



TOKYO JOHO UNIVERSITY

東京情報大学

Since 1988

2019年度

Webシステムプログラミング^a

PHPの基礎(1)

講義内容

- 動的なWebページ(復習)
- PHPとは？
- 動的なWebページの実際

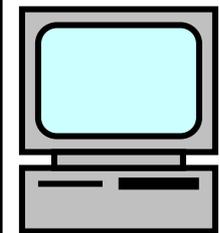
動的なWebページ(復習)

■ 動的なWebページ

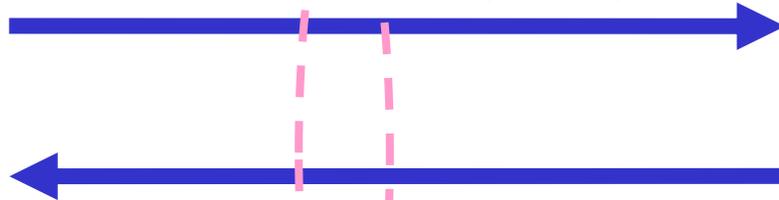
- ➡ クライアント(Webブラウザ)のリクエスト(+入力データ)に対して、**その指定されたプログラムを実行してHTMLデータを作成(送られた入力データごとに異なる)し、そのデータを送る**

「フォームデータ」
(form data)と呼ばれる

①Webページをリクエスト(+入力データ)
(`http://www.edu.tuis.ac.jp/~fujiiwara/xyz.php`)

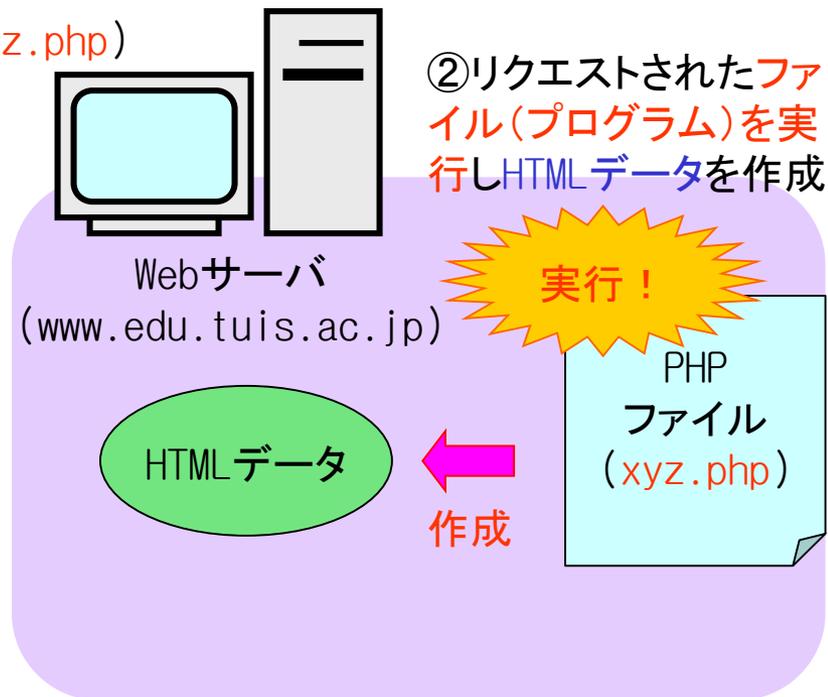


クライアント



③作成されたHTMLデータを送る

インターネット



PHP

■ PHP (PHP: Hypertext Preprocessor)

➡ 動的なWebページにおいて, Webサーバ側で実行されるプログラム(ファイル)を記述するプログラミング言語

特徴

- ◆ 習得が簡単
- ◆ HTML (Webページを記述するための言語) の中に埋め込める
- ◆ 無償 (オープンソース)
- ◆ ライブラリが豊富
(例えばデータベースとのやりとりが標準で可能)

このような目的のプログラミング言語では他にも Perl, Ruby が有名

PHPの基礎(1)

■ 記述の仕方(HTML内での)

```
<?php  
    ...この間にPHPのプログラムを記述する...  
?>
```

- ※複数行のプログラムが書ける
- ※各文の終わりは必ず「;」

```
<html>  
<body>  
<?php  
    echo "hello. world!!"; // PHPのプログラムです  
?>  
    <span style="color:red"><?php echo "何カ所でもOK"; ?></span>  
</body>  
</html>
```

PHPの基礎(2)

■コメント

```
// この'//'以下から行末までがコメント(1行のコメント)
```

```
/*  
  この'/*'から'*/'までがコメント(複数行のコメント)  
*/
```

PHPの基礎(3)

■ 変数

\$変数名

- ※頭に「\$」をつける
- ※型指定はなく, **宣言なし**で使用できる
- ※変数名は自由(1文字目はアルファベットかアンダーバー, 2文字目以降は数も可)

```
// 変数「$hoge」に「325」を代入  
$hoge = 325;
```

■ 演算子

+ : 足し算
- : 引き算
* : かけ算
/ : 割り算
% : 余りを求める
= : 代入
++ : $?=?+1$ の省略形
-- : $?=?-1$ の省略形
() : 演算の優先順位
 を変える

※ Java言語, C言語とほとんど同じ演算子が使用可能

```
// 「5 × 3 + 2」の結果を変数「$result」に代入  
$result = 5 * 3 + 2;
```

```
// 変数「$result」と変数「$atai」の値を足してその結果を「$kekka」に代入  
$kekka = $result + $atai;
```

PHPの基礎(4)

■ 表示(出力)

```
echo 表示したい「値」;
```

- ※「値」には数値や文字列, 変数, 計算式など
- ※複数の「値」をつなげて表示したい場合は各「値」を
文字列結合演算子「.」(ドット)でつなげる
- ※「print 表示したい「値」;」も同様.

```
$hoge = 325;  
echo $hoge;
```



325

```
$hoge = 325;  
echo "変数の値は" . $hoge . "です";
```



変数の値は325です

```
$hoge = 325;  
$foo = 48;  
echo $hoge . " × " . $foo . " = " . ($hoge*$foo) . " です";
```



325 × 48=15600です

PHPの基礎(5)

■ 書式付き表示

```
printf( 書式, 値, 値... );
```

※C言語の「printf」関数とほとんど同じ

※書式は

「%d」…整数

「%f」…実数

「%s」…文字列

「%x」…値を16進数で表示

※「%」直後には表示桁を指定することもできる

```
$hoge = 5;  
$moji = "こんにちは";  
printf( "文字は %s , 値は %d", $moji, $hoge );
```



文字は こんにちは , 値は 5

PHPの基礎(6)

■ 文字列

“なんらかの文字”

'なんらかの文字'

- ※ 文字列はダブルクォーテーション(")か、シングルクォーテーション(')で囲う。
- ※ 2つの違いは変数展開がされるかどうか(変数は中括弧{}で囲う)。
 - ダブルクォーテーション・・・変数展開される
 - シングルクォーテーション・・・変数展開されない
- ※ ダブルクォーテーションの文字列の中では、シングルクォーテーションはただの文字(逆も同じ)。HTMLタグ(属性)の出力に多用。

```
$hoge = 325;  
echo “変数の値は {$hoge} です”;
```

変数の値は325です

```
$hoge = 325;  
echo '変数の値は {$hoge} です';
```

変数の値は {\$hoge} です

```
echo “<span style=' color:red' >ほげ</span>”;
```

PHPの基礎(7)

■ 配列

● 作成

```
array(要素1, 要素2, . . .)
```

- ※各要素のデータ型はバラバラでよい.
- ※空の配列を `array()` で作成することもできる.



```
$hoge = array( 235 , "情報大" , 6.73 );
```

● 要素へのアクセス

```
配列名[要素番号]
```

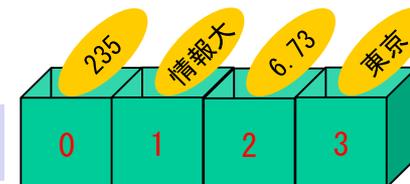
- ※要素番号は0から始まる.
- ※要素番号の指定は変数や計算式などが使える.
- ※配列の最後に新たに要素を追加することもできる.

```
配列名[] = 追加したい要素; $hoge[] = "東京";
```

```
echo $hoge[1];  
$i = 5;  
echo $hoge[$i-3];
```



情報大6.73



● 要素数の取得

```
count(配列名)
```

```
echo count($hoge);
```



4

PHPの基礎(8)

■ 連想配列

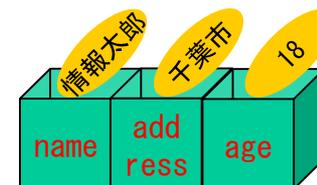
● 作成

```
array( "キー1" => 要素1 , "キー2" => 要素2 , . . . )
```

※(要素番号ではなく)キー名で各要素にアクセスできる配列.

※キー名は文字列.

```
$hoge = array( "name" => "情報太郎", "address" => "千葉県", "age" => 18 );
```



● 要素へのアクセス

```
配列名[キー名]
```

```
echo $hoge["address"];
```



千葉県

PHPの基礎(9)

■ 条件分岐

```
if( 条件 ){  
    ...条件が「真」(成り立っている場合)の処理...  
}
```

```
if( 条件 ){  
    ...条件が「真」の処理...  
} else {  
    ...条件が「偽」(成り立っていない場合)の処理...  
}
```

```
if( 条件1 ){  
    ...条件1が「真」の処理...  
} elseif( 条件2 ){  
    ...条件1が「偽」で、条件2が「真」の処理...  
}
```

さらに「elseif」や「else」を
続けることもできる

※「条件」には比較演算子を使う

「==」...等しい

「!=」...等しくない

「>=」...以上

「<=」...以下

「>」...より大

「<」...より小

「===」...等しい(かつデータ型も等しい)

「!==」...等しくない(またはデータ型が等しくない)

※複数の「条件」をつなげる場合は論理演算子を使う

「&&」...かつ

「||」...または

「!」...否定

```
$hoge = 325;  
if( $hoge == 300 ){  
    echo "変数の値は300です";  
} elseif( $hoge >= 300 && $hoge <= 1000 ){  
    echo "変数の値は300以上1000以下です";  
} else {  
    echo "変数の値はよく分かりません";  
}
```

PHPの基礎(10)

■ 繰り返し処理(その1)

```
for( 初期化処理 ; 繰り返し条件 ; 繰り返しごとの処理 ){  
    ...  
    ...繰り返し処理...  
    ...  
}
```

※「初期化処理」は、繰り返しを行う前に1度だけ行う処理

※「繰り返し条件」は、この条件が「真」(成り立っている場合)とき繰り返し処理が一回実行される

※「繰り返しごとの処理」は、繰り返し処理を1回行うごとに行われる処理

```
for( $i=1; $i <= 100; $i++ ) {  
    echo "繰り返し第" . $i . "回目¥n";  
}
```

PHPの基礎(11)

■ 繰り返し処理(その2)

```
while( 繰り返し条件 ){  
    ...繰り返し処理...  
}
```

```
do{  
    ...繰り返し処理...  
} while( 繰り返し条件 );
```

※「繰り返し条件」は、この条件が「真」(成り立っている場合)とき繰り返し処理が一回実行される

※「do~while」では、必ず1回は繰り返し処理が行われる

```
$i = 1;  
while( $i < 100 ){  
    echo "繰り返し第" . $i . "回目\n";  
    $i++;  
}
```

PHPの基礎(12)

■ 関数の呼び出し

引数(=パラメータ)がない場合

関数名()

引数(=パラメータ)がある場合

関数名(引数1, 引数2, ...)

※PHPは豊富な関数が利用できる。

※各関数の詳細は「繰り返し条件」は、PHP関数リファレンス

<http://php.net/manual/ja/funcref.php>

を参照。

```
$val = rand(1, 100); // rand関数は乱数を生成する（最小値1から最大値100までの範囲）  
echo "乱数" . $val;
```

```
$pos = strpos($val, "3"); // strpos関数は第1引数の文字列の中で、第2引数の最初の位置を返す  
//（先頭の位置は0. 無い場合はfalse）
```

```
if( $pos === false ){ // ==での比較はダメ（===でデータ型の一致も調べる）  
    echo "乱数は3を含まない";  
} else {  
    echo "乱数は3を含む（位置は{$pos}）";  
}
```

動的なWebページの実際(1)

- 実習0: 本日の演習用作業フォルダを作成しましょう.
 - ◆ 手順1: 以下のフォルダの中に, 新規フォルダ「php01」を作成する.

Z:\xampp\htdocs\

※以下, 本日の演習でのファイルはすべてこの「php01」フォルダに作成すること.

動的なWebページの実際(2)

■ 実習1: 「1」から「100」までの数について偶数を太字にして表示するWebページを作成しましょう.

◆ 手順1: 以下(リスト1)のPHPプログラムのファイルを「`hundred.php`」という名前で作業フォルダに作成する.

※ `???` のところは適宜自分で考える.

文字コードは UTF-8 で保存すること

◆ 手順2: Webブラウザで手順1で作成したページにアクセスし確認する.

「`http://localhost/php01/hundred.php`」

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>実習 1</title>
</head>
<body>
```

```
<h2>実習1</h2>
```

```
<table border=' 1' >
```

```
<?php
// ここからPHPのプログラムが始まります（「//」以下はコメント）

// 「for」を使った繰り返し処理（変数$iが「1」～「100」になるまで）
for( $i=?????? ; ??????? ; ??????? ){ // 繰り返し処理
    echo "<tr>";
    echo "<td>";
    if( $i % 2 == 0 ){ // $jを2で割って余りを求める。それが0かをチェックする。
        echo "<b> . $i . "</b>"; // 偶数の場合
    } else {
        echo $i; // 奇数の場合
    }
    echo "</td>";
    echo "</tr>¥n";
}
// ここまでがPHPのプログラムです
?>
```

```
</table>
```

```
</body>
```

```
</html>
```

動的なWebページの実際(3)

■ 実習2: 「九九」の表を表示するWebページを作成してみましょう.

◆ 手順1: 以下(リスト2)のPHPプログラムのファイルを「[kuku.php](#)」という名前で作業フォルダに作成する.

※ **???** のところは適宜自分で考える.

文字コードは UTF-8 で保存すること

◆ 手順2: Webブラウザで手順1で作成したページにアクセスし確認する.

「<http://localhost/php01/kuku.php>」

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>実習 2</title>
</head>
<body>

<h2>実習 2</h2>

<table border=' 1' >

<?php
// ここからPHPのプログラムが始まります（「//」以下はコメント）
echo "<tr style=' background-color:#99FF99'>"; // 「echo」を使って文字列の表示（出力）
echo "<td></td>";

// 「for」を使った繰り返し処理（変数$iが「1」から「9」になるまで）
for( $i=1; $i<=9; $i++ ){
    echo "<td>" . $i . "</td>"; // 文字列と変数値の表示
}
echo "</tr>¥n";

for( $i=1; $i<=9; $i++ ){ // 繰り返し処理
    echo "<tr><td style=' background-color:#99FF99'>" . $i . "</td>";
    for( $j=1; $j<=9; $j++ ){ // 繰り返し処理（2重ループ）
        $atai = ??????????; // かけ算の結果を計算し変数$ataiに代入
        echo "<td>" . ???????? . "</td>"; // 掛け算の結果を表示
    }
    echo "</tr>¥n";
}
// ここまでがPHPのプログラムです
?>

</table>
</body>
</html>
```

提出課題

- 課題1: 1から100までの乱数で作成した2つの整数の足し算を表示するWebページを作成しましょう。

- ◆ 乱数を作成する関数: `rand(範囲の最小値, 範囲の最大値)`
- ◆ ファイル名: 「`add.php`」
- ◆ アクセスURL: 「`http://localhost/php01/add.php`」
- ◆ 備考: ブラウザ再読み込み(F5キー)をするたびに式と結果が変わることを確認しましょう。



提出課題

■ 課題2: 「九九」の表において

- ・3の付く数字は**太字**(例: 3 13 23 ...)

- ・4の倍数は**赤字**(例: 4 8 12 ...)

- ・3の付く数字で, かつ, 4の倍数は**赤字で太字**

になるようなWebページを作成してみましょう.

※実習2で作成した「kuku.php」ファイルをコピーして名前を変更, および内容を少し変更する.

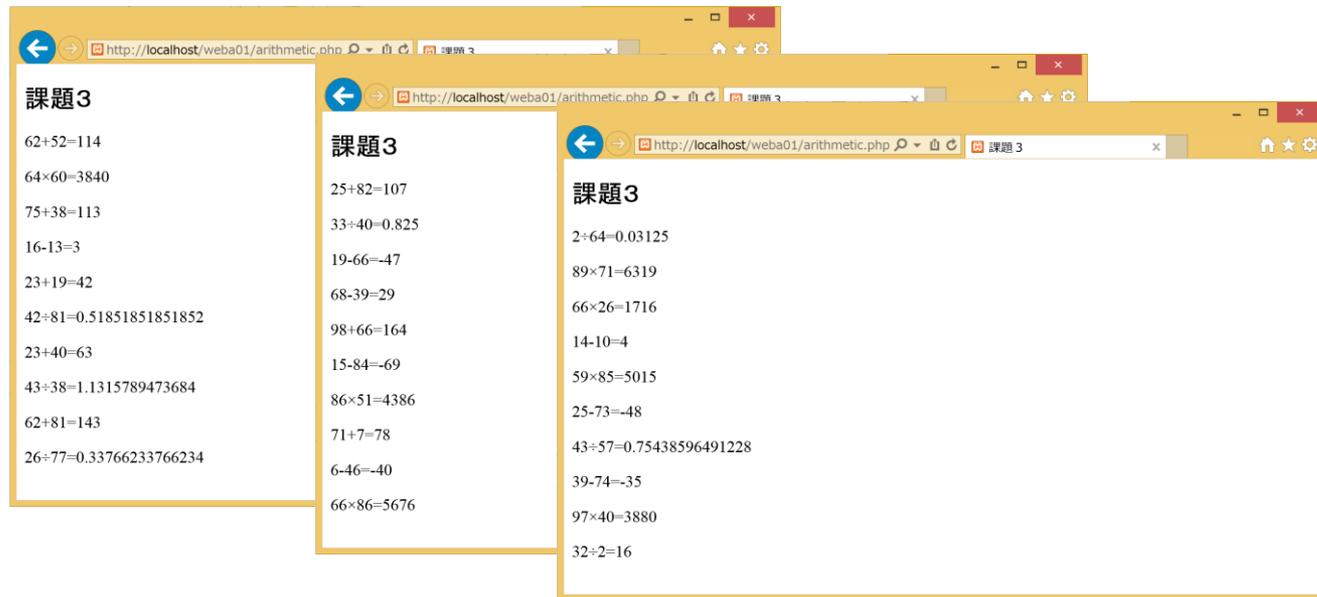
◆ ファイル名: 「[kuku2.php](#)」

◆ アクセスURL: 「<http://localhost/php01/kuku2.php>」

提出課題

- 課題3: 課題1を改造し, 1から100までの乱数で作成した2つの整数の四則演算のどれか(これも乱数を使う)の式と結果を10個表示するWebページを作成しましょう.

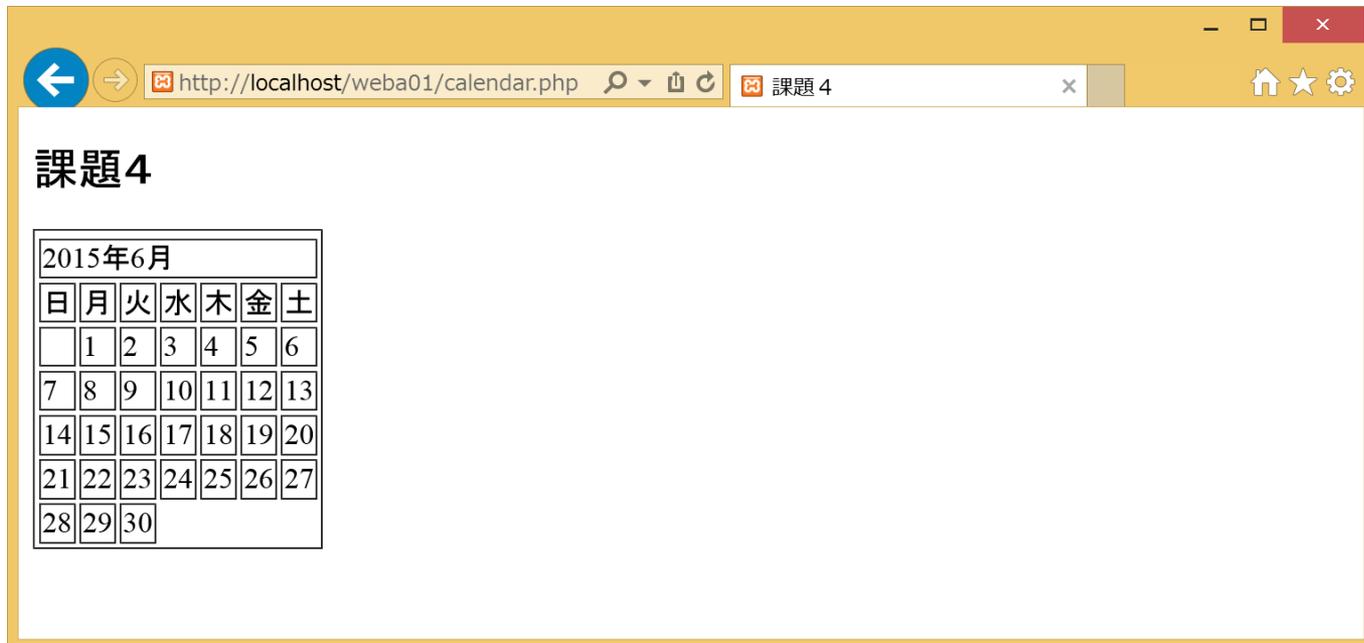
- ◆ ファイル名: 「[arithmetic.php](#)」
- ◆ アクセスURL: 「<http://localhost/php01/arithmetic.php>」
- ◆ 備考: ブラウザ再読み込み(F5キー)をするたびに式と結果が変わることを確認しましょう.



提出課題

- 課題4 (上級問題): 現在の月のカレンダーを表示するWebページを作成しましょう.
※次ページのプログラムの???を考える.

- ◆ ファイル名: 「calendar.php」
- ◆ アクセスURL: 「<http://localhost/php01/calendar.php>」



```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>課題 4</title>
</head>
<body>

<h2>課題 4</h2>

<p>
<?php

$year = date("Y"); // 現在の年を取得
$month = date("n"); // 現在の月を取得

// 現在月の始めの日の曜日を整数で取得 (0:日曜・・・6:土曜)
$first_dweek = date("w", mktime(0, 0, 0, $month, 1, $year));

$ndays = date("t"); // 現在の月の日数を取得

echo "<table border='1'><tr>";
echo "<tr><td colspan='7'>" . $year . "年" . $month . "月</td></tr>";
echo "<td>日</td><td>月</td><td>火</td><td>水</td><td>木</td><td>金</td><td>土</td></tr><tr>";
```

次のページにつづく

```

// 最初の週の処理（始めの日の曜日まで空白セルにする）
for ($i=0;$i<$first_dweek;$i++){
    echo "<td> </td>";
}

$current_dweek = $i; // 表示する日にちの曜日（整数. 0:日曜・・・6:土曜）
$day = ??????; // 表示する日にちの初期化

while($day<=?????) { // 日にち表示の繰り返し
    if($current_dweek>?????) { // 土曜を超えたら改行
        echo "</tr>¥n";
        echo "<tr>";
        $?????? = 0;
    }
    echo "<td>" . $?????? . "</td>"; // 日にちを表示
    $??????++;
    $current_dweek++;
}
echo "</tr></table>";
?>
</p>

</body>
</html>

```