

# プログラムに関連する基礎用語

## ・PHPタグ

<?php

//ここにプログラムを書く

?>

## ・コメントアウト = プログラムとして読まれないもの

一行をコメントアウト:「//」

複数行をコメントアウト:「/\* から \*/」



```
1 <?php
2 echo "これはテストです。<br />";
3 //echo "この部分はコメントのため見えません";
4 # echo "これもコメントのため見えません";
5 /*echo "実はこんなふうになると";
6 echo "複数行をコメントとして扱えます";*/
7 echo "コメント機能を使えば、「この一行必要かな」というスイッチが可能!";
8 ?>
```

ブラウザ上では・・・

これはテストです。

コメント機能を使えば、「この一行必要かな」というスイッチが可能!

# プログラムに関連する基礎用語

- ・文字列: 文字の羅列、数値とは扱い方が違う

- ・変数 = 値を入れておく箱

```
$name = "牧口";
```

```
echo $name;
```

- ・配列 = 複数の値が入っている箱

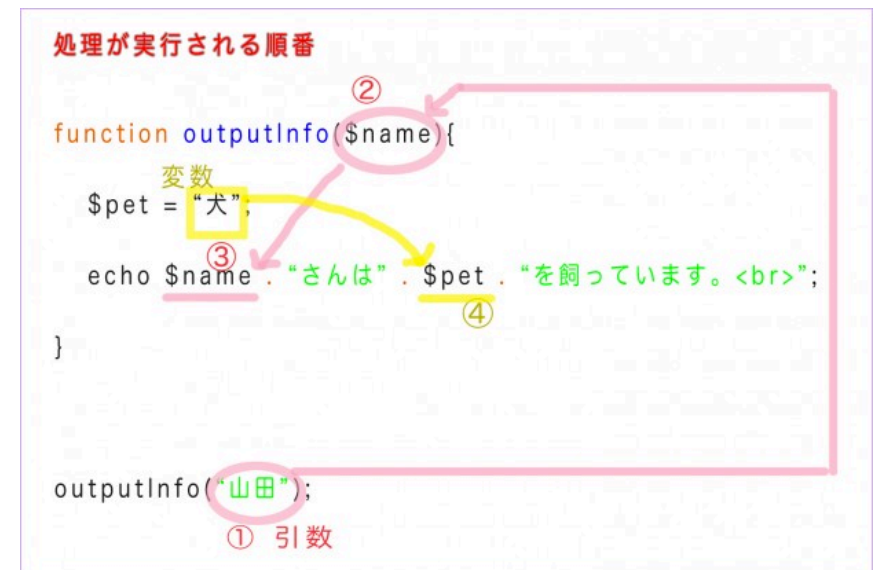
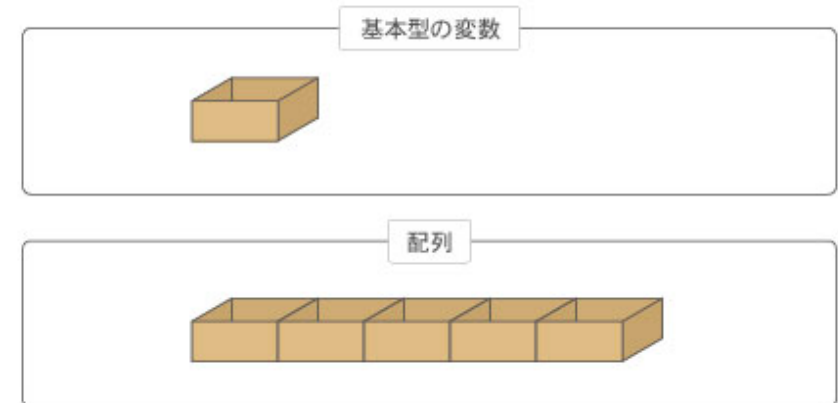
```
$array = array("foo", "bar", "hello", "world");  
var_dump($array);
```

⇒変数と違って添字がつく（\$array[0]などの[0]）

- ・関数 = 事前に登録された処理をする

最初から登録されている関数とは別に

関数は自分で作成も可能



# プログラムに関連する基礎用語

---

- ・文字コード = ファイル、HTMLヘッター、ブラウザ別で指定ができる  
文字化けが発生した時 → 文字コードの揃え方を覚えよう
- ・キャッシュ = ブラウザ上に古い内容が残る  
更新されない場合 → ブラウザのキャッシュ更新を覚えよう  
⇒ IEなら「InPrivate」、Chromeなら「シークレットウィンドウ」などを使ってみよう
- ・条件分岐とループ : if関数により、場合分けが可能。  
ループとは繰り返し処理をさせること。forやwhileが使われる。

一回で成功しないのが普通  
ゆっくり一行ずつ確認しよう。

# 分かりやすいプログラムの書き方

鉄則：自分が後で見えて分かるように書く！（出来れば他の人も分かるように）

```
<html>
  <head>
  </head>
<body>
<!-- ここから下の</body>までがブラウザで表示される-->
<!-- この記号の間に入れるとブラウザでは表示されない-->
  <?php
    //echoで文字列を表示する
    $moji = "hello world"; //行末に ; を入れるのを忘れない
    echo $moji; //変数の場合は "" で囲まない
    //条件分岐: $mojiが文字列とそうでないときで別の表示を出す
    //条件分岐やループを使う場合はインデント(字下げ)を行う
    if (is_string($moji)) {
        echo "文字列でした！";
    } else {
        echo "文字列じゃなかった";
    } //次はフォームを作ってフォームから送った文字が$mojiに入るようにする
  ?>
</body>
</html>
```

- //で処理内容を書いておくと分かりやすい
- 行の途中からでもコメントアウト可能
- インデントで揃えるとどこまでが一連の処理かが分かりやすい
- 次の予定や現状を書くのもアリ

# 研修 1：～ミッション2-6

## ▶ 掲示板機能の実装

### ▶ ① WEB上でのフォーム送信

- ・HTMLでフォームを作る

- ・データを送信する

GETとPOSTの違いとは？

- ・POSTの際のURL

`http://XXXXXXXXXXXXX/form.php`

- ・GETの際のURL

`http://XXXXXXXXXXXXX/form.php?text=123456&btn=GET%E9%80%81%E4%BF%A1`

『?』はパラメータの始まりです。『&』はパラメータの区切れ目です。

『=』を挟んだ左側がGETの変数名で、右側が渡される値です。

なお日本語はURLに使用できないため変換が自動で行われます。

## フォームデータの送信

# 研修 1：～ミッション2-6

## ▶ 掲示板機能の実装

### ▶ ②テキストファイルへの保存

ヒント)

フォームから送信された文字列を  
書き込むようにしてみよう！

fopen 『第二引数の使い方がポイント』  
fwrite  
を活用してみる

改行はダブルコーテーション「”」で囲う。

参考) 「”」と「'」の違い

「”」… 文字列中の変数を変数値に置換する  
「'」文字列中の変数を変数値に置換しない

「.」は 文字列 や 変数 を結合します。 \$contents . "¥r¥n"

例)

```
<form method="post" action="write2.php">
<textarea name="contents" cols="40" rows="20"></textarea>
<input type="submit" value="送信">
</form>
<a href="write.txt">write.txt</a><br>
<?php
$contents = $_POST['contents'];
if ($contents) {
    $fp = fopen('write.txt', 'a+');
    fputs($fp, $contents."¥r¥n");
    fclose($fp);
    print "書き込み完了しました。";
}
?>
```

# 研修 1：～ミッション2-6

---

## ▶ 掲示板機能の実装

### ▶ ③テキストファイル内の情報表示

保存されているテキストファイル  
(ファイル名.txt)をPHPで呼び出してみよう

ヒント)  
File関数を使ってみる

例)

```
<?php
$test = "./sample.txt";

$hyouji = file($test);

print $hyouji[0]."<br />";
print $hyouji[1]."<br />";
print $hyouji[2]."<br />";

?>
```

# 研修 1：～ミッション2-6

---

## ▶ 掲示板機能の実装

### ▶ ④WEB上でのファイル編集・削除

ヒント)

Unlink

fopen

Fwrite

を活用してみる

例)

```
<?php  
unlink("./test.txt");  
  
?>
```



## 研修2：ミッション2-7 PDOで接続

---

### ▶ PHPでMySQLデータベースに接続しよう(課題2-7)

「php pdo データベース接続」でググろう

(例)

```
<?php
//データベースへの接続

// mysql:dbname=データベースの名前;host=localhost(DBのホスト) // ユーザー名 // パスワード

$pdo = new PDO(接続情報);
```

ヒント)  
PDO

# 研修2：ミッション2-8 テーブルを作る

## ▶ PHPでMySQLデータベースのテーブルを作ろう

「php pdo テーブル作成」でググろう

(例)

```
<?php
$pdo = new PDO(接続情報);

$sql = 'CREATE TABLE テーブル名
(
    フィールド名1 データ型1(長さ1),
    フィールド名2 データ型2(長さ2),
    :
    PRIMARY KEY(フィールド名)
);';

$result = $pdo->query($sql);
```

ヒント)

PDO

CREATE TABLE

テーブル名にハイフン「-」などを使わない

## 研修2：ミッション2-9 テーブルの一覧表示

### ▶ PHPでMySQLデータベースのテーブルを一覧表示しよう

「php SHOW TABLES mysql」でググろう

(例 ヒント SHOW TABLES)

```
<?php
$pdo = new PDO(接続情報);
$sql = 'SHOW TABLES';
$result = $pdo->query($sql);
foreach ($result as $row) {
    echo $row[0];
    echo '<br>';
}

?>
```

## 研修2：ミッション2-10 テーブルの中身確認

### ▶ PHPでMySQLデータベースのテーブルのカラム型を確認

「php show create table mysql」でググろう

(例 ヒント)

show create tableを利用

```
$pdo = new PDO(接続情報);  
$sql = 'show create table テーブル名';  
$result = $pdo->query($sql);  
foreach ($result as $row) {  
    echo $row[0];  
    echo '<br>';  
}
```

## 研修2：ミッション2-11 テーブルにデータ挿入

### ▶ PHPでMySQLデータベースのテーブルにデータ挿入しよう

「php pdo テーブル データ挿入」でググろう

(例 ヒント)

PDOでINSERT文を使ってMySQLにデータを挿入してみよう！

```
$pdo = new PDO(接続情報);  
$sql = "INSERT INTO テーブル名 (name, value) VALUES (:name, :value)";  
$result = $pdo->query($sql);
```

PDOには

『query』 以外 『prepare』 & 『execute』

もある

## 研修2：ミッション2-12 テーブルのデータ表示

### ▶ MySQLデータベースのテーブル内のデータを取得しよう

「php pdo テーブル データ表示」でググろう

(例)

```
<?php
$pdo = new PDO(接続情報);

$sql = 'SELECT * FROM テーブル名';//クエリ
$result = $pdo->query($sql);//実行・結果取得
//出力
foreach ($result as $row) {
    echo $row['id'].' ';
    echo $row['text'].'<br>';
}
?>
```

## 研修2：ミッション2-13 テーブルのデータ編集

### ▶ PHPでMySQLデータベースのテーブルのデータを編集しよう

「php pdo テーブル データ編集」でググろう

(例)

```
// UPDATE文を実行

$pdo = new PDO(接続情報);

$sql = "update test_tbl set name='$nm' , age=$age where id = $id";
$result = $pdo->query($sql);
```

実際に編集が来ているか、2-12の方法で確認してみよう

## 研修2：ミッション2-14 テーブルのデータ削除

### ▶ PHPでMySQLデータベースのテーブルのデータを削除しよう

「php pdo テーブル データ削除」でググろう

(例)

```
// DELETE文を実行

$pdo = new PDO(接続情報);

$sql = "delete from test_tbl where id=$id";
$result = $pdo->query($sql);
```

実際に削除が来ているか、2-12の方法で確認してみよう

ただし、1度消すと復活できないので慎重に！