# Shin x blog

PHP 2010/01/27 09:25

## PHP基礎文法最速マスター ★15





PerlとRubyの文法一覧がとても良くまとまっていたので、便乗してPHPでもやってみました。

- Perl基礎文法最速マスター - Perl入門〜サンプルコードによるPerl入門〜
- Route 477 - Ruby基礎文法最速マスター

他の言語をある程度知っている人はこれを読めばPHPの文法を何となく理解できると思います。

間違い、不足等々あれば教えて下さいm(_ _)m

バージョン

PHP5.3系がリリースされていますが、ここではPHP5.2系を対象としています。

### 1.基礎

コードブロック

PHPコードは「<?php」という開始タグから始まります。終了タグは「?>」です。HTMLにPHPコードを埋める際は終了タグを使いますが、ライブラリのようにPHPコードのみを記述する際は終了タグを省くことが慣例となっています。

終了タグを省く理由について id:Kiske さんに解説頂いています。ありがとうございます。
PHP基礎文法最速マスターの補足 - Absolute Playing!

PLAIN TEXT

```
PHP:

1. <?php hoge(); ?>
2. <?php
3. $i = 1;
4. hoge($i);
5. ?>
```

PLAIN TEXT

```
PHP:

1. <?php
2. hoge();
```

print文

print/echoを使います。

PLAIN TEXT

PHP:

```
<?php
print "Hello World!";
echo "Hello World!";
```

デバッグにはvar_dump()をよく使います。var_dump()では変数の内容が出力されます。

PLAIN TEXT

PHP:

```
<?php
$array = array(1,2,3);
var_dump($array);

array(3) {
  [0]=>
  int(1)
  [1]=>
  int(2)
  [2]=>
  int(3)
}
```

コメント

一行コメント

PLAIN TEXT

PHP:

```
// コメント
# コメント
```

複数行コメント

PLAIN TEXT

PHP:

```
/*
コメント
コメント
*/
```

変数の宣言

変数の宣言です。

PLAIN TEXT

```php
$a = 'string';
$i = 1;
```

スクリプトの実行

コマンドラインでPHPファイルを実行します。

$ php hoge.php

PHPコードを直接記述することもできます。<?php ?>は不要です。

$ php -r "var_dump('a');"

出力結果をファイルに書き出すにはリダイレクトを使います。

$ php hoge.php > out

スクリプトの文法チェック

phpコマンドで文法がチェックできます。

$ php -l hoge.php

## 2. 数値

数値の表現

数値には整数、浮動小数点があります。

PLAIN TEXT

```php
<?php
$int = 100;
$float = 100.123;
```

四則演算

PLAIN TEXT

```php
<?php
$i = 1 + 1;
$i = 1 - 1;
$i = 1 * 1;
$i = 1 / 2;
```

余りと商。商を求めるには普通の除算を行った後にintval関数で整数部を取り出します。

PLAIN TEXT

```php
<?php
$div = intval(3 / 2); // 商
$mod = 3 % 2; // 余り
```

インクリメントとデクリメント

```php
<?php
$i++;  // インクリメント
$i--;   // デクリメント
```

## 3. 文字列

### 文字列表現

文字列はシングルクォートかダブルクォートで囲みます。ダブルクォートの中では¥t(タブ)や¥n(改行)などの特殊文字を利用することができます。またダブルクォートで囲まれた文字列の中では変数展開することができます。

```php
<?php
$str1 = "abc\tcde"; // abc  cde（\tがタブ[0x09]）
$str2 = 'abc\tcde'; // abc\tcde（\tが文字列）

$str3 = "$str1 100"  // abc  cde 100 //$str1が展開される
$str4 = "{$str1}100"  // 変数名に文字列が繋がる時は{}で囲む
```

### 文字列操作

```
```

PHP:

```php
<?php
// 結合
$join1 = 'aaa' . 'bbb';
$join2 = implode(',', array('aaa', 'bbb', 'ccc'));

// 分割
$split = explode(',', 'aaa,bbb,ccc');

// 長さ
$length = strlen('abcdef');
// 長さ（マルチバイト）
// 内部エンコーディングの設定が必要
// mb_internal_encoding('UTF-8');
$mb_length = mb_strlen('あいうえお');

// 切り出し
$substr = substr('abcd', 0, 2); // ab

// 検索
$index = strpos('abcd', 'bc'); // 見つかったらその位置（先頭が
```

## 4. 配列、連想配列

PHPには連想配列しかありません。配列はキーが数値の連想配列として表現されます。また順序を持っています。

PLAIN TEXT

PHP:

```php
<?php
$array1 = array(1, 2, 3); // 配列（キーが0から始まる連想配列）
$array2 = array('a' => 1,  'b' => 2, 'c' => 3); // 連想配列
$array3 = array(1, 'a' => 1, 2); // 混在もok
```

要素の参照と代入

PLAIN TEXT

PHP:

```php
<?php
$i = $array1[0];
$s = $array2['a'];
```

PLAIN TEXT

PHP:

```php
<?php
$array1[3] = 1;
$array2['z'] = 'zzz';
```

要素の個数

PLAIN TEXT

```php
<?php
$len = count($array1);
```

配列の操作

PLAIN TEXT

```php
<?php
$array = array(1, 2, 3);
// 先頭を取り出す
$first = array_shift($array); // $first = 1 / $arrayは、(2,
// 先頭に追加
array_unshift($array, 5); // $arrayは、(5, 2, 3)
// 末尾を取り出す
$last = array_pop($array); // $last = 3 / $arrayは、(5, 2)
// 末尾に追加
array_push($array, 9); // $arrayは、(5, 2, 9)
```

連想配列に関する関数

PLAIN TEXT

```php
<?php
// キーの取得
$keys = array_keys($array);
// 値の取得
$values = array_values($array);
// キーの存在確認
$boolean = array_key_exists('key', $array);
// キーの削除
unset($array['key']);
```

## 6. 制御文

if文

PLAIN TEXT

```php
<?php
if (条件) {
}
```

HTML内で記載する際は以下のような記法も用いられます。

PLAIN TEXT

```
PHP:

1. <?php if (条件): ?>
2.   <span>hoge</span>
3. <?php endif; ?>
```

if 〜 else文

```
PLAIN TEXT
PHP:

1. <?php
2. if (条件) {
3. } else {
4. }
```

HTML内で記載する際は以下のような記法も用いられます。

```
PLAIN TEXT
PHP:

1. <?php if (条件): ?>
2.   <span>hoge</span>
3. <?php else: ?>
4.   <span>foo</span>
5. <?php endif; ?>
```

if 〜 else if 文

elseif or else ifが可。

```
PLAIN TEXT
PHP:

1. <?php
2. if (条件) {
3. } else if {
4. }
```

HTML内で記載する際は以下のような記法も用いられます。

```
PLAIN TEXT
PHP:

1. <?php if (条件): ?>
2.   <span>hoge</span>
3. <?php elseif (条件): ?>
4.   <span>foo</span>
5. <?php endif; ?>
```

while文

PLAIN TEXT

PHP:

```php
<?php
$i = 0;
while ($i <5) {
  // 処理
  $i++;
}
```

HTML内で記載する際は以下のような記法も用いられます。

PLAIN TEXT

PHP:

```php
<?php while ($i <5): ?>
  <span><?php echo htmlspecialchars($i); ?></span>
  <?php $i++; ?>
<?php endwhile; ?>
```

for文

PLAIN TEXT

PHP:

```php
<?php
for ($i = 0 ; $i <5 ;$i++) {
}
```

HTML内で記載する際は以下のような記法も用いられます。

PLAIN TEXT

PHP:

```php
<?php for ($i = 0 ; $i <5 ; $i++): ?>
  <span><?php echo htmlspecialchars($i); ?></span>
<?php endfor; ?>
```

foreach文

連想配列の各要素を処理できます。

PLAIN TEXT

PHP:

```php
<?php
foreach ($array as $v) {
  // $v が要素の値
}
foreach ($array as $k => $v) {
  // $k が要素のキー、$v が要素の値
}
```

HTML内で記載する際は以下のような記法も用いられます。

```
PLAIN TEXT
PHP:

<?php foreach ($array as $v): ?>
  <span><?php echo htmlspecialchars($v); ?></span>
<?php endforeach; ?>
```

## 7. サブルーチン（関数）

PHPには関数があります。戻り値を返却するにはreturnを使用します。

```
PLAIN TEXT
PHP:

<?php
function sum($v1, $v2) {
  return $v1 + $v2;
}
$total = sum(1, 2); // $total = 3

// 配列で多値を返すこともできます
function get_multi($v1, $v2) {
  $v1 += 100;
  $v2 += 200;
  return array($v1, $v2);
}

list($ret1, $ret2) = get_multi(1, 2); // $ret1 = 101 / $re
```

## 8. ファイル入出力

ファイル入出力にはいくつかの方法があります。

fopen関数

ファイルポインタを使ってファイルの入出力を行います。

```
PLAIN TEXT
```

```php
<?php
// 読み込み
$fp = fopen("/path/to/file", "r");
if (!is_resource($fp)) {
  die("can't open file");
}

while (!feof($fp)) {
  $line = fgets($fp, 4096);
  // 何か処理
}
fclose($fp);

// 書き込み
$fp = fopen("/path/to/file", "w");
if (!is_resource($fp)) {
  die("can't open file");
}

fputs($fp, $buff);
fclose($fp);
```

file関数

ファイル全体を読み込んで配列に格納します。

```php
<?php
$list = file("/path/to/file"); // ファイルの各行を連想配列で取
```

file_get_contents関数 / file_put_contents関数

file_get_contents関数はファイル全体を読み込んで文字列として格納します。file_put_contents関数は変数の値を全てファイルに書き込みます。

```php
<?php
// 読み込み
$contents = file_get_contents("/path/to/file"); // ファイル
// 書き込み
file_put_contents("/path/to/file", $buff); // ファイルに $bu
```

## 知っておいた方が良い文法

### 真偽値

PHPでは以下の場合、偽と判断されます。

- boolean の FALSE
- integer の 0 (ゼロ)
- float の 0.0 (ゼロ)
- 空の文字列、および文字列の "0"
- 要素の数がゼロである 配列
- メンバ変数の数がゼロである オブジェクト (PHP 4のみ)
- 特別な値 NULL (値がセットされていない変数を含む)
- 空のタグから作成された SimpleXML オブジェクト

==と===

==/!=といった比較演算子では、数値・文字列の自動変換が行われます。よって意図しない結果をもたらす場合があります。

PLAIN TEXT

PHP:

```
<?php
var_dump(1 == 1); // true
var_dump(1 == '1'); // true
var_dump(0 == 'a'); // true
var_dump(100 == '100a'); // true
var_dump('+1' == '1.0'); // true
```

こういった場合、===/!==を使うと変数の型も厳密に比較することができます。

PLAIN TEXT

PHP:

```
<?php
var_dump(1 === 1); // true
var_dump(1 === '1'); // false
var_dump(0 === '0'); // false
var_dump(100 === '100a'); // false
var_dump('+1' === '1.0'); // false
```

変数が定義されているかどうか

変数が定義されているかどうかを調べるにはisset関数を使用します。定義されている場合はtrueが返ります。ただしisset関数では変数の値がNULLの場合もfalseが返ります。

PLAIN TEXT

PHP:

```
<?php
isset($a);
```

コマンドライン引数

コマンドライン引数を受け取るには$argv変数を使用します。

PLAIN TEXT

```
PHP:

1. <?php
2. var_dump($argv);
```

array_map

array_map関数を使うと、連想配列の各要素に処理をして新たな連想配列として受け取ることができます。

```
PLAIN TEXT
PHP:

1. <?php
2. $array = array(1,2,3);
3. $mapped = array_map(create_function('$v', 'return $v *= 10
```

array_filter

array_filter関数を使うと、条件に一致した要素のみを新たな連想配列として受け取ることができます。

```
PLAIN TEXT
PHP:

1. <?php
2. $array = array(1,2,3,4);
3. $filtered = array_filter($array, create_function('$v', 're
```

複数の変数への代入

```
PLAIN TEXT
PHP:

1. <?php
2. list($v1, $v2, $v3) = array(1, 2, 3);
```

php.ini

PHPには設定ファイルがあります。設定に応じて挙動が変わるので注意が必要です。この設定はphp.iniという設定ファイルの他に、httpd.conf、.htaccess、そしてソースコードにて設定が可能です。

設定方法は、項目に応じて変わりますが、ソースコードで設定を行う際はini_set()を使うことが多いです。

```
PLAIN TEXT
PHP:

1. <?php
2. ini_set('include_path', '.:/path/to/libs'); // include_pat
```

現在の設定は、phpinfo関数もしくはphpコマンドで確認できます。

```
PLAIN TEXT
```

```
PHP:

1. <?php
2. phpinfo();
```

```
# 全ての設定値を出力
$ php -i
# grep で絞る
$ php -i | grep include_path
```

クラス定義

classでクラスを定義できます。

```
PLAIN TEXT
```

```
PHP:

1. class User {
2.   protected $name = null;
3.
4.   public function __construct($name) {
5.     $this->name = $name;
6.   }
7.
8.   public function hello() {
9.     printf("%s: Hello!\n", $this->name);
10.  }
11. }
12.
13. $user = new User('Mike');
14. $user->hello();
```

継承もできます。単一継承のみ可能です。

```
PLAIN TEXT
```

```
PHP:

1. class MyUser extends User {
2. }
```

例外

throwで例外を投げることができます。try/catchで例外をキャッチします。他の言語にあるfinallyに相当するものはありません。

```
PLAIN TEXT
```

```php
function foo() {
  throw new Exception();
}

try {
  foo();
} catch (Exception $e) {
  echo $e->getTraceAsString();
}
```

## PHP参考資料

### 公式マニュアル

PHPに関する書籍は多く出版されていますが、やはり一番参考になるのは公式マニュアルです。

PHP: PHP マニュアル - Manual

公式マニュアルを使う際にちょっとしたTIpsを。

ブラウザで公式マニュアルを開く際は、http://php.net/の後ろに調べた関数名を入力すると直接そのページが開きます。
合致するものがなければ類似するキーワードが一番表示され、候補の中から選択することもできます。

http://php.net/array

### コーディング規約

コーディング規約にはいくつか流派があるのですが、Zend Frameworkのコーディング規約が参考になるでしょう。

Zend Framework PHP 標準コーディング規約 - Zend Framework Manual

### モダンPHP

PHPにはオブジェクト指向言語としても機能があります。以下の資料が参考になります。

モダンPHP勉強会を開催しました & 資料 - 肉とご飯と甘いもの @ sotarok

### フレームワーク

PHPを使ったWebシステム開発ではフレームワークを用いることがメジャーになりつつあります。

多くのオープンソースフレームワークがありますが、主要なものは以下です。

- CakePHP
- symfony
- Zend Framework
- Ethna
- CodeIgniter

■Related Posts

- Wiiリモコンが反応しない
- PHP 5.2.4 の新機能
- Zend PHP 5 Certification self test
- PHP5.2.4ではPHPエラーでHTTP 500を返す
- PHPフレームワーク

## 11 Responses to “PHP基礎文法最速マスター”

1. on 27 1月 2010 at 18:15 1.PHP基礎文法最速マスター | Shin x blog ≪ とっても！　ちゅどん（雑記帳） said …

   [...] PHP基礎文法最速マスター | Shin x blog PHP基礎文法最速マスター | Shin x blog [...]

2. on 28 1月 2010 at 10:52 2.Absolute Playing! said …

   [php]PHP基礎文法最速マスターの補足...

   id:shin1x1 さんがPerl、Rubyに続いてPHP基礎文法最速マスター | Shin x blogで基礎文法をまとめてくれています。ありがとうございます。空いた時間で書いてみようかなと思っていたら先越されてしまいました。記事を読んでみてコードブロックの説明を最初読んだときに少し...