

## 目次

1. 導入 ～ プログラミングは難しくありません～
  - 1-1 様々なプログラミング言語
  - 1-2 PHPとは？
  - 1-3 PHPの優位な点 ～HTMLとの親和性～
  - 1-4 覚えればどんなことができる？
2. 文字を出力してみよう
  - 2-1 準備 -PHPが動作する環境を作る-
  - 2-2 HTMLファイルをPHPファイルに変換
  - 2-3 PHPを組み込んで文字を表示
  - 2-4 時刻を表示する
  - 2-5 PHPに組み込まれる色々な関数
3. 条件によって表示を変えてみよう
  - 3-1 条件分岐
  - 3-2 分岐することによって変えられる便利な処理
  - 3-3 ブラウザの判別とCSSの制御
4. 発展 ～もう一歩進んだPHPの処理～
5. 最後に (2008/11/29) ～PHPフレームワーク 「CodeIgniter」～
6. 補足

導入 ～プログラミングは難しくありません～

日頃HTML+CSSでページを作成されている方に、「システム」や「プログラミング」という言葉について聞くと、「難しいから無理」とか、「プログラマーの仕事だ」といった答えを頂きます。

ブログやSNSを作りたい、という大規模なものから、お問い合わせフォームを作る、といった簡易なものまで、ページを表示する、という静的な部分以外の動きには必ず何らかのプログラムが動いています。そしてそれをちゃんと動作させるには、やはりそれを作る人が必要になります。

ではそれは完全にプログラマーの仕事なのでしょうか。

それは必ずしもそうだとは言いきれません。

確かにシステム開発には専門家がいますし、そこには「プログラミング」という専門外の工程が存在します。

しかしそれはあくまで「専門家がいる」というだけであり、皆さんが手を出してはいけない分野ではないのです。さらに言えば、お問い合わせフォーム等は、専門家に頼まないといけないほど複雑な事をしているわけではありません。

「フリースクリプト」という言葉をご存知でしょうか。これらはメール送信プログラムなどをパッケージにしたものであり、無償で配布しているサイトがたくさんあります。皆さんも導入した事があるかもしれません。

そこで、導入に当たって問題が浮上します。

1. ソースを見てもちんぷんかんぷん。
2. デザインが上手く組み込めない。サイトの雰囲気にならせない。

このような所でしょうか。特にデザイナーの方にとって2は大きな問題だと思います。著作権表記等で困ったケースもよく聞きます。

では、1はどうでしょうか？実際にソースを見てもちんぷんかんぷんでしょう。実はそれほど複雑なことをしている訳ではないのに、全く意味が分からないと思います。

例えば、

```
<?php echo htmlspecialchars($message, ENT_QUOTES);?>
```

こんな事がHTML上に書かれていたら、迂闊に触ってはいけないような気がしてしまいますよね。でも気にする必要はありません。何度も言いますが、決して難しい処理をしている訳ではありません。

ただ「知らない」だけなのです。

だから一度知ってしまえば、取るに足らない問題となります。さらに、これらを知ってしまえば、あとは皆さんのデザイン力を十分に生かし、ページを装飾するだけ。その点はプログラマーでは及ばない部分であり、皆さんが自信を持って作業をしていただける部分でもあります。

まとめれば、

「プログラミングを少しでも分かっているデザイナーは、よりリッチなページを作成できる」

これが結論です。さらに Web ディレクションをされている方であれば、デザイン面だけでなく、プログラム面からも構成を考えられる筈です。これはコーダーにとってもプログラマーにとっても、さらにディレクターにとっても有意義なことであると言えるでしょう。

本勉強会が、デザイナーの方のより良い Web サイト作りの糧となれば幸いです。

日本 CodeIgniter ユーザ会  
杉本 吉章

## 1-1 様々なプログラミング言語

Web システムにおいて動作しているアプリケーションは、本当に多様な言語で書かれています。例を挙げましょう。

- A S P
- J a v a
- P e r l
- R u b y

前述した「フリースクリプト」と呼ばれる物は Perl (パール) という言語で書かれている場合が多いです。「CGI」という言葉を耳にした方もおられると思いますが、「CGI」というのは Web 上で動作するプログラムを総称しており、後述する PHP で書かれたものもありますが、Perl で記述された物が圧倒的に多いので CGI と言えば Perl を指すことが殆どです。

では、今回学んでいただく PHP 言語についてです。

## 1-2 PHP とは？

PHP とは、サーバー上で動作する「サーバーサイドスクリプト」です。ブラウザで動作することを前提とした、JavaScript や ActionScript などの「クライアントサイドスクリプト」とは対称的となる存在ですが、PHP を用いることにより、JavaScript などではできない処理を実現することが出来ます。メールの送信やデータベースとの連携など、とてもシステムティックな処理を行うことが出来ます。

## 1-3 PHP の優位点 ～HTML との親和性～

デザイナーの方にプログラミングを覚えていただく上で、最もお勧めしたいのが PHP です。

その理由が「HTML との相性の良さ」です。

1-1 で紹介したような言語もありますが、PHP の良さは HTML 上に書き込める、という点です。

今までに作成された HTML ファイルを少し変えるだけで PHP として動作させることも可能ですし、

逆にPHPからHTMLを出力させることも勿論可能です。つまり、HTML+CSSでページを作成されている方には最も始めやすい言語だと私は思います。

また、他の言語に比べて習得が比較的容易だという点や、大抵のレンタルサーバでは、既にPHPがインストールされているというのも、PHPが他の言語に比べて優位な点といえるでしょう。

#### 1-5 覚えればどんなことが出来る？

突き詰めて言えば、現在稼働している殆どのシステムが開発できます。ブログサイトやソーシャルネットワーキングサービス（SNS）。これらの多くは、PHPで記述されたアプリケーションです。

とはいえ、そこまで大規模な物はそう簡単に開発することは出来ません。まずは簡単なアプリケーションから開発していきましょう。

趣旨としては、フリーCGIを使わなくてもよくなるくらいまでの知識を持っていただきたいと思います。大規模なシステムにはやはり専門家の力が必須でしょうから。

## 2. 文字を出力してみよう

### 2-1 準備 ～PHPの動作する環境を作る～

PHPはサーバーサイドスクリプトであり、通常の手を加えていないローカルPCで動作させることはできません。そこで、ローカルでも動かせる「XAMP (MAMP)」というソフトを入れて環境を作りましょう。

#### ▶ ダウンロード、インストール

本勉強会ページのリンクから各マシン用のXAMP (MAMP) をダウンロードしましょう。あとは、そのファイルを解凍すれば使用することができます。インストール先はCドライブ直下でOKです。

Macの場合は、dmgファイル形式になっていますので、ダウンロード後のdmgファイルをダブルクリックすると、インストールのための準備が始まり、XAMPをインストールすることができます。

#### ▶ 各サーバーに合わせた設定

インストールすればPHPは動作しますが、普段皆さんは自分でサーバーを立てたりはしないでしょう。レンタルサーバーなどにアップロードする事が殆どだと思います。そこで、先ほどインストールしたXAMP (MAMP) の設定をレンタルサーバーに合わせた設定に変更します。これらの作業は複雑になるので、一緒にお配りしたCD内のファイルを上書きする形で対応します。

Windows マシンの場合 (Macの方は適宜読み替えてください)

マイコンピュータ → Cドライブ → xampp → php

と移動すると、phpフォルダ内に「php.ini」というファイルが下の方にあります。ここにCD内のphp.iniファイルをコピーして上書きしてください。これで完了です。ローカルマシンがサーバーのように動作します。レンタルサーバーで言うドキュメントルートは、

マイコンピュータ → Cドライブ → xampp → htdocs

です。

### 2-2 HTMLファイルをPHPファイルに変換

先に作成してあるHTMLファイルとCSSファイル、イメージフォルダをhtdocsフォルダ内に配置します。その後、HTMLファイルの拡張子を変更します。

test.html → test.php

こんな感じです。アラート画面が出ますが気にしないでください。では、ブラウザからアクセスしてみましょう。サーバーのように動作させているので、ファイルをダブルクリックではダメです。ブラウザのアドレスバーに、

<http://localhost/test.php>

このように打ち込んでアクセスしましょう。きちんと表示されましたか？見かけ上は普通のHTMLファイルと同じですが、これでこのファイルにPHPスクリプトを埋め込む準備ができました。

それでは、実際にPHPを埋め込んでみましょう。

### 2-3 PHPを組み込んで文字を表示

PHPを組み込む前に、PHPのルールについて知っておきましょう。

#### PHPの埋め込み方

PHPを埋め込むには、「ここからここまでの部分はPHPのコードですよ」とコンピュータに認識させないといけません。そこで、

```
<?php ~ ?>
```

こんなタグを用います。この<?php ~ ?>で囲まれた部分に書かれたコードはPHPスクリプトであると認識されます。一部分だけでもかまいませんし、ソース全てがPHPスクリプトであれば、最初から最後まで囲っても問題ありません。

#### 処理文の最後は「; (セミコロン)」を書く必要がある

基本的にPHPでは1行に1処理、という流れが続きます。

そこで、各処理の終わりがどこであるかを教えるために、処理の最後にセミコロンを置い

て処理の区切りをつけます。

```
<?php
```

```
    $string = 'サンプル文字列';
```

```
    echo $string;
```

```
?>
```

このように、各行末にセミコロンをおきます。これを忘れてエラーが発生するケースはよくあるので注意してください。

また、PHP 内では、半角スペースとタブは無視されます。上記のコードではソースを見やすくするために半角スペースで隙間を取ってあります。できるだけ見やすいソースを書くように心がけましょう。ちなみに、全角のスペースは文字列と認識されるのでエラーとなります。気づかないケースがよくあるので注意してください。

それでは、試しに文字列を表示させてみましょう。指定した場所に、

```
<?php echo 'Hello world!';?>
```

このように入力して上書き保存します。その後、ブラウザでページを更新すると、上記のコードを挿入した部分に「Hello world!」と表示されます。これでPHPで文字を出力できるようになりました。

ここで注意点です。文字列の場合はそれらを「(シングルクオート)」または「”(ダブルクオート)」で囲む必要があります。数字の場合は必要ありません。シングルクオートとダブルクオートの違いもありますが、これは後述します。

## ➤ 「echo」文

PHPで何か文字を出力させるには、echo文を使用します。

この関数の引数に指定された文字列・変数はブラウザ上にそのまま出力されます。

この関数は基本なので、覚えて置いてください

プログラム内部でとても複雑な処理を行ったとしても、結局のところ出力はこのようなになります。

そういう意味ではPHPの末端部分を知っていただいたといっても過言ではありません。

ここで、注意点があります。

<?php ~ ?>で囲った部分に通常のHTMLを入れるとエラーが出る

という点に注意してください。

```
<?php
    <div id="box">
        <p>サンプルコード</p>
    </div>
?>
```

このようなコードがあったとします。一見して何の問題もないように思えますが、これはエラーとなります。

PHPのプログラミング上では、HTMLタグは‘文字列’として扱われます。そのため、ソース中にいきなり文字列が現れると、PHPはどうしていいか困り、エラーを出します。そこで、正しくは、

```
<?php
    echo '<div id="box">';
    echo '        <p>サンプルコード</p>';
    echo '</div>';
?>
```

このように逐一文字列をシングルクォートで囲み、echo関数で表示させる必要があります。これで出力されたタグはブラウザ上ではちゃんとHTMLタグだと認識するので安心してください。

#### ▶ 変数に格納して表示

PHPで文字や数字を扱う時、毎回100文字の言葉を打ち込むのは手間がかかります。そこで、プログラムではよくある「変数」というものに代入して、次からはその変数を使用することで処理を簡素化できます。

例えば、

```
$string = 'こんにちは、今日もいい天気ですね';
```

と記述すれば、変数\$string には右側の文字列が格納されます。表示する時はこの変数を指定すればOKです。

```
<?php
    $string = 'こんにちは、今日もいい天気ですね';
    echo $string ;
?>
```

### ▶ 文字列の連結

文字列を出力する時に、例えば苗字と名前が別々にあって、それをくっつけて表示したい場合があります。そのときは、文字列同士を「. (ドット)」でつなげます。

```
<?php
    echo '苗字' . '名前';
?>
```

こうすればドットでつなげた文字列は一つの文字列となって出力されます。PHP では上に挙げたような手法を組み合わせる望む文字列を出力させることができます。

ここで先ほどのシングルクォートとダブルクォートの違いを説明しましょう。シングルクォートの場合は、文字列がそのまま文字列として表示されますが、ダブルクォートで囲んだ場合は、その中に含まれる変数は展開されます。そのため、シングルクォート内で変数を入れても変数に文字がそのまま出力される為、変数がある場合はシングルクォートを終わらせて、ドットでつなげる、という処理を行います。分かりにくいので例を挙げましょう。

```
<?php
    $string = 'b';
    echo 'a' . $string . 'c'; //この場合は abc が出力
    echo 'a$stringc'; //この場合は a$stringc がそのまま出力されてしまう
    echo "a$stringc"; //この場合はエラー (致命的ではない)
    echo "a{$string}c"; //この場合は abc が出力
?>
```

1 番目と 2 番目は先ほど説明したとおりですね。では説明とおりならば 3 番目も期待通りの出力が得られそうですが、結果はエラーです。何故でしょうか？

その答えは、「どこまでが変数名なのか分からない」からなのです。3 番目を見ると、変数名は \$stringc のようにも見えませんか？この為、PHP は \$stirngc という変数だと認識して、実際そのような変数は定義されていないのでエラーとなるのです。そこで、PHP にどこまでが変数名かを教える為に「{ }」で囲みました。これで分かりますよね。

実際の出力には { } は表示されません。また、変数名の後ろが全角文字の場合はこのような事は起こりません。なので、アルファベットや半角英数を扱う時にだけ注意してください。

しかし、ここでも問題があります。

echo 関数は一行分しか表示させられない

ということです。

つまり、先ほどの HTML を出力させる部分ではわざわざ 3 つの echo 文を用いました。これを一つの echo 文で出力させるならば、

```
<?php
    echo '<div id="box"><p>サンプルコード</p></div>'
?>
```

というように一行にまとめてしまわないといけません。これでは分かりにくくなってしまいます。さらに変数も入れ込みたいのであれば、

```
<?php
    $string = 'サンプルコード';
    echo '<div id="box"><p>' . $string . '</p></div>';
?>
```

というようにさらに複雑になり、シングルクォートが抜けてたり、といった些細なミスが起こりがちになります（私はそうです）。

そこで、下のヒアドキュメントの出番です。

## ▶ ヒアドキュメント

100文字程度なら先ほどまでの方法で何とかいけますが（それでもかなりシンドイですね）、例えばメールの文面なんかを一括して出力したいときに全て入力し、さらに名前などの変数を逐一 { } で囲って…という工程はとても面倒です。HTML タグ全てを PHP で書き

出すなどという日には途方にくれてしまいます。

そこで、「ヒアドキュメント」と呼ばれるとても便利な仕組みがあります。

これを使えば、長い文章も通常と同じように記述でき、中に変数が含まれる場合でも{変数名}とすれば展開してくれます。

```
<?php
    $name = 'あなたの名前';
    $string = <<<END
これはテスト構文です。
とても長い文をここに入れたい場合は、このようにヒアドキュメントを使いましょう。
{$name}←これで名前もちゃんと出てくれます。
END;
    echo $string ;

?>
```

実際はもっと長い文もありえるでしょう。その場合もヒアドキュメントを用いれば簡単になりますね。

「<<<END」がヒアドキュメントを始める部分です。この END の後ろにはセミコロンがつかないので注意してください。

そして最後の END で終了、この間の文章がすべて文字列として格納されます。

終わりの END は必ず行頭に書くようにしてください。

実際の記述では、そこまで長い文章を出力させるケースは稀かもしれませんが、これは覚えておいて頂きたい処理です。

以上で、様々な出力方法を覚えていただきました。色々な方法で文字を自由に出力して見ましょう。次は、文字列だけでなく、関数で処理を行った結果を出力させる方法について学びます。

## 2-4 時刻の表示

サイト訪問時に更新時間や現在の時刻を表示するページはよくあります。掲示板などでは投稿日時を出力していますね。実はこれも PHP で簡単にできるのです。

先ほどのスクリプトの部分に、

```
<?php echo date("Y年m月d日 H時i分s秒にアクセスしました",time());?>
```

と記述して更新すると、更新した時刻が表示されます。

ここでは二つの関数を使用しています。

まず、「date 関数」です。その名のとおり、現在の時間を計算する関数です。「Y年～」の部分は表示の形式を指定しています。2008年と表示させたり、08年、といったように表示形式も細かく指定できます。

二つ目が「time ()」関数です。これは現在の時刻を「タイムスタンプ」という特殊な数値で返します。これを date 関数が自動的に解析して現在の時刻にして形式通りに表示してくれます。

指定した時刻を表示させたいときは、「mktime()」関数を使います。

```
mktime( 時 , 分 , 秒 , 月 , 日 , 年 )
```

各形式の指定の順番に注意してください。これを time()の代わりに入れれば、指定した時刻を表示してくれます。

プログラム内部で処理を行った結果を出力する際に、何らかの形式に指定して出力ケースもよくあります。例えば、得られた価格を3桁区切りの値段表示にして表示する、入力されたパスワードを「※」に変換して隠蔽する、などです。これらも別段特殊な処理を施す必要は無く、ちゃんとそれ専用の PHP 関数が用意されているので、それを用いるだけで OK なのです。

## 2-5 PHPに組みこまれる色々な関数

PHPで利用できる関数は膨大で、全てを覚えることは困難です。しかし、CSS プロパティで頻繁に用いられるプロパティがあるように、PHPにも頻繁に使われる関数があります。

- htmlspecialchars 関数（冒頭に書いた物です）
- count 関数
- str\_replace 関数
- print 関数
- print\_r 関数

他にも沢山あります。どのように使われ、どのような意味を持つのかはネット上にリファレンスがあるので、そちらを参考にしてみてください。今回使用する関数は適宜説明します。

### 3. 条件により表示を変えてみよう

ここからよりプログラミングっぽいことをやっていきましょう。  
動的に出力させる、といっても今までの方法では常に同じ表示しかされません。そこで、今度は条件を設定して表示を変えてみましょう。

#### 条件分岐には「if」を使う

ある指定した条件により表示する内容・行う処理を変えたい時には、「もし～ならば」という意味の if を使用します。

```
$a = 3 ;  
if( $a > 0 )  
{  
    echo 'a は 0 より大きいです' ;  
}  
else  
{  
    echo 'a は 0 より小さいです' ;  
}
```

この場合、if の条件評価は ( ) 内で行います。その結果が正 (true) ならば直ぐ下の処理を行い、それ以外ならば else 内の処理を行います。条件判定には「比較演算子」という記号や、boolean を返す関数を用いて評価します。比較演算子には数学のような記号を用いることができます(表1)。

表1. 比較演算子

記号	意味
==	左辺と右辺は等しい
>	左辺は右辺より大きい (大なり)
<	左辺は右辺より小さい (小なり)
>=	左辺は右辺と等しいまたは大きい (以上)
<=	左辺は右辺と等しいまたは小さい (以下)
!=	左辺と右辺は等しくない
===	左辺と右辺は等しく、型も同じ※
!==	左辺と右辺は等しくないか、型が同じでない※

表1の※に着目してください。「型」という言葉が出てきました。

この「型」という物はその対象の変数がどのような形式であるかを表します。

例えば、1,2,3…といった数字は数値型、'あいうえお'といった文字は文字列型です。

これらは当たり前ののですが、この認識でいると少し困ったことになります。

例を挙げます。

```
$hensuu = 1;           //これは数値型
$hensuu = '1';        //これは文字列型になります！
```

このように、同じ数字でも、シングルクオートで囲むとそれらは文字としての「1」であると認識されます。しかしPHPは賢い言語で、たとえ文字列としての数字でも、ちゃんと計算はしてくれるのです（実はそれ故に厄介な事も起こりえるのですが…）。

```
$hensuu = '1';
$hensuu_2 = '2';
$answer = $hensuu + $hensuu_2;
```

この計算結果はちゃんと3が返ってきます。これはPHPが自動的に型変換を行ってくれるからなのですが、この認識は少し危ないので、しっかりと型について知っておく必要があります。ちなみに、文字列なので連結もできます。文字列の連結は「.(ドット)」を使いましたね。

```
$hensuu = '1';
$hensuu_2 = '2';
$answer = $hensuu . $hensuu_2;
```

この結果は「1 2」という文字列が返ります。

ここで、PHPで扱う型をご紹介します。

数値(number)型	数字。主に整数はこの型です。
不動小数点型(float)型	小数点が付くものはこの型になります。
文字列(string)型	文字列はこの型です。数値としての文字もあります。
論理(boolean)型	正しい(true)か、正しくない(false)かのどちらか。
配列(Array)型	配列はこの型です。
オブジェクト(Object)型	オブジェクトの型です。

上から3つは当たり前なのでお分かりになると思いますが、下の3つは少し特殊です。下2つの今回説明を省略しますが、次回以降はこの型を扱うことになります。少しイメージが付きにくいものですが、慣れればとても簡単、しかも頻繁に使用するので覚えていただきたいです。

では比較演算子の話に戻りましょう。if文の結果はboolean型が返ります。つまり、正しいか、正しくないか。シンプルな答えが返ります。それによって処理を分岐させることができます。

もう一度式を見てみましょう。

```
$a = 3 ;
if( $a > 0 )
{
    echo 'a は 0 より大きいです' ;
}
else
{
    echo 'a は 0 より小さいです' ;
}
```

この「if( \$a > 0)」の部分では、\$aは0より大きいので…そう、trueが返りますね。よってtrueでは直下の処理が実行されます。

では\$aが-3だったらどうでしょうか？この場合はfalseが返りますね。よって、else以下の処理が実行されることになります。もっと細かく条件を指定することが出来ます。

複雑な条件を指定（あまりお勧めはできませんが…）する場合には、比較演算子を組み合わせ使います。先に組み合わせの演算子も紹介しておきましょう（表2）。

表2. 組み合わせに用いる演算子

&&	二つの条件が共に正しいならば (and)
	二つの条件のどちらかが正しいならば(or)

これらを組み合わせて使います。

```
$a = 5;
if( $a > 0 && $a <= 10) //$a が 1 以上、かつ 10 以下ならば true
{
    処理 ;
}
else
{
    処理 ;
}
```

これらはいくつでもつなげられます。

```
if( 条件 && 条件 && 条件 || 条件 && ...)
```

このようにできますが、これだと正直どの条件が true になるのかわかりませんよね。  
あまりに複雑な条件指定はしないほうが賢明です。

さらに、もっと処理を複数に分岐させることも出来ます。次のソースをご覧ください。

```
$a = 10;
if( $a > 0 && $a <= 5) //$a が 1 以上かつ 5 以下ならば true
{
    処理 ;
}
else if( $a > 5 && $a <= 10) //$a が 5 より大きく、かつ 10 以下ならば true
{
    処理 ;
}
else //それ以外
{
    処理 ;
}
```

else if~という部分が追加されました。これにより、最初の if 文で true にならなかったら、次の else if 文で評価します。それでも true にならなかったら、仕方がないので else 文の処理を行う、という流れが出来上がります。

このように、いくらでも条件を分岐させることが可能ですが、やはりこれも複雑になり過ぎない程度にしましょう（実はもっと複数の条件比較に適した書き方があるのですが、今回は省略させていただきます）。

この条件比較は PHP ではもっとも良く使用する文であると思います。しっかりと使いこなせるようになれば、あらゆる処理が可能になるでしょう。

### 3-2 条件により分けられる便利な処理

さて、長々と説明して来ましたが、早速この if 文で処理を分岐してみましょう。ちょうど前に作成した時刻を表示させるプログラムがありました。これを使って、アクセスした時間によって挨拶文を変える、という動的な処理を行ってみましょう。

まず、時刻を表示させるプログラムは以下のようでした。

```
<?php echo date("Y年m月d日 H時i分s秒にアクセスしました",time());?>
```

では、取得した時刻をそのまま表示させずに、条件評価するために一時的に変数に格納しましょう。さらに、今回は時間だけを評価したいので、時間だけを取り出します。

```
$jikoku = date( " H " , time() )
```

この「 H 」は、取得した時刻から 24 時間表記の時間だけを返してくれます。

その結果を変数 \$jikoku に格納しています。

では、これを条件評価してみましょう。こんな感じになります。

```

<?php
    $jikoku = date( " H " , time() );
    if( $jikoku > 6 && $jikoku < 12) //時刻が 7 時～11 時の間ならば true
    {
        $message = 'おはようございます';
    }
    else if( $jikoku >= 12 && $jikoku < 17) //時刻が 12 時～16 時の間ならば
true
    {
        $message = 'こんにちは';
    }
    else //それ以外は夜の挨拶
    {
        $message = 'こんばんは';
    }

    echo $message ;

?>

```

これで時間によって挨拶文を変えるプログラムが完成しました。

else 節に注目してください。このプログラムでは 17 時～23 時、0 時～6 時の間は夜の挨拶としています。

時刻は 0～23 の間（午後 24 時は 0 時となります）で返ってきます。よって、この夜の部分を if 節で実行させようとする若干手間のかかる処理となります。

```

if( ($jikoku >= 0 && $jikoku <= 6) || ($jikoku >= 17 && $jikoku < 24))

```

このように条件式がかなり複雑になります。これは無駄なエラーを排出する原因にもなりかねないので、複雑になりそうな部分は後回しにして、比較的簡単な条件を先に評価して行き、あとは「それ以外」としてまとめて処理を入れてしまいましょう。その方が分かりやすいですし、何より簡単になります。

複雑にプログラムを構築するのが素晴らしいのではありません。いかに簡素に、それでいて確実に実行されるプログラムこそが素晴らしいのです。

話を戻しましょう。このプログラムをファイル内の好きな位置に配置して実行してみましょう。時刻は皆さんの PC の時間を参照しますので、時刻を適宜変更してみて、挨拶文が変

わることを確認してみましょう。

### 3-3 ブラウザの判別と CSS の制御

さて、ここからが本題です。今回の勉強会で最もやりたかったことはこの部分になりますので、正直今まで説明したことは、このブラウザ判別と CSS 制御を行う為の布石でした。

デザイナーの方が CSS を組む上で最も手間がかかり、かつ厄介な問題が、「プロパティの各ブラウザのレンダリングの違い」だと思います。

Internet Explorer のマージン解釈の問題、標準モード、互換モードの違い、:after 疑似プロパティの解釈の有無の問題などなど、CSS 制御を行う上で各種ブラウザ別に対策を施す必要があり、CSS ハックを入れる、コンディショナルコメントで IE を制御する、といった二重の手間がかかってしまいました。

しかしながら、CSS ハックでは CSS バリデータに通らない、など、付随する問題も浮上します。「ブラウザのレンダリングを統一してくれたらどんなに楽だろうか…」などと考えた事が一度はあると思います。私は何度もあります。

しかし、それはあくまでクライアント側での処理に限った問題であり、PHP を含めて考えれば意外にもそれらの問題はあっさりと解決してしまいます。

少しでも「それは便利だ」と思っていただけならば、以下に進みましょう。

CSS の解釈の違いについて説明するととても長くなるので、今回はブラウザがページにアクセスした際にどのような事を行っているか、という点にのみ着目したいと思います。

各ブラウザが「このページを表示してください」とサーバーにリクエストした際に、実は自分がどんなブラウザなのか、どんな OS なのか、どこから接続しているのか、といった多様な情報を教えています。現実でも素性の分からない人からの電話は怖いですね。

とにかく、アドレスを入力して、ページが表示されるまでの間のコマ数秒の間にそんなやりとりが行われています。

そこで、サーバーにリクエストする際に、「自分がどんなブラウザなのか」という情報を教えている、これを利用してサーバー側で処理を行うことが出来るんです。はい、PHP の出番ですね。

少し見たことのない物が出てきますが、一度知ってしまえば次からは当たり前のように使

えます。CSS プロパティを 5 個覚えるより簡単です。

まずは準備です。各ブラウザに読み込ませたい CSS ファイルを必要数だけ作成します。

ie.css , firefox.css , opera.css , safari.css

これくらいあれば十分です(Mac の IE5 とかも考慮に入れるならばその分のファイルも作成してください)。あとは各ファイル内に各ブラウザ用に最適化した CSS を書けば OK です。では次に肝心のブラウザ判別に進みましょう。

ブラウザからサーバーに与える情報は、全て「\$\_SERVER」という「スーパーグローバル変数」というものに格納されています。誰かが値を入れたわけではなく、自動的にサーバーが格納します。

その多様な変数の中で、ブラウザ情報を入れているポイントは、

```
$_SERVER[ ' HTTP_USER_AGENT ' ]
```

という配列のキーになります。配列については今回省略させて頂きましたが、ともあれブラウザ情報はここに入っている、という認識をお願いします。

では、実際どのような情報が入っているのか、echo 関数で出力してみましょう。

出力は…`<?php echo $_SERVER[ ' HTTP_USER_AGENT ' ]; ?>`でしたね。

結果は、Internet Explorer なら、

```
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727; InfoPath.1)
```

Firefox なら、

```
Mozilla/5.0 (Windows; U; Windows NT 5.1; ja; rv:1.9.0.4) Gecko/2008102920  
Firefox/3.0.4
```

Opera なら、

```
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; ja) Opera 9.21
```

Safari なら、

```
Mozilla/5.0 (Macintosh; U; Intel Mac OS X; ja-jp) AppleWebKit/523.12.2 (KHTML,
```

like Gecko) Version/3.0.4 Safari/523.12.2

こんな感じの表示がされます。一見して解読不能のように見えますが…ある一点に注目してください。

Internet Explorer なら「MSIE」の文字列が、Firefox なら「Firefox」の文字が入っていますね(ちなみに windows マシンなら「Windows」の文字が、Mac なら「Mac」の文字も入ります)。つまり、代表されるブラウザ名がちゃんと入っている事が分かります。

もうお分かりですね。あとはこの文字列が入っているかどうかで条件分岐をさせればいいんです。

文字列に特定の文字が含まれているかを判別する為の関数も PHP はちゃんと用意されています。ereg 関数がそれです。

ereg ( '含まれる文字のパターン', 対象の変数 , 配列で返すかどうか)
--

このような形式で引数を指定します。3つ目の引数はとりあえず使いません。

では、早速ブラウザの判別を試してみましょう。

```

<?php
    $browser = $_SERVER[ 'HTTP_USER_AGENT' ]; //ブラウザ情報を変数に
格納
    //条件分岐開始
    if(ereg( "Opera" , $browser)) //Opera の文字が含まれるならば
    {
        $stylesheet = 'opera.css'; //Opera 用の CSS ファイルを指定
    }
    else if( ereg("Safari" , $browser)) //Safari の文字が含まれるならば
    {
        $stylesheet = 'safari.css'; //Safari 用の CSS を読み込む
    }
    else if( ereg("Firefox" , $browser)) //Firefox の文字が含まれるならば
    {
        $stylesheet = 'firefox.css'; //Firefox 用の CSS を読み込む
    }
    else //それ以外は IE とする
        $stylesheet = 'ie.css'; //IE 用の CSS を読み込む
    }
?>

```

このようにして各ブラウザ用に CSS を指定します。

ここで注意点です。前述の `$_SERVER[ 'HTTP_USER_AGENT' ]` の値を見て頂くと分かるのですが、Opera でアクセスした場合も「MSIE」の文字が含まれてる場合があります。このままでは IE か Opera か判別できないので、先に Opera の条件判定を行ってしまいます。よって、それで true にならなかったブラウザの MSIE は必然的に IE となります。先にも書きましたが、より簡単な条件から先に判定してしまうのは鉄則です。複雑な条件は else にまとめてしまうのが賢明です。

さて、まだ不完全なのはお分かりでしょうか。

そうです、CSS でも結構な難関となる IE の 6 と 7 のバージョンの判別が出来ていません。ここでさらに `$_SERVER[ 'HTTP_USER_AGENT' ]` の値に注目して頂きたいのですが、MSIE の後ろに「6.0」とか「7.0」とか数字がついていますね。実はこれが IE のバージョン情報なのです。よって、先ほどの条件判定の最後に、この数値について条件判定を行って完成です。

```

<?php
    $browser = $_SERVER[ 'HTTP_USER_AGENT' ]; //ブラウザ情報を変数に
格納
    //条件分岐開始
    if(ereg( "Opera" , $browser)) //Opera の文字が含まれるならば
    {
        $stylesheet = 'opera.css'; //Opera 用の CSS ファイルを指定
    }
    else if( ereg("Safari" , $browser)) //Safari の文字が含まれるならば
    {
        $stylesheet = 'safari.css'; //Safari 用の CSS を読み込む
    }
    else if( ereg("Firefox" , $browser)) //Firefox の文字が含まれるならば
    {
        $stylesheet = 'firefox.css'; //Firefox 用の CSS を読み込む
    }
    else if(ereg( "MSIE 6\\.0" , $browser)) //6.0 ならば IE6 である
        $stylesheet = 'ie_6.css'; //IE6 用の CSS を読み込む
    }
    else if(ereg( "MSIE 7\\.0" , $browser)) //7.0 ならば IE7 である
        $stylesheet = 'ie_7.css'; //IE7 用の CSS を読み込む
    }
    else
    {
        $stylesheet = 'default.css'; //どのブラウザにも当てはまらない
    }
?>

```

あとは変数として格納した CSS ファイル名を<link>タグ内に仕込むだけです。

```

<head>
    <link rel="stylesheet" type="text/css" href="<?php echo $stylesheet; ?>" />
</head>

```

結構苦労しましたが、この方式を使うと余計なハックやコンディショナルコメントも必要なく、非常にクリーンな HTML ソースが出来上がります。

とはいえ、実際にここまで厳密にブラウザ判別を行うケースは稀かもしれませんが（2,3 個のハックを埋め込むだけで大丈夫なときも多いですね）。

さらに、Mac 版のブラウザの判別も先ほどのソースに追記するだけで OK です。考えてみてください。

さらに、このブラウザ判別法は、携帯サイトで大きな効力を持ちます。携帯サイト製作で念頭に置くべきキャリア別の処理。絵文字なんかはキャリアによって違いますよね。

その場合でもこの方法を使えばキャリア判別が簡単にできます。あとは判定したキャリア用に HTML ファイルを読み込ませれば良い訳です（詳細は補足にて）。

#### 4. 発展 ～もう一步進んだ PHP の処理～

さて、ここからはもう一步進んだ処理を見てみましょう。詳細はまた次回以降に解説いたしますので、「ふ～ん、こんなものもあるんだ」程度に思っただけであれば幸いです。

- ・ アクセス毎に違う画像を表示させる

最近はあまり見かけませんが、アクセスする度に、またはアクセスした時間によって表示される画像を変えているサイトがあります。これはどのようにやっているのでしょうか？

実はこれもごく簡単な処理しかしていません。基本は、

何か関数で処理 → 結果を条件判定 → 出力

これだけです。そしてその処理用の関数も PHP ではちゃんと用意してくれています。

では、まずアクセスする度に画像を変える方法から行きましょう。

- ・ 乱数の生成

「乱数」というのはその名のとおりランダムな数値のことです。

PHP には乱数を生成してくれる関数があります。mt\_rand( )関数がそれです。

```
$random = mt_rand(最小値 , 最大値)
```

これだけで OK です。あとは最大の数値と最小の数値を指定すれば乱数が生成されて変数

\$random が格納されます。ここでは 1 から 5 までの乱数を生成してみましょう。

```
<?php
    $random = mt_rand( 1 , 5 );
?>
```

これでアクセスする度にランダムな 1 ~ 5 の数字が生成されます。あとは条件判定です。1~5 の何れかによって表示させる画像ファイルを変えればいいだけです。

```
<?php
    $random = mt_rand( 1 , 5 );
    if( $random == 1) // $random が 1 だったら
    {
        $image_file = 'image_1.jpg';
    }
    else if( $random == 2)
    {
        $image_file = 'image_2.jpg';
    }
    //以下、5 まで繰り返し
?>
```

あとは対応する画像ファイルを作成し、<img>タグの中に仕込むだけです。

```

```

これは HTML ファイル上の好きな場所に配置できます。バナー表示をこれにしても面白いですね。

さらに、時間によって判別したい場合は 3-2 で作った時間判定のプログラムが使えます。これを挨拶文ではなく、画像ファイルに指定すれば、あとは先ほどと同じ処理で時間によって昼の画像だったり、夜の画像だったり出来ます。夜用の CSS ファイル、昼用の CSS ファイルを読み込ませて雰囲気自体を変えてもいいですし、あとはアイデア次第ですね。

このように、簡単な処理を覚えるだけで、一つのページをいくらでも動きのあるものに変える事ができます。あまりに複雑な処理まで覚えなくても、そこはプログラマーさんに任

せて、出力を変えて、表現の方法に注力出来れば、安易に Flash に頼らなくても面白味のあるサイトが増えてくるのではないのでしょうか。

## 5. 最後に ～PHP フレームワーク 「CodeIgniter」～

「フレームワーク」という言葉をご存知でしょうか？

フレームワークとは、分かりやすく言えば「ある一定の処理系等をまとめたプログラム群を一つのパッケージにまとめたもの」です。まだ分かりにくいでしょうか？

例えば、メールの送信。これには PHP でも「mail 関数」というものがありますが、日本語特有の文字化けや、文字コードの問題などが浮上してきます。

しかしこのフレームワークを使えば、メール送信プログラムが予め組み込まれていて、必要な情報（送信情報や送信形式、送信内容など）を渡せば、あとは自動的にメール送信をしてくれる、といったものです。便利だと思いませんか？

他にも色々と機能は沢山あり、近年の Web 開発ではこのフレームワークの使用が必須といえる程になってきています。

数ある PHP のフレームワークの中でも、私は「CodeIgniter(コードイグナイタと読みます)」をお勧めします。詳細は後日お話をさせて頂きたいですが、かなりデザイナーさん向けに分かりやすいものになっています。今回は名前だけでも覚えていただけると嬉しいです。

「じゃあ始めからコレ使ったほうが早いかも」と思った方もいらっしゃるでしょうが（実際私も使っていたきたいです）、やはりフレームワークと言えど、まずは PHP の基本的な処理を知って頂きたく、このような内容とさせて頂きました。

PHP の基本は今回お話したような条件判定や値の取得の繰り返しで構成されています。

どんなに複雑な処理を行っていても、結局はこれらの繰り返しであり、「分からない」というのは使用している関数が多すぎて、その関数の処理内容が分からないからなのです。

よって、一定の規則とあとは関数を知ってしまえば、それをいかに使うか、というセンスの問題となり、その積み重ねで大規模なシステムは完成されます。

まずは基礎を知っていただき、PHP はこんなものだ、という部分を知って頂けたなら幸いです。また、「面白そうだ」と思って頂けたならば光栄です。

## 6. 補足

- ・ 携帯の\$\_SERVER['HTTP\_USER\_AGENT']について

携帯電話も固定の文字列を持っています。よって、それらを判別することにより、キャリア別の処理を行うことが出来ます。最近の携帯電話、及び各種機種に含まれる文字列を以下に示します。

各種機器の文字列

ドコモ	「DoCoMo」の文字列が含まれます。
au※	「KDDI」の文字列が含まれます。
ソフトバンク	「SoftBank」の文字列が含まれます。
ボーダフォン※	「Vodafone」の文字列が含まれます。
PSP	「PlayStation Portable」の文字列が含まれます。
PlayStation 2	「PlayStation BB」の文字列が含まれます。
PlayStation 3	「PLAYSTATION 3」の文字列が含まれます。
Wii	「Nintendo Wii」の文字列が含まれます。

※ やや古いモデルの場合、機種名ではなく、「UP Browser」という文字の場合があります。ご注意ください。

やはり機種名がそのまま出てくる場合が多いです。ゲーム機器用にCSSなんかを入れる場合には良いですね。

- ・ お問い合わせフォームについて

HP上で「お問い合わせフォームの作成」と書きましたが、内容的にハードになる可能性があった為、順を追って作成していきたいと思えます。期待して頂いた方は申し訳御座いません。

次回以降、お問い合わせフォームの作成に際して、とても重要なセキュリティの問題を時間を掛けてお話ししたいと思います。あと数回の勉強会で徐々にメール送信アプリケーションを完成させていく、というプロセスで行きたいと考えております。

また、サンプルとして配布ファイルにメール送信プログラムを入れてあります。実用には耐えません（実際にメールが送信できるサーバーでないと動作しません）が、今回学んでいただいた内容と照らし合わせて、どのような処理を行っているかを追っていただくのも良いかと思えます（実際if文ばかりです）。

また、本勉強会では勉強内容についてのリクエストを受けております。  
PHP ではなく、JavaScript や ActionScript を学んでみたい、という方がおられましたら  
お申し付けください。また、逆にデザイナーさんからの発表も歓迎しています。  
デザインのポイントや面白いサイト作りの意見交換の場となれば幸いです。逆に教えて頂  
きたい位です。

本日は、「デザイナーのためのプログラミング入門」にご参加くださり、誠にありがとうござ  
いました。また次回の開催の際にもご参加くださると嬉しいです。