Python のオンライン復習問題集の作成について

奥村　雄太郎

目　次

——Python のオンライン復習問題集の作成について——

1　はじめに

(1) 誰のために何を作るのか

我々のゼミで最初に学ぶプログラミング言語は、Python である。

私はゼミに入るまでプログラミングに触れたことが無かったため、初めて学ぶプログラミング言語は Python であった。

前の項目で説明した通り Python は、初心者に適した言語であるが、初心者の私には、学び始めてから感じた苦手意識で、積極的に学ぶ意欲が低下し、習得は簡単では無く授業後の復習は、必須であった。

その経験から私は、ゼミでの学習内容に沿った問題集があればと思い、この卒業制作のテーマに辿り着いた。この制作物は、ゼミに入って初めてプログラミングを学んだ学生、苦手意識を持った学生等の、私と同じような境遇の学生を中心に役立てて貰いたいとの思いで、臨んでいる。

(2) Python とは

Python とは開発者の Guido van Rossum 氏が、参加していた教育用プログラミング言語「ABC」を実用目的向けに、１９９０年頃から開発したプログラミング言語で、Guido がファンである英国 BBC 放送のコメディ番組「モンティパイソン」から「Python」と名付けられた、フリーなオブジェクト指向プログラミング言語である。主な特徴としては、

1. オープンソース (商用利用も含め、無償で利用・再配布可)
2. マルチプラットフォーム (Unix, Windows, Macintosh, etc...)
3. モジュール機構
4. リストや辞書、複素数など、豊富な組み込みデータ型

等がある。

このような教育用プログラミングという背景から生まれた Python には、「シンプル」・「習得が容易」という目標を重点に置いた言語設計が成されています。そのため、初心者が学ぶプログラミングとして適しているといえます。

(3) それを作ることがどのように役に立つのか

私自身が、プログラミングに対して、苦手意識を持っており、授業でも多くの単元・問題に躓いたことから、苦手意識を持つ学生と同じ目線で、問題作りに取り組んでいけると考える。

その結果、分かりやすい問題内容・構成になり、理解し易い復習問題集を作製する事が出来ると考えている。この復習問題集を学習する事により、プログラミングの基礎を理解し、プログラミングへの苦手意識を克服すること。さらには、プログラミングの基礎を理解することで、苦手意識を克服し、今後授業で学ぶであろう PHP、javascript や自主的に学ぶ新たなプログラミング言語学習へのモチベーションアップに繋げる。この二点で役立てることを目標に製作に取り組む。

2 計画

(1) 問題の構成について

まず製作の構想を考えていくにあたって、この問題集は、復習問題集である。

そのため、授業で学んだ事が理解できずに躓いたり、プログラミングに対して苦手意識を持っている学生を対象に製作していく。そのため分かり易さを重視する点でも、問題の項目と順序は、授業で学ぶ構成と同

じであった方が分かり易くて良いと考えた。

私の学んでいた去年の、Python 学習の授業の主な教材は、教授から配布されるテキストであった。しかし、私の持っている教材は、古い教材であるため、教授に今年使っている教材を頂くことにした。学ぶ単元とその順序については、去年のテキストと比べ、大きな変更は無かった。来年の授業の構成について変更があるか、教授に伺った所、来年も大幅な変更は無いだろう。との確認を取り、今年の教材を参考に、問題の構成を考えていくことを決定した。

(2) 問題の形式について

問題の解答方法をどのような物にしていくかを考えていく。

まず、始めに考えられたのは、授業のテキスト内の問題と同様に、作るプログラムが、どの様なものであるかを説明した後、一からプログラムを作り上げていき、出来上がったプログラムの正誤を判断する方法。二つ目は、作るプログラムが、どのようなものであるかを説明した後、問題プログラムの要点を、穴埋め形式の問題にし、回答欄に記述し、正誤を判断する方法が考えられた。

前者は、問題の内容を完全に理解していなければ、答えることが出来ない。そのため理解力は、後者より深まると思われる。しかしプログラムが苦手な学生には、白紙の状態からプログラムを書き上げる事が、出来ない人がいる可能性がある。または、この問題を覚えていないため手の付けようが無いという自体になる可能性もあると考えた。さらに、解答の正誤を判断する面でも、プログラムの書き方は、書く人によって様々であるため、全ての解答に対して、正誤の判断をどのようにして付けるかが難しいという問題がある。

一方後者は、前者に比べると初めから要点以外のプログラムは書いているため、簡単になるだろう。しかしプログラムが苦手な学生でも、要点以外のプログラムから、問題内容を理解し、答えられる可能性がある。解答の正誤を判断する面から見ても、正誤を判断するのは、要点である穴埋め部分一つ一つを判断すればいいので、判断しやすいと考えられる。

このことから、プログラムが苦手な人を対象とする、この復習問題集においては、問題のプログラムの要点を穴埋めで、解答する問題形式の方が適していると考えるため、穴埋め形式で問題作りに取り組んでいくこととする。

(3) 作業計画

1. 授業で作製した Python ファイルの動作確認・Python の復習
2. 問題にするファイルの選択
3. 問題集の作製・動作チェック・修正
4. テストプレイ後アンケートの実施
5. アンケート結果を踏まえての更なる改良
6. 最終確認の後、提出

この問題集は私の考えだけでなく、アンケートで使ったユーザーの意見を取り入れて改良してこそ、ユーザーの立場に立った製作になると考えるため、アンケートに万全の態勢で臨むことの出来るよう計画性を持って取り組んでいく。

3　制作物について

(1) トップページの作製

最初に問題集のトップページを、どのようなものにするか考えた。

そこで、ページのフレームを横に右と左で分割する方法を取ることにした。フレームを分割するためには、フレームを設定する HTML ファイル、右に表示するための HTML ファイル、左に表示するための HTML ファイルの三つがになる。そしてフレームを設定する HTML ファイルを作る。HTML タグは<body>タグの中に書くことが普通である。しかしフレームを設定するので、代わりに<frameset>タグを使用した。さらに左右に分割する場合は、この<frameset>タグには clos という属性与える必要があり、clos 属性をつけることで左右のウインドウの大きさを指定することができるようになる。今回は左側のウインドウを、それ程大きくする必要が無かったため、画面に対して左のウインドウを２０%、右を８０%と指定した。次に、<frame src==”” >を使用して右と左それぞれに表示する HTML ファイルを指定した。ここまでの説明をソースで表示すると以下のようになる。

```
<html>
<head>
<meta http-equiv=” Content-Type”  content=” text/html;
charset=shift_jis” >
<title>Python 復習教材</title>
</head>
<frameset clos=” 20%,80%” >
<frame src==” menu.html”  name=” menu” >
<frame src==” main.html”  name” main” >
```

```
</html>
```

上記のフレームを設定する HTML ファイルを index.html とした。

次に左側に表示する HTML ファイルを menu.html として、この問題集の目次の役割を果たすように縦並びで、各単元ごとにテキストと同じ順番で、章ごとのタイトルと問題番号を表示した。問題番号には、リンクを貼って問題ページへ飛べるよう作成することにした。残りの右側に表示する HTML ファイルは、main.html としてこの問題集の使い方を説明することにした。(図 1)

(2) ファイル形式の選択

問題の形式は前の項目の「問題の形式について」で説明した通り、問題とするプログラムの要点を穴埋めで、解答する問題形式を目標とした。

問題を作製するにあたって、どのようなファイル形式で問題のページを作製するのかを自分の使うことのプログラミング言語を確認しながら考える必要があった。

私の使うことの出来るプログラミング言語は、授業で学んだことのある HTML、CSS、Python、PHP、Javascript である。これらの私の持っている技術面から考えた結果、PHP を用いて作製することが、考えられた。

PHP は、オープンソースの汎用スクリプト言語である。特にサーバーサイドで動作する Web アプリケーションの開発に適している。

PHP は基本的には、HTML ファイルの中に処理内容を記述したスクリプトを埋め込み、結果を HTML として Web 画面に表示することができる。その他にも CGI というプログラムを Web サーバー外で起動させ、その出力をクライアントに送信する方式と処理速度を比べて

も、CGI は Web サーバーの外部プログラムなので、クライアントからリクエストがあるたびに、新たにプログラムを起動する必要がある。そのため CGI を実行するたびに、CPU やメモリに負担がかかり結果、サーバーに負担がかかってしまう。しかし PHP ではクライアントのリクエストのたびにプログラムを起動することがなく、サーバーに負担がかかりにくい。結果、CGI に比べて処理速度が速くなる。以上の事から今回作製しているオンライン復習教材には、Web アプリケーションの開発に適している、処理速度が速い等の事から、問題解答のページは PHP で作製することとした。

(3) 問題ページの作製

PHP で作製した問題ページのレイアウトは、問題のタイトル、どのようなプログラムを作るのかをページの上部分で説明し、それより下を本題の穴埋め問題が表示されるというデザインにした。(図２)

問題のタイトル、問題内容を説明する文章には<h1>タグや<p>タグを使用したが、それより下の問題部分は全て<pre>で囲いその中に記述していくこととした。

問題ページは説明を除けば、プログラムのソースコードが、大半を占めるページであるため、<pre>タグを使用するのに適したページだと判断した。

<pre>は、PREformatted text の略で<pre>で囲っている範囲内のソース中の改行やスペースを、そのまま表示することの出来るタグであり、ソースコードなどを表示するには便利である。ただし¡や¿や＆は特殊文字として認識されているため、< や> や& で記述する必要がある。

使用例としては以下のようににになる。

HTML ソース

```
<pre>
if (xx &gt; 5) {
    print "大きい!!\n";
}
</pre>
```

ブラウザ表示

```
if (xx > 5) {
    print "大きい!!\n";
}
```

穴埋め形式のシステムについては、<form>というタグを使用した。

<form>とは<form>から</form>までで、一つの入力フォームを形成しており、<form>の中には<input><select><textarea>などの部分フォームを配置することができる。部分フォームの<input>の後に続く、type 属性の値によって見栄えや動作も異なったものとなり、以下のような属性がある。[type 属性]

1. text　一行テキストの入力を行なうフィールド
2. password　パスワードの入力フィールド
3. file　ファイルをアップロードするためのファイル入力フィールド
4. checkbox　チェックボックスで複数項目の選択
5. radio　ラジオボタンで一つの項目の選択
6. submit　実行ボタンを形成して action 指定の URL にフォーム

データを渡す

7. reset　取消ボタンを形成してフィールド内すべてをデフォルト値にリセット
8. button　汎用ボタンを形成してスクリプトを起動
9. hidden　表示させないで、サーバー側にコントロールデータを渡す

[属性]

1. name　フィールドのデータ名で submit と reset 以外すべてのタイプに必要
2. value　スクリプトに渡される値で submit、reset はボタン名として表示
3. checked　選択フォーム checkbox、radio の選択された初期値
4. size　text や password フィールドの長さを文字数で指定。初期値は 20

<form>は通常一つのサブミット（実行ボタン）があります。各入力部分に値を入力した後、サブミットボタンを押すと、method 属性で指定した転送方法で、action 属性で指定したアクション（転送先 URL）呼びだす。method 属性値はデータの送信方法より、GET と POST の二種類があり、これらはそれぞれ HTTP 要求メソッドの GET と POST に対応している。GET は action 属性に指定した URL に？をつけデータをつけて URL として送られる。それぞれ受け取ったサーバーは、？以降を環境変数に入れて CGI プログラムを渡す。送信できる情報量にも制限がある。

POST は、URL の一部ではなく独立した本文として送られる。セキュリティの面でも有利な POST が、Web ページの入力フォームではよく使われる。このような性質を持ったフォームを次のように使用

した。

まず、正解の場合の指示を作製していく。

もし正解すれば、この動作を行うという指示をするため、PHP タグの中に if 文を記述した。そしてその中に適当な変数作る。ここでは、$q1 とする。この変数をイコールで結び、($p1="def") というように回答欄の正しい答え置き換える。

続けて、正解の場合に問題ページに表示される入力フォームを作製していく。

<input type>で入力フォームを作製し、type 属性 hidden を使うことで、正解のプログラムが回答欄に入力する以外では、答えが表示されることの無いようにした。なお正しい答えは<u>タグで囲み、<u>タグで囲んだ文字は、CSS を使用する事で緑色で表示されるよう、指定した。これにより、入力した答えが正しかった場合は、一目で見分けることの出来るよう工夫した。(図３)

補足のため、ここまでのソースを表示すると以下のようになる。

```
<?php if ($q1 == "def") { ?>
<u>def</u> bmi(shincho, taiju):
<input type="hidden" name="q1" value="<?php echo $q1;?>">
<?php }
```

次に間違った場合の指示を作製していく。

先程の if と対になる else を PHP タグの中に記述する。後は<input type>で入力フォームを作成し、type 属性 text と記述することで一行の入力フィールドを作り、回答欄ごとの答えが、入力できる適切なサイズにするため、type 属性 size を記述することで指定した。

補足のため、ここまでのソースを表示すると以下のようになる。

```
<?php } else { ?>
<input type="text" name="q1" size="4"
value="<?php echo $q1; ?>">
bmi(shincho, taiju):
```

そしてこの正解の場合の指示と間違いの場合の指示を合わせると以下のようになる。

```
<?php if ($q1 == "def") { ?>
<u>def</u> bmi(shincho, taiju):<input type="hidden"
name="q1" value="<?phpecho $q1;?>">
<?php } else { ?>
<input type="text" name="q1" size="4"
value="<?php echo $q1; ?>">
 bmi(shincho, taiju):
```

このようにしてできる正解の場合と間違えた場合のプログラムを記述する事で、回答欄を作製し、同じ要領で変数を 1.2.3.4…. と作っていき、問題ページごとに解答欄の数だけ繰り返していくことにより、穴埋めのシステムを作り上げていった。

回答欄と、その他の穴埋め問題にしなかったソースは CSS を使用し、オレンジ色の枠で囲み、問題文として仕上げていった。問題文より下には、<input type>より type 属性 submit を使用することで、解答欄の問題が正しいものであるか判断するチェックボタンと、reset を使用することで、解答欄に入力中の内容を削除するクリアボタンを設けた。

そして、いくら考えても問題の答えが、解らない場合に答えを確認できるよう解答ページへのリンクとトップページに戻ることの出来るようトップページへのリンクも設けた。

(4) 解答ページの作製

解答ページは特に特別な処理を必要としなかったため、HTML ファイルで作製に取り組んだ。

問題内容のソースを全て記述し、問題ページでは解答欄が正解した場合、答えが一目でわかるよう緑色で表示したので、解答ページと同様に CSS と<u>タグを用いて、解答となる文字は、緑色で表示されるよう指定した。さらに解答ページ全てにでは無いが、答えの横には答えを導き出すヒントを記述した。

例えば、この問題は指定したファイルの本文を全て表示するプログラムを作製する問題の解答である。

```
#!/usr/bin/env python
# -*- coding:utf-8 -*-
#関数定義
import sys
# ファイルをオープンする。
# ファイル名はコマンドラインの引数で指定する。
input_file = open(sys.argv[1])

# 一行読み込む。
line = input_file.readline()
```

```
# 何もなくなるまで繰り返し読み込む。
while line != "":
        # 処理を書く。
        print line, # ここの処理は読み込んだものを表示するだけ。
        # 次の行を読み込む。
        line = input_file.readline()

# 終わったら、ファイルをクローズ。
input_file.close()
```

上記のように＃の後に、答えを導き出すための考え方を記述する事で、ヒントを作製した。

後は、対応する問題ページへのリンクとトップページへのリンクを設け、これを解答ページとした。(解答ページ図４を参照)

(5) 問題集の仕様について

このようにして作り上げた問題ページは、上記に記述通り、ページ上にはこのページの単元のタイトル、どのようなプログラムを作製するかの指示した説明文を記述し、その下に問題本文を CSS を使用し、オレンジの枠内に表示するようにした。解答方法は、どのようなプログラムを作製するかの指示した説明文を読んだ後、問題本文の回答欄以外のソースを頼りに、空欄に当てはまる答えを導き出すというものである。

回答欄に回答者が正しいと思った解答を入力し終えるたら、問題ページの下部分に設置されているチェックボタンをクリックする。もし回答欄に記述した解答が正解であれば、正しい答えが緑色で表示される。こ

の解答済みの答えは、一度入力すると再びチェックボタンをクリックしても、解答済みの状態が継続されるようになっており、何度も同じ問題の解答をする二度手間を防いでいる。解答済みの状態を解除する場合には、問題ページの解答リンクから各問題に対応した解答ページに飛んだ後、問題ページに戻って来る事で、解除する事ができる。

回答欄に記述中の解答を全て消したい場合には、問題ページ下部分のチェックボタンをクリックすることで、問題ページ内の全ての回答欄の記述中の解答を削除する事が出来る。また、問題ページ下部分にはＴＯＰページへのリンクが設置されている。

もし解答した答えが間違った場合は、回答欄は再び空欄になり、もう一度考え直すようにしている。何回問題に挑戦しても正しい答えがわからず、詰まってしまった場合には上記のように問題ページの解答リンクから解答ページに飛ぶ事で答えを確認する事が出来る。

解答ページでも問題ページと同じように正しい答えは緑色で表示されるようになっている。問題によっては、行ごとに＃の後に、答えを導き出すための考え方が記述されており、ヒントの役割をしている。解答ページ下部分には各解答ページに対応した、問題ページへのリンクが設置されており、再び問題に挑戦する事が出来る。また、解答ページ下部分にはＴＯＰページへのリンクが設置されている。

以上がこの復習問題集の正解・間違いごとの反応と、ページごとの仕様である。

(6) 利用者を考えての工夫点

まず、一つ一つ問題に回答欄が多くなりすぎないよう心掛けた。何故なら問題形式を選択した時も考えていたように、あまり空白が多すぎる

と白紙の状態からプログラムを組み立てていくのと変わらなくなってしまい、折角白紙の状態から作れない利用者の事を考えて穴埋め問題にした意味が、無くなってしまうと考えたためである。そのためどうしても要点が多く回答欄が増えてしまう問題以外は、穴埋め部分以外のソースの割合増やし、答えを導きやすくなるよう心掛けた。

次に同じ単元の一つの問題で穴埋めにした部分は、他の問題でも穴埋めにするよう気を付けた。

例えば、この問題集には BMI 値を計算する問題が二つ載っている。一つは、身長・体重を計算しＢＭＩ値を計算するプログラムを作る。さらに BMI 値が 18.5 以下だったら「痩せすぎ」、18.5 以上 25 未満なら「適正」、25 以上なら「肥満」という判定結果も表示するプログラムを作製せよというものである。

もう一つは身長・体重を計算しＢＭＩ値を計算するプログラムを作る。さらに BMI 値が 18.5 以下だったら「痩せすぎ」、18.5 以上 25 未満なら「適正」、25 以上なら「肥満」という判定結果も表示するプログラムを作製せよ。ここまでは、さっきのプログラムと同じであるが、これをコマンドラインを用いて行えというものである。そのため途中のソースが全く同じなのである。

以下がそのソースである。

```
        shi_m = float(shincho) / 100
        kekka = float(taiju) / (shi_m * shi_m)
        return kekka
def bmi_hantei(bmi_chi):
        if bmi_chi <= 18.5:
                hantei = "痩せすぎ。"
```

```
elif bmi_chi <= 25:
        hantei = "肥満。"
else:
        hantei = "適正。"
return hantei
```

そのためどちらかで、穴埋めにした部分はもう一方でも穴埋めにしておかないと片方の問題を解いたときに答えがわかってしまう可能性がある。これでは、問題集としての欠陥が出来ると考え、同じ単元の一つの問題で穴埋めにした部分は、他の問題でも穴埋めにするよう心掛けた。

4　まとめ

(1) アンケート結果とその後の改良

当初の計画である問題集の全ての内容を完成させてから、アンケートに臨むことは、叶わず、製作途中でのアンケートを取ることとなった。アンケートに協力してもらったのは、今年プログラミン演習で Python を学んだ、同学科で第三学年の福田ゼミ生五人である。

アンケート用紙には、問題集の使いやすさを {使いやすい・やや使いやすい・普通・やや使いにくい・使いにくい} の六段階の評価に丸を付けてもらい、改善点を自由に記入してもらうスペースを用意した。その他に問題の難しさ、ページのレイアウトに関しても同様に六段階の評価に丸を付けてもらい改善点を記入するスペースを用意した。良かった点は、正解している解答に色がついて見やすいことが挙げられていた。問題の難しさは、「復習問題なのでこのくらいの難しさが丁度いい」との意見もあり、六段階評価でも普通が四人、やや簡単が一人と問題の難しさは、ほぼ適切であることが確認できた。

一方悪い点は、入力した解答が間違っていた場合に何もアクションが無く、分かりづらいとの意見が挙げられていた。例として間違った問題には、間違いっています表示してほしいとの要望もあった。他には、問題集のトップページが殺風景であるシンプルすぎるとの不満が挙げられていた。

このアンケート結果を踏まえて改良を行うことにした。

最初に挙げられていた不満の間違った問題へのアクションが欲しいとの意見には、アンケートの要望を参考に、間違った後の解答欄に間違っていますとの表示が出るよう改良をしようと考えたが、解答欄に間違っていますと表示するようにすると、もう一度問題に答える度に、間違っていますという表示を消してから、解答しなければならなくなるため手間がかかると考えた。

他にも間違えた解答欄に色を付けるなども考えたが、正解の解答は緑色になるため、一目で判断できる、すると必然的に残りの空欄になっている解答欄が間違いであることは、何かのアクションを付け加えなくても容易に判断することが可能である。との結論に至ったため、間違った問題への何らかのアクションを付け加えるという改良に関しては、トップページに解答例を載せて間違えると空欄になるとの説明を加えることで対応することとした。

次に、トップページが殺風景シンプルすぎるとの不満は、アンケート実施のギリギリまで問題製作に取り掛かっており、トップページが未完成のままアンケートに臨んだことが、原因であると考えられる。

アンケート終了後、トップページは問題集の使い方を言葉での説明と問題画面をキャプチャーした画像を掲載した。種類は二種類で、正解の時の動作と間違いの時の動作を説明するものである。画像を交えて問題

集の使い方をなるべく分かり易い心掛け説明を加えたトップページに仕上げたため、この不満点にも対応できたと考える。

(2) 自己評価と今後の課題

この問題集はプログラミングが苦手で躓いていた、私だからこそ同じプログラミングが苦手な人の気持ちを理解し、分かり易い復習問題集が作くれるのではないかという考えから、始まった製作であった。

目標はこの問題集を使った人が、プログラミングに対して理解を深める、苦手意識を克服してのプログラミング学習意欲向上のきっかけとなることを目標として取り組んできた。製作を終えた今でもその気持ちは、変わっていない。問題集としてはアンケート結果をもとに、その後の改良も加えたため、使いやすいものになり満足のいく出来栄えになったと感じている。

製作を終えて改めて自分の目標としていたものが、いかに難しいものであるかを感じた。今回の問題集は、授業のテキストに掲載せれている問題の中から、各単元の内容をプログラミング理解するのに、良いと感じたものを私の判断で選び、穴埋め形式の問題も、各単元から私が思う要点を穴埋めに設定して作製した。もちろん使用するユーザーのことを考えた分かり易い問題作りを心掛け、アンケートの結果では、問題の難しさは、復習問題としては丁度いいとの意見もあった。しかし、これでは物足りない簡単すぎるという意見やこれでは、難しいもっと簡単なものがいいという意見のように、この問題集がすべての人にとって分かり易く丁度よい難易度であるとは言えないと思う。

穴埋めの設定にしてもそうである。この穴埋めは、私がこの単元の要点であり、これが解ければ、この単元は理解できたと言えるだろうと考

えたものを穴埋めにしたが、問題を解いた後でも、その単元について理解できていない人もいるかもしれない。そう考えると、この問題集には、まだまだ課題・改良すべき部分があると言えるだろう。そうした課題を一つ一つクリアしていくことで、プログラミングに対して理解を深める、苦手意識を克服してのプログラミング学習意欲向上のきっかけとなるという目標に近づけるのではないかと考える。

文献表

ホームページ作成講座　http://www.homepagecreation.net/

日本 Python ユーザー会　http://www.python.jp/Zope

とほほのWWW入門　http://www.tohoho-web.com/www.htm