

CoderDojo Kashiwa PHP教材

※HTMLをやってからトライしてください！

目標：PHPの基本文法を理解する

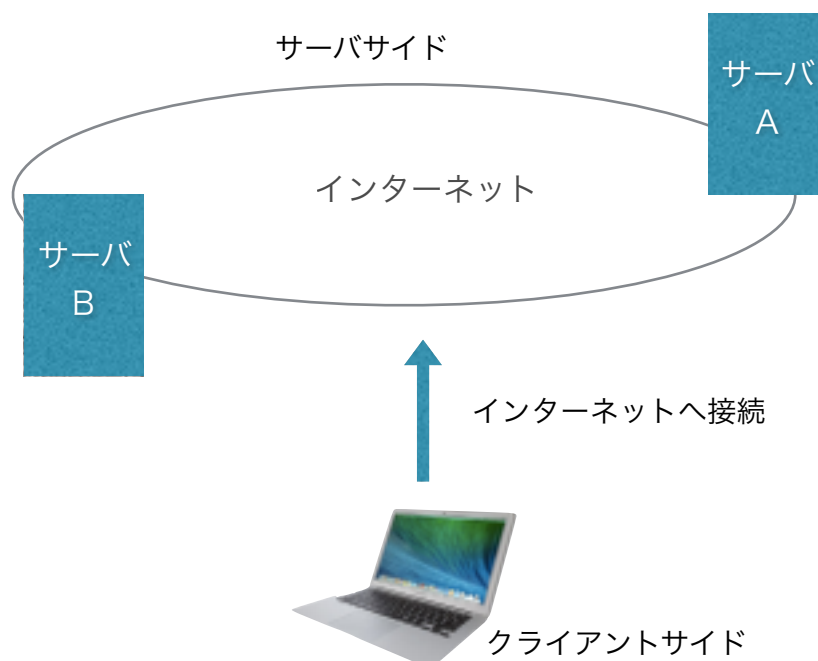
【1】PHPとは？

プログラミング言語には、クライアントスクリプトとサーバサイドスクリプトの2種類がある。JavaScriptはクライアントスクリプト、これから学ぶPHPはサーバサイドスクリプトに分類される。

サーバサイドスクリプト	クライアントスクリプト
PHP, Perl	JavaScript

【2】インターネットの構造

サーバサイドとクライアントを知る上で、インターネットの構造を知る必要がある。クライアントとは、コンピュータやスマートフォンなどユーザが実際に使う物のことを言う。サーバサイドとは、インターネット上に存在しているサーバのことを指す。



Webサイトをインターネット上で公開するには、htmlファイルをサーバ上に設置する必要がある。サイトに設置したフォームを例に考えてみよう。フォームに値を入力するのがサーバサイド、送信されたデータを処理するのがサーバサイドとなる。

ここまで理解したところで、次はPHPを動作させるための環境を構築していこう。

【3】ローカル開発環境の構築

このパートは自分のPCのOSによってやるが変わるので注意！

①Windowsの場合

<http://techacademy.jp/magazine/1722>

このサイトを参考に、XAMPPをインストールする。

インストールが成功したら、htdocsフォルダ内に"myphp"というフォルダを作っておこう。

②Mac OSXの場合

<http://pc-karuma.net/mamp-install-mac/>

こちらのサイトを参考に、MAMPPをインストールする。

インストールが成功したら、htdocsフォルダ内に"myphp"というフォルダを作っておこう。

ここから先はローカル開発環境がセットアップされていることを前提に話をすすめる。

テキストエディタを開いて次の章から進めていこう。

【4】PHPの解説

①：Hello,World!と表示させる

```
<?php
    echo "Hello, World";
?>
```

②：コメントアウト

ソースコード内のある一行を無効化したいときはコメントアウトをする。

```
<?php
    //echo "Hello, World";
?>
```

③：複数行のコメントアウト

```
<?php
    /*
    echo "ここからさきの";
    echo "ソースコードを";
    echo "すべてコメントアウト！";
    */
?>
```

④：Pタグの中にPHPを使って文字を表示させる

PHPはHTMLの中に組み合わせて書くことができるので、このような書き方もできる。

```
<p><?php echo("こんにちは！"); ?></p>
```

⑤：変数の設定

PHPはプログラミング言語なので、変数を使うことができる。変数名は基本的にすべて小文字で書くのがルール。「\$」マークを先頭につけることで変数の宣言ができる。

```
<?php
$message = “こんにちは”;
echo($message); //==>こんにちは
?>
```

また、変数は文字通り「変えられる数」なので、後から格納されている値を変えることもできる。

```
<?php
$message = “こんにちは”;
echo($message); //==>こんにちは

$message = “さようなら”;
echo($message); //==>さようなら

?>
```

⑥：変数の型

コンピュータは1と”1”を区別する。例えば、整数の1は「整数値」、文字の”1”は「文字列」と呼ばれる。

型名（日本語）	型名（英語）	例
文字列	string	“おはようございます”
整数	int	1 3 100
小数	float	3.14
真偽値	bool	true（正しい） false（正しくない）
null	null	null

nullとは、「なにもない」を意味する。

⑦：定数

変数と違って、一度宣言すると後から変えることの出来ないもの。ユーザ名など大事な値を保管するときに使用する。また、定数名は大文字で書くのがルール。

```
<?php
define(‘USER_NAME’, ‘KIRIE MIYAJIMA’);
echo(USER_ID); //==>KIRIE MIYAJIMA
?>
```

⑧：四則計算と文字列の連結

コンピュータは電子計算機であるから、当然計算することができる。

```
<?php
$a = 7;
$b = 49;
$c = 51;

//a+b
echo($a + $b); //==>56

//c-b
echo($c - $b); //==>2

//a×c
echo($a * $c); //==>357

//b÷a
echo($b / $a); //==>7

//c÷aのあまり
echo($c % $a); //==>2
```

また、文字列同士を連結することもできる

```
<?php
$e = "CoderDojo";
$f = "Kashiwa";
echo($e . $f); //==>CoderDojoKashiwa
```

⑨：文字列の中に変数の値を組み込む。

文字列の連結を使うことで以下の様なことができる。

```
<?php
$username = "MJK";
echo("こんにちは".$MJK."さん!"); //==> こんにちはMJKさん
```

また、以下のように展開させることもできる。

```
<?php
$username = "MJK";
echo("こんにちは{$MJK}さん!"); //==> こんにちはMJKさん
```

中括弧 { } を使うとその文字が展開されて表示される。

⑩：比較・条件分岐 if

値を比較する際に使用する符号を紹介する。

```
<?php
$a = 10;
$b = 20;

$a > $b //aはbよりもおおきい → false
$a < $b //aはbよりも小さい → true
$a == $b //aとbは等しい → false
```

また、以上以下を表す符号も存在する。

```
<?php
$a = 12;
$b = 12;

$a < $b //aはbより小さい → false
$a <= $b //bはa以上 → true
```

これらの比較を用いて処理を分けるのが条件分岐 ifである。

ある条件のときはAの処理を、違うならばBの処理をしたいとき、if文を使う。（ifとは英語で「もし」という意味）Scratchの「もし～なら」と同じ。

```
<?php
$a = 10;
$b = 20;

if($a < $b){
    echo("aはbより小さい"); //==> aはbより小さい
}
```

条件が正しかったら（trueならば）以下の処理を実行しなさいというのが基本。

ちなみに、Scratchでいう「もし～なら〇〇、でなければ」という命令もある。

```
<?php
$a = 40;
$b = 20;

if($a < $b){
    echo("aはbより小さい");
}else{
    echo("aはbより小さくない"); //==> aはbより小さくない
}
```

⑪：switch文

ある1つの値によって複数の処理に分岐させたいとき、if文ではなくswitch文を使う。

```
<?php
    $color = "red";

    switch($color){
        case "yellow":
            echo("黄");
            break;
        case "blue":
            echo("青");
            break;
        case "red":
            echo("赤");
            break;
        default:
            echo("その他");
            break;
    }
```

条件に合致したときにだけ処理を実行させることができる。caseの後に書いてあるのが条件。breakはその処理が終わった時にswitch文のループを終わらせるという意味。defaultは変数が条件のどれとも合致しなかった場合のみに実行する命令を定義する。

⑫：while文を使った繰り返し処理

条件がtrueになるまで繰り返し処理を行うのがwhile文だ。Scratchでいう「○○まで繰り返す」がこれに該当する。

```
<?php
    $i = 0;

    while($i <= 10){
        echo($i);
        $i++;
    }
```

この場合では、\$iという変数が10以上になるまで\$iを出力し続けている。
また、\$i++は \$i = \$i + 1と同じことで、これをインクリメントという。

⑬：for文を使った繰り返し処理

while文とfor文はどちらも繰り返し処理を行うことができるが、その用途は若干異なる。
Scratchでいう「○回くりかえす」がfor文にあたる。

```
<?php
    for($i=0; $i<=10; $i++){
        echo($i);
    }
```

⑭：配列

配列とはScratchでいうリストのことである。変数は1つの変数に対して1つの値しか保存できなかったが、配列は複数の値を保存することができる。

```
<?php
$subject = array(“国語”, “算数”, “理科”, “社会”, “英語”);
```

配列の宣言は変数と似ており、配列名の先頭に「\$」を付ける。arrayとは配列という意味で、array()の中に値を入れる。（今回は文字列なので”をつけている）
1つの配列に複数の型の値はいれることができないので、以下のコードはエラーとなる。

```
<?php
//これはエラー
$number = array(1, 10, “100”);
```

配列の要素は先頭から0, 1, 2, ...と増えていく。例えば、\$subject配列の”国語”を取り出すときは、\$array[0];、理科は\$array[2];となる。

⑮：foreach

配列の中身を1つずつ取り出して表示したり、なにか処理をしたいと思った時に使うのがforeachという繰り返し処理だ。

```
<?php
$subject = array(“国語”, “算数”, “理科”, “社会”, “英語”);
foreach($subject as $key => $value){
    echo($value);
}

//==> 国語 算数 理科 社会 英語
```

イメージとしては、\$subjectの要素を1つずつ\$valueに入れて、処理が終わったら次の要素にするといった感じ。

⑩⑪：関数

プログラムが複雑になってきたり、ソースコードが長くなってくると、同じ処理を様々な場所で実行したくなることがある。このとき、関数という機能を作ることによって何度も同じことを書くことなく処理を実行することができる。

関数は英語でfunctionという。以下がもっとも簡単な関数の作りかただ。

```
<?php
//関数の定義
function funcName(){
    //処理

    echo("Hello, World");
}

//関数の実行
funcName();
```

funcName は関数の名前で、自分で決めてよい。できるだけ、その関数がなんのプログラムなのかがわかるような名前をつけよう。

{ } で囲まれた部分に処理を書く。

関数を実行するときは、関数名 () とする。

⑩⑫：関数の引数

関数の処理の中に、値によって処理を変えたりする場合、引数（ひきすう）という機能を使おう。

例えば、名前を入れるとその人に対して挨拶をしてくれるプログラムを作ってみよう。

```
<?php
//関数の定義
function sayYourName($name){
    echo("こんにちは、{$name}さん");
}

sayYourName("みやじま");    //==>みやじま
```

このように()の中に引数をセットすると、引数をプログラム内で使用することができる。