
PHP

PHP 基礎目次

Lesson 2

変数

変数とは数値は文字列などのデータを格納する箱のようなもので、メモリ上に保存される。格納したデータを変更したり取り出したりできます。変数は$ + [変数名]で表現します。
変数名にはルールがあり

- 使用できるのはアルファベット、数値、アンダースコアのみ
- 先頭はアンダースコアかアルファベットで始まること
- 大文字、小文字は区別される
- 予約語は使用できない(functions や return など)

```
$a; // OK
$_a; // OK
$1a; // NG
$return; // NG
$returnB; // OK
```

変数を用意するのが **[宣言]**、変数に値を入れることを **[代入]** と呼びます。
宣言は$num;のように記述し
代入は$num = 10;と記述する 宣言と代入を同時に行うことを **[初期化]** と呼びます。
変数は初期化しなくても使用できるが必ず初期化するように気を付けること
初期化することで別のファイルの同名変数の汚染リスクを避けられます(後述する外部ファイル読み込み時の話し)。

sample01.php

```php
<?php
$num = 123;
echo $num;
echo '<br />';

$str = 'hello';
echo $str;
echo '<br />';
```

出力結果

```
123
hello
```

多次元配列

配列の要素に別の配列を格納することができます。複雑な構造のデータを格納したいときに便利です。

前に項で解説した配列は 1 次元配列(単に配列)と呼び配列の中に配列を作成した場合は 2 次元配列と呼びます。さらに配列を作成していくと 3 次元、4 次元となっていきます。通常は 2 次元配列をよく使用するので、まずは 2 次元配列をしっかり理解しましょう。2 次元廃列の宣言は下記のように配列記号を 2 つ連ねて記述します。

```
// 2 次元廃列の宣言
$配列変数 = [][];

// 2 次元廃列の初期化
$配列変数 = [
  [10, 20, 30, 40],
  [50, 60, 70, 80]
];
```

2 次元配列の要素にアクセスする場合は、下記のように要素番号を指定します。

```
// 2 次元廃列の初期化
$配列変数 = [
  // 0, 1,  2,  3(2 次元目)
  [10, 20, 30, 40],  // 0(1 次元目)
  [50, 60, 70, 80]  //  1(1 次元目)
];

echo $配列変数[1][2];
// 70 が出力される
```

Sample19.php

```
<?php
$fruits = [
  'apple' => ['name' => 'りんご', 'stock' => 2, 'price'=> 150],
  'orange' => ['name' => 'みかん', 'stock' => 5, 'price'=> 120],
  'banana' => ['name' => 'バナナ', 'stock' => 10, 'price'=> 180],
  'grape' => ['name' => 'ぶどう', 'stock' => 4, 'price'=> 200]
];

echo '<pre>';
print_r($fruits);

echo "{$fruits['banana']['name']}の在庫は{$fruits['banana']['stock']}個です。";

echo '</pre>';
```

Lesson03

条件文

条件文とはソースコード内である条件によって処理する内容を分けて記述できます。この時に使用するのが条件文です。条件式は比較演算子を用いて 2 つの値を比較し、その結果が true(真)もしくは false(偽)になる式のことです。

比較演算子

演算子	式が true になる場合
==	型の相互変換後、右辺と左辺が等しい
===	右辺と左辺が同じ型で等しい
!=	型の相互変換後、右辺と左辺が等しくない
<>	型の相互変換後、右辺と左辺が等しくない(!=と同様)
!==	右辺と左辺が同じ型でないか、等しくない
<	左辺が右辺より小さい
<=	左辺が右辺より小さいか等しい
>	左辺が右辺より大きい
>=	左辺が右辺より大きいか等しい
<=>	左辺が右辺と等しい場合 0, 左辺が右辺より大きい場合 1、左辺が右辺より小さい場合-1(三方比較演算子もしくは宇宙船演算子)

if 構文

if 構文を使うと、特定の条件が true の場合にのみ処理を実行することができます。状況に応じて動作を変えることができます。条件式は必ず bool 値(true, false)になる式でなければなりません。

また処理する文が 1 文しかない場合は {} (ブロック)を省略してもかまいません。

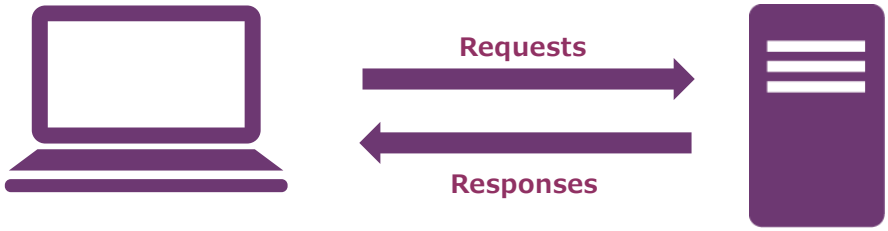
```
if (条件式) {
  // 条件式が true の場合のみ処理される
}

if (条件式)
  // 処理する文が 1 文の場合
```

Lesson10

クライアントとサーバ

これからの Lesson に必要な WEB の知識を少しおさらいしておきましょう。 まずは、WEB ブラウザで WEB ページを表示する仕組みを簡単に解説します。
WEB に接続されたコンピュータは **クライアント** と **サーバ** という役割で呼ばれます。
クライアントは、コンピュータやスマートフォンで利用できる WEB ブラウザ(Chrome や Firefox など)のことです。
サーバは WEB ページや WEB アプリを格納しているコンピュータのことです。
クライアントが WEB ページにアクセスする場合は、WEB ページのコピーがサーバからクライアントにダウンロードされ、クライアントの WEB ブラウザに表示されます。

クライアントからサーバにデータを要求することを request と呼び、サーバからの応答を response と呼びます。



IP アドレスとドメインと DNS

コンピュータが通信をする場合は **IP アドレス**と呼ばれる番号が必要になります。
IP アドレスには IPv4 と IPv6 がありますが、今回は IPv4(以下 IP アドレスと表記する)のみの説明に留めます。
インターネットに直接接続するコンピュータには、インターネット内でただ一つの、他のコンピュータとは決して重複しない IP アドレスが割り当てられます。
そのため、これを指定することで、通信する相手を特定できるわけです。
コンピュータは IP アドレスでしか通信相手のコンピュータを判断していません。
ですが、人には 63.245.215.20 のような IP アドレスは理解しづらく、代わりにドメイン名を 通信の相手を指定するのに、IP アドレスの代わりにドメイン名と呼ばれるコンピュータの名前を指定することもできます。
ドメイン名は mozilla.org のような名前で IP アドレス(63.245.215.20)と紐づける DNS という名前解決という仕組みがあります。

データの更新

データ更新もデータの挿入と手順は一緒で、exec()メソッドに UPDATE 文を渡し、結果は更新対象行の件数が取得できます。

Sample08.php

```
<?php
if ($_POST) {
  $host = 'localhost';
  $user = 'root';
  $pass = '';
  $dbname = 'php_study';
  $charset = 'utf8mb4';

  $dsn = 'mysql:host=' . $host . ';dbname=' . $dbname . ';charset=' . $charset;

  try {
    $con = new PDO($dsn, $user, $pass);
  } catch (PDOException $e) {
    echo '接続に失敗しました。', '<br />';
    echo $e->getMessage(), '<br />';
    exit;
  }

  $con->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
  $stmt = $con->prepare("update todos set title=:title where id=:id");
  $stmt->bindValue(':id', $_POST['id'], PDO::PARAM_INT);
  $stmt->bindValue(':title', $_POST['title'], PDO::PARAM_STR);

  $cnt = $stmt->execute();

  $stmt = $con->prepare('select * from todos');
  $stmt->execute();
  $rows = $stmt->fetchAll();

  echo '<pre>';
  echo $cnt . '件のレコードを更新しました。';
  print_r($rows);
  echo '</pre>';
}
?>

<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>Lesson11 Sample05</title>
</head>
<body>
  <form action="./sample08.php" method="POST">
    id:<input type="number" name="id"><br />
    タイトル:<input type="text" name="title"><br />
    <input type="submit" value="送信">
  </form>
</body>
</html>
```

上記のサンプルだと、JSON 形式のため PHP からはただの文字列として扱われます。
そこで、json_decode 関数を使用して JSON を PHP で扱えるようにデータ構造を変換します。

sample03.php

```
<?php
$params = ['zipcode' => 1500002, 'limit' => 5];
$url = 'https://zipcloud.ibsnet.co.jp/api/search?' . http_build_query($params);
$response = file_get_contents($url);
$decode = json_decode($response);

echo '取得した住所は' . $decode->results[0]->address1 . 'です。', '<br />';

print_r($decode);
```

出力結果

```
取得した住所は東京都です。
stdClass Object ( [message] => [results] => Array ( [0] => stdClass Object ( [address1]
=> 東京都 [address2] => 渋谷区 [address3] => 渋谷 [kana1] => ﾄｳｷｮｳﾄ [kana2] => ｼﾌﾞﾔｸ
[kana3] => ｼﾌﾞﾔ [prefcode] => 13 [zipcode] => 1500002 ) ) [status] => 200 )
```

cURL

file_get_contents 関数は幅広い HTTP リクエストが可能ですが、HTTP リクエストと HTTP レスポンスを詳細に制御したい場合は cURL 関数を使用します。

cURL 関数は、アクセスした URL を curl_init 関数に渡し、ハンドルを取得します。ハンドルは new PDO()や fopen 関数が返すオブジェクトと同じようなものです。curl_setop 関数は URL を取得する際の動作を制御(GET か POST などや、Cookie を渡すかなど)し、curl_exec 関数で実際にリクエストを実行します。

sample04.php

```
<?php
$params = ['zipcode' => 1500002, 'limit' => 5];
$url = 'https://zipcloud.ibsnet.co.jp/api/search?' . http_build_query($params);
$c = curl_init($url);
curl_setopt($c, CURLOPT_RETURNTRANSFER, true);
curl_setopt($c, CURLOPT_HTTPHEADER, ['Content-Type: application/json']);
$response = curl_exec($c);

$decode = json_decode($response);

echo '取得した住所は' . $decode->results[0]->address1 . 'です。', '<br />';

print_r($decode);
```

出力結果

```
取得した住所は東京都です。
stdClass Object ( [message] => [results] => Array ( [0] => stdClass Object ( [address1]
=> 東京都 [address2] => 渋谷区 [address3] => 渋谷 [kana1] => ﾄｳｷｮｳﾄ [kana2] => ｼﾌﾞﾔｸ
[kana3] => ｼﾌﾞﾔ [prefcode] => 13 [zipcode] => 1500002 ) ) [status] => 200 )
```