



---

# 中村研究室ゼミ

## CGIとPHP

---

中村 聡史



# 本日の内容

- アクセスのたびに動作が変わるページの実現
- CGI (Common Gateway Interface)
- PHP



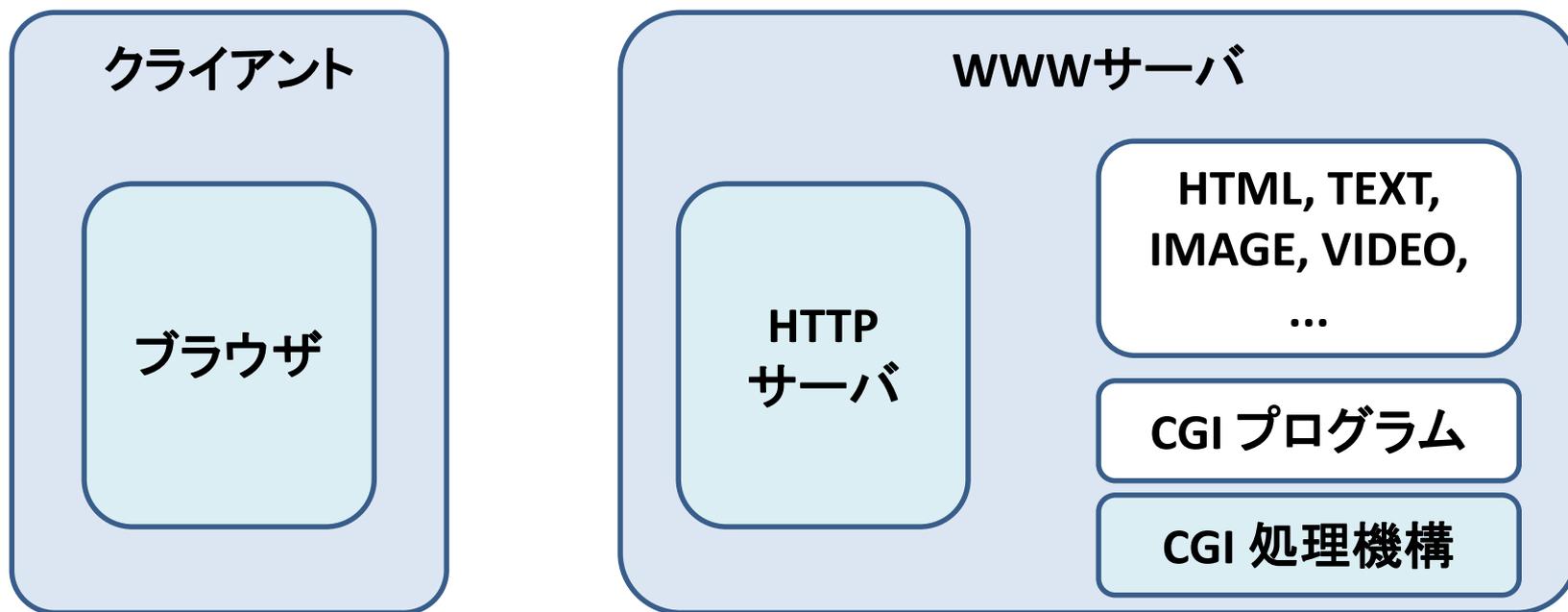
# 動的なコンテンツ

- アクセスするたびに結果が変わったり, 問い合わせをするようなウェブページをどのようにして実現するか?
  - ウェブ掲示板やウェブアンケート
  - アクセスカウンター
  - ウェブログ
  - 検索サービスや物販サービス  
などなど

**CGI (Common Gateway Interface)**

# CGI (Common Gateway Interface)

- ウェブサーバ上で、ユーザが記述したプログラムを動作させる仕組み



# CGI (カウンタの例)

index.html

```
<html><head>  
<title>中村のページ</title>  
</head>  
<body>  
<h1>中村聡史</h1>  
  
中村のページです  
</body>  
</html>
```

## 中村聡史

3 1 1 6 7

中村のページです



# CGI (カウンタの例)

## index.html

```
<html><head>
<title>中村のページ</title>
</head>
<body>
<h1>中村聡史</h1>

中村のページです
</body>
</html>
```

## counter.cgi

```
#!/usr/bin/perl
$target = $ENV{'QUERY_STRING'};
print "Content-type: Image/gif";
print "¥n¥n";
open( IN, $target.".dat" );
$count = <IN>;
close( IN );
$count++;
open( OUT, ">".$target.".dat" );
print OUT $count;
close( OUT );
:
ShowCountImage( $count );
```



# CGI とプログラミング言語

- C言語
  - Perl
  - PHP
  - Ruby
  - Python
- などなど

```
#include <stdio.h>
```

```
void main() {
```

```
{
```

```
void main() {
```

```
<h1>
```

```
<?php
```

```
echo
```

```
?>
```

```
</h1>
```

```
}
```

```
}
```

```
print "¥n¥n"
```

```
print "<html><body>"
```

```
print "Hello World"
```

```
print "</body></html>"
```



# CGI とプログラミング言語

- プログラミング言語はサーバに環境があればなんでもよい
  - コンパイラ言語 (C, C++, Javaなど)
    - コンピュータが解釈できるように予め変換 (コンパイル) しておき, そのまま実行される (高速)
  - インタプリタ言語 (Perl, PHP, Rubyなど)
    - 毎回実行の際にインタプリタがプログラムを読み込んで内容を理解し, 変換しつつ実行する (手軽だがコンパイラ言語に比べて低速)



# PHP

- PHP: HyperText Preprocessor
  - Personal Home Page Tools
  - HTMLにプログラムを埋め込むことができるため御手軽にプログラミング可能
  - 割と高速に動作（CGI以外のモジュールでの実行形態をとることが可能なため）
  - データベースとの連携が容易
  - ライブラリが充実している



# PHP の基礎

- 拡張子は .php
- HTML をそのまま書いてもOK
- `<?php ここにプログラムを記述 ?>`
- `<?php ~ ?>` の外はそのまま表示される

```
<html><body>  
<?php  
echo "kariyushi";  
echo 40+18;  
?>  
</body></html>
```



```
<html><body>  
kariyushi58  
</body></html>
```

test.php として作ってみましょう



# PHP の基礎

## • 考え方

- PHPはHTMLの中に埋め込むもの
- HTMLの中で動的に変化させたい部分だけを, PHPのプログラムとして書く
- もちろん, PHPでHTMLコードを書くことも可能
  - echo という表示メソッドを利用してひたすら書く

```
<html>  
<body>  
こんにちは  
</body>  
</html>
```

```
<?php  
echo "<html>";  
echo "<body>";  
echo "こんにちは";  
echo "</body>";  
echo "</html>";  
?>
```

```
<html>  
<body>  
<?php  
echo "こんにちは";  
?>  
</body>  
</html>
```

# ProcessingとPHP

- 同じ所: プログラミング言語なので変数, 計算, 条件分岐, 繰返し, メソッド, クラスなど同じ

## 12345の約数の数を数えるプログラム

```
int i = 1;
int count = 0;
while( i <= 12345 ){
    if( (12345 % i) == 0 ){
        // 12345をiで割った余りが
        // 0だったらcountを増やす
        count++;
    }
    i++;
}
println( "約数の数は"+count );
```

```
<?php
$i = 1;
$count = 0;
while( $i <= 12345 ){
    if( (12345 % $i) == 0 ){
        // 12345を$iで割った余りが
        // 0だったら$countを増やす
        $count++;
    }
    $i++;
}
echo "約数の数は".$count;
?>
```



# ProcessingとPHPの違い

- PHPでは変数名の最初には必ずドルが付く
  - 例: \$value, \$hoge, \$ball などなど
  - 変数には型 (int, Stringなど) が無い (勝手に解釈)
  - 変数への代入は「=」でつなぐだけ
  - ユーザに対する出力は echo で行う
  - ユーザからの入力 HTMLフォームやGETメソッドなどから行う
  - <?php から ?> までの部分が実行コードとなる
  - コンパイラ言語とインタプリタ言語の違い



# PHP の基礎

- 命令の最後は「;」で終わる

- 変数は \$ と英数字で表現

\$value (\$と英数字)

– value という名前の変数を定義

`$x = 5;`                    (=で繋ぐと代入)

`$y = 10;`

`$z = $x + $y;`            (計算は + - \* / など)

**変数は記憶するもの／箱**



# PHP の基礎

- echoで文字を表示
  - 数字や計算結果はそのまま表示
    - echo 100;    echo 3.14\*2.5\*2.5;
  - 文字列を表示するには半角の「"」 「'」 で囲む
    - echo "中村聡史";
  - 文字列を繋げるときはピリオドで繋ぐ
    - echo "平均 = " . (95+88+64)/3;
  - 変数の内容はそのまま表示可能
    - echo \$value;
  - 変数と文字列を繋げるときもピリオドで繋ぐ
    - echo \$value . "hoge";



# 計算の方法

演算子	意味
$10 + 5$	10と5を足しあわせた値
$10 - 5$	10から5を引いた値
$10 * 5$	10と5を掛けあわせた値
$10 / 5$	10を5で割った値
$10 \% 5$	10を5で割った余り
$\$value = 10$	$\$value$ に 10 を代入

$(10 + 5) / 2$  は 10と5を足した結果を2で割る



# GET / POST

- データの渡し方

- methodで方法をnameで入れ物を指定
- GET は URI に埋め込む

```
<form action="input.php" method="GET">  
<input type="text" name="message">  
</form>
```

- POST は URI に埋めこまない

```
<form action="input.php" method="POST">  
<input type="text" name="message">  
</form>
```



# PHP の基礎

- 入力の取得

method でGETを使った場合

<http://lab.nkmr.io/~ユーザ名/input.php?message=Hello>

```
<?php  
echo $_GET["message"];  
?>
```

method でPOSTを使った場合

```
<?php  
echo $_POST["message"];  
?>
```



# PHP の基礎

- 入力の取得 (GETの場合)

<http://lab.nkmr.io/~ユーザ名/bbs.php?nickname=nakamura&message=hello>

```
<?php  
echo $_GET["nickname"];  
echo $_GET["message"];  
?>
```

- 入力の取得 (POSTの場合)

```
<?php  
echo $_POST["nickname"];  
echo $_POST["message"];  
?>
```



# 演習

- hello.phpを改良し自分の名前を表示しましょう
- 数式の計算結果を表示してみましょう

```
echo 5 * 5 * 3.14;
```

```
echo (10 + 5) * 3 / 2;
```

```
$x = 10;
```

```
$y = 5;
```

```
echo $x * $y;
```



# [演習] 円の面積を求める calc.php

```
<html>
<body>
円の面積を求めます <br>
<form action="calc.php" method="get">
半径 = <input type="text" name="hankei">
</form>
</body>
</html>
```

```
<html>
<body>
円の面積を求めます <br>
<form action="calc.php" method="get">
半径 = <input type="text" name="hankei">
<input type="submit" value="計算" />
</form>
<?php
$r = $_GET['hankei'];
echo "面積は";
echo $r * $r * 3.14;
echo "です";
?>
</body>
</html>
```



# 演習(おまけ)

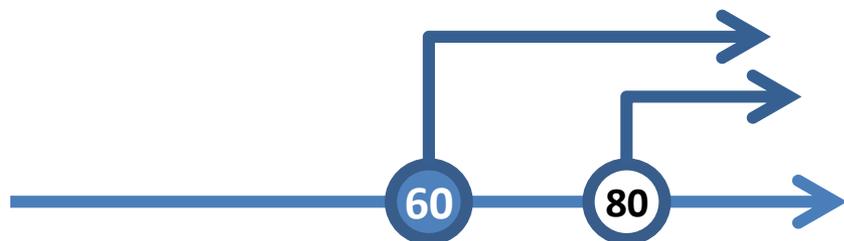
- 計算する値をGETで入力してみましょう
  - <http://lab.nkmr.io/~user-name/calc.php?hankei=5>
  - などのようにして値を指定できます
- 上底と下底と高さを入力してもらい、台形の面積を求めて結果を表示する



# PHP の基礎

- 条件分岐

```
if ( $score > 90 ){  
    echo "成績は優です";  
}  
elseif( $score >= 60 ){  
    echo "成績は可です";  
}  
else {  
    echo "不可です";  
}
```



もし \$score が  
90より大きければ

もし \$score が  
上記の条件を満たさ  
ず60より以上ならば

もし \$score が  
上記の条件を満たさ  
ないならば



# 演算子と意味

演算子	意味	プログラム上
$x > y$	$x$ が $y$ より大きい	左記のときに true それ以外で false
$x < y$	$x$ が $y$ より小さい	同上
$x \geq y$	$x$ が $y$ 以上	同上
$x \leq y$	$x$ が $y$ 以下	同上
$x == y$	$x$ と $y$ が等しい	同上
$x != y$	$x$ と $y$ が等しくない	同上



# [演習] PHP の基礎

- 条件分岐

```
$number = rand( 1, 5 );  
if ( $number == 1 ){  
    echo "大吉です";  
}  
elseif( $number == 2 ){  
    echo "中吉です";  
}  
else {  
    echo "凶です";  
}
```

1から5までの  
ランダムな値を  
取得する

<?php と ?> の間に  
書きましょう



# [演習] PHP の基礎

- 条件分岐

```
$number = $_GET["kuji"];  
if ( $number == 1 ){  
    echo "大吉です";  
} elseif( $number == 2 ){  
    echo "中吉です";  
} else {  
    echo "凶です";  
}
```

<?php と ?> の間に  
書きましょう

# 演習

- 占いのプログラムを作ってみましょう
  - 数字を入力してもらって，入力してもらった内容に基づき条件分岐して結果を表示
- input.html というHTMLを用意してニックネームとメッセージを入力させ，info.php というプログラムで入力情報を表示しましょう
  - ニックネームをnicknameという箱に，住所をmessage という箱に入れるようにしましょう



# 演習

- ニックネーム (nickname) とメッセージ (message) を入力するフォームと, その結果を表示する PHP のプログラムを作成しましょう (bbs.php)
  - 結果を出力する場合, 表形式にしましょう. また, ニックネームやメッセージ以外も入力できるようにしてみましょう.
  - 時間があれば, メールアドレスなども入力できるようにしてみましょう.



# 繰り返し

- 1から10000までを足し算した結果を知りたい  
echo 1+2+3+4+...+10000;
- 入力された数字の全ての約数を求めたい
- 1から100000までの全ての素数を求めたい
- 100個の変数の値を表示したい
- 掲示板に書き込まれた内容を20件表示したい



# 繰り返し(1から順に表示)

```
while( 繰り返し条件 ){  
    処理内容  
}
```

まずは入力して  
実行してみましょう

```
$i = 1;  
while( $i <= 10000 ){  
    echo $i . "<br>";  
    $i++;  
}
```

10000になるまで  
繰り返す

$i$  と  $\langle br \rangle$  を表示

$i$  を1増やす



# 繰り返し

```
while( 繰り返し条件 ){  
    処理内容  
}
```

条件を満たしている間  
ずっと処理される

```
$i = 1; $total = 0;  
while( $i <= 10000 ){  
    $total = $total + $i;  
    $i++;  
}  
echo $total;
```

10000まで足して表示  
sum.php

$\$i$  に1を追加

$\$i$  が 10000 以下なら  
 $\$total$  に  $\$i$  を足す



# 演習 (loop.php)

- 1から1000までの数字を表示してみましょう
  - `while( $i <= 1000 ){ ... }`
- 1から10万までの数字の和を計算してみましょう
  - `while( $i <= 100000 ){ ... }`
- 繰り返す回数(1から幾つまで足し合わせるか)を変数 `$value` で指定しましょう
  - `while( $i <= $value ){ ... }`
- `$value` をフォームから入力できるようにしてみましょう



# [参考] 繰り返し

```
for( 初期化; 繰り返し条件; 繰り返しの際の処理 ){  
    処理内容  
}
```

---

```
$total = 0;
```

```
for( $i=1; $i<=10000; $i++ ){
```

```
    $total = $total + $i;
```

```
}
```

```
echo $total;
```



# 繰り返し＋条件分岐

偶数のみを表示

```
<?php
$i = 1;
while( $i <= 1000 ){
    if( $i % 2 == 0 ){
        echo $i . "<br>";
    }
    $i++;
}
?>
```

2で割った余りが0の  
場合に表示する

$\$i \% 2$  は  $\$i$  を  
2で割った余り

$\$i$	$\$i \% 2$
1	1
2	0
3	1
4	0



# 繰り返し(yakusu.php)

1000の全ての約数を求める

```
<?php
```

```
$i = 1;
```

```
while( $i <= 1000 ){
```

```
    if( 1000 % $i == 0 ){
```

```
        echo $i ."は1000の約数です<br>";
```

```
    }
```

```
    $i++;
```

```
}
```

```
?>
```

まずは入力して  
実行してみましょう

1000を*\$i*で  
割った余り



# 繰り返し(yakusu.php)

`$value` の全ての約数を求める

```
$value = 10000;  
$i = 1;  
while( $i <= $value ){  
    if( $value % $i == 0 ){  
        echo $i ."は".$value."の約数です<br>";  
    }  
    $i++;  
}
```

`$value` を form タグで  
入力するようにしてみましょう



# 繰り返し(余りを表示)

```
$i = 1;  
while( $i <= 10000 ){  
    echo $i . "を2で割った余りは";  
    echo $i%2 . "です<br>";  
    $i++;  
}
```



# 演習

- 3の倍数のみを表示してみましょう
- 5の倍数のみを表示してみましょう
- 2000年1月から2012年12月までひたすら出力してみましょう

(calendar.php)

```
2011年8月 ←  
2011年9月 ←  
2011年10月 ←  
2011年11月 ←  
2011年12月 ←  
2012年1月 ←  
2012年2月 ←  
2012年3月 ←  
2012年4月 ←  
2012年5月 ←  
2012年6月 ←  
2012年7月 ←  
2012年8月 ←  
2012年9月 ←  
2012年10月 ←  
2012年11月 ←  
2012年12月 ←
```



# 繰り返しの多重化

1000 までの全ての数の約数を求める

```
$i = 1;
```

```
while( $i <= 1000 ){
```

```
    $j = 1;
```

```
    while( $j <= $i ){
```

```
        if( $i % $j == 0 ){
```

```
            echo "$j は $i の約数です";
```

```
        }
```

```
        $j++;
```

```
    }
```

```
    $i++;
```

```
}
```



# 繰り返しの多重化

九九の表を作ってみる(kuku.php)

```
$i = 1;  
while( $i <= 9 ){  
    $j = 1;  
    while( $j <= 9 ){  
        echo $i * $j;  
        echo "|";  
        $j++;  
    }  
    $i++;  
    echo "<br>";  
}
```

		かける数								
		1	2	3	4	5	6	7	8	9
かけられる数	1	1	2	3	4	5	6	7	8	9
	2	2	4	6	8	10	12	14	16	18
	3	3	6	9	12	15	18	21	24	27
	4	4	8	12	16	20	24	28	32	36
	5	5	10	15	20	25	30	35	40	45
	6	6	12	18	24	30	36	42	48	54
	7	7	14	21	28	35	42	49	56	63
	8	8	16	24	32	40	48	56	64	72
	9	9	18	27	36	45	54	63	72	81

入力して実行してみましょう



# 繰り返しの多重化

九九の表を作ってみる

```
$i = 1;
```

```
while( $i <= 9 ){
```

```
    $j = 1;
```

```
    while( $j <= 9 ){
```

```
        echo $i * $j;
```

```
        echo "|";
```

```
        $j++;
```

```
    }
```

```
    $i++;
```

```
    echo "<br>";
```

```
}
```

左記の九九をテーブルタグで  
表示してみましょう

```
<table border=1>
<tr><td> ... </td><td> ... .. </td></tr>
<tr><td> ... </td><td> ... .. </td></tr>
<tr><td> ... </td><td> ... .. </td></tr>
      :
</table>
```

# 演習

- 10000までの全ての素数を求めるプログラムを作ってみましょう
  - 素数とは, 1と自分自身以外に約数を持たない数
  - 約数の数が1とその数自身の2のとき素数

# 演習

- FORMから数字を入力してもらい, その数の全ての約数を表示するプログラムを作しましょう

約数を表示します. 数字を入力して下さい



# 宿題

- HTMLフォームで値を入力すると、結果を返すプログラムを作成せよ
  - (例) 自動販売機, 計算機, などなど
- 本スライドで提示している演習のプログラムを作成せよ

# 参考サイト

- PHPマニュアル:
  - <http://www.php.net/manual/ja/index.php>
- PHP入門
  - <http://www.scollabo.com/banban/php/index.html>