

サービス指向システム設計

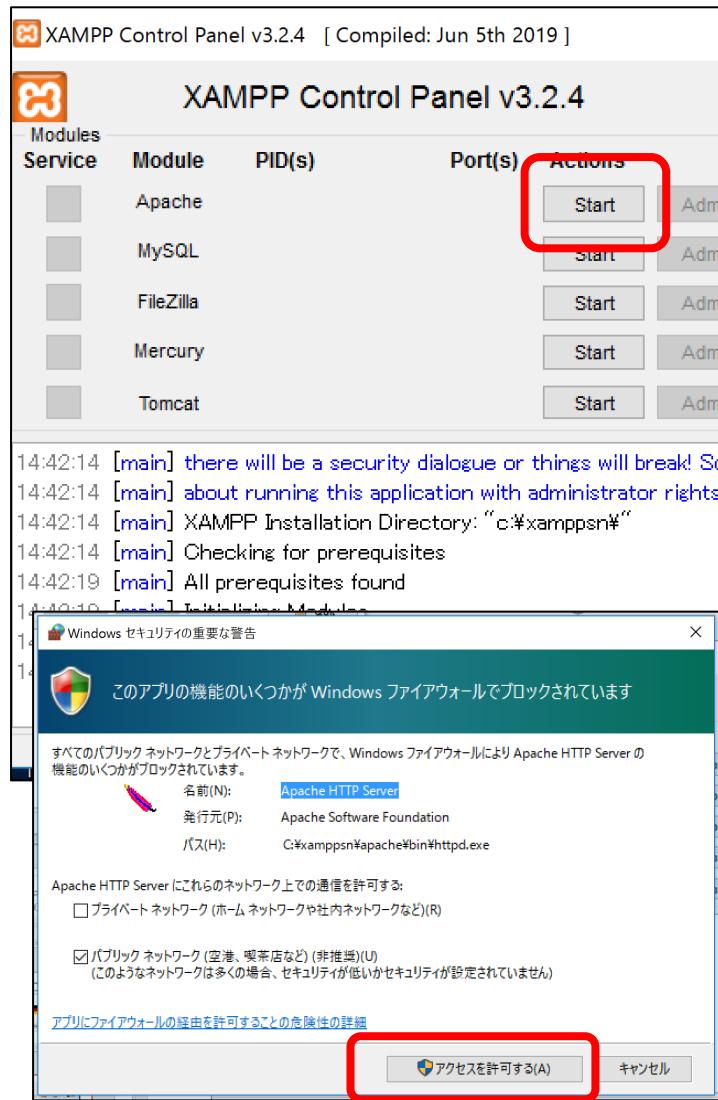
～PHPによるFormの処理～

Today's Goal

- PHPの動作確認
- PHPによるWeb Form
- Sticky Formの作成に挑戦

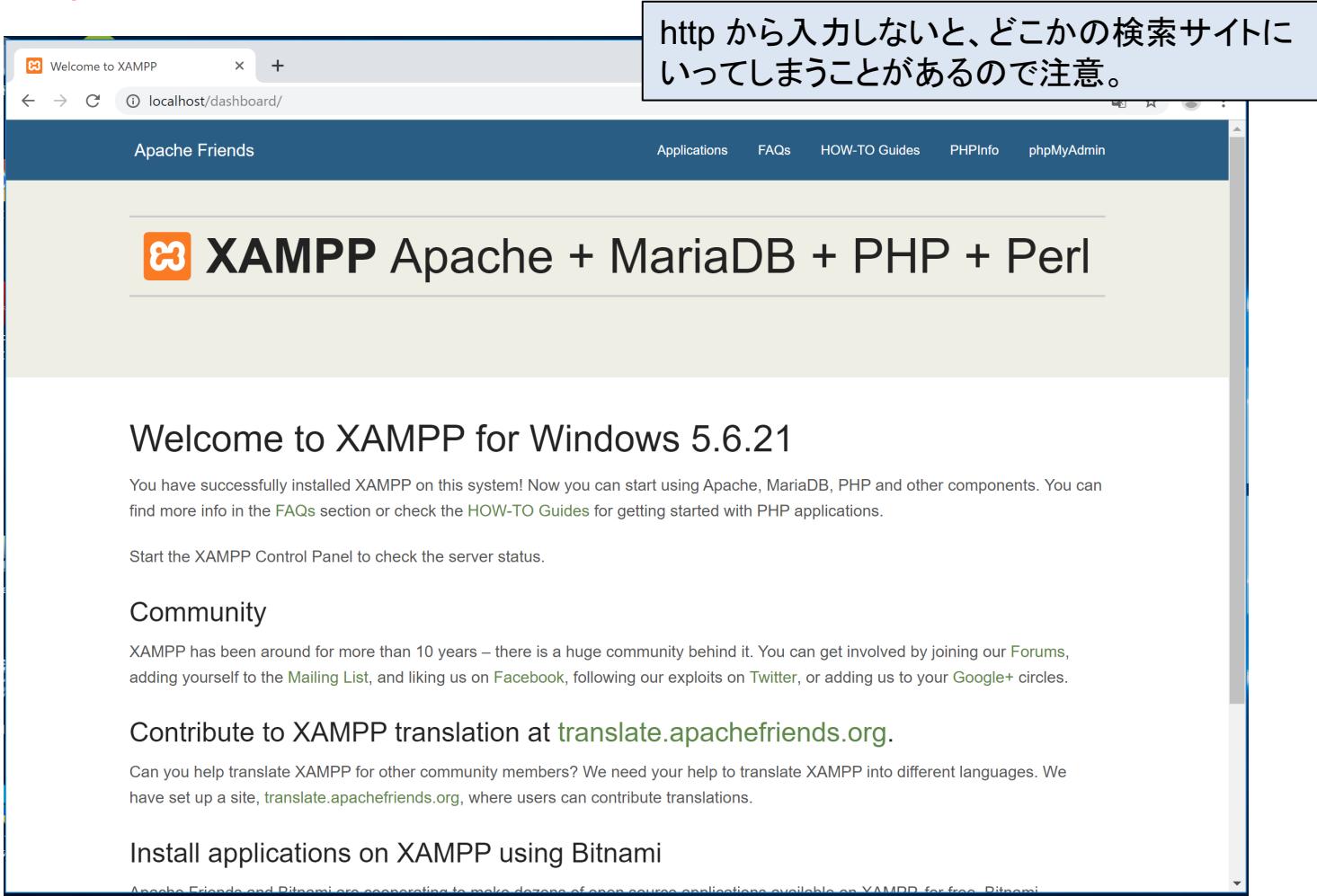
Webサーバの起動と終了

Windows: XAMPPの起動



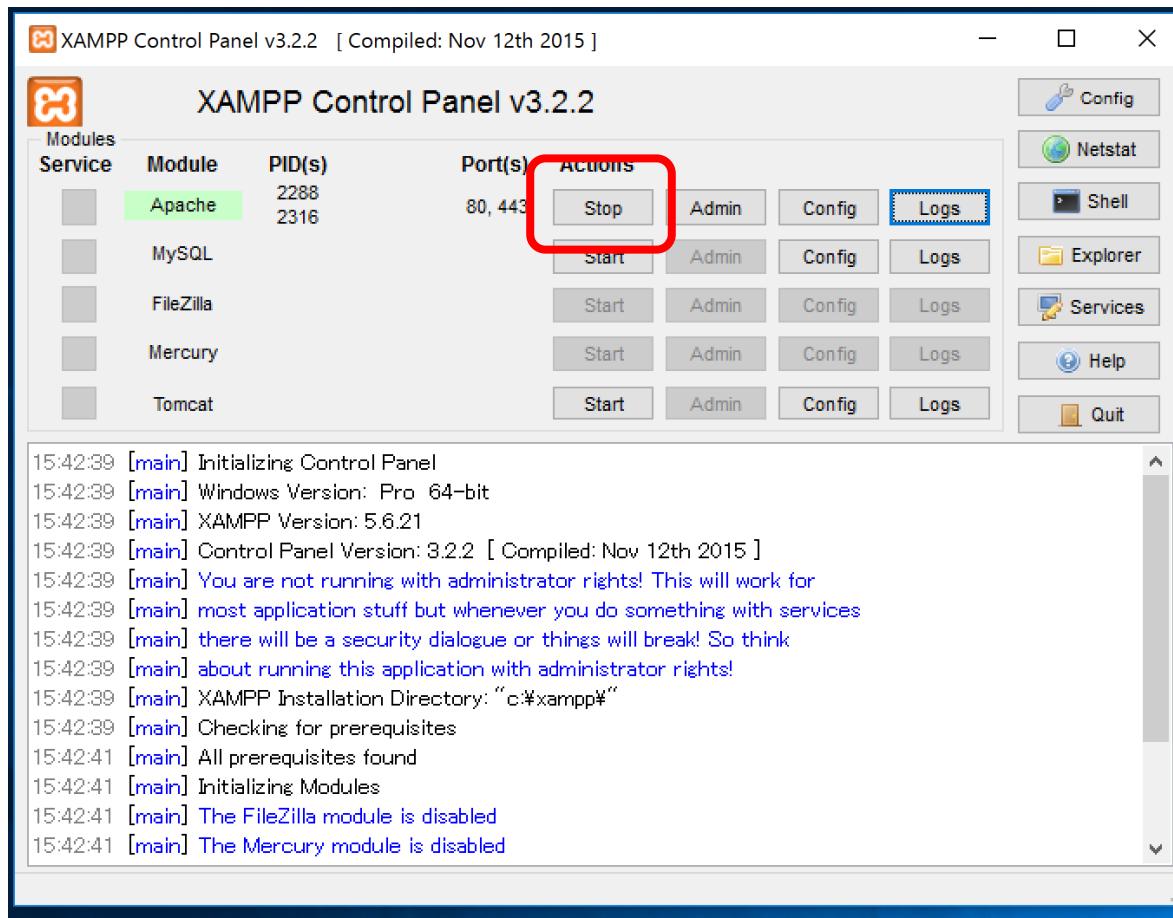
Windows: ブラウザでアクセス

- **http://localhost/ にアクセス**



Windows: 終了

- "Stop"をクリックして終了する。

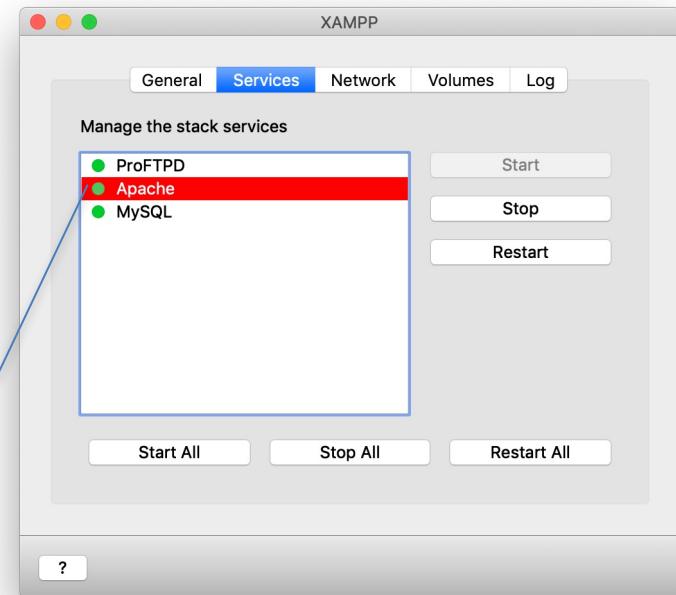


macOS : XAMPPの起動

- XAMPP.appを起動する



XAMPP.app を起動し、GeneralタブのStartをクリックすると、Apache と MySQLが自動的に起動する



少し待つと緑になる

macOS X : ブラウザでアクセス

The screenshot shows the XAMPP Control Panel on macOS. The 'Network' tab is selected, and the 'Enable' button is highlighted with a red box. A callout bubble points from this button to a text box containing the URL <http://localhost:8080>. Below the Control Panel, a browser window displays the XAMPP Apache Friends dashboard at localhost:8080/dashboard/. The dashboard header reads 'XAMPP Apache + MariaDB + PHP + Perl'. At the bottom of the dashboard, there is a note in a box: 'http から入力しないと、どこかの検索サイトにいってしまうことがあるので注意。' (If you enter it via http, it may appear in some search results, so be careful.)

http://localhost:8080
にアクセスすると
Apacheに接続できる

Welcome to XAMPP for 7.3.6-1

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You
the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

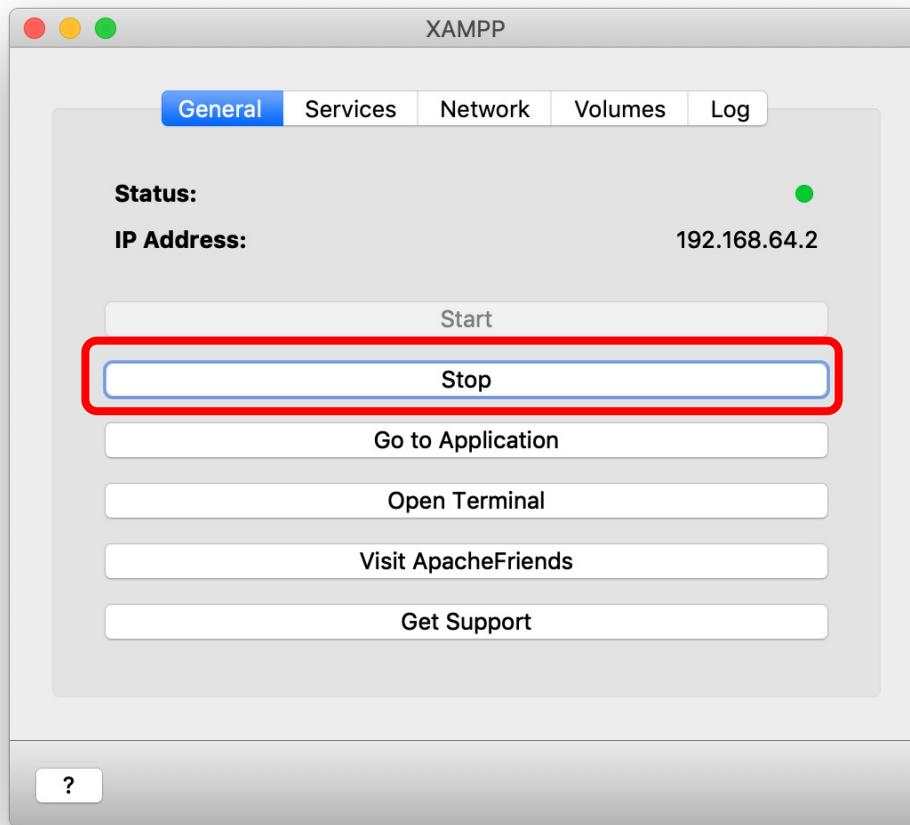
for development purposes. It has certain configuration settings that make it easy to develop locally but that are
to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make

sure you understand the implications and you checked the [FAQs](#) to learn how to protect your site. Alternatively you can use [WAMP](#),
[MAMP](#) or [LAMP](#) which are similar packages which are more suitable for production.

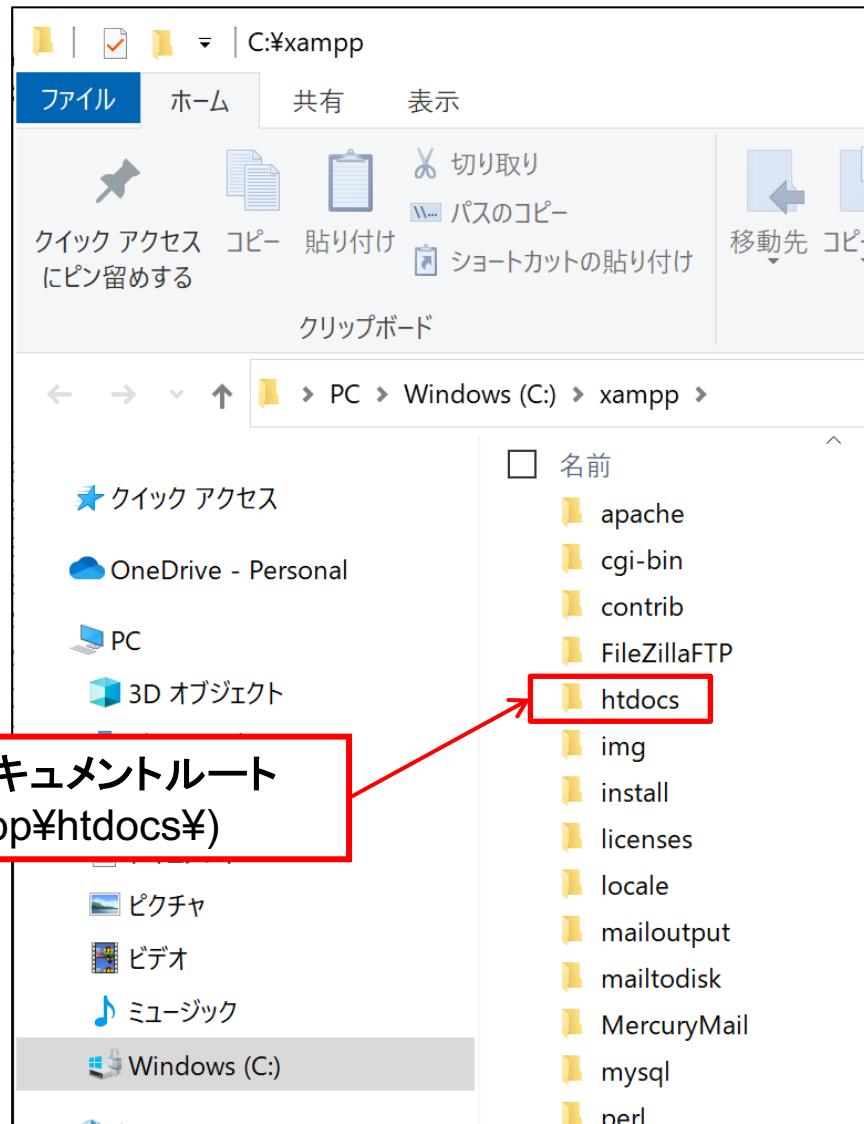
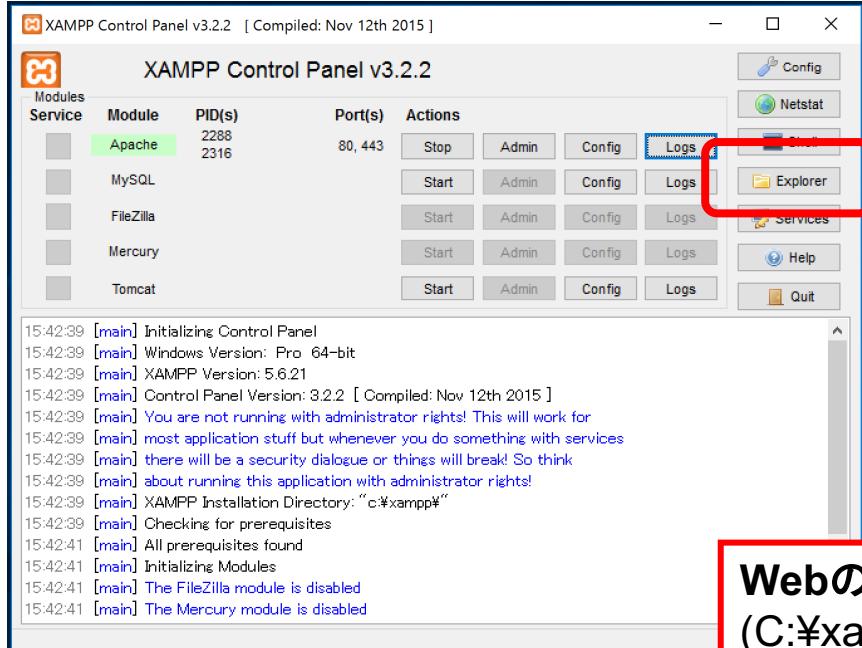
http から入力しないと、どこかの検索サイトに
いってしまうことがあるので注意。

macOS : 終了

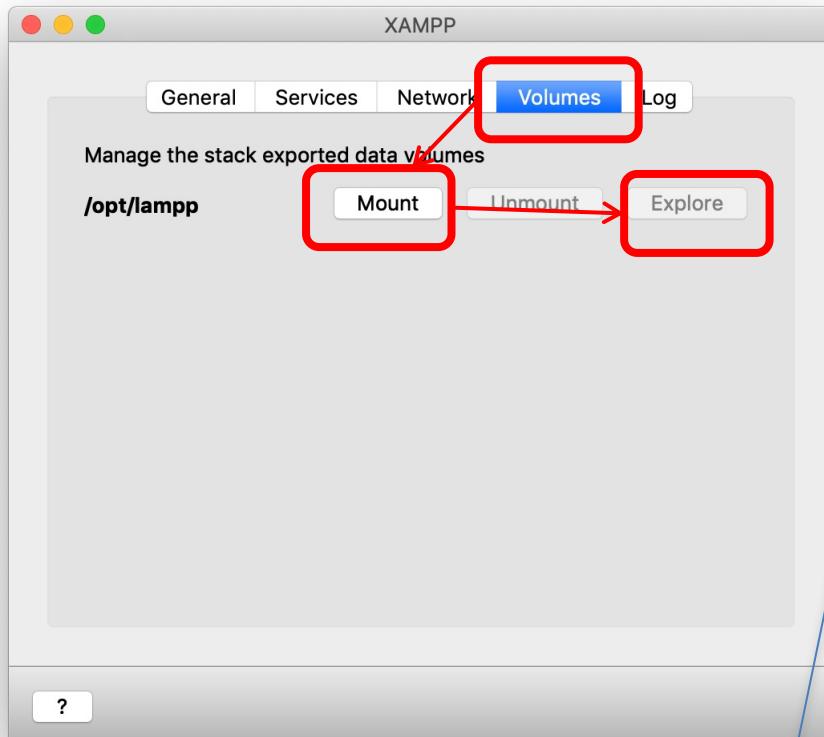
- General の Stop をクリックする。



Windows: DocumentRoot



macOS: DocumentRoot



ドキュメントルートはここ

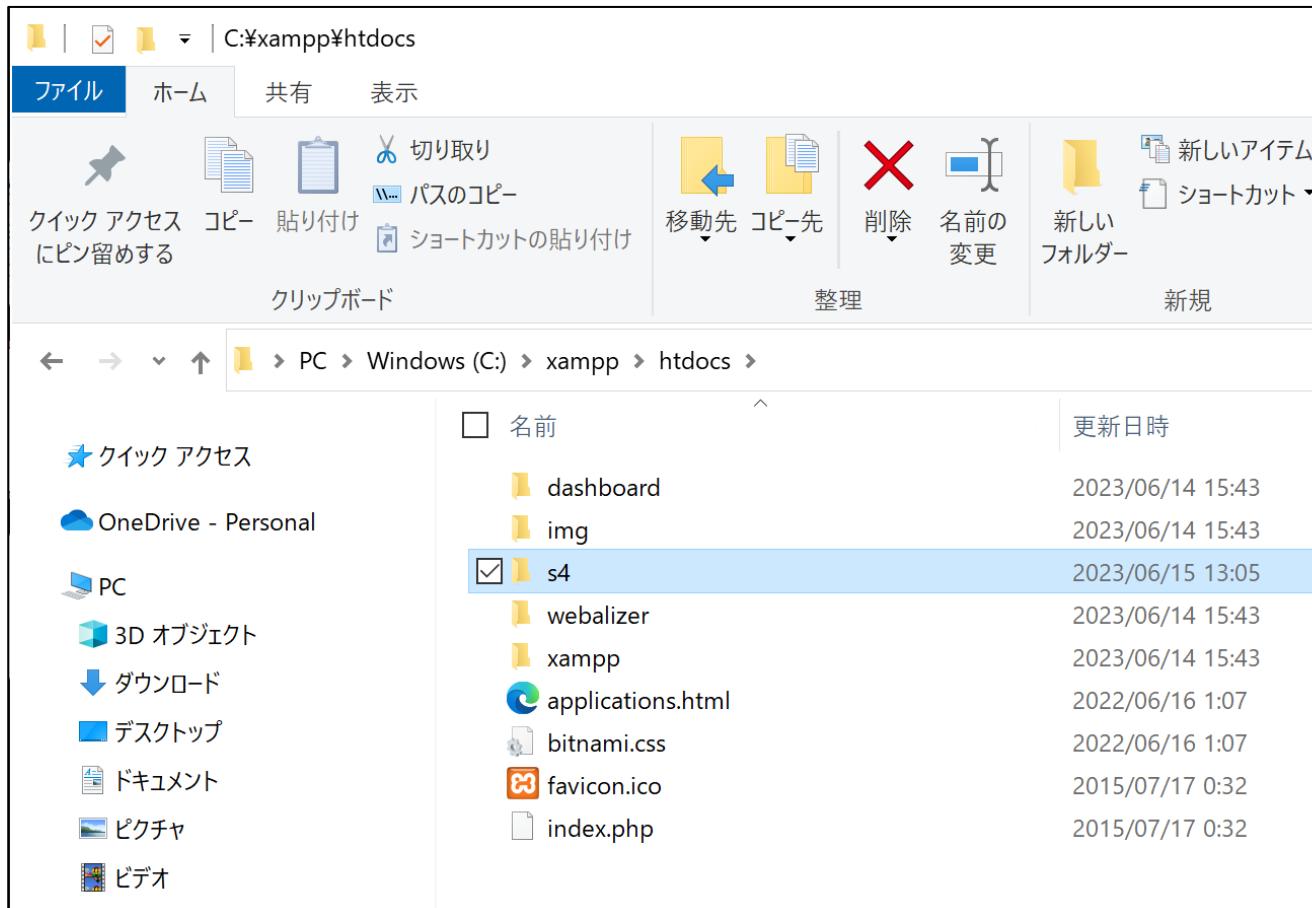
名前	変更日	サイズ
apache2	2019年6月14日 19:17	--
bin	2019年6月14日 19:17	--
cgi-bin	2019年6月14日 19:17	--
ctlscript.sh	2019年6月14日 19:17	27 KB
error	2019年6月14日 19:17	--
etc	2019年6月14日 19:17	--
htdocs	2019年6月14日 19:19	--
icons	2019年6月14日 19:19	--
img	2019年6月14日 19:17	--
killprocess.bat	2019年3月8日 20:20	299 バイト
lamp	2019年6月14日 19:17	16 バイト
lib	2019年6月14日 19:17	--
libexec	2019年6月14日 19:17	--
licenses	2019年6月14日 19:17	--
logs	今日 14:07	--
manager-linux-x64.run	2017年2月27日 18:04	3.4 MB
modules	2019年6月14日 19:17	--
mysql	2019年6月14日 19:17	--
pear	2019年6月14日 19:17	--
php	2019年6月14日 19:17	--
phpmyadmin	2019年6月14日 19:19	--
proftpd	2019年6月14日 19:17	--
properties.ini	2019年6月14日 19:18	906 バイト
README-wsrep	2019年5月14日 3:13	20 KB
README.md	2019年5月14日 3:13	3 KB
RELEASENOTES	2019年6月14日 18:52	2 KB
sbin	2019年6月14日 19:17	--
scripts	2019年6月14日 19:18	--
share	2019年6月14日 19:17	--
temp	2019年6月14日 19:17	--
THIRDPARTY	2019年5月14日 3:13	86 KB
uninstall	2019年6月14日 19:17	8.8 MB
uninstall.dat	2019年6月14日 19:18	239 KB
var	今日 14:07	--
xampp	2013年7月22日 22:16	15 KB

この授業でのXAMPPの使用場所

- C:\xampp\htdocs\s4\
<http://localhost/s4/>
- C:\xampp\htdocs\s4\test\
<http://localhost/s4/test/>
- C:\xampp\htdocs\s4\r2\
<http://localhost/s4/r2/>

【演習】自分のコンテンツを見る①

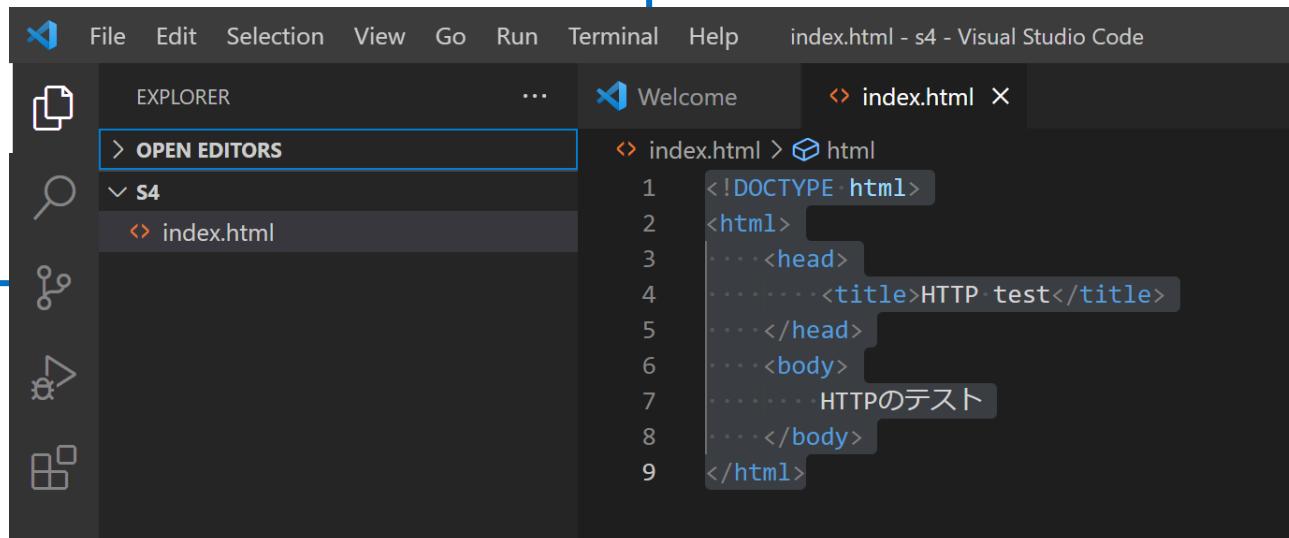
- DocumentRoot の下に本講義用のフォルダ s4 を作成する。



【演習】自分のコンテンツを見る②

- テキストエディタを用いて、以下の内容を **index.html** というファイル名で作成し、s4フォルダの中におく。

```
<!DOCTYPE html>
<html lang="ja">
<head>
<title>HTTPテスト</title>
</head>
<body>
HTTPのテスト
</body>
</html>
```

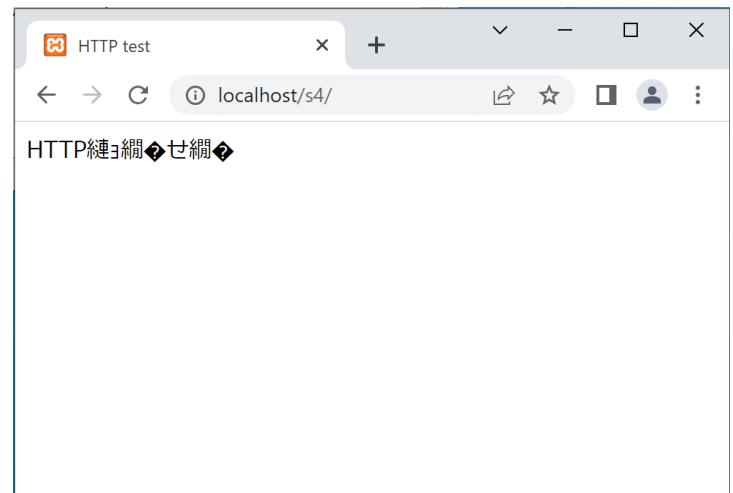
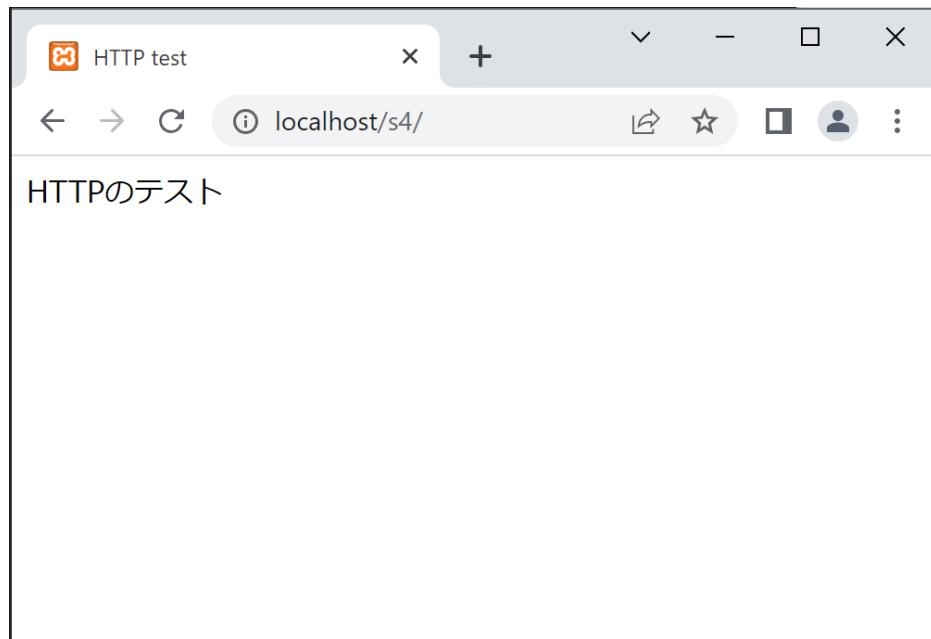


The screenshot shows the Visual Studio Code interface. The title bar reads "index.html - s4 - Visual Studio Code". The left sidebar is the Explorer view, which has "OPEN EDITORS" highlighted. Below it, there is a tree view of the "s4" folder, with "index.html" selected. The main editor area on the right displays the HTML code. The line numbers 1 through 9 are visible on the left side of the editor. The code itself is:

```
1 <!DOCTYPE html>
2 <html>
3 ...<head>
4 ...<title>HTTP test</title>
5 ...</head>
6 ...<body>
7 ...<... HTTPのテスト
8 ...</body>
9 </html>
```

【演習】自分のコンテンツを見る③

- Webブラウザを使い、以下のURLで自分のページが表示できることを確認する。
 - <http://localhost/s4/> (windows)
 - <http://localhost:8080/s4/> (macOS)



【演習】自分のコンテンツを見る④

- 以下の内容を **date.php** というファイル名で作成し、s4フォルダの中におく。

```
<!DOCTYPE html>
<html lang="ja">
<head>
<title>PHP sample</title>
</head>
<body>
<h1>PHPのサンプル</h1>
今日の日付は
<?php
date_default_timezone_set("Asia/Tokyo");
echo date("Y/m/d"); ?>
です。
</body>
</html>
```

date() :
指定したフォーマットで日時を返す

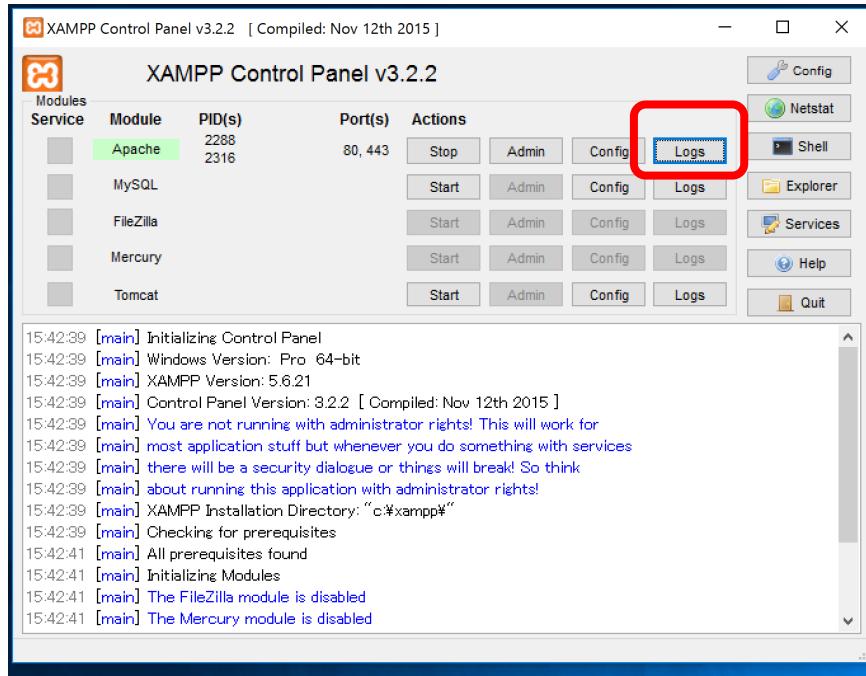
Y,y	年(4ケタ or 2ケタ)
m,M	月(01-12 or Jan,Feb,...)
d,j	日(01-31 or 1-31)
D,N	曜日(Mon,Tue,... or 1-7)
A,a	AM,PM or am,pm
g,G,h,H	時(1-12, 0-23, 01-12, 00-23)
i	分(00-59)
s	秒(00-59)

【演習】自分のコンテンツを見る⑤

- Webブラウザを使い、以下のURLで自分のページが表示できることを確認する。
 - <http://localhost/s4/date.php> (windows)
 - <http://localhost:8080/s4/date.php> (macOS)

The screenshot shows a web browser window with a blue header bar containing the text "PHP sample". The main content area displays the following text:
PHPのサンプル
今日の日付は2024/06/24です。

Windows : ログの確認



127.0.0.1 -- [16/Jul/2024:14:02:42 +0900] "GET /s4/index.html HTTP/1.1" 200 143 "-"
"Mozilla/5.0 (Windows NT 5.1; rv:14.0) Gecko/20100101 Firefox/93.0.0.1"

今後の作業で意図した通りに動かない場合には、
アクセスログやエラーログをみると原因がわかることがある。

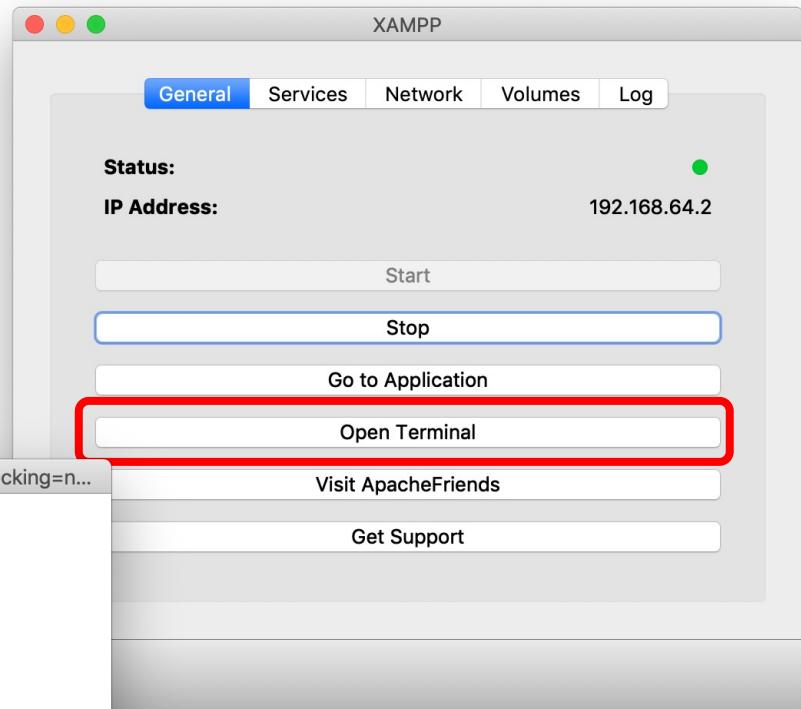
macOS : ログの確認

- Open Terminal をクリック
- cd /opt/lampp/logs
- tail -f access_log
(ctrl-C で終了する)

```
Linux debian 4.9.0-9-amd64 #1 SMP Debian 4.9.168-1+deb9u2 (2019-05-13) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~# cd /opt/lampp/logs/
root@debian:/opt/lampp/logs# ls
access_log  cgisock.808 error_log  php_error_log
cgisock.710  cgisock.918 httpd.pid  ssl_request_log
root@debian:/opt/lampp/logs# tail -f access_log
127.0.1 - - [18/Jun/2019:05:45:21 +0000] "GET /dashboard/stylesheets/normalize.css HTTP/1.1" 200 6876
127.0.0.1 - - [18/Jun/2019:05:45:21 +0000] "GET /dashboard/javascripts/all.js HTTP/1.1" 200 188385
127.0.0.1 - - [18/Jun/2019:05:45:21 +0000] "GET /dashboard/images/xampp-logo.svg HTTP/1.1" 200 5427
127.0.0.1 - - [18/Jun/2019:05:45:21 +0000] "GET /dashboard/images/fastly-logo.png HTTP/1.1" 200 1770
127.0.0.1 - - [18/Jun/2019:05:45:21 +0000] "GET /dashboard/images/bitnami-xampp.png HTTP/1.1" 200 22133
127.0.0.1 - - [18/Jun/2019:05:45:21 +0000] "GET /dashboard/images/social-icons@2x.png HTTP/1.1" 200 5360
127.0.0.1 - - [18/Jun/2019:05:45:25 +0000] "GET /ES// HTTP/1.1" 200 82
127.0.0.1 - - [18/Jun/2019:05:45:25 +0000] "GET /favicon.ico HTTP/1.1" 200 30894
127.0.0.1 - - [18/Jun/2019:05:45:36 +0000] "GET / HTTP/1.1" 302 -
127.0.0.1 - - [18/Jun/2019:05:45:40 +0000] "GET /ES/ HTTP/1.1" 200 82
```



ここまで の チェック

- Webページのデータを置く場所がわかるか
- Webページは表示されたか
 - 文字化けしたか？
- アクセスログを確認できたか

文字化けした？

```
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>PHP sample</title>
</head>
<body>
<h1>PHPのサンプル</h1>
今日の日付は
<?php echo date("Y/m/d"); ?>
です。
時刻は<?php echo date("H:i"); ?>です。
</body>
</html>
```

ファイルの文字コードを指定する metaタグをheadの最初に置く。

実際のファイルの文字コードと一致していないと、文字化けする。

PHPの基礎演習

【演習】フォームデータの受信

- s4フォルダの下に、以下の内容のform.htmlというファイルを作成する。

```
<html>
<body>
<h1>一言どうぞ</h1>
<form action="form.php" method="GET">
  あなたの名前:<input type="text" name="user"/><br/>
  一言:<input type="text" name="comment"/><br/>
  <input type="submit">
</form>
</body>
</html>
```

スライドの面積の都合で、
HTMLの先頭部分は
かなり省略しています。
ちゃんと書いてね！

【演習】フォームデータの受信

- s4フォルダの下に、以下の内容のform.phpというファイルを作成する。

```
<?php  
error_reporting( E_ALL );  
$uname=$_REQUEST['user'];  
$comment=$_REQUEST['comment'];  
  
echo<<<EOD  
<html>  
<body>  
{$uname}は、「{$comment}」としました。  
</body>  
</html>  
EOD;  
?>
```

スライドの面積の都合で、
HTMLの先頭部分は
かなり省略しています。
ちゃんと書いてね！

【演習】フォームデータの受信

- ・ ブラウザで以下のURLにアクセスし、動作を確認せよ。
 - <http://localhost/s4/form.html>
 - <http://localhost:8080/s4/form.html> (macOS)

The image shows two side-by-side browser windows. Both have a title bar with 'localhost/s4/form.html' and a status bar with 'localhost/s4/form.html'. The left window shows a form with fields 'あなたの名前:' containing 'ももたろう' and '一言:' containing 'おひとつどうぞ'. A '送信' button is visible. The right window shows the same URL but with a query parameter '?user=ももたろう'. It displays the message 'ももたろうは、「おひとつどうぞ」と言いました。'.

localhost/s4/form.html

localhost/s4/form.html

あなたの名前 : ももたろう

一言 : おひとつどうぞ

送信

localhost/s4/form.php?user=ももたろう

localhost/s4/form.php...

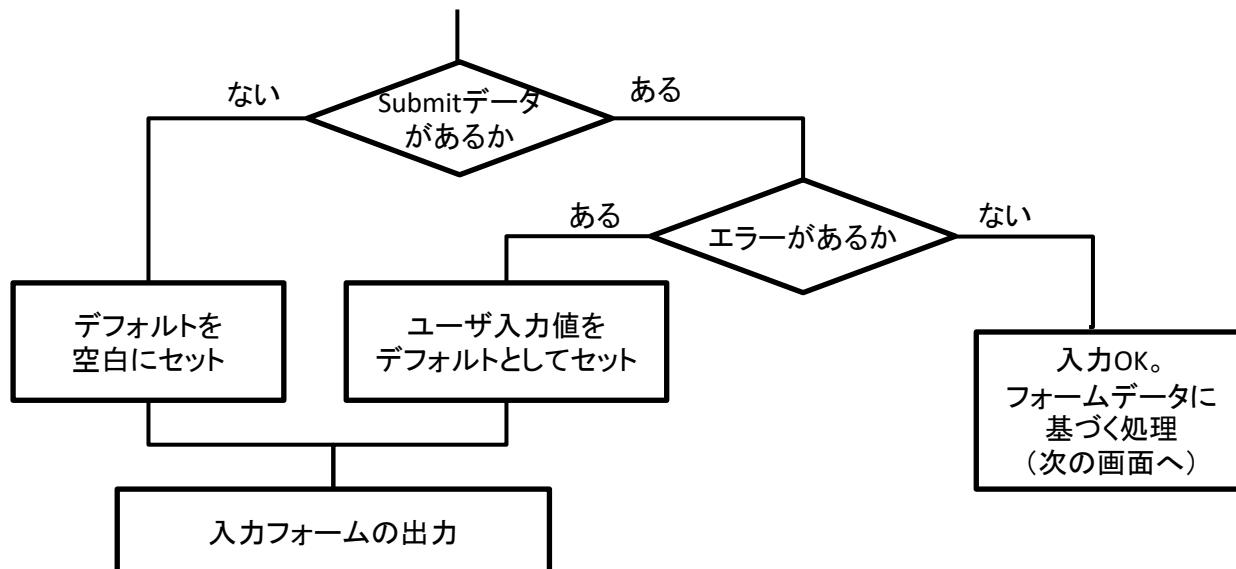
ももたろうは、「おひとつどうぞ」と言いました。

Sticky Form

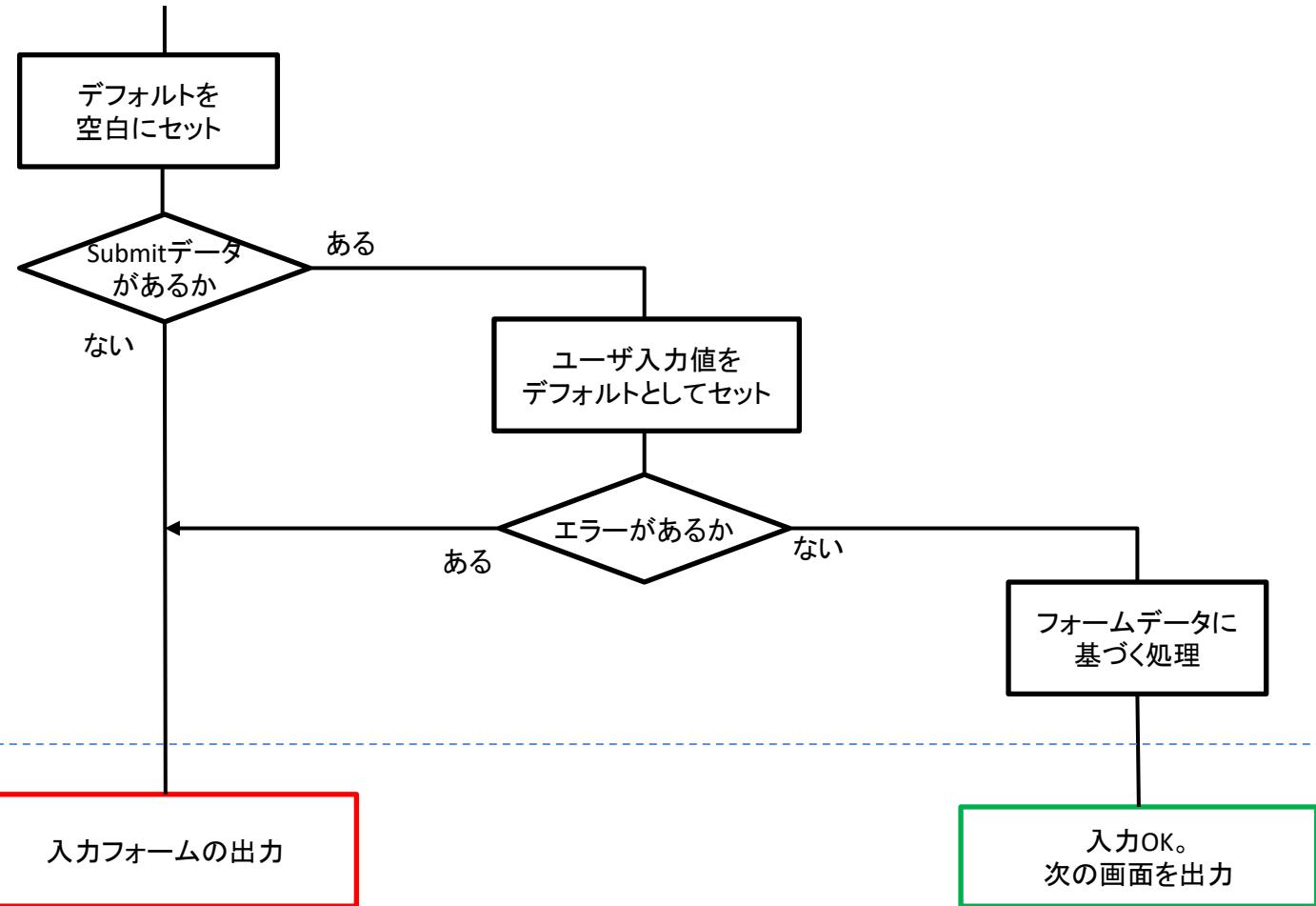
- Formで入力エラーがあったときに、全てのフォームデータを再入力させるのはユーザビリティを損なう。
- エラーメッセージを表示するとともに、
ユーザが前に入力した値が編集可能な状態で
あらかじめセットされた入力フォームが再表示されると、修正が最小限で済む。
- このように、
エラーがある間は 次の画面に
遷移しないタイプのフォーム
を Sticky Formと呼ぶ。

Sticky Formの構成

- Sticky Formは「入力フォームの表示」と「送信されたフォームデータの処理」を一つのスクリプトにまとめることで実現できる。
 - フォームは、データを自分自身に向けてsubmitする。
 - ユーザが前に入力したデータをフォームにセットするには、フォームの「デフォルト値」をセットすることで実現する。



Sticky Formの構成2



Sticky formの疑似コード

submitなしを想定した初期値設定をする
フォームデータがsubmitされてきたか？
もしsubmitされていたら
再入力に備えて初期値設定を修正する。

全てのフィールドの正当性を検証する。
全てのフィールドが記入されていて、正当性もOKか？

OKなら
フォームデータに基づく処理を行う
次の画面を表示
おわり

入力フォームを表示

Sticky formの疑似コード

エラー表示 + 処理 / 表示の分離版

submitなしを想定した初期値設定をする

フォームデータがsubmitされてきたか？

もしsubmitされていたら

再入力に備えて初期値設定を修正する。

全てのフィールドの正当性を検証する。

全てのフィールドが記入されていて、正当性もOKか？

OKなら

フォームデータに基づく処理を行う

NGなら

エラーの内容を記録する(まだ表示はしない)

フォームデータがsubmitされているなら

フォームエラーがないなら

次の画面を表示する

おわり

ELSE

エラーメッセージを表示する

ELSE

入力フォームを表示

投稿データ
処理部

表示部

【演習】Sticky form

- 次の動きをするように、sticky.phpを作成せよ。
 - form.html, form.phpの両方の役割を sticky.php で行う。
 - ユーザ名とコメント欄の両方が入力された時にのみ投稿を受け付け、「●●さんは■■■と言いました。」と表示する。
 - 片方しか入力されなかった場合には、適切なエラーメッセージを表示し、再入力させる。
 - そのとき、すでに入力された方のフォームについてはユーザの入力をあらかじめセットしておく。
 - 正常に投稿が行われた時にも、次の入力のための入力フォームを表示する。初期値を空欄とする。

sticky.php (1/2)

```
<?php  
$me=$_SERVER['PHP_SELF'];  
$submitted=false; $errmsg="";  
$defuname=""; $defcomment="";  
  
if ( isset($_REQUEST['user']) ) {  
    $submitted=true;  
    $uname=$_REQUEST['user']; $defuname=$uname;  
}  
if ( isset($_REQUEST['comment']) ) {  
    $submitted=true;  
    $comment=$_REQUEST['comment'];  
    $defcomment=$comment;  
}  
  
if ($submitted){  
    if ($uname == ""){ $errmsg .= "ユーザ名が入力されていません.<br>"; };  
    if ($comment == ""){ $errmsg .= "一言が入力されていません.<br>"; };  
}  
if ( $errmsg === "" ){ //エラーが起きてなかったら入力欄は空白に戻す。  
    $defuname = ""; $defcomment="";  
}
```

投稿データ処理部

(続く)

```
echo<<<EOD
<!DOCTYPE html>
<html>
<body>
EOD;

if ($submitted){
    if ( $errmsg === ""){ // seems ok.
        echo "<div class='msg'>{$uname}は、「{$comment}」と言いました。</div>";
    } else { // submitted and error occured
        echo "<div class='errmsg'>{$errmsg}</div>$n";
    }
}

echo<<<EOD
<h1>一言どうぞ</h1>
<div id='myform'>
    <form action="$me" method="GET">
        あなたの名前:<input type="text" name="user" value="$defuname"/><br/>
        一言:<input type="text" name="comment" value="$defcomment"/><br/>
        <input type="submit">
    </form>
</div>

</body>
</html>
EOD;
?>
```

表示部

CSSの追加

```
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="my.css" />
</head>
<body>
<h1>一言どうぞ</h1>
```

- my.css

```
div errmsg {
  color:red;  font-size:large;
  border:solid; border-width:1px; background-color:#ffeeee;
}

div msg {
  color:blue;
  font-size:large;
  border:solid; border-width:1px; background-color:#eeeeee;
}
```

【演習】sticky.php の改造

- ・他の種類の入力を加えてみよう。
 - テキスト入力欄を増やす
 - ラジオボタン
 - セレクトボックス ←難易度高い

selectbox型のフォームの sticky formは難しい

```
<select name="example1">  
<option value="Fruit1">りんご</option>  
<option value="Fruit2">みかん</option>  
<option value="Fruit3">ばなな</option>  
<option value="Fruit4">ぱいなつぶる</option>  
</select>
```

```
<select name="example1">  
<option value="Fruit1">りんご</option>  
<option value="Fruit2">みかん</option>  
<option value="Fruit3" selected>ばなな</option>  
<option value="Fruit4">ぱいなつぶる</option>  
</select>
```