

PHP 01



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

■webの仕組み

■PHP概要

■PHP基礎

- ・ 変数など
- ・ 練習

■サーバへデータ送信

- ・ getとpost

■ファイルへデータを書き込む

■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

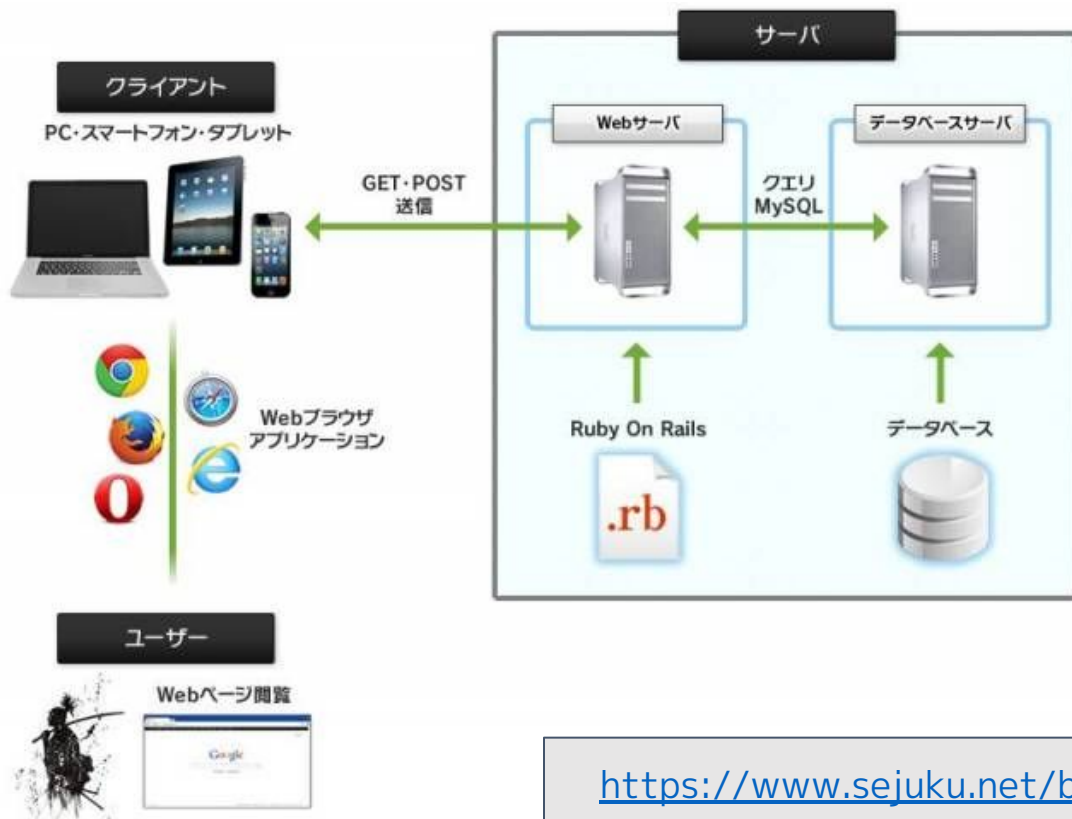
- 書いたら保存しよう！

command + s

ctrl + s

webの仕組み

webの仕組み



<https://www.sejuku.net/blog/1046/>より引用

URL

■URLとは

- ・ web上にある情報（ファイル）の場所を指し示す住所.
- ・ Uniform Resource Locatorの略.

■例

サーバ名

ファイル名

<http://www.〇〇〇〇.jp/△△△△/index.html>

ディレクトリ名

サーバとクライアント

■サーバで動作する言語（サーバサイド）

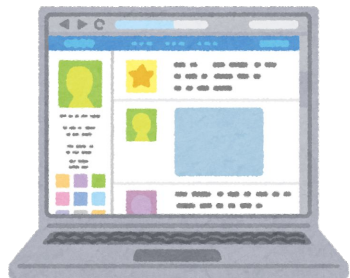
- ・ サーバ上でプログラムが実行される.
- ・ PHP, ruby, python, JAVA, (node.js), etc...

■クライアント（webブラウザ）で動作する言語（クライアントサイド）

- ・ webブラウザがプログラムを実行する.
- ・ html, css, javascript

サーバサイド言語の仕組み

※ 言語によらず、ファイル（プログラム）はサーバ上に存在



送られてきたhtmlを実行

- ・ こういう情報がほしい
- ・ こういう処理をしたい
- ・ 例：index.phpにアクセス

http通信

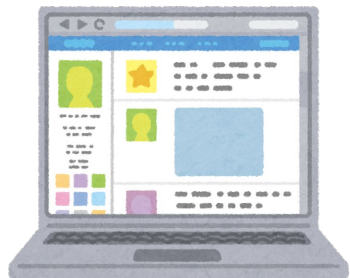
http通信

- ・ 処理した結果のデータ
- ・ 構成したhtml



クライアントサイド言語の仕組み

※ 言語によらず、ファイル（プログラム）はサーバ上に存在



送られてきたhtmlを実行

- ・ こういう情報がほしい
- ・ こういう処理をしたい
- ・ 例：index.htmlにアクセス

http通信

http通信

- ・ リクエストしたhtml



index.html

ファイル（の中身）を送信

処理は実行しない！

javascriptもこのタイミングで実行！！

PHP概要

サーバサイド（PHP）で実行されること

■例

- ・ twitter → ツイート, 検索, タイムラインの表示, etc
- ・ facebook → 投稿, 検索, 記事の更新, コメント, etc
- ・ wordpress → ブログ記事の投稿, 編集, 削除, etc

※上の例はPHPで作られているわけではない！！

サーバサイド (PHP) で実行されること

■例

- ・ twitter → ツイート, 検索, タイムラインの表示, etc
- ・ facebook → 投稿, 検索, 記事の更新, コメント, etc
- ・ wordpress → ブログ記事の投稿, 編集, 削除, etc

「登録」 「表示」 「更新」 「削除」

サーバサイド（PHP）で実行されること

■例

- ・ twitter → ツイート, 検索, タイムラインの表示, etc
- ・ facebook → 投稿, 検索, 記事の更新, コメント, etc
- ・ wordpress → ブログ記事の投稿, 編集, 削除, etc

「Create」 「Read」 「Update」 「Delete」

サーバサイド (PHP) で実行されること

■例

- ・ twitter → ツイート, 検索, タイムラインの表示, etc
- ・ facebook → 投稿, 検索, 記事の更新, コメント, etc
- ・ wordpress → ブログ記事の投稿, 編集, 削除, etc

「Create」 「Read」 「Update」 「Delete」



「CRUD」

PHP基礎

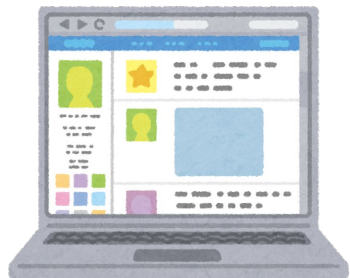
準備

■ サンプルフォルダを下記のディレクトリへ移動しましょう！

- ・ windows : C\MAMP\htdocs
- ・ mac : /Applications/MAMP/htdocs

htdocsの仕組み

htdocsディレクトリをwebサーバとして扱う！！



ブラウザ

PHPファイルにアクセス

PHP実行結果

- ・ 処理した結果のデータ
- ・ 構成したhtml



htdocsディレクトリ
擬似的なサーバ (localhost)
ここにPHPファイルを配置

PHPの基礎

■PHPファイルの作成

- ・ 拡張子が「.php」
- ・ 例：「index.php」 「insert.php」

■開始タグと終了タグ

- ・ 「<?php」で始まり「?>」で終了
- ・ 開始タグと終了タグの間に書かれた処理がサーバ上で実行される！

PHPの基礎

■変数など

- ・ 変数は「**\$******」 ← \$で始める！
- ・ 例：

<code>\$num = 1;</code>	<code>// ← 処理の終わりは「;」 必須</code>
<code>\$name = 'engineer';</code>	<code>// ← 文字列は「'」 で囲む</code>
<code>\$1lang = 'php';</code>	<code>// ← 数値スタートはNG</code>
<code>\$lang2 = 'python';</code>	<code>// ← OK</code>
<code>\$_lang = 'javascript';</code>	<code>// ← OK</code>
- ・ 変数と文字列は「.」 でつなげる.
`$str = '今日は' . $lang2 . 'を勉強中';`

■条件分岐

```
$num = 1;           //←rand(0, 1);でも試してみよう！  
if ($num == 1) {  
    echo '値は1です';//←「echo」でブラウザに表示  
} else {  
    echo '値は1以外です';  
}
```

■練習1

rand(min, max);を使用しておみくじを作ろう！

```
$num = rand(**, **);  
if ($num == 1) {  
    $result = '大吉';  
} elseif (...) {  
    ...  
}
```

■PHPの処理結果をhtmlに埋め込み

<?=変数名?>でhtmlに埋め込みが可能

■例

```
<?php
```

```
    $result = '大吉';
```

```
?>
```

```
...
```

```
<h1>今日の運勢は<?=$result?>です！</h1>
```

【参考】

■ 配列

```
$arr = ['javascript', 'php', 'python', 'COBOL'];  
echo $arr[2];    //python
```

■ console.log的なやつ

var_dump(); で変数, 配列などの構造と中身を出力可能！

例：var_dump(\$arr);

※echoでも出力できるが, var_dump()では構造も見られる！

サーバに情報を送る

フォームを使って情報を送ろう！ ~get編

get.html

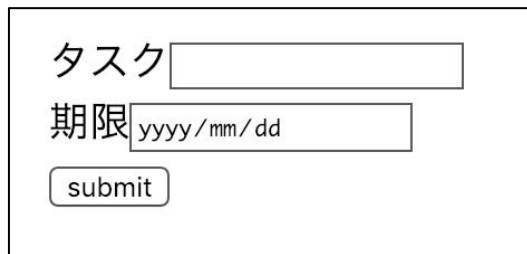
■フォームで宛先と方法を記述

```
<form action="get_confirm.php" method="get" >
```

```
  タスク<input type="text" name="task">
```

```
  期限<input type="date" name="deadline">
```

```
</form>
```



タスク

期限

■ポイント

- ・ **宛先のファイル**が必要！（action="*****.php"）
- ・ 送信方法を記述．「**get**」「**post**」の2種類！（後述）
- ・ **name属性**で受け取り側が処理する！

送られた情報を受け取ろう！ ~get編

get_confirm.php

■フォームの送信先に指定されたファイルで受け取り

`$task = $_GET['task'];` ← formのname属性

`$deadline = $_GET['deadline'];`

あとはPHPの変数として処理可能！

■getってなんだ？？

- ・ **URLに情報を追加**して送信する方法
- ・ 送信結果のページをブックマークさせたいときなどに便利！！



PHPの勉強を2019-01-31までに完了させましょう！

フォームを使って情報を送ろう！ ~post編

post.html

■フォームで宛先と方法を記述

```
<form action="post_confirm.php" method="post" >
```

```
  タスク<input type="text" name="task">
```

```
  期限<input type="date" name="deadline">
```

```
</form>
```

■ポイント

- ・ 「get」が「post」に変わるだけ！！
- ・ （宛先の変更も必要！）

送られた情報を受け取ろう！ ~post編

post_confirm.php

■フォームの送信先に指定されたファイルで受け取り

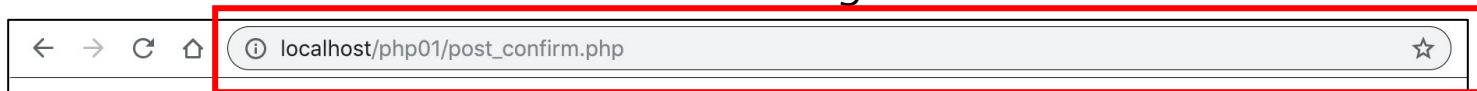
```
$task = $_POST['task'];
```

```
$deadline = $_POST['deadline'];
```

こちらも「GET」 → 「POST」の変更だけ！！

■postってなんだ？？

- ・ 情報を**見えないように送信**する方法
- ・ 個人情報など．（送れるデータ量がgetと比較して多い）



PHPの勉強を2019-01-31までに完了させましょう！

【参考】XSS : クロスサイトスクリプティング

■セキュリティを高める

- ・ formに悪意あるスクリプトを埋め込まれる場合がある.
- ・ 表示する際に対策できる.

`<?=$task?>`



`<?=htmlspecialchars($task, ENT_QUOTES); ?>`

※現時点では優先度低いので特に意識しなくてOK !

※最近はブラウザが優秀なので記述しなくてもわりと防いでくれる.

ファイル操作

送信したデータをファイルに書き込んで保存する

■やること

- ・ 入力したデータをtxtファイルに書き込む
- ・ 書き込んだデータを読み込んで表示する
- ・ 保存場所は「data」ディレクトリの「data.txt」

■必要なファイル

- ・ データを入力して送信するファイル (txt_form.php)
- ・ データを受け取ってファイルに書き込むファイル (txt_write.php)
- ・ ファイルのデータを読み込んで表示するファイル (txt_read.php)

txtファイルに書き込みたいデータを送信する

txt_form.php

■データ送信の流れ

- ・ 送信先のファイルを指定する（今回はtxt_write.php）
- ・ 送信の方式を指定する（get or post）←今回はpost
- ・ formにname属性を指定する

txtファイルに書き込みたいデータを送信する

txt_form.php

■データ送信の流れ

<form **method="post" action="txt_write.php">** //方式と宛

先

名前<input type="text" **name="name">** //name属性追加

日時<input type="date" **name="indate">**

コメント<input type="text" **name="comment">**

<button type="submit">Submit</button>

</form>

送信したデータをtxtファイルに書き込む

txt_write.php

■書き込みファイルの流れ

- ・ データを受け取る
- ・ 書き込み先のファイルを開く（なければ新たにファイルを作成）
- ・ 他の人が書き込まないようにロックする
- ・ データを書き込む
- ・ ロックを解除する
- ・ ファイルを閉じる

送信したデータをtxtファイルに書き込む

txt_write.php

■書き込みファイルの流れ

```
$name = $_POST['name'];           //データ受け取り
$indate = $_POST['indate'];
$comment = $_POST['comment'];
$str = $indate . ',' . $name . ',' . $comment;    //受け取ったデータを「,」で結合
$file = fopen('data/data.txt', 'a');             //ファイルを開く 引数はa
flock($file, LOCK_EX);                           //ファイルをロック
fwrite($file, $str . '\n');                       //データをファイルに書き込み
flock($file, LOCK_UN);                           //ロック解除
fclose($file);                                    //ファイルを閉じる
```

【参考】ファイルの操作

■引数の種類

r 読み込みのみで開く

r+ 読み込み/書き込み用に開く

w 書き込みで開く&内容を削除→ファイルがなければ作成

w+ 読み込み/書き込みで開く&内容を削除→ファイルがなければ作成

a 追加書き込みのみで開く→ファイルがなければ作成

a+ 読み込み/追加書き込みで開く→ファイルがなければ作成

送信したデータをtxtファイルに書き込む

■練習1

- ・ txt_form.phpのform欄の設定を記述しよう！
- ・ txt_write.phpでデータを受け取り，ファイルに書き込もう！
- ・ 書き込み処理の結果をdata.txtで確認しよう！

txtファイルを読み込んで表示する

txt_read.php

■読み込みファイルの流れ

- ・ 出力用の変数を用意する
- ・ csvファイルを開く（読み取り専用）
- ・ ファイルをロックする
- ・ txtファイルのデータを読み込んで出力用の変数に入れる
- ・ ロックを解除する
- ・ ファイルを閉じる
- ・ （html上で表示）

txtファイルを読み込んで表示する

txt_read.php

■読み込みファイルの流れ

```
$str = ""; //出力用の空の文字列
$file = fopen('data/data.txt', 'r'); //ファイルを開く（読み取り専用）
flock($file, LOCK_EX); //ファイルをロック
if ($file) {
    while ($line = fgets($file)) { //fgets()で1行ずつ取得→$lineに格納
        $str .= '<p>' . $line . '</p>'; //取得した1行ずつを$strに入れる
    }
}
flock($file, LOCK_UN); //ロック解除
fclose($file); //ファイル閉じる
```


txtファイルを読み込んで表示する

■練習2

- ・ txt_read.phpでdata.txtの内容を読み込もう！
- ・ 読み込んだデータをブラウザで表示しよう！

課題

【課題】 csvファイルでアンケート集計

■構成

- ・ 入力画面 (index.php)
- ・ 書き込みファイル (write.php)
- ・ 読み込みファイル (read.php)

■最低限ここまで！

- ・ 名前, email, 任意の質問を入力
- ・ 入力内容をcsv形式で「data/data.csv」に保存
- ・ 読み込みファイルでcsvファイルの内容を表示

※例によってアンケート項目とか適当でOK！

【課題】 csvファイルでアンケート集計

■ アップグレード

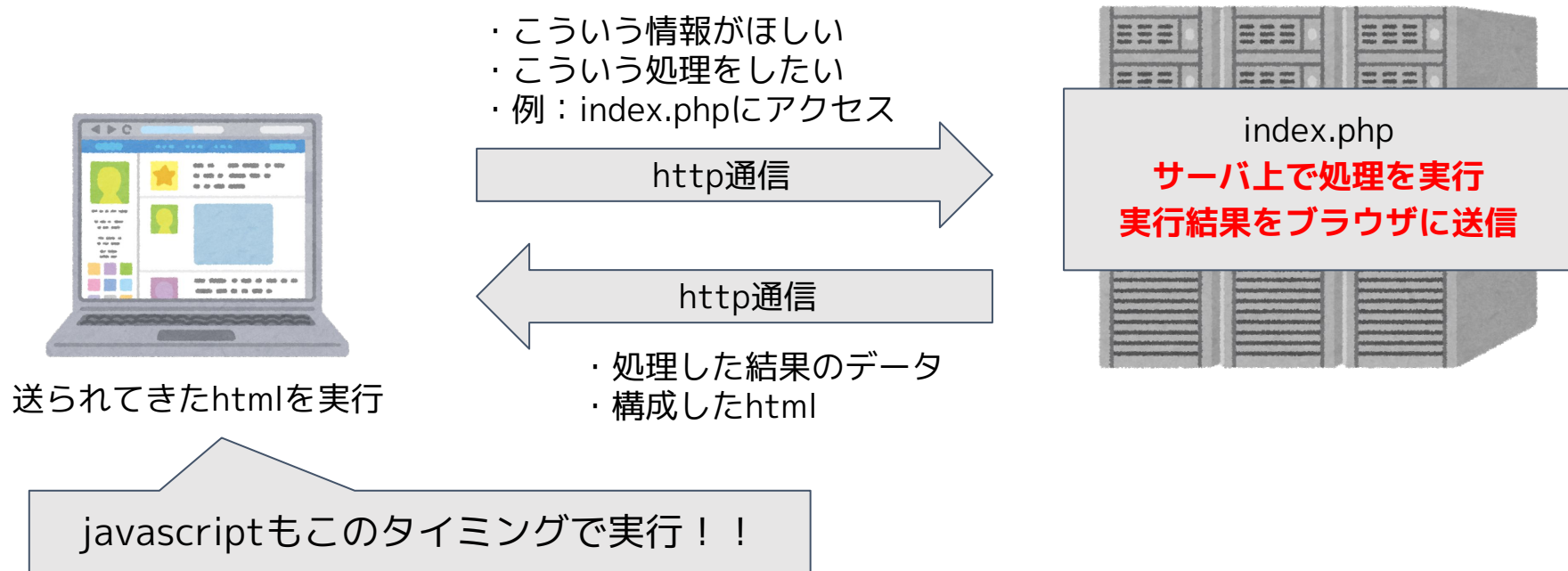
- ・ csvファイルの内容の集計結果をグラフ表示
- ・ 結果に応じた評価機能
- ・ 統計解析とか

■ 考え方

- ・ PHPはjavascriptと比較して自由度が低い. . . !
- ・ 狙ったデータを確実に表示できるように!!

順番が大事！！

サーバでPHPが実行 → ブラウザでjavascriptが実行



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！