
forever 朝活

php 講座 クラスの巻



【目次】

php の基本.....	エラー! ブックマークが定義されていません。
php とは?	エラー! ブックマークが定義されていません。
Web サーバーの代用品 xampp.....	エラー! ブックマークが定義されていません。
htdocs がドキュメントルート	エラー! ブックマークが定義されていません。
プログラムの基本.....	7
変数の復習	エラー! ブックマークが定義されていません。
フォームデータを php で受け取る.....	エラー! ブックマークが定義されていません。
セッションでデータを渡す	エラー! ブックマークが定義されていません。

関数を作ってみる

今日は php の関数を作ってみましょう。

関数とは？

php の関数は、教科書ではファンクションと読んでいました。

教科書でやったのは、`htmlspecialchars()`を簡単に記述するためにファンクションで `h()`でした。

これは、DOCTYPE の前に記述しました。

```
<?php
function h($value) {
    return htmlspecialchars($value, ENT_QUOTES,'UTF-8');
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

関数は、`function` で宣言します。しかし、宣言しただけでは、実行されません。かならず、それをコールする部分が必要です。

1箇所からだけでなく、何箇所からコールされても、動きます。ですから、よく使う関数はまとめておくと、あとから使用するとき便利です。

関数を作ってみよう

では、php にない機能を関数として作ってみましょう。

では、「西暦を和暦に変換する」関数を計画します。

データベースを使っては、面倒になりますので、あらかじめ切り替わりの日をまず覚えておきましょう。

明治:1868年1月25日

大正:1912年7月30日

昭和:1926年12月25日

平成:1989年1月8日

これらが、切り替わりの日付になります。日付は、unix 時間に変換して計算する方法があります

が、もっと単純に日付を8桁の整数として扱う方法があります。

例えば、2014年2月4日を20140204とするやり方です。こうすると、比較をする時などは簡単になります。

まず、この方法を考えてみましょう。

今日の日付を年、月、日で取得して、それを sprintf で組み立てるとすると

```
$year = date('Y');
$month = date('n');
$day = date('j');
echo '<p>西暦 : '.$year.'/'.$month.'/'.$day.'</p>';
$date = (int) sprintf('%04d%02d%02d',$year,$month,$day);
```

sprintf の桁数固定の変換で月が2でも%02d に変換すると、02 と、4日の4は%02d で 04 となります。それを(int)でキャストすると、整数の\$date にすることができます。

では、これを使って、今日の日付を取得して、西暦と整数を表示するプログラムを組んでみましょう。

sample1.php

```
<body>
<?php
$year = date('Y');
$month = date('n');
$day = date('j');
echo '<p>西暦 : '.$year.'/'.$month.'/'.$day.'</p>';
$date = (int) sprintf('%04d%02d%02d',$year,$month,$day);
echo '<p>整数にしたら : '.$date.'</p>';
?>
```

これができたら、この機能を関数にしてみましょう。

関数名は最終的に和暦にするので、toWareki()にしましょう。

sample2.php

```
<?php
function toWareki($y,$m,$d) {
    $date = (int) sprintf('%04d%02d%02d',$y,$m,$d);
    return $date;
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>西暦を和暦に変換してみる</title>
</head>
<body>
<?php
$year = date('Y');
$month = date('n');
$day = date('j');
echo '<p>西暦 : '.$year.'/'.$month.'/'.$day.'</p>';
echo '<p>整数にする : '.toWareki($year,$month,$day).'</p>';
?>
</body>
</html>
```

return で返すと、それをコール元で取得できます。そのまま echo すれば表示されます。整数値になれば、あとは簡単です。切り替わりの年数と比較して、和暦に当てはめていきます。

```
if ($date >= 19890108) {  
    $label = '平成';  
    $localYear = $y - 1988;  
} elseif ($date >= 19261225) {  
    $label = '昭和';  
    $localYear = $y - 1925;  
}
```

大きい値から順に割り当てていきます。\$localYearはその和暦の中の年数です。

では、エラーチェックなども入れて作り上げていきましょう。

sample3.php

```
<?php
/*
    明治:1868年1月25日
    大正:1912年7月30日
    昭和:1926年12月25日
    平成:1989年1月8日
*/
function toWareki($y,$m,$d) {
    //明治5年以降のみ
    if (!checkdate($m,$d,$y) || $y < 1873) {
        return false;
    }
    $date = (int) sprintf('%04d%02d%02d',$y,$m,$d);
    //日付で分割
    if ($date >= 19890108) {
        $label = '平成';
        $localYear = $y - 1988;
    } elseif ($date >= 19261225) {
        $label = '昭和';
        $localYear = $y - 1925;
    } elseif ($date >= 19120730){
        $label = '大正';
        $localYear = $y - 1911;
    } else {
        $label = '明治';
        $localYear = $y - 1868;
    }
}
```

sample3.php 続<

```

//1 年は元年
if ($localYear == 1) {
    $wareki = $label . '元年';
} else {
    $wareki = $label . $localYear . '年';
}
return $wareki . $m. '月' . $d . '日';
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>西暦を和暦に変換してみる</title>
</head>
<body>
<?php

$year = date('Y');
$month= date('n');
$day = date('j');
echo '<p>西暦 : '.$year.'/'.$month.'/'.$day.'</p>';
echo '<p>和暦 : '.toWareki($year,$month,$day).'</p>';
?>
</body>
</html>

```

まず、日付のチェックを `checkdate` で行います。引数が月、日、年ですので間違えないように。また、1 年を元年に変えます。

クラスにしてみる

では、今度はこの関数をクラスにしてみましょう。

DOCTYPE の前で宣言するのではなく、ひとつのまとまった機能ごとにクラスを作ります

暦クラスの作成

クラスを作るのは有益な関数を機能としてまとめたいからです。

そこでさきほど作った和暦変換を Koyomi クラスに入れてみましょう。

```
class Koyomi {
    public function Koyomi() {
        date_default_timezone_set('Asia/Tokyo');
    }
    //西暦和暦変換：年、月、日
    public function toWareki($y,$m,$d) {
```

クラスは別ファイルにしておきましょう。

ひと目で分かるように、class.php の添字を付けます。

クラス名の頭文字は大文字にして、コンストラクタとメソッドを作ります

Java のクラスと少し違って、public でも内部参照で \$this-> を使用します。

Koyomi.class.php

では、クラスを宣言します。

Koyomi.class.php

```
<?php
class Koyomi {
    /*
        明治:1868年1月25日
        大正:1912年7月30日
        昭和:1926年12月25日
        平成:1989年1月8日
    */
    global $MEIJI = 18680125;
    public $TAISYO = 19120730;
    public $SYOWA = 19261225;
    public $HEISEI = 19890108;

    public function Koyomi() {
        date_default_timezone_set('Asia/Tokyo');
    }
}
```

続きます。

```

//西暦和暦変換：年、月、日
public function toWareki($y,$m,$d) {
    //明治5年以降のみ
    if (!checkdate($m,$d,$y) || $y < 1873) {
        return false;
    }
    $date = (int) sprintf('%04d%02d%02d',$y,$m,$d);

    //日付で分割
    if ($date >= $this->HEISEI) {
        $label = '平成';
        $localYear = $y - 1988;
    } elseif ($date >= $this->SYOWA) {
        $label = '昭和';
        $localYear = $y - 1925;
    } elseif ($date >= $this->TAISYO){
        $label = '大正';
        $localYear = $y - 1911;
    } else {
        $label = '明治';
        $localYear = $y - 1868;
    }

    //1年は元年
    if ($localYear == 1) {
        $wareki = $label . '元年';
    } else {
        $wareki = $label . $localYear . '年';
    }
    return $wareki . $m . '月' . $d . '日';
}
}
?>

```

クラスを呼ぶ側

そして、コール元で `require_once` でクラスソースを読み込んでおきます。
そして、コンストラクタを使って、オブジェクト変数を作ります。

sample4.php

```
<?php
require_once('Koyomi.class.php');
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>西暦を和暦に変換してみる</title>
</head>
<body>
<?php
$koyomi = new Koyomi();

$year = date('Y');
$month = date('n');
$day = date('j');
echo '<p>西暦 : '.$year.'/'.$month.'/'.$day.'</p>';
echo '<p>和暦 : '.$koyomi->toWareki($year,$month,$day).'</p>';
?>
</body>
</html>
```

では、時間が余った人は、別のメソッドも作りこんでみましょう。

forever 朝活 php 講座

2014 年 1 月 20 日 初版発行

著作/制作：株式会社フォーエバー

〒890-0053 鹿児島市中央町 22-16 アエールプラザ 4F

TEL:099-296-9118 FAX : 099-250-2333 <http://www.forever.co.jp>

●本書は、構成・文書・プログラム・画像・データなどのすべてにおいて、著作権上の保護を受けています。

本書の一部あるいは全部について、いかなる方法においても複写・複製など、著作権法上で規定された権利を侵害する行為を行うことは禁じられています。