

初心者のための PHP オンライン教材に ついて

兼田 真弓

目 次

1	テーマと計画	1
1	1 何を作るのか	1
2	2 PHP とは	2
3	3 対象とする人	2
4	4 従来の教材	3
2	説明と作業	5
1	1 構想や規模、技術的な問題	5
2	2 背景となるコンピューター技術	6
3	3 具体的な資料と使用するソフト	6
4	4 作っていく作業の見通し・計画	7
3	内容	8
1	1 制作物について	8
2	2 制作過程での問題点とその解決策	10
3	3 利用者を考えて工夫した点	11
4	まとめ	17
1	1 評価	17
2	2 自己評価	18
3	3 今後の課題	19

1 テーマと計画

(1) 何を作るのか

福田ゼミのテーマは、「人に役立つものを作る」である。私がこの4年間学んできた上で「こんなものがあったら良かったな」と思ったものを、自分が実際に作ってみようと考えた。

最近注目されている Web プログラミング言語に PHP がある。私は全くのプログラム初心者で、PHP の授業では難しい用語や文字コードの意味がわからず、PHP そのものに苦手意識を持ってしまふことが多かった。

プログラミングは1つ1つの積み重ねであり、途中で1つでも分からなくなってしまうと、その後も分からないまま進んでいってしまうため、初めは学ぼうといった姿勢で取り組んでいても、プログラム初心者にとっては苦手意識も手伝い、だんだんとその姿勢は萎んでいってしまう。その学習意欲を継続させることができたなら、PHP プログラムをもっと有効に活用していくことができるのではと考えた。

また PHP の基本を学んでも、どのような場合にこういった活用ができるのかもわからずにいた。PHP を使うことによって、アクセスカウンタや掲示板、その他にも高機能なサイトまで幅広く作成することが可能となる。PHP は他のプログラミングに比べて言語も簡単である。つまり基本を学ぶことができたなら、誰でも簡単に幅広い Web を作成することができるようになるのである。

PHP を実際に動かしながら、PHP の基礎や大事なポイントを覚えていくためのさまざまなプログラムを、身近である Web 上で学ぶことができる教材を作成し、初心者でも取り掛かりやすく、そして理解しや

すい教材にしたいと思ったことが、この教材を制作しようと思ったきっかけである。

そしてこの PHP 教材を作ることで、プログラミングを途中で躓いてしまった人や、これから初めてプログラミングを学ぶ人、そして他のプログラミングを学んでいたけれど、PHP を初めて学ぼうとする人が、PHP プログラミングを理解するきっかけになればと考えた。

(2) PHP とは

PHP は、Web サーバに置かれる Apache モジュールとして動作するスクリプト言語を実行する環境である⁽¹⁾。HTML 文書の中に記述されたプログラムを PHP が実行し、結果を HTML として Web 画面に出力できるのである。

PHP は基本的に、HTML ファイルに埋め込む言語であるため、簡単なテキストエディタがあれば、誰でも書くことができる。その他にも以下のような利点がある。

- ・ Web アプリケーション開発に特化した言語である。
- ・ 文法が比較的容易に誰でも習得することができる。
- ・ CGI と比較すると、実行速度が早い。
- ・ どのような種類のデータベースとも接続可能である。

このように PHP は高機能な作成が可能にも関わらず、まだまだ馴染みの浅い言語なのである。

(3) 対象とする人

これからプログラミングを学ぼうとしているプログラミング初心者、そして今までプログラミングを使っていた人や挫折してしまった人に対

して、PHP 初心者としてテーマに沿ったプログラムの説明をし、PHP プログラミングの基本的な使い方を簡単で分かりやすい教材として役立つようにしたい。また、Perl や Python との違いを知るためにも、この教材で PHP プログラミングの楽しさや、奥深さを知ってもらいたい。そしてこの PHP プログラミングを理解することで、他のプログラミングを理解するきっかけにもなればと考えている。

(4) 従来 of 教材

PHP は他のプログラミング言語に比べて習得しやすいが、他の言語に比べてプログラミング初心者を対象とした教材が不足しているため、まだまだ習得しにくい言語である。またアクセスカウンタや掲示板など、Web サイトで動作を行う CGI は、Perl というプログラミング言語で書かれることが多いが、PHP は Perl に比べて優れている点が多くある。ここでは、PHP と Perl を比較しながら、もう少し具体的に PHP の特徴をつかみたいと考えている。

プログラムの中で最もよく使われるプログラムの例は、画面に「Hello,World!」という文字を表示させるプログラムである。どちらも同じく簡単なプログラムであるが、Perl の場合は、プログラム部分だけでなく、HTML 全体を Perl で書く必要がある。それに対して PHP の場合は、HTML ファイル内の必要な部分だけ PHP で記述するだけで済むのである。

もちろん、HTML ごと全て PHP で記述しても良いが、HTML と組み合わせることで、複雑なプログラムも非常に見やすく簡単に作成することが可能なのである。

Perl の場合

```
#!/usr/local/bin/perl
print "content-type: text/html\n\n" ;
print "<HTML>\n" ;
print "<HEAD>" ;
print "<TITLE>Hello,World!</TITLE>\n" ;
print "</HEAD>" ;
print "<BODY>\n" ;
print "Hello,World!\n" ;
print "</BODY>\n" ;
print "</HTML>\n" ;
```

PHP の場合

```
<HTML>
<HEAD>
<TITLE>Hello,World!</TITLE>
</HEAD>
<BODY>
<?php
print "Hello,World!" ;
?>
</BODY>
</HTML>
```

2 説明と作業

(1) 構想や規模、技術的な問題

この教材を作る際の目的は、初心者にとっていかにわかりやすく解説できるかである。そのために PHP を使う上での基本と、その基本を用いたプログラムの例を表示し、その説明とともに自分でも実際に PHP プログラムが作れるような構想にしたいと考えた。

PHP は、数あるプログラミング言語の 1 つである。PHP はほかのプログラミング言語に比べてそれほど難しくはなく、プログラミングをこれから始める人にとっても、学びやすい言語であるといえる。また、簡単なプログラミング言語では高度なプログラムを作り難いと思われがちであるが、PHP でもかなり高度なプログラムを作成することが可能である。一般的にプログラムには、処理（設計）のパターンがある。

なぜその関数を使うのか、そしてそれはどのような働きをし、結果どうなるのかを簡単で難しい用語などはあまり使わずに説明するようにした。また知っておきたい用語に対しては、別途の説明を付け加えてある。私自身、プログラミングに取り組む際、分からない用語に戸惑ったことがあった。そしてその度にプログラミングに対して「難しい」という印象が付いてしまったためである。まずは、教材として学びに掛かりやすい内容にすることを心掛けた。

以上のことから、PHP プログラミングの十分な知識はもちろん、その以外にも Perl や Python などの他のプログラミングについても理解する必要があった。そしてプログラムの例は、初めは基本の簡単なもの、それから段階を踏んで難しいプログラム例を作成していった。

(2) 背景となるコンピューター技術

コンピューターの技術としては、まずはやはり PHP プログラミングの根本的理解である。自分自身が PHP プログラミングを学んでいくことでどういった所が難しかったか、また理解しにくかったのかを知ることができるのである。そして Perl や Python など、他によく使われているプログラムの知識も必要である。それは、他のプログラムに比べどういった違いがあり、PHP のメリット・デメリットを明確にするためである。

PHP はメリットの多い言語であるが、逆に PHP ならではの問題もある。それは、バージョンアップが早いということである。バージョン間で大きな変更があると、前バージョンの PHP スクリプトが実行できない可能性がある。しかし、PHP のバージョンアップを待つことよりも、今の状態から始めておくことが、新しいバージョンが出てから勉強するより効率がいいと考えた。

(3) 具体的な資料と使用するソフト

制作にあたり『PHP で Web アプリケーションをつくろう』という資料を主に参考にした。PHP の入門書で、基礎解説が書かれている内容であった。また付録の CD-ROM に実際のスクリプトを収録しているので、うまく動作しない場合はサンプルと比較することで、原因が容易に特定できるようになっている。

使用したソフトに関しては、大谷版 knoppix を使い、apache という Web サーバーの中に組み込まれている PHP を起動した。

(4) 作っていく作業の見通し・計画

まずはこの制作にかかるために PHP の知識が必要であった。授業で学んだことを含め、その時に理解できなかったことなどをもう一度勉強し直した。PHP プログラミングの基本から、応用までを資料を使い理解していった。その他にも Python や Perl など、PHP 以外のよく使われるプログラミングの基本も勉強することにした。それは、PHP と他のプログラミングを比較することで、他に比べてどこが違い、どのような点が使いやすいかを説明するためである。この教材は、プログラム初心者はもちろん、今まで他のプログラムを使っていた人がこれから PHP を学んでもらえることも目的としているからである。

3 内容

(1) 制作物について

まずは、解説するプログラムに表題を付けて目次のページを作成した（図 1 参照）。そのページで解説しているプログラムが一覧になっており、項目ごとにその解説ページへとアクセスできるよう設定した。項目の順番は上から順に、だんだんと高度なプログラム内容になるようにした。解説ページは全て解答例と解説を同じページに表示しており、解答例を見ながら解説を読めるように作成した。そしてページの最後には、トップページへと戻れるような選択ボタンを設定してある。

はじめに、文字列を表示するプログラムの解説のページである（図 2 参照）。解説の前に、ここでは「hello！」という文字を表示することを目的とし、解答例を表示している。これは簡単な PHP プログラムの初歩で、基本の形を覚えることのできるようにしたためである。echo 文で文字列を表示する命令をできることや、`<? PHP ~ ?>` タグが PHP スクリプトであることを示していることなど、解答例を見ながら解説を読むことができるので、基礎の知識を得ることができる。

次に現在の日時の表示方法である（図 3 参照）。文字列として表示させるだけでなく、日にちや時間を、現在と同じように合わせて表示させる方法で、`$ datetime` という関数を使うことで、関数についての解説もすることができるようこのプログラムを選択した。また、この関数を使う際に少しではあるが応用ができるよう、解答例で書いたもの以外のタイムスタンプを表にして表示した。

この教材で、初心者に分かりやすく解説することが一番難しかったのが、環境変数の解説である（図 4 参照）。環境変数はプログラム内で参

照するデータベースで、その内容はサーバ自身の情報やアクセスしたクライアントの情報など様々である。そのデータベースを参照する際に、関数を使う。前の現在日時の表示の解説で関数の説明をしたのは、この環境変数で使われる関数の意味を少しでも理解しやすくするためであった。この解説の中で、再び HTML 文として表示させる内容が出てきている。そのときには最初の「文字列を表示する」のページに飛べるように工夫をした。

また、if ~ else 文を使い、条件分岐についても少し触れている。しかし条件分岐は、環境変数のところで一緒に解説してしまうと、内容が分かりにくくなってしまったため、別の解説ページを作った (Jyouken.html 参照)。

関数や条件分岐の解説をしたところで、次は繰り返し (ループ) 処理の説明をしている (図 5 参照)。繰り返し (ループ) 処理は、プログラムを作り、使い続けるために必要な知識である。決まった処理を効率良く使うことを学べるように解説した。そのなかで for 構文にも触れている。これもまた、繰り返し (ループ) 処理というプログラムの中で for 構文を使うことで、どのようなときに使うのかを知ってもらえるようにした。それとは別に for 構文に関して詳しい解説が必要であるため、別ページに解説を作成することにした (図 6 参照)。ここでは for 構文を使う利点は何かを説明し、より高機能なプログラムを作成できるようにするためである。

最後は演算子の初歩である代数演算子の解説にした (図 7 参照)。この教材はあくまでも初心者向けの教材である。そのため、ここで代数演算子という初歩ではあるが、演算子を学んでおくことで、今後さらに PHP の知識を得ようとしたときに役立つのではないかと考えたからで

ある。解答例では簡単な足し算の表示を書いており、演算子を変えることで、引き算など他の簡単な演算ができるよう、代数演算子の表も解説と一緒に記載した。

(2) 制作過程での問題点とその解決策

制作者である私自身、PHP の知識が浅く初心者に近い状態であった。これは既存の教材などで勉強するしかなかった。授業で配布されたプリントや、実際に自分が読んでみて理解しやすかったものを参考にし、基本から応用までの知識を得ていった。

そして制作に取り掛かるにあたり、教材の中で PHP プログラミングを学ぶ上で必要な知識であるプログラムの解説をしなければならなかった。PHP の基本知識に対してどこまでの説明をし、どの部分を省くかという問題があった。PHP の基本知識は莫大な量があるので、それをどこまでが知っておかなければならない知識とすることを見極めることが困難だった。そしてまた、初心者向けにどの問題を解説するか、私 1 人で決めるには不安があった。教材をつくる際に気を付けたことは、「いかに簡潔でわかりやすく、学んでいけるようにするか」ということであつた。

そこで、私が今までに何度か作成したプログラムの中で、使う機会が多かったり、いくつかの既存の教材ほとんどに掲載されていたプログラムを解説していくことにした。使う機会が多かったり、既存の教材に掲載されているということは、それだけ一般的に使用頻度が高く、またプログラムで Web をつくる際に役立つ内容になっているからだと考えた。

また何故プログラミング自体にこんなにも苦手意識を持ってしまうのか理由を考えた結果、プログラムを組み立てる際に、一番苦労する点

は、プログラムの構成を考えることである。そしてプログラミング自体に慣れていないという問題があった。

そのためこの2つの問題を重視して教材を作成していけば、初心者にとって理解しやすい教材ができるのではないかと考えた。

初心者に近い状態で勉強をし直したことで、今から PHP プログラミングを始める人がどういったところに疑問や難しさを感じるかが分かった。そのため制作していくにあたり、その疑問点をカバーし作っていくことができたと感じている。

(3) 利用者を考えて工夫した点

初心者を対象としているため、基本的な説明をなるべく詳しく書くことを心掛けた。プログラムの解説をするときには、同じページに解答例を表示し、そうすることで利用者が自分で実際に同じように入力し、出力できるようにした。これは私が実際に感じたことだが、解説だけ見てもプログラムと見比べないと、どの部分を解説しているのかわからずに、理解しにくいのではないかと考えたからである。説明を読むだけでは理解しにくいことも、入力してもらい使い方を理解してもらえば練習になると考えた。そして、何よりも説明が長すぎると、読む気もなくなるので出来るだけ簡潔に長くなりすぎないように気を付けた。

どんな内容のプログラムを基本として学ぶ必要があるか、そして初心者に分かりやすい解説を考えた。この関数を使うことで、どんな働きと表示結果が出るのかを示し、解説では関数の説明から、プログラム用語に関しては別途の説明を付け足した。文字列や日時の表示、環境変数、繰り返し処理（ループ）と、プログラムを作っていく上で、必要であるプログラムを問題にした。

(i) 文字列の表示

まずは Web アプリケーションをつくる際に、1 番初めに必要な知識である文字列を表示するプログラムの説明から始めた。

`<?php ~ ?>` のタグ内が PHP スクリプトであることを示しており、その中で `echo` という関数を使うことによって文字列が表示される。つまり `echo` は文字を表示するという命令である。`echo` のような命令のことを関数という。関数とは命令の内容をまとめて指示できる便利な言葉である。そして文章の終わりに「。」をつけるように、PHP では文の終わりに必ず「;」をつけなければならない。; をつけないと文の終わりともみなされないため、途中で改行が入っても、表示では改行されないのがある。

```
<?php
echo "hell
    o!";
echo "こん
に
    ちは!";
?>
```

上記のように記述しても問題はないが、分かりにくいいため一般的には一行に一文を記述する。これは、他のあらゆるプログラムを作成する際に必要なもので、最も基本となる部分である。

次からはさまざまな違った条件を基に違った結果を実行する条件分岐について説明していく。

(ii) 日時の表示

ここでは、時刻により違ったメッセージを表示させることを通じて、さまざまな条件により違った結果の実行を学ぶ。日時の表示には \$ datetime を使用し、その際にタイムスタンプを利用することで異なった表示方法が可能である。

```
<?php
    $ datetime = date("Y年m月d日 H:i:s");
    echo ("現在の日時は $datetime");
?>
```

これを例えば nitiji.php というファイル名で保存する。この時拡張子は必ず php とする。これを実行すると、ブラウザには「現在の日時は 2007年01月01日 24:00:00」といった形で現在の日時が表示される。

echo は出力の命令、date () は日付や時刻を文字列として表示する関数である。date () に与える文字列により次の値が表示される。

(iii) 環境変数の取得と表示

環境変数はプログラムなどが参照するデータベースのことで、その内容はサーバ自信の情報や、アクセスしたクライアントの情報など様々である。これを使うと、任意に環境変数を参照して、各種の情報を取得したり、動作を変更したりすることができる。また、自分の環境において、環境変数を新規に作成したり、既存の環境変数の値を変更したり、削除したりすることができる。

つまり、環境によってさまざまな条件を基に違った結果を実行する必

タイムスタンプ

a	”am”または”pm”を表す。
A	”AM”または”PM”を表す。
y	70 や 99 のような 2 桁の数字による”年”を表す。
Y	1980 や 2001 のような 4 桁の数字による”年”を表す。
d	05 や 22 なのような 2 桁の数字による”日”を表す。
D	Sun や Fri のような 3 文字のテキスト形式による”曜日”を表す。
l	Sunday や Friday のような長いテキスト形式の”曜日”を表す。
F	January や June のような長いテキスト形式の”月”を表す。
m	04 や 10 のような 2 桁の数字による、”月”を表す。
M	Jan や Oct のような 3 文字のテキスト形式による”月”を表す。
h	05 や 11 のような 2 桁の数字による 12 時間単位の”時”を表す。
H	09 や 20 のような 2 桁の数字による 24 時間単位の”時”を表す。
i	05 や 35 のような 2 桁の数字による”分”を表す。
s	09 や 52 のような 2 桁の数字による”秒”を表す。
U	Unix 基準時 (1970 年 1 月 1 日 00:00:00GMT) からの秒数を表す。

要があるため、条件分岐をしなければならない。条件分岐では if ~ else 構文を使う。

これは、時間によって朝か夜、どちらかの表示を実行する。if は「もしも……」、else は「そうでなければ……」という意味である。

ある式の結果が正しい場合は TRUE (真)、正しくない場合は FALSE (偽) とする。これはユーザーが使用する際に可能性のある環境を、作成した側があらかじめ設定しておくもので、プログラムをうまく起動さ


```
<?php
    if ($h < 10)
        echo "朝";
    else
        echo "夜";
?>
```

せるためにも必要な知識である。

(iv) 繰り返し (ループ) 処理

全く同じ処理、あるいは似たような処理をある一定回数繰り返すことが、プログラミングではよく使われる。決まった処理を効率良く繰り返すループ処理を学べるように解説した。そして for 構文を使うことにより、配列を便利に使用できるようになる。

for 構文における処理の流れは以下ようになる。

1. 初期値が設定される。
2. 条件式の判定が行われ、条件式が満たされる (TRUE) なら実行文が実行され、満たされなければ (FALSE) ループを抜ける。
3. 条件式が満たされ、実行文が実行された後に、増減式が実行される。
4. 条件式が満たされる間、2 と 3 が繰り返される。

(v) 代数演算子

プログラム内で、足し算や引き算など、数値と数値を計算する際には代数演算子を使う。PHP でも、他のプログラム言語と同様に、簡単な計算から複雑な科学計算まですることが可能である。そうした計算の基となるのが代数演算子である。

演算子と一言でいっても、いくつかの種類がある。PHP では簡単な

足し算や引き算などのような加減乗除の演算子、あるいは比較演算子、論理演算子など、利用できる演算子が多くある。この教材では加減乗除の、最も基本的で簡単な演算子の解説をすることにした。特に足し算は数値だけでなく、文字列を結合することにも利用されているからである。

例	名前	意味
$\$ a + \$ b$	加算	$\$ a$ と $\$ b$ の和
$\$ a - \$ b$	減算	$\$ a$ と $\$ b$ の差
$\$ a * \$ b$	乗算	$\$ a$ と $\$ b$ の積
$\$ a / \$ b$	除算	$\$ a$ と $\$ b$ の商
$\$ a \% \$ b$	剰余	$\$ a$ を $\$ b$ で割った余り

4 まとめ

(1) 評価

制作途中で、実際に教材を使ってもらい、同学科の第3学年ゼミ生にアンケートをとった。そして、完全なプログラミング初心者の意見も聞くために、人文情報学生以外の友人にも協力してもらった。アンケートでは良かった点・悪かった点を自由に記述してもらった。

良かった点では、ゼミ生からは実際にプログラムの解答例が書かれていたことと、その解答例と一緒に解説を見れるのが良かったということ、問題別の目次があって使いやすかったことが挙げられていた。初心者である友人からは、難しい用語がほとんど使われていないので、解説がわかりやすかったことや、最初は簡単なところから始まるので、取り掛かりやすく応用も理解できた、といった意見があった。

この結果から、私が見やすいように作成した解答例と解説を同じページに表示していたことが、同じように見やすいと感じる人が多くいたのでこの表示方法を変えずに進めていった。他にも、不安に思っていた初心者の人にとって分かりやすい内容になっているかという点で、アンケートしたところ分かりやすいといった貴重な意見を得ることができた。これは、私が初心者向けに教材を作っていく上で大変参考になった。

悪かった点では、ゼミ生からはコンパイルしたときにエラー表示がされてしまうプログラム例があったという意見が挙がった。また、見やすいと思い解答例と解説を同じページで表示していたが、それが反対に見にくいので、違うページにした方が良いのではないか、という意見もあった。友人からは、もう少しプログラムそのものを詳しく説明してほ

しいという意見も挙がった。今まで全くプログラムに触れたことのない人にとっては、プログラム自体がどんな働きをするものなのかが、わからない人もいるのである。

初めてプログラミングを学ぶ人にとっても、自分自信が分かりにくいと感じていたことと同じで、慣れない用語の意味を難しく思う人が多かったことを改めて感じた。また表示画面については、私が見やすいと思い作成した、解答例と解説を同じ画面で表示することに対して2つの意見があった。同じ画面で表示することが解説を見ながら復習を兼ねて学べるので見やすいという人と、反対に見にくく、解答例を先にみてしまうため違うページに表示してほしい、という意見も挙がった。良い意見ばかりではなく、悪い点の意見を聞いたことで、教材をさらに良くするために役立ったと感じている。

(2) 自己評価

当初の目的は初心者でもわかりやすく、簡単にプログラミングを学ぶことのできるような教材を制作することが目的であった。制作後も、もちろんその目的は変わっていなかったが、初心者に対して分かりにくい内容となってしまっているところがあった。アンケートの結果から、私が初心者の立場になってみて考えたことが、全ての初心者に当てはまることでは無いのだと気付くきっかけとなった。分かりやすい・分かりにくいといった判断は人によって様々であり、全ての人に共通して使いやすいものを作るのは難しい。それでも、私と同じような考えでプログラミングに関わっている人は少なくないと感じた。そのためこの制作は制作者である自分自身が初心者という視点で見たときに、どうすれば分かりやすく理解できるかを考え制作していった。

またこの教材ではエラー表示プログラムの説明していない。それはこの教材を使った人にまずは基本を学んでもらい、応用としてエラー表示を学ぶ必要があると考えていたからである。しかしこの教材で学んだ後に、何か1つのプログラムを作ろうとした場合、やはりエラー表示の仕方が分からないのではより高機能なプログラムが作れない。基本をしっかり学べるようにエラー表示は説明しなかったが、自分自身でプログラムを作り上げた、という達成感を感じてもらうためにはエラー表示の説明をするべきであった。その達成感を感じることで、プログラミングの楽しさやおもしろさをわかることができるはずなのである。

(3) 今後の課題

この教材の中で制作したプログラムは、PHP 3.0.9 と PHP 4.0.6 のバージョンで動作確認を行っている。特定の環境に依存するような処理はしていないので PHP 3.0.9 以上の環境であれば動作する。しかし PHP はたくさんの PHP 利用者の協力のもと、たびたびバージョンアップが行われる。バージョンが上がることにより、PHP がより便利で快適に使用することが可能となるが、少しのバージョンの違いであってもプログラムが動作しなくなることがあるので注意しなければならない。

また、エラー表示やアップロード処理などの説明を増やすことで、さらに幅の広いプログラムの理解につながるはずである。プログラムの幅広い理解は応用につながり、さらに楽しんでプログラムを作成していくことができるようになる。

そして何よりも、この教材を作るきっかけとなった「初心者に分かりやすい教材」をつくることである。その目的が果たせたのかというと、

実際はまだ課題が残っている。この教材で PHP プログラミングの基本が身に付き、プログラミングに対する苦手意識が減り、楽しくプログラムを作ることができるような解説はできたであろう。しかし前にも述べたように、この教材が本当にプログラム初心者にとって理解してもらえるのかといった点には疑問が残る。解説の中では説明しているとはいえ、ある程度のプログラミング用語を使っている部分もあり、教材を使う人によっては説明不足なのではないかと感じたからである。プログラムの解答例に関しては、やはり完全では無いところもあるため、うまくコンパイルが通らないこともあった。

しかしこの教材では、1つのプログラムを理解することができたら、それが基となりまた新しいプログラムを理解することができるようになる。それはプログラムの組み立て方の関連性を理解することにつながるのである。そして、プログラムをさらに学ぼうといった姿勢は、このように1つ1つを理解することでの楽しさからうまれるのであり、この教材はその目的を果たしているのではないだろうか。

今回、この教材を制作した上で、教材をつくることがどれだけ重要であるかがよく分かった。ただ単に言葉を連ねていくだけでは何も伝わらず、図を入れたりどこを強調して説明するかによって内容は大きく変わってくる。その内容によりどれだけ人の役に立つものができるのかが分かったように感じた。

PHP プログラミングはたくさん的高機能を利用することができるにも関わらず、まだまだ習得しにくい言語である。そしてまた、初心者の人にとってはプログラミング言語というだけで困難なイメージが付いてしまうものである。この教材をきっかけとして、困難だというイメージが減り、PHP プログラミングがもっと多くの人に習得され、普及さ

——初心者のための PHP オンライン教材について——

れてほしいと考えている。しかし、そのイメージはこの制作では和らげる程度で、無くならせることはできていない。この困難であるというイメージを無くすためのさらなる工夫や改良をしたときに、私が目的としていた「プログラミング初心者にとって役立つもの」ができるのではないだろうか。

——初心者のための PHP オンライン教材について——

注

- (1) Apache は、世界中で最も普及しているオープン・ソフトであるため、ほとんどの Web サーバに実装されている。

文献表

イクスブレイン著

2002 『PHP で Web アプリケーションをつくろう』MYCOM

アラン・ゴールド著 松葉素子訳

2001 『Python で学ぶプログラム作法』ピアソン・エデュケーション