じく\$.getJSON()を使用.

- ・ PHPのコード（**ajax_get.php**）

DBからデータを取得する処理.

前回までのselect.phpとほぼ同様.

取得したデータをjson形式で返す.

DBに登録されている情報を取得する

■処理の流れ

- ①javascriptからPHPファイルにリクエストを送る．(js)
- ②DBからデータを取得する．(PHP)
- ③取得したデータをjson形式にして出力する．(PHP)
- ④javascriptでデータを受け取る．(js)←ここまでつくろう！
- ⑤（受け取ったデータをブラウザに表示する）

①④javascriptからPHPファイルにリクエストを送る→受け取り

ajax_index.html

```
function selectData() {  
    const url = 'ajax_get.php';  
    $.getJSON(url)                                //①リクエスト送信  
        .done(function (data, textStatus, jqXHR) {  
            console.log(data);                    //④受け取り→表示  
        })  
        .fail(function (jqXHR, textStatus, errorThrown) {...})  
        .always(function () {...});  
}
```

※必ず関数実行しましょう！

②③DBからデータを取得→出力（select.phpとほぼ同じ！）

...（ここまでselect.phpと同じ）

ajax_get.php

```
if ($status==false) {  
    errorMsg($stmt);  
} else {  
    $res = [];  
    while ($result = $stmt->fetch(PDO::FETCH_ASSOC)) {  
        $res[] = $result;           //配列に入れる  
    }  
    echo json_encode($res);        //json形式にして出力  
}
```

ajax練習①

■処理の流れ

- ①\$.getJSON()でリクエストを送ろう！
- ②DBからデータを取得しよう！
- ③json形式にして出力しよう！
- ④受け取ってconsoleでデータを確認しよう！

情報の登録

入力した情報をDBに登録し，登録後の最新情報を取得

■必要なもの

- ・ javascriptのコード（**ajax_index.html**）

PHPファイルに対してリクエストを送る処理．

データを送る必要があるので，**\$.ajax()**を使用．

- ・ PHPのコード（**ajax_post.php**）

データを受け取る処理．（`$_POST`を使用）

DBにデータを登録する処理．（ここまでinsert.phpと同じ）

登録後のデータを取得する処理．（ajax_get.phpと同じ）

取得したデータをjson形式で返す．（ajax_get.phpと同じ）

入力した情報をDBに登録し，登録後の最新情報を取得

■処理の流れ

- ①javascriptからPHPファイルにリクエストを送る．(js)
- ②DBにデータを登録する．(PHP)
- ③登録後の最新のデータをDBから取得する．(PHP)
- ④取得したデータをjson形式にして出力する．(PHP)
- ⑤javascriptでデータを受け取る．(js)←ここまでつくろう！
- ⑥（受け取ったデータをブラウザに表示する）

※DBのデータがconsoleに出力されればOK！

【解説】 ajaxの基本的な構文

```
$.ajax({  
    dataType: 'json',           //取得するデータの形式. 今回はjsonで受け取る  
    url: 'test.php',           //情報を送信する先のアドレス  
    type: 'POST',              //送信の方法  
    data: 'hoge'               //送信する内容（この辺り他にもたくさん存在！）  
})  
  
    .done(function(data){...}) //成功時  
    .fail(function(...){...})  //失敗時  
    .always(function(){...});  //どちらでも（この辺りは$.getJSONと同じ）
```

①⑤javascriptからPHPファイルにリクエストを送る→受け取り

関数の定義

ajax_index.html

```
function insertData() {  
    const url = 'ajax_post.php';           //urlの指定  
    const value = {  
        task: $('#task').val(),           //キーと値の指定  
        deadline: $('#deadline').val(),   //PHPでは$_POST['キー名']で受取  
        comment: $('#comment').val()  
    };  
};
```

続く...

①⑤javascriptからPHPファイルにリクエストを送る→受け取り

ajax_index.html

```
$.ajax({  
  dataType: 'json',  
  url: url,  
  type: 'POST',  
  data: value,
```

```
}).done(function (data) {
```

```
    console.log(data);
```

//最新データの表示

```
    })
```

```
.fail(function (XMLHttpRequest, textStatus, errorThrown) {...})
```

```
.always(function () {...});
```

```
}
```

①⑤javascriptからPHPファイルにリクエストを送る→受け取り

関数の実行

ajax_index.html

```
$('#send').on('click', function () {
```

```
    insertData();
```

```
    $('#task').val("");
```

```
    $('#deadline').val("");
```

```
    $('#comment').val("");
```

```
});
```

//送信ボタンクリック時に

//送信関数を実行

//入力部分を空にする

②③④DBからデータを登録→取得→出力（実は全部コピー！！）

ajax_post.php

...（前半はinsert.phpと同じ）

...（データの受取，SQLの作成など）

```
$status = $stmt->execute();
```

...（後半はajax_get.phpと同じ）

...（SQLの作成部分から同じように記述）

```
$sql = 'SELECT * FROM php02_table ORDER BY deadline DESC';
```

...

```
echo json_encode($res);
```

※PHPファイルの中で2回SQLの作成と実行を行う点に注意！！

ajax練習②

■処理の流れ

- ①\$.ajax()でリクエストを送ろう！
- ②DBにデータを登録しよう！
- ③最新の情報を取得しよう！
- ③json形式にして出力しよう！
- ④受け取ってconsoleでデータを確認しよう！

※送信したデータ含めて全部consoleに表示されればOK！

課題

【課題】PHPで好きなものを！！

PHPとDB使っていれば自由！！

例：

- ・ ブログ記事投稿アプリ
- ・ 掲示板アプリ
- ・ 投票アプリ
- ・ 写真共有アプリ
- ・ twitter風

※DBの情報のフィルタリングや順番の変更など工夫しよう！

提出は次週木曜日「23:59:59」まで！！

チュータリングタイム

17:00までは一人でもくもく
後半は近くのメンバーで教え合おう！