PHP プログラミング

基礎

#1

1. はじめに

このテキストでは ,Web アプリケーションのプログラミングの初歩を学びます . プログラムの学習は個々の学習進度の差が大きくなりがちです . 課題が早く終った人は周りの人を手助けしてあげて下さい . 他の人の作ったプログラムの不具合を発見する能力もプログラムに関わる業務では必要になります . 失敗や間違いを恐れないで下さい . 失敗や間違いを見つけ , 修正する経験は実務でも必要になります .

凡例

コード例において , グレーの字は既に記述されている部分 , 黒字は新しく変更された部分を表しています . 前後のグレーの部分を参考にして , 実際のファイルの変更箇所を探してください .

注意

このテキストでは一つの目標に対して , 簡単に正しい結果を得られる手順と , 不具合を発生させてそれを解決することでより深い理解を得られるようになっている手順との二つの手順を用意しています . 実務では必ず不具合が発生するため , その不具合の原因と対処法をあらかじめ知っておくことは実際に仕事をする上でとても役に立ちます . また , プログラマがどういった点に気を付けて作業をしているかを知っていただければと思います . 授業中に余裕があれば , ぜひ不具合解決ルートを試してみてください . 簡単成功ルートは背景が白 , 不具合解決ルートは背景がグレーになっています .

実習環境について

プログラムは実際に書いて動かしてみないとなかなか身に付きません .「実習環境構築」を参考に , 実際に Web アプリケーションを動かせる環境を作って , 手を動かしながら学習してください .

ドキュメントルートについて

Web ブラウザで対象になる Web サーバのホストアドレスだけを指定した場合 (http://www.example.com/ など) に表示される , 基準となるフォルダのことをドキュメントルート (Document Root) と呼びます . このドキュメントルートが Web サーバのどのフォルダを指しているかは Web サーバの設定によって異なります .

このテキストではドキュメントルートを "/" で表わし ,"/form/" と書いてある場合はドキュメントルートフォルダの下の form フォルダを指す , とします .

ドキュメントルートにファイルをアップロードと書いてある場合は ,Web サーバ上のドキュメントルートにファイルをコピーすることを意味します . 実習環境として手元のコンピュータで XAMPP を使用している場合は ,/xammp/htdocs がドキュメントルートですので , そのフォルダにファイルをコピーしてください .

PHP の情報を探すには

・PHP 公式ページ (英語)
http://www.php.net

・PHP 言語リファレンス (日本語)
http://www.php.net/manual/ja/langref.php

・Google
http://google.co.jp

'PHP htmlentities' や 'PHP 文字列を連結する ' など、PHP+ キーワードで検索すると有益な情報が見つかりやすくなります。

ただし、PHP のバージョンや環境によって検索結果がそのまま使えない場合もありますので注意してください。また、記述が間違っている可能性もありますので、必ず複数の情報で確認するようにしてください。

2." プログラム " について

プログラムとは

コンピュータ上で動くソフトウェア (OS, アプリ ,etc...) は全てプログラムと言えます . コンピュータを動かすためには , コンピュータに対して " 足し算 " や " 掛算 " などの計算や ," 画面に色のついた点を表示する " などの細かい「命令」をする必要があります . 細かい命令はひとつでは仕事にならないため , 複数の命令をまとめるなくてはなりません .

定義的にはコンピュータに対する命令のまとまり (手続き) を記述したものがプログラムと呼ばれます .

コンピュータを人間に例えるならプログラムとは業務手順書や , 料理のレシピに近いでしょうか .

プログラムは最終的には電気信号に変換されて実行されますが , 人間にはその操作をするのがまず不可能なので , プログラミング言語と呼ばれる人工言語を使用してプログラムを記述します .(日本語や英語などの会話に使うような言語は自然言語と呼ばれます)

プログラミング言語によって書かれたプログラムは , プログラムを変換するプログラム (コンパイラ , インタプリタ等) によって変換 / 実行されます .

プログラミング言語

プログラミング言語は現在 , 数千種類あると言われていますが ,Web アプリケーションの作成をするときに実際に使用するのは 2 ～ 4 種類程度です . Ｗｅｂ業界でよく使われるプログラミング言語には表 1 のようなものがあります .

プログラミング言語の種類がこれほどまでに沢山あるのは , 言語毎に得意不得意があり , 必要に応じて使い分ける必要があるからです .

このテキストでは PHP を使って Web プログラミングを学びます .

プログラムを作成するために必要なこと

プログラムを作成するためにはプログラムを記述する能力が必要なのは当然ですが , それよりも重要なのが , 目的を把握して , その目的を細かく分割する能力です .

“ やりたいこと (目的)” を ," 出来る事 " に分割する ." 出来る事 " を組み合せて " やりたい事 " を実現する . それがプログラミングです . そのためには ," 出来る事 " は何か？を知らなくてはなりません .“ 出来る事 " は , プログラミング言語 , 環境 , 時代によってそれぞれ異なります . ひとつのやり方に固執するのではなく , 絶えず新しい情報に触れてより効率的な方法を身に付けるようにしてください .

Web アプリケーションとは

プログラムが動作する環境として , パソコンやスマートフォンの OS の上で直接動作するプログラムをネイティブアプリケーション ,Web ブラウザの上で動作するプログラムを Web アプリケーションと呼びます .Web アプリケーションにおいて実際にプログラムが実行されるのは ,Web サーバ上か Web ブラウザ上 , またはその両方です . 基本的にユーザデータをローカル PC には置かず , サーバ側に保存します .

Web アプリケーションを作成するためのプログラムを行なうことを Web プログラミングと呼びます .

表 1. 代表的なプログラミング言語

言語名	実行箇所	特徴
PHP	サーバサイド	HTML との親和性が高い．最近の Web アプリケーション開発での利用が多い
Javascript	クライアントサイド	主に HTML 要素を操作するために Web アプリケーション開発ではほぼ必ず使用される
ActionScript	Flash 専用	Javascript と親戚の間柄
Java	サーバサイド	Javascript と名前が似ているが別物なので注意
Ruby	サーバサイド	中規模な Web アプリケーションで使われることが多い
C#	サーバサイド	中規模～大規模な Web アプリケーションで使われることが多い

3.Web プログラマに必要な知識

Web の仕組み

大前提として ,Web(World Wide Web) の仕組みをよく理解していることが大切です . Web アプリケーションで出来ること , 出来ないことの判断を付けられるようになってください .

プログラミング言語

Web プログラマは , 最低でもクライアントサイドプログラミング言語とサーバサイドプログラミング言語のうち , 一つずつは習得している必要があります .

脆弱性にもつながるため , 安全なプログラミングのための知識が必要です .

また ,Web アプリケーションは場合によっては月間数十万回〜数千万回も使用されるため , 効率の良いプログラムが書けることも重要です .

セキュリティ情報

脆弱性を作り込まないためには言語 , 各種サーバアプリ ,Web ブラウザ , データベースなど , 広い範囲で最新のセキュリティ情報をチェックしておく必要があります .

ネットワーク

Web ブラウザと Web サーバ ,Web サーバとデータベースサーバ等を繋ぐのはネットワークです . 基本的な知識は持っておく必要があります .

データベース

Web アプリケーションではユーザデータの保存に主にデータベースを使用します . しかし , 正しく使い方を理解し効率の良い仕組みを作成できないと , ユーザを待たせたり , データが破壊されたりします . また , データベースの情報 , とくに個人情報などが漏洩すると大問題になりますので , 注意が必要です .

法律

著作権法 , 個人情報保護法 , 不正アクセス禁止法など , Web やコンテンツ関連法規の知識が無いと , 意図せず違法なシステムを構築することになりかねません . 詳細に知っている必要はありませんが , ポイントだけは押さえるようにしてください .

英語

情報を探していると , 英語圏の Web ページを参照することが多くなります . 殆どが専門用語なので , 専門用語さえ理解していれば , 意味を取るだけならそれほど難しさは無いとも言えます . また , プログラムの中でいろいろな要素に名前を付けることがありますが , 基本的に英語を使用します . スペルミスをすると正しく動かなかったり , 他のプログラマに伝わらないこともありますので , 正しいスペルを憶えておく , または都度調べるようにしてください .

プログラミング言語を勉強するには

プログラム（プログラミング言語）を学ぶ上で , 効率良く勉強するポイントを挙げておきます .

- 入力と出力の方法を知る
- 基本構造 (繰り返し , 条件分岐 , 関数等) を知る
- データ (文字列 , 数値 , 変数 , オブジェクト等) 操作の方法を知る

プログラムの参考書を買うときには , 最初の方は理解できて , 後ろの方は難しく感じるぐらいの本を選ぶのことをお勧めします . また , プログラム言語については入門書と辞書的な書籍の 2 冊を揃えておくと効率良く勉強できます .Web 上の情報は断片的なものが多いため , 最初に学ぶときには体系だててまとまっている書籍を基に学習を進めたほうが良いでしょう . 新しい言語を学ぶことに慣れてくれば ,Web 上の情報だけでも十分になると思います .

4. 実習環境の構築

一般に Web アプリはサーバホスティング会社等の Web サーバ上で動作させます． 直接本番サーバ上にファイルをアップロードするのではなく , 開発 / テスト用の環境を作ることによって本番環境を間違えて編集すなど取り返しの付かない事故を防げます .

XAMPP

XAMPP(ザンプ , エグザンプ) とは ,Web 制作で良く使われるサーバ環境を PC に簡単にインストールできるようパッケージ化したものです .

アプリケーション開発に限らず ,Web ページの表示確認にも使えますので , プログラマ以外でも使い方を憶えておくことをお勧めします .

Windows 用 ,Linux 用 ,Mac 用がありますので , お持ちの環境にあわせて使用するパッケージを選択して下さい . このテキストでは ,USB メモリ上にインストールして動作する Windows 用のパッケージ (XAMPP USB Lite 1.7.7) を例に話を進めます .

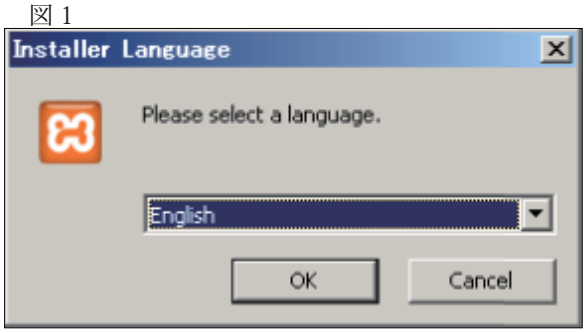
インストール

XAMPP 日本語ページ (http://www.apachefriends.org/jp/xampp-windows.html) にアクセスして ,"XAMPP ポータブル " から "EXE" をクリックしてインストーラをダウンロードして下さい .

ダウンロードしたインストーラをダブルクリックしてインストーラを起動してください .

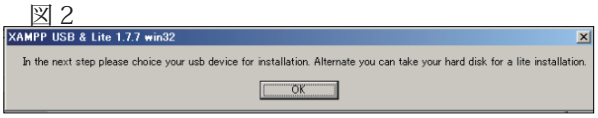
インストーラが起動するとまず言語選択のダイアログが表示されます (図 1) が ,English 以外は選択できませんので , そのまま「OK」をクリックしてください .

図 1



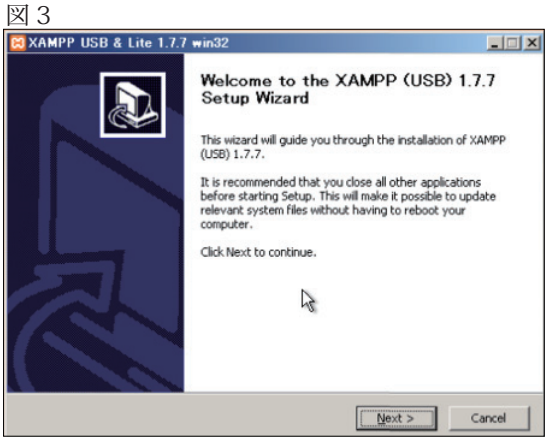
次のステップでは " インストール先に USB メモリを選択して下さい " と言うような注意書きが出ます (図 2) ので ,「OK」をクリックして先に進みます .

図 2



セットアップウィザードが起動します (図 3).「Next」をクリックして下さい .

図 3



インストール先選択ダイアログ (図 4) では ,「Browse」ボタンをクリックして (図 5)USB メモリ (リムーバブルディスク) を選択して下さい .

図 4

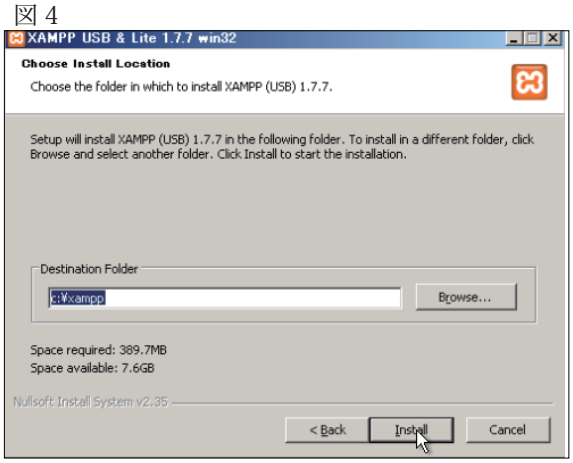
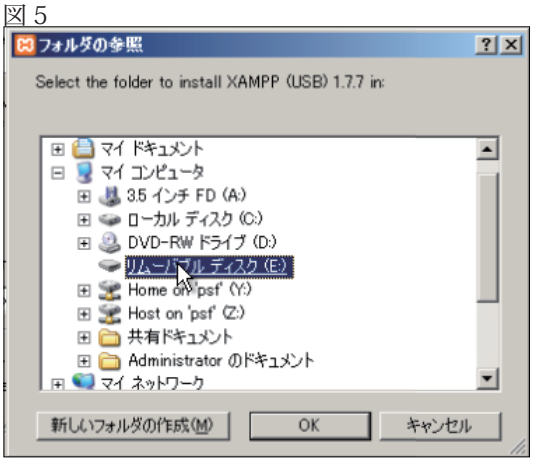


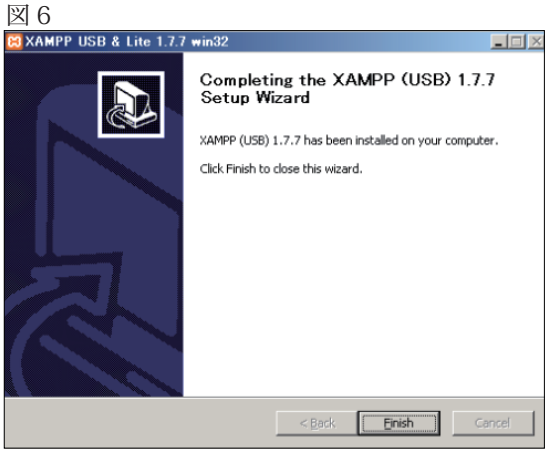
図 5



インストール先を選択したら「Install」ボタンをクリックします .

ファイルが USB メモリにコピーされます . 転送速度によっては時間が掛りますので気長に待ちましょう . 終了ダイアログが表示されたら (図 6)「Finish ボ」タンをクリックして終了します .

図 6



起動

図 7

インストールしたフォルダ (図 7) を開き ,xampp-control を起動します .

XAMPP Control Panel(図 8)　で Apache/MySQL の Start ボタンを押下して Running 状態にして下さい .Web ブラウザを起動し , アドレスバーに localhost と入力して XAMPP のトップページ (図 10) が表示されていればインストール成功です .

図 9

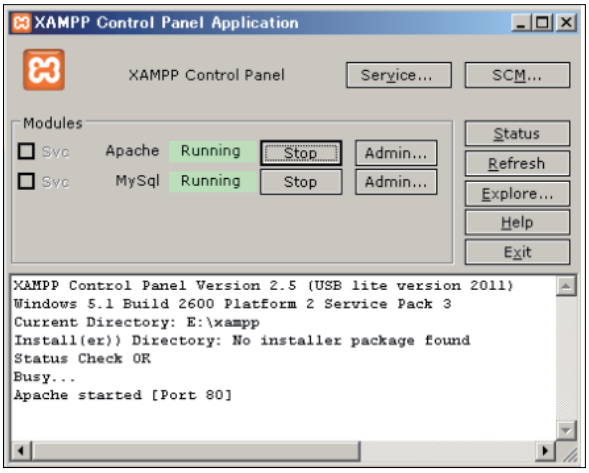


図 10

注意点

インストール時のフォルダ選択では , パスに日本語を含まないフォルダにインストールするようにしてください .

ファイル名 , フォルダ名には日本語を使用すると海外製のソフトウェアで不具合が発生することがあります .Web 制作の現場では原則としてファイル名 , フォルダ名には英数字 (と一部の記号) のみ使用してください .

インストール後にフォルダを移動すると , 設定に不具合が発生して起動しなくなります . フォルダを移動しないようにしてください .

Skype がインストールされている環境では ,TCP の 80 番ポートが競合するため ,Apache が起動しないことがあります . その場合には ,Skype のポート設定を変更して下さい .OS やバージョンによって異なるので , 具体的な対処方法は "XAMPP Skype" などのキーワードで検索してください .

5. ＰＨＰ基礎

「プログラムは思ったようには動かない . 書いた通りに動く」

実際にプログラムを記述するとき ,(特に慣れないうちは) 一気に沢山書いてしまうと , 思ったように動かない場合に原因を特定することが難しくなります .

最初のうちは「少し書いては実行して動作を確認」を繰り返すようにしてください .

PHP を何故学ぶのか

Web プログラミングでは Javascript が一番多く使われていると言えます . では , 何故 PHP を学ぶ必要があるのでしょうか .

PHP はサーバサイドのプログラムを作成するために使用します . 申込フォームを作成する実習では , メール送信などサーバ側でしかできない機能を実装します .(データ保存もクライアントサイドでは基本的にできません)

PHP のプログラムの書き方

- ファイルの拡張子は .php にします .
- PHP を実行できる Web サーバにファイルを設置してブラウザでアクセスすると実行されます .
- ファイルの中身は <?php　プログラム本体　?> という書式で記述します .
- 文字列やコメントなど一部を除いて , 半角英数字 , 記号で記述します . 俗に言う全角文字があるとエラーになります .
- 保存時の文字コードは原則として UTF-8:BOM なしにします . 以降の実習では , ファイルは全て UTF-8：BOM なしで保存してください .

具体的な手順はこの後ステップ毎に説明します .

BOM とはなにか

Byte Order Mark の略で ,Unicode テキストファイルにおいて , バイトの並び順を表すためにファイル先頭に付けられる印 . 通常のテキストエディタでは有無を確認できない上に ,BOM の有無でブラウザでの表示が変化することもあり , 原則付けないほうがよい .BOM の有無を指定して保存する方法はエディタによって異なるため , 注意が必要 . チェックボックスで指定 , 文字コードを選択 (保存時の文字コードに UTF-8,UTF-8N の両方があった場合 ,UTF-8N が BOM なし) など .

図 1.program01a.php

```
<?php
  echo ‘ 文字化けする最初のプログラム ’;
?>
```

出力

まず最初に出力を憶えましょう . プログラムが動いても , その結果が目に見えなくては意味がありません . プログラムの結果を目に見えるようにすることを出力と言います .(他のプログラムのための出力など , 目に見えない形での出力もあります .) 出力の方法は画面表示 , ファイル書き出し , 音声再生等さまざまです .

課題 1

目的：Web ブラウザ上に ” 最初のプログラム !” と表示する .

出来る事：echo ' 表示したい文章 ';　と書くと文章を表示できる .

【不具合解決コース】

テキストエディタを開きます .

「名前を付けて保存」でファイル名を ”program01a.php” として保存して下さい .

図 2.program01a.php の実行結果

図 1 を参考にプログラムを記述して , ファイルを上書き保存してください　保存したファイルを , ドキュメントルートにアップロードします . アップロード出来たら ,Web ブラウザで /program01.php にアクセスしてみて下さい . 変な表示になってしまっていると思います (文字化け).

ブラウザ毎のエンコード変更方法

IE はページの上で右クリックして「エンコード」
Chrome はスパナアイコンから「ツール」「エンコード」
Firefox はメニューから「表示」「文字エンコーディング」

もし文字化けしていない場合は , 他の Web ブラウザでも同様にアクセスしてみてください . ブラウザによって表示が違うことが判ると思います .

変な表示になってしまっている理由はブラウザの表示に使っている文字コードが , プログラムの出力の文字コードと違っているからです .

ブラウザの表示設定を変更して , 文字コードを一致させてみましょう .Unicode(UTF-8) を選択します .
“ 文字化けする最初のプログラム !” と正しく表示されたでしょうか .

エンコードを変更して正しく表示された場合 , 文字コードの不一致が原因であると判断できます . では , 何故文字コードの不一致が発生したのでしょうか .

通常 ,HTML を表示する場合には文字コードの指定が含まれています (HTML 内の <meta http-equiv="content-type" content="text/html; charset=UTF-8"> や ,Web サーバが自動で設定する場合もあります)

しかし今回作成した PHP プログラムには含まれていません . そのためにブラウザは文字コードの判別に失敗しました .

文字化けを防ぐために一番簡単で融通が効くのは ,HTML に PHP プログラムを埋め込む方法です .(HTML の文字コード指定をそのまま使います)

PHP は拡張子が .php で , プログラムが <?php 〜 ?> で囲まれていれば , 他の部分 はそのまま出力されます . また , 一つのファイル内に <?php 〜 ?> はいくつでも書くことができます . この特徴を利用して HTML に PHP のプログラムを埋めこみます .

【簡単解決コース】

エディタで新規作成して ,”program01.php” というファイル名を付けて保存して下さい .(文字コードは UTF-8：BOM なしです)

エディタに限らず , 新しく作業をするときは早めにファイル名を付けて保存しておくと , トラブルでアプリが終了しても途中までは作業が残ります .

今回は文字コードの設定を確認するだけな ので , 最低限の HTML しか使用しません . 正しい表示のために文字コードの指定を追加します . そして PHP のプログラムを埋め込んでみましょう .(図 3)

ファイルを上書き保存して , 保存したファイルをドキュメントルートにアップロードしてください .

アップロードが終了したら Web ブラウザで http://ホスト名 /program01.php にアクセスしてみてください . 図 4 のように正しく表示されているでしょうか ?

図 4:program01.php の実行結果

応用課題 1

目的：”1 行目 ” ”2 行目 ” と改行して表示する
ヒント：PHP の出力をブラウザは HTML として表示します . 方法は一つでは無いので , いろいろ HTML のタグと組み合せて動作を確認してみてください .

図 3:program01.php

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  </head>
  <body>
    <?php echo ‘ 最初のプログラム ’; ?>
  </body>
</html>
```

意図した通りに動かない場合

もしも正しく表示されない場合は , 次の点に注意して , プログラムを見直してみてください .

- プログラム中に全角半角が混在していないか
- 括弧 ( ) {} [] が正しく対応しているか
- "" (ダブルクォーテーション)," シングルクォーテーションが正しく対応しているか
- 文の最後に ;(セミコロン) を付け忘れていないか

また ,Web ブラウザに何らかのエラーメッセージが表示されている場合は , ほとんどの場合 , 行番号が表示されていますので , その行の近辺でエラーになっている可能性が高いので , 重点的にチェックしてみて下さい .

echo の省略

PHP 実行環境の設定によっては <?php echo 'abcd' ?> を <?= 'abcd'; ?> と書ける場合もありますが , 移植性 (他の環境へコピーした場合など) を考えると推奨できません .

入力

日本語表示を正しく出力できるようになりました . 次は入力方法を学びます .

リクエストとレスポンス

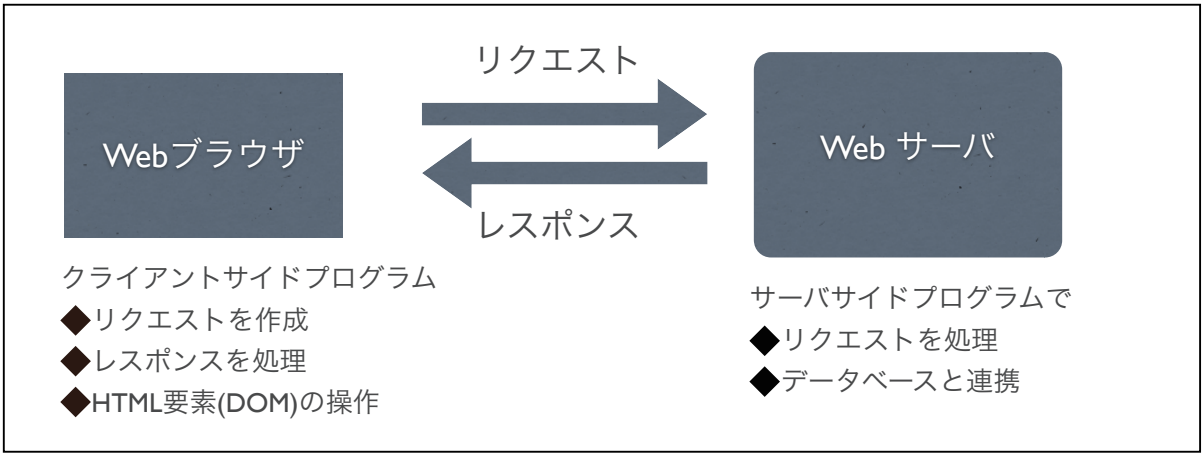
サーバサイドプログラム言語においては , 入力とは基本的に Web ブラウザからの GET/POST リクエストによるものになります . リクエストとは ,Web ブラウザが Web サーバに対して ,Web ページを表示するために必要な HTML や画像 (リソース) などを要求 (リクエスト) することを言います . ちなみに , リクエストに対してサーバから返答を返すことをレスポンスと呼びます (図 5).

GET/POST の基本的な使い分け方

GET…リソースを取ってくる
POST…リソースを登録する
ただし GET では URL のパラメータがアドレスバーに表示されるので , パラメータをアドレスバーに表示したい場合以外は POST を使用します

GET/POST リクエストでは URL とパラメータを設定することが出来るので ,URL で機能を示し , パラメータで必要なデータを送る設計が一般的になっています
PHP(に限らず , サーバサイドプログラム言語) で入力するためには ,HTML の form 機能を使用してデータをサーバに送ります .

図 5:Web アプリの動作概略



form 機能とは ,Web ページ上でテキストフィールドやボタン , チェックボックスなど , ユーザが操作する UI 部品を提供する機能です .

実際にプログラムを作成して動作を確認してみましょう . エディタで新規作成して ,"program02.html" というファイル名を付けて保存して下さい .(文字コードは UTF-8：BOM なしです)

最初に , 文字化けが発生しないように ,HTML のヘッダ等を記述します (図 6).

form タグ

準備ができたら <form> タグを使ってテキストフィールドとボタンを設置します .<form> タグでは , 以下の二つの属性を記述します .

method …リクエストの種類を指定します

action …リクエストする対象の URL を指定します .

action は同じ Web ページの URL でも良いですし , 別の Web ページの URL を指定することも出来ます .

Input タグ

<form> タグでリクエスト送信の設定をしたら ,<form> タグの中に <input> を追加します . ひとつの <form> タグの中に含まれる <input> タグはひとまとまりのデータとして送信されます .Web アプリケーションの要件によっては ,1 ページに複数の <form> タグを配置することもあります .

<input> タグでは次の二つの属性を設定します .

type … UI 表示の種類を指定します

name … データの名前を指定します .

PHP では name 属性で指定された名前を元にデータを探しますので , 書き間違えないように気をつけてください .

value … type によって意味が異なります . テキストフィールドではデフォルト値 , ボタンではボタンではボタンに表示される文字列になります .

課題 2

目的：

Web ページ上のテキストフィールドに文章を入力してボタンを押すと ,「"< 文章 >" が入力されました」と表示する

出来る事：

HTML の機能

<form method=" リクエスト種別 " action=" リクエスト先 URL" > タグでＵＲＬとリクエスト種類の指定

<input type="text"> タグでテキストフィールドの表示

<input type="submit"> タグでボタンの表示

PHP の機能

PHP タグ内で $_POST['name 属性 '] を使って入力されたデータの取得

図 6:program02.html

```
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
 </head>
 <body>
 </body>
</html>
```

ｐｒｏｇｒａｍ０２．html に , 図 7 のように <form> タグと <input> タグを追記してください . 追記したら上書きして保存したあと , サーバにアップロードし ,Web ブラウザで http:// ホスト名 /program02.html にアクセスしてみてください .

無事 , テキストフィールドとボタンは表示されたでしょうか (図 8)?

テキストフィールドに文字を入力して送信ボタンを押すと , エラーが表示されます (図 9). これはリクエストされた URL に該当するリソースが存在しないからです .

それでは ,PHP プログラムを作成しましょう .program02.php(拡張子に注意してください) というテキストファイルを用意してください .

図 10 を参考に , リクエストを表示する PHP プログラムを追記してください .program02.php をアップロードして , 再度 Web ブラウザで http:// ホスト名 /program02.html にアクセスし , テキストフィールドに何か適当に文字を入力して (図 11), ボタンを押してみて下さい . 無事 ,"< 文章 > が入力されました " と表示されたでしょうか ?(図 12)

PHP のプログラムを詳細に見ていきましょう . $_POST['<form タグ内の UI 部品の name>'] で POST したデータを取り出せます . そして PHP では , ドット記号で , その前後の二つの文字列を連結することができます .($_POST['<name>'] で取り出せるデータは文字列です)

つまり ,<?php echo $_POST['data'] . " が入力されました "; ?> の意味は ,「POST された data と言う名前のデータがあれば , そのデータと " が入力されました " という文字列を連結して出力しなさい」になります .

出力した結果は HTML に埋め込まれて , サーバか

図 7:program02.html

```
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
 </head>
 <body>
  <form method="POST" action="./program02.php">
   文章 :
   <input type="text" name="data">
   <input type="submit" value=" 送信 ">
  </form>
 </body>
</html>
```

図 8:program02.html 表示結果

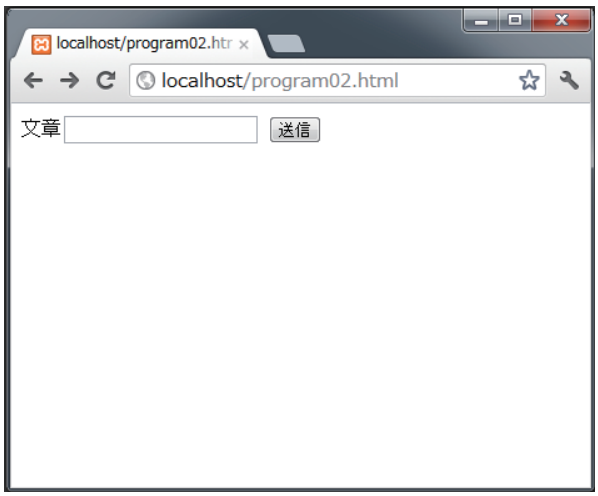


図 9:program02.php エラー表示

らクライアントにレスポンスされるので ,Web ブラウザは通常の HTML として表示することが出来ます .(“ ソースの表示 ” をしても PHP の痕跡はありません)

図 11:program02.html 文字入力

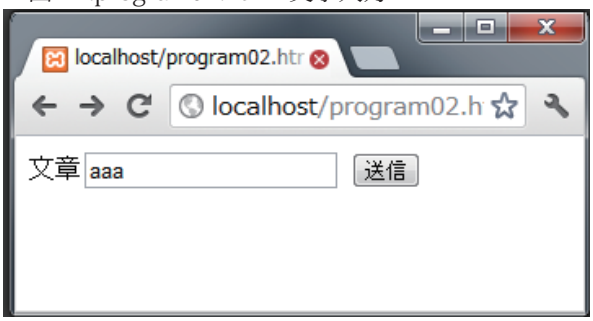


図 12:program02.php 入力結果表示

コメント

プログラムでは " 何をするかを書くことになりますが , 何故そのようなプログラムを記述するのかを記述することはできません .

例えば 100 * 1.05 は 100 という数にに 5% 上乗せをする式ですが , これが消費税の計算なのか ,100ml の商品の 5% 増量なのかはこの式からだけでは読み取れません .

もしも消費税が 10% になったら , このプログラムは 100 * 1.1 として良いのでしょうか . プログラムの前後を見れば判断できるかも知れませんが , いちち確認するのも面倒です .

そこで , プログラムが何故その処理をするのか , 説明を付けておきましょう . これをコメントと呼び ,2 つの書き方をすることが出来ます .

ひとつめは行コメントで ,"//" から行末までがコメントとして扱われます . もうひとつは "/*" と "*/" で囲まれた部分がコメントとして扱われるブロックコメントです .

コメントはプログラムとして認識されないので , 日本語も使えますし , コメントの終了文字以外は好きなように記述できます .

ただしブロックコメントはネスト (入れ子) にはできませんので注意が必要です .(/* /* コメント */ */ とは書けません)

コメントにはプログラムの説明以外にも用途があります . それはプログラムを一時的に無効にするために使われ , コメントアウトと呼ばれています .

図 13 を参考に program03.php を作成してブラウザでアクセスしても , 何も表示されません .

プログラムの動作確認を行なうための命令や , オプション的に使用される部分をコメントアウトすることで , 本番で不要な動作をさせずにすみます .
また , プログラムにエラーが出た際に , 怪しい箇所をコメントアウトして , エラーの発生箇所を特定する場合にも使用します .

図 10:program02.php

```
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
 </head>
 <body>
  <?php echo $_POST['data'] . ' が入力されました '; ?>
 </body>
</html>
```

図 13: program03.php

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>program03: コメント </title>
  </head>
  <body>
    <?php
      // 行コメント
      // ←から改行までがコメントとして扱われます

      /* ←ここからブロックコメント
        ここもコメント
        ここもコメント
      ここまで→ */

      // echo ' コメントの範囲内にプログラムを書いても実行されません ';

      /*
        ブロックコメントも同様
        echo ' コメントの範囲内にプログラムを書いても実行されません ';

      */
    ?>
  </body>
</html>
```

6.Web サーバ

Web ブラウザからのリクエストを受けて ,HTML や画像 ,CSS ファイルなどをレスポンスするためのコンピュータを Web サーバと呼びます . ただし , コンピュータそのものを Web サーバと言うこともありますが , コンピュータの上で動いているアプリケーションを指して Web サーバと言うこともあるので注意が必要です . アプリケーションを指して言う場合には ,HTTP サーバとも呼ばれます .

Web サーバは Windows などと同じように , ブラウザから要求されたファイルの拡張子によって , 動作を変えることができます . 例えば , 拡張子が .php の場合には ,PHP のプログラムファイルとして PHP 処理系 (PHP のプログラムを受けとって実行し , 結果を出力するプログラム) に処理を委譲します (図 15). 基本的には拡張子 .html の場合には ,PHP のプログラムとして処理を行ないません (図 14) が , 設定によって .html のファイルでも PHP のプログラムとして実行されることがあります . これは Web サーバの設定しだいなので作業をするまえに確認をしてください .

図 14 HTML ファイルの処理

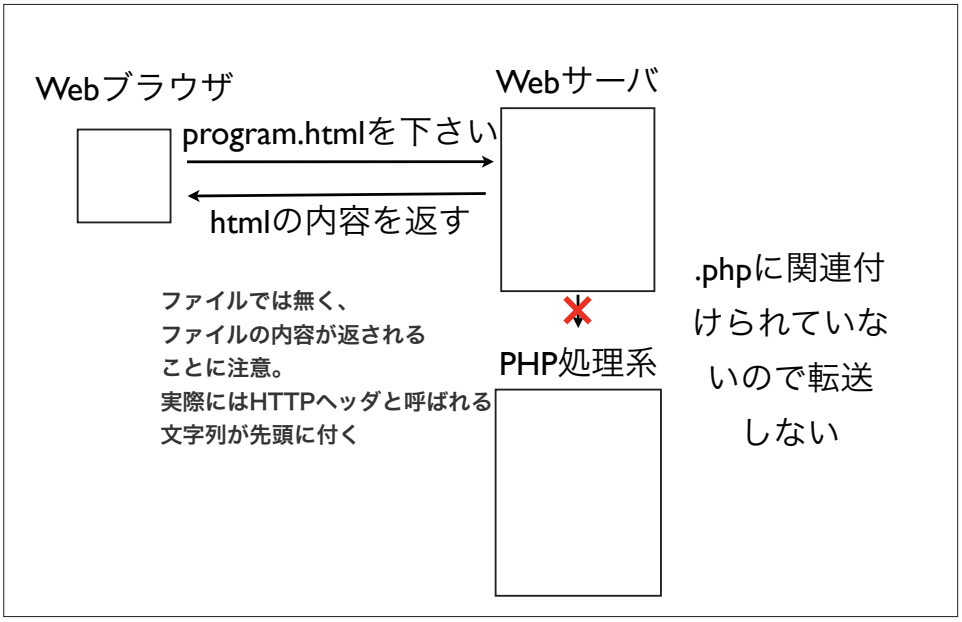
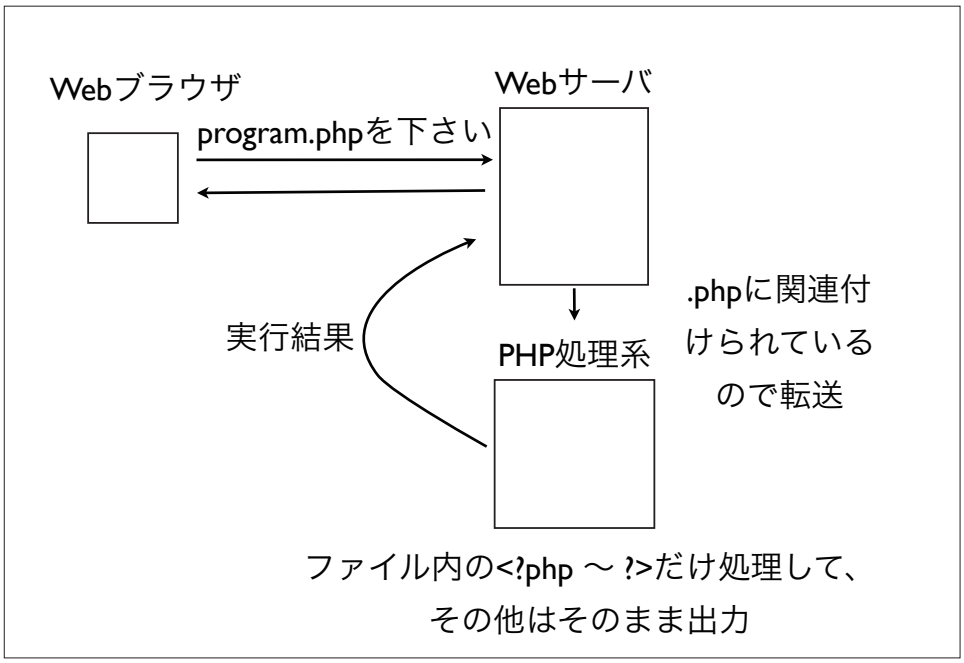


図 15 PHP ファイルの処理



7. セキュリティ

実はこれまで学んだ内容には危険が潜んでいます . program02.html にアクセスして , テキストフィールドに ”<script>alert(“BANG”);</script>” と入力してボタンを押してみて下さい (図 16).

図 16:program02.html Javascript を入力

“BANG” と表示されたダイアログが表示されたでしょうか（図 17）(Web ブラウザによっては表示されません). これは , 悪意あるユーザが攻撃スクリプトを実行できてしまうと言うことでもあります .

攻撃の一例を挙げると , この文章が表示されるタイミングで Javascript によってこのサイトのログイン画面を偽造し , 一般ユーザに ID とパスワードを要求した上で , 攻撃者にその情報を送信することが出来ます .

図 17:program02.php Javascript が実行された

サーバサイドプログラムの出力は通常 HTML として Web ブラウザは解釈しますので , そこに javascript として見なされる文字列が含まれていると Web ブラウザはそのまま実行してしまいます .

このように , 攻撃を受ける可能性のある不具合を脆弱性 (ぜいじゃくせい) と呼びます . この場合は , 入力された文字列をそのまま表示してしまうことが脆弱性にあたります . 脆弱性を放置しておくと , 知らない間に顧客や会員のデータを盗まれたり , 他のサイトへの攻撃の足掛かりにされたりと大きな問題になります .

詳しくは別に章を立てて解説しますが , とりあえずは完全では無いものの , 攻撃を受けにくくする対処をします . program02.php の表示部分を以下の様に変更します .

```
echo htmlentities($_POST['data'], ENT_QUOTES, 'UTF-8').' が入力されました ';
```

htmlentities(文字列 , フラグ , エンコーディング) は , 文字列に含まれる ”<“,”>”,”&” などの HTML における特殊な文字を ”<”,”>”,”&” に変換し , 無効化します . フラグ ENT_QUOTES は、シングルクオートを変換するよう指示します . またエンコーディングは、プログラムの文字コードを指定します (一般的には UTF-8).

修正をしたファイルをアップロードしたうえで , 再度 ”<script>alert(“BANG”);</script>” と入力して , ボタンを押してみて下さい .

今度は ”<script>alert(“BANG”);</script> と入力されました ” と表示されたと思います (図 18).

図 18:program02.php Javascript が実行されない

ソースを見ると , 記号が変換されていることが確認できます (図 19).

図 19:program02.php HTML ソース

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  </head>
  <body>
    &lt;script&gt;alert(&quot;BANG&quot;);&lt;/script&gt;が入力されました </body>
</html>
```

入力された文章をそのまま出力するのでは無く , 安全な形式に変更して出力するようにしてください .

Web アプリケーションでは不特定多数のユーザが対象になります . 悪意あるユーザへの対処は必ず行なってください .

8. まとめ

ここまで Web プログラムとは , と言う話と ,PHP プログラムの書き方 , 入力と出力を解説しました .
復習としてもう一度見直してみましょう .

web プログラムとは

PHP などで作成されるサーバサイドプログラム ,Javascript で記述されるクライアントサイドプログラム . これらは HTML だけでは実現できない , 動的な Web ページ ,Web サイトを作るために使用されます . 主にブラウザ上での見た目を操作する Javascript と違い ,Web サーバ側で動作する PHP のプログラムはどのような役割があるのか判り難いですが ,EC サイトやブログ (WordPress は PHP で書かれています) などはサーバサイドプログラムのデータベースアクセス機能や HTML 生成機能 , メール送信機能などが無くては実現できません .

PHP プログラムの書き方

PHP のプログラムは <?php ?> という PHP タグで囲む必要があります .

PHP の特徴として , タグで囲まれていない部分はそのまま出力されるので ,PHP プログラムの外側に HTML を記述することが出来ます . この機能によって , あたかも HTML が主 ,PHP プログラムが従であるかの様にプログラムを記述できます .

そのため , プログラマとデザイナーが分業して Web サイトの制作が行ないやすくなっています .

出力

プログラムの実行結果を出力するためには echo 構文を使用します .
また "." でメッセージを連結することも学びました .

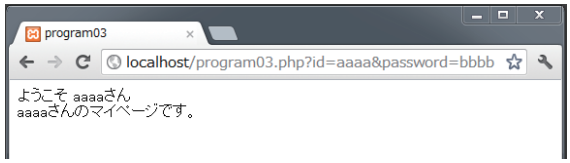
入力

ユーザからの入力を取得する方法は主に 2 種類あり ,HTML の form タグを使用して ,GET, または POST リクエストをブラウザから送信することで入力を行ないます .

基本的に GET は取得 ,POST は登録という建前ですが , 取得する場合でも POST を使うことがあります . その理由を実際のプログラムで見てみましょう .(図 22 program04.html, 図 23 program04.php)

GET の場合はアドレスバーにパラメータが表示されます .(図 20)

図 20



この様に ID やパスワードを GET で送信するとそのまま表示されてしまいますので , 横から覗かれでもしたらパスワードが他人に知られてしまいます .

POST リクエストにしたところでネットワークでの盗聴の危険性はありますが , 多少なりともリスクの低減が期待できます .(図 24 program05.html, 図 25 program05.php) form タグの method 属性を変更したら PHP 側の $_GET,$_POST も忘れずに変更してください .

GET リクエストの利点としてはブックマーク (お気に入りに登録) すると , パラメータごと登録される点が挙げられます . たとえばブログなどでは検索結果を指し示すパラメータごとブックマークできたほうがユーザにとっては便利ですので ,GET リクエストで検索できるようにプログラムを作成します .(図 21)

図 21

逆に POST リクエストではブックマークしてもパラメータが保存されないため , ブックマークから開くと必ずパラメータ無しで表示されます .

実現したい機能に合せてリクエストの種類を選択するようにしてください .

図 22: program04.html

```
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Program04:GET リクエスト </title>
 </head>
 <body>
  <form method="GET" action="./program04.php">
   ID:<input type="text" name="id"><br />
   Password:<input type="password" name="password"><br />
   <input type="submit" value=" 送信 "></p>
  </form>
 </body>
</html>
```

図 23: program04.html

```
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>program04:GET リクエスト </title>
 </head>
 <body>
  <?php echo ' ようこそ '.htmlentities($_GET['id'],ENT_QUOTES,'UTF-8') . ' さん '; ?><br />
  <?php echo htmlentities($_GET['id'],ENT_QUOTES,'UTF-8').' さんのマイページです .'; ?><br />
 </body>
</html>
```

図 24: program05.html

```
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Program05:POST リクエスト </title>
 </head>
 <body>
  <form method="POST" action="./program05.php">
   ID:<input type="text" name="id"><br />
   Password:<input type="password" name="password"><br />
   <input type="submit" value=" 送信 "></p>
  </form>
 </body>
</html>
```

図 25: program05.php

```
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>program05:POST リクエスト </title>
 </head>
 <body>
  <?php echo ' ようこそ '.htmlentities($_POST['id'],ENT_QUOTES,'UTF-8') . ' さん '; ?><br />
  <?php echo htmlentities($_POST['id'],ENT_QUOTES,'UTF-8').' さんのマイページです .'; ?><br />
 </body>
</html>
```