

プログラミング演習用教材の作成につ
いて

子浦 未央

目 次

1	はじめに	1
1	1 動機と目的	1
2	2 考察	2
3	3 従来の教材の問題点	4
4	4 使用する対象	5
2	2 計画	6
1	1 使用するソフト	6
2	2 工夫すべき点	7
3	3 実際の制作	8
1	1 アシスタントとしての経験	8
2	2 1度目のアンケート	9
3	3 制作過程	11
4	4 評価	16
1	1 アンケート	16
2	2 教材の評価	17
3	3 解答の評価	17
4	4 そのほかの指摘	18
5	5 自己評価	19
5	5 課題	21
1	1 改良する部分	21
2	2 改善すべき点	21
3	3 これからの課題	22
4	4 最後に	23

1 はじめに

(1) 動機と目的

私が福田ゼミに入ることを決めたのは、人文情報学という特殊な学問を、技術と共に学ぶことができるといった点であった。そして、そこで知ったのが、このゼミのテーマである。2回生までは、私はこの大学に入ってどんなことがしたかったのか分からなかったが、ゼミに入り知識を高めていくこと、で自分が持ち合わせているスキルを向上させていきたいと思った。それは、プログラミング演習という授業を受けたことによるものだった。

ゼミでは、プログラミング演習を必ず受けることになる。授業を受けていく中で、必ず触れるものがPythonであり、おそらくゼミに入って大半の人が、今まで学んだことのないものだと思われる。そこで私は、授業で使用していたプリントを基盤にし、模範解答や自分なりの説明を加え、初めて習うことでもできるだけ分かりやすく、また、後で読み返して、復習ができるような教材を作成しようと決めた。これは、ゼミのテーマである「人の役に立つものを作成する」という目的に当てはまる。

きっかけは、私自身が、授業を3回生のときに受講し、ここはどうしてこうなるのか、説明を聞いても分からず、授業アシスタントの先輩に聞いてやっと理解するものの、その後はどうすればいいのか分からない、と考える機会が多いと思ったことだった。実際にプリントを読んでも、文字の羅列に翻弄されるばかりで、いっこうに頭に入らない。そのため、わからないままに授業を終えてしまうことが多く、後になってとても後悔していた。

なぜこうなるのか、という疑問を抱いた時、どうしてもっと先生に聞かなかっただろうか。たとえ口頭での説明をされて、その時は理解し

でも、いざ同じ問題をやろうとした時に、どうすればいいのか分からない。プログラムの意味も、その場限りの知識にしかならないことを痛感したのである。

そういった経験から、受講時にとにかく自分が必要だと思ったものを考えた。それは練習問題の解答である。解答のデータやファイルはUSBメモリやハードディスクに保存はできるのだが、自宅でいざ復習しようと思ってみても、Pythonを動かすなどといったことは、初心者を対象にすると容易なことではない。とにかく見やすく、分かりやすいものになることを目指した。

(2) 考察

初心者のためのプログラミング講座、Pythonを扱うためのチュートリアルなど、サイトや本はたくさんある。調べていくと、Pythonはその中でもシンプルなものなので、比較的初めてでも扱いやすいということだった。しかし、初心者向けということで考えると、マニュアル全体の表現がやや難しいように思う。それはおそらく初心者といっても、ある程度の知識を備えた者への説明だからであろう。

専門学校に通っていた経験がある、などというならば話は別だが、授業を受けるほとんどの者がその知識はないに等しい。そんな初めての授業の時に、C言語の説明などをいきなりされても、すぐに理解して動かしてみようというのは、大変困難である。

実際にサイトを見ながら、その内容が活用できるか否かを確認してみた。参考にしたのは、『Pythonの紹介』⁽¹⁾、『Pythonチュートリアル』⁽²⁾というサイトである。

前者の『Pythonの紹介』については、始まりから見えていくと、専門用

語が飛び交い、紹介という文面であっても、ぱっと見る限りでは混乱する内容であった（図1参照）。Pythonの特徴を紹介している部分も、プログラマを目指す者への説明であり、どうにも扱いづらい印象が強い。これは、やはり知識を要するサイトの内容であると思う。

しかし、プログラミング経験のない人のために、という目的で貼られていたリンクにあった、『インスタント・ハッキング』⁽³⁾という項目（図2参照）は、プログラミングそのものの説明が書かれており、料理レシピを使用して例題に挙げていた部分は比較的分かりやすいものだった（図3参照）。print文の説明も、授業序盤に行くものなので、参考にする順番としては、Pythonの紹介というトップページよりも、こちらを先に読んだほうが、骨組みなどは分かるかもしれない。

『Python チュートリアル』というサイトでは、序という項目を読むと、説明文はまわりくどいが、Pythonそのものの説明は、読んでいけば「Pythonは簡単に習得できる」という意味合い自体は理解できる（図4参照）。

しかし次の項目である「やる気を高めよう」という項目については、自分が開発者ならば、という仮定を示しているが、これは未経験の者への例えとして本当に適切なのか、疑問を抱く（図5参照）。シェルスクリプト、などという言葉を用いられても、注釈もないので、どういったものなのかも分からず、ひとつひとつを自ら調べていかなければならない。

一通り目を通して思ったことは、「プログラミングの勉強をしている者、知識がある者にとって面白く読める内容」であった。チュートリアルと称するならば、やはり初心者への対応も表記したり、参考になるサイトのリンクが欲しいと感じる。これはどんな参考書や問題集を読んで

いても思うことではないだろうか。教材のように、例題があつて、それについての説明を書く、というような流れではないと、初めてプログラミングを学ぶ者にとっては習うことに不安を覚えてしまうのでは、と私は思う。

以上のことを踏まえて様々な資料を見比べていくと、授業で使用しているプリントの内容は、本当に基礎のものであることが分かる。『Python チュートリアル』というサイトは特に、読むべき項目の量も多く、授業で使用している教材の倍以上の章の構成である。大学の授業はたった 90 分で、ある程度の内容を進めることになる。そう考えると、教材の 1 回 1 回の授業内容は、凝縮されているとはいえ妥当な構成であるのではないだろうか。現にこのサイトを活用して、大量な資料を読みつつ授業を行うのは、困難であると思われる。

他に気付いたことは、電卓計算がどのサイトや本でも例として取り上げられていることである。この電卓計算はプログラミングの基礎を学ぶ上で必要な項目であると感じた。もう少し具体的に説明を付ければ、参考になるのではないかと思う。

私が目指すものは、あくまで授業で使用できる初心者用のマニュアルなので、「基礎中の基礎」という根本を見直していきたい。基礎というものは、単に用語を並べ立てることではない。「りんごが 1 つとりんごが 1 つ、合わせると 2 つ」というのが、 $1+1=2$ である、という説明をすることと同じものではないか。応用は、慣れればいくらでもできるのだから、基礎の部分を固めることが重要であると私は考えた。

(3) 従来の教材の問題点

授業で使われた教材の特徴として、以下の点を挙げる。

- ・ 長文が多く、読みづらい
- ・ 例題の内容がわかりづらく、練習問題が解けない
- ・ プログラムを起動した結果がない

長文であるという点は、読み手を意識するという書きかたにするべきだと感じた。改行を入れたり、大事なタグや説明の文体を変えたりしてレイアウトをデザインすることはもちろん、説明の解釈を自分なりに付け加えたり、または省いたり、結果を書き足すことが必要である。

練習問題で応用しづらいということについては、例題での説明が少なく、応用問題を解こうとしても参考しづらいと感じる。もちろん口頭での説明を聞くことも大事なのだが、どうにか解きやすいようにできないものか思案する必要がある。

プログラムを起動した結果が表記されていないので、ターミナル上でどう表示されれば成功なの分からないということも、問題である。説明を並べ立てているだけで、「こうすればこうなる」という答えがないため、授業アシスタントに確かめてもらうことでしか、正解かどうかを判断できない。

これらの問題点をどう工夫すべきか、考えていかなければならない。

(4) 使用する対象

対象人物は、プログラミング演習という授業を初めて受ける 3 回生を中心とし、なおかつ Python についての知識があまりない者を対象とする。もし、わからないままに進めてしまっても、後々読み返して復習ができるようなものを制作したい。これは、自分自身の経験に基づいた考えである。

また、プログラミングに関して多少の知識がある者は、模範解答のプ

プログラムと自分の解答を比べたり、自ら練習問題を考えてみたりと、いろいろと応用するためのマニュアルとして使って欲しい。Python 自体が応用の効くものなので、後々に見返すことで基礎力の向上に繋がるものになると思われる。

2 計画

(1) 使用するソフト

では、具体的にどういったものを作っていくか。計画としては、Microsoft Word、TeX、もしくは HTML で作成する。

Microsoft Word ならば、レイアウトの変更も比較的簡単に行えるので、時間をかけずに作成することができる場所にメリットがある。従来のようなプリント形式なので、保存はファイルなどで整理すればよい。

TeX を用いる場合は、現在使われている教材プリントが TeX を使用したものなのだが、福田ゼミの必修授業のひとつである DTP 演習の形式と似たものになり、同時に授業を受けている者が多いため、プリントの保存の際に混乱するかもしれないと考えたので、TeX での作成は今回は見送る。

HTML はオンライン教材として使用できるので、例題や練習問題の工夫もできる。

Microsoft Word を主として使用し、時間があれば様々な工夫を凝らすことのできる HTML での教材を作りたいと考えた。

(2) 工夫すべき点

まず、この授業を受けたときに、3回生の時の自分がどう思ったかを改めて考え直した。私はまったくのプログラミング初心者で、このゼミに入った時がついていけるかが不安だった。

第1回目の授業は、まずはPythonの起動の仕方などから習った。しかし、今プリントを見返してみても、その説明は長文で、これでは読んでいてもインストールの方法などが初心者にはよくわからないのではないかと、という印象が強い。これは、起動の仕方の段階で、番号をつけてわかりやすくすればよい。

それぞれのセクションでの説明文は、もう少し詳しい解説を付け加えるべきだと考えた。例えば基盤となるプリントの1枚目を見ると、計算問題で使用する演算子の説明がなかった。これは私自身の知識がないこともあるが、「掛け算はアスタリスクの記号を使う」などという説明が問題を解く前に先にあれば、迷うこともないのではと感じた。また、「こんなプログラムがあるので、こうしよう」と大雑把に書かれているだけで、例題がないセクションがある。これでは、仮に動かしたいプログラムがあってもどう起動すればいいか、プリントを見ただけでは分からなくなる。例題を付け加え、それについての説明文を書き加えることが必要である。

これで、先述した従来の教材の問題点に挙げた事についても、工夫ができる。

3 実際の制作

(1) アシスタントとしての経験

卒論のテーマをプログラミング演習用教材の作成だと決めたときに、先生から「授業のアシスタントをやってみないか」という依頼のメールが届いた。正直なところ、私は迷った。私自身がちゃんとプログラミングを理解していないのに、人に教えることができるのか。人に教えるという立場になることに抵抗があったのである。その意思を先生に伝えたところ、先生は授業を教えていく上でもう一度勉強ができること、教えることについては先生もいるので問題はない、と背中を押して下さった。何より、自分自身にとっても役に立つテーマであったので、アシスタントを行うことで何かヒントを得られるかもしれないと思い、依頼を引き受けることにした。

3回生と共に授業を受けていく中で分かったのは、やはりプログラミングについての経験がない者が大半だった。初めのほうの簡単な問題では、応用ができず、同じところで立ち止まる者が多かった。これは、いきなりの問題のステップアップについていけなくなったことが原因であると推測する。私自身も実際、いきなり問題が難しくなったと感じることがあったので、この点は改善すべきだろうと考えた。

また、コマンドについての知識もあまりなかった。Mac での場合 `command` キーと `S` キーを押すと保存ができるということや、`#` はコメントアウトの記号であるということを知るのにも、メモを取ることを忘れていたりして、何度も繰り返して教えるという機会が多かったように思う。授業でコマンド表のプリントを渡すこと自体が遅かったせいもあるが、この先もよく使うコマンドはプリントに表記したほうが良いと考えた。

授業を教える上で、つまづいていたものが多いと感じた授業があった。「リストというデータ型」という内容である。これは、リストを処理する機能であり、括弧内にあるものを取り出すというプログラムなのだが、リストの番号が0から始まるという説明が理解できず、私自身も頭では分かっているのに説明ができないという状況だったため、教えることにとても苦労した覚えがある。

アシスタントを経験して、自分の力不足を改めて感じた。授業をしながら復習はできるものの、自分がWindowsでのやり方でしか学んでいないため、解答の応用が利かないのである。従来のやり方と違う答えもあったので、とても戸惑った。その答えを導かせるためにも、今までの解答をもう一度動かすことは必要不可欠な要素であるように感じた。

(2) 1度目のアンケート

前期終了の時点で、私は一度、プログラミング演習を受講している生徒全員に、授業で使用したプリントについての意見を聞くためのアンケートをとった。質問項目は、以下の3点になる。

1. 改善してほしい部分や分かりにくい部分はあるか
2. 追加してほしい解説や内容はあるか
3. そのほか気付いたところ

改善してほしい部分で一番多かった意見が、「長文で読みづらい」ということだった。これは私が3回生のときに感じたことと同じなので、改行を加えたりすることで改善ができる。「プリントのナンバーが混同する」という意見もあった。訂正されていない日付などがあったので、確かに分かりづらいと思われる。

追加してほしい解説は、「解答がほしい」「どのように表示されれば正解なのか分からない」という意見が目立った。

確かに、問題では実行した結果が書かれていない。これは先述した、従来の教材の問題点で挙げたことと同じである。やはり、皆分からないままに授業を終わらせてしまうことが多いのであろう。自分が感じたことと同じ意見が多いと思った。「具体例が欲しい」という意見も同様である。

また、昨年からプログラミング演習を受講している生徒は、「私たちが去年行っていたのは Windows での操作だったので、Mac での違いを説明してほしい」という意見を述べてくれた。

そのほか気付いたところは、「プログラムの日本語解説が分かりやすかった」という意見が多かった。授業の後半での問題で、日本語の通りにプログラムを打とうといった内容があったが、確かに個々のプログラムの意味を理解することに繋がると感じた。これについては、「プログラミングはあまり数学的な要素を使わないのが意外だった。読解力が必要とされるということが重要だった」という意見もあったので、解答を作る際は日本語の解説をつけようと思った。

一回目のアンケートを終えた段階で思ったのは、やはり同じ意見が偏っていたということと、自分の考えていたことが皆と一致していることが多いということだった。授業を受ける人が何を求めているかという理由が明確だったので、是非とも取り入れたい意見が多いように感じた。

(3) 制作過程

マニュアルの制作にあたり、どこから始めれば良いのかが分からなかったため、計画を練り直した。序盤は、自分自身がプログラミング演習の復習を行った。やはり忘れていたり分からないことが多かったため、かなりの時間を要した。とにかく最初から学び直すことが重要で、最低限の基礎をきちんと固めた上での作成を目標にした。

(i) 基礎力を自ら高める

アシスタントをしていたとは言え、それだけで基礎力がつくということはない。資料を参考に、もう一度 Python を使うプログラミングを勉強し、「初心者」という人物の視点から考える。とはいっても、やはり市販のマニュアルやサイトは知識不足の私には難しいものなので、もう一度授業の内容をすべて理解するという目的で、元々のプリントを使用した。

市販のマニュアルである『Python で学ぶプログラミング作法』という本は、序盤は何とか理解できる基礎の内容だったので、教材を作成するときにヒントを得られるかもしれない、と思い、その本を参考にしながら問題を解いていくことにした。基盤は十分に整った上で、その都度必要な説明やコマンドをプリントに実際に書き足していく作業を行った。

(ii) 構想

HTML で制作を行うと CSS の使用などの知識があまりない自分は、サイトの構造、デザインなどを一から勉強し直さなければならないため、制作する時間というリスクを負うことになるので、Microsoft Word のみで作ることを決めた。本来ならば HTML での作成は、実際に起動させながら一つ一つを確認することができるので利便性が高いのだが、

私の技術が足りないために断念せざるを得なかった。

すべてのプリントを何度も読み返し、どういったものが必要なのかを自分なりに整理した。Mac での操作は今年度が初めてであったため、授業でも Mac でのインストール方法や起動の仕方を一から説明を受け、時間がかかっていた記憶がある。具体的な Python のインストール法が載っていなかったため、それを書き加える。

あとは例題、説明の追加と解答の制作を行っていく。解答については、自分がもう一度理解し直すためと、解答の正当性を確かめるために同じプログラムを Python で起動する。間違った答えをそのままマニュアルに載せてしまっただけでは何の意味もないこと、そして、Windows で動いたプログラムが Mac で起動できるかどうかの確認を行いたかったことからである。

(iii) マニュアルの作成

基盤となるプリントの 1 枚 1 枚に目を通していくと、やはり文章の改行と段落分けをするだけでだいぶ見栄えが変わると思った。先生から教材のデータをいただいて、文章を Microsoft Word で打ち込んでいく際に、改行と段落を増やしていくことにした。必要なタグはフォントを変え、太字にしたり、覚えてほしい解説については下線を引くことで重要性をアピールする。

見栄えは上記のことでかなり改良されたが、それでも難解な文章が多いと感じていた。用語解説の意味について挙げる。基礎を固める為に重要な第 1 回目の授業内容のプリントでは、まず、1 行目の Python についての説明文を変更した。基盤のプリントでは、Python そのものについて「Perl に変わったスクリプト言語である」という説明があったのだが、知識のない者にその説明を記しても、よく分からないのではと感じ

た (図 6 参照)。

そこで私は、「初心者にも扱いやすいプログラミング言語」という言葉を用いて、簡潔にした。Python そのものの詳しい意味が知りたければ個人で調べることが出来るし、まずは Python とは一体どんなものなのか、実際に触れてみるのが大切だと思ったので、重要視しなかった。

同様の理由で、この授業以降に使用するプリントでも用語についての説明がある部分はほぼ省いている。しかし、演習プリント 6 にある `sys.argv` という変数の説明に関しては、その意味の説明をしている。これは、実際に `sys.argv` という変数を用いて処理を行うという内容なので、省略してしまうと扱うものの意味を理解できないからである。

解説文の説明に関しては、演習プリント 7 を例に挙げる (図 7 参照)。ここでは関数の説明について書かれているのだが、その例えの基盤のプリントを見ると、句点が多く、さらに言い回しが理解しづらいのでは、と私は考えた。まわりくどい説明を簡単に纏めることで、少しでも「文章が長い」という印象を和らげたことになったのではないだろうか (図 8 参照)。

使用するマニュアルには、Mac での使用法を書き加えた。Windows の Python の起動の仕方については、学校でのインストールの方法しか書かれていなかったため、簡単ではあるが自宅で使う場合の方法も付け加えた。これで自宅でも授業の復習ができると思われる。Mac は使用している人物が全体的にも少ないと考えたので、自宅での Mac の使用法は省略した。

解答は、その授業でのプリントにそのまま載せる予定であったが、それでは文章が長くなりすぎる上に、目に入るところに解答があつては問題を解く意味がない。そのため別紙で解答を作り、見直しができるよう

に工夫した。

(iv) 説明文の追加

以前に勉強した部分でも、覚えていないということは誰にでもあるので、演習プリントの何枚目を参考に、という説明を付け加えた。これはアンケートにもあったように、プリントを配布する日付と授業内容がずれることもあるため、数字をつけたほうが保存もしやすい、という要望も同時に聞き入れることができたように思う。

他に、startexampp という機能についての説明と、そのダウンロード法についての文章を作成していたが、これは省くことになった。この機能は、Mac を使用する場合に実行するのだが、ダウンロードすることは、先生自身が口頭での説明をすることで解決するという理由と、自宅での需要があまりないという理由である。

また、演習プリント 10 は、HTML タグ (HTML の要素名) を扱った内容である。これは、HTML の要素ごとに関数を作っていく、という内容なので、HTML タグの意味を理解することが重要である。だが、使用する HTML タグの意味をすべて書いていくと読み手には理解し難いものになると感じたので、HTML タグの解説が載せられている「とほほの www 入門」⁽⁴⁾ というサイトのアドレスを参考として表記した。HTML タグについては、1 回生のときに専門の技法という授業でも少し学ぶのだが、記憶に残っている者もいれば、忘れている者もいる。この内容は後の授業でも応用されるものなので、復習が必要である。その際にこのサイトはかなり役に立つので、その趣旨を書いておくことにした。

同じ演習プリント 10 にある問題 1 は、例題の表示がなかったので新たに書き足した。この内容は、説明を読んですぐに応用問題を解け、と

言っているに等しいものだったからである。内容は見返しても難しい説明ではなかったので、プリントを見直して問題を解く、という趣旨の元での問題文にした。この問題は、HTML タグの表示をそのままターミナルに反映させるという内容なので、あらかじめどう表示するかを問題に表記した。これで起動結果が自分で確かめられることになる。

(v) 模範解答の作成

解答を作る際に、解答がターミナル上で起動できるかどうかを確かめるために、すべての練習問題の解答を動かし、プログラムの間違いがないかを確認した。結果は、文字コードのエラーがあり日本語の表示ができないということだけで、あとは成功した。エラーの原因は私では対応できない問題だったので、漢字で表記されているものをローマ字に書き換えることで対応した。

序盤の演習プリント 3 では、授業中、特につまづく者が多いと判断したため、模範解答のほうに解説を新たに追加した。このプリントでの「番号は 1 から始まるのではなく、0 から始まる」という説明は分かりづらい表現なので、できるだけ詳しく文体を考えた。これは、演習 6 でも同じような状況になる。

3 枚目の内容は基礎の部分に値するので、覚えることを重要視するが、授業が進むにつれて以前の内容を覚えていない、という者がいることを仮定した上で、演習 6 では画像を追加した。ドーナツを 5 つ並べて、その 1 つ目のドーナツは 0 と数え、2 つ目は 1、3 つ目は 2…というように表現した。そうした図解があれば視覚的にも見やすく、記憶に残りやすい内容になるのではないかと、という理由からである。

プログラムの日本語解説は模範解答の一例に付け加え、コメントアウトの記号をつけることで区別した。特に、演習プリント 9 の内容は、た

くさんのプログラムが列挙されているので、一度に理解することは難しいと思われる。そこで、セクションごとにプログラムを分け、列挙されたプログラムひとつひとつに日本語の意味をつけることで、見やすいものになるようにした。

この模範解答の解説はコメントアウトされているため、ターミナルには反映されないで、そのまま実行して、それぞれのプログラムの持つ意味や言葉を、解答を見ながら確認するということにも使える。また、`=`が一つのみだと、代入するという意味が、`=`を二つ並べるのは、「等しい」という意味合いになることも、混乱しないように注釈を入れた。

4 評価

(1) アンケート

マニュアルがある程度形になった時に、意見を伺う為に福田ゼミに所属する3回生にアンケートを取ることにした。実際にマニュアルを見て、活用できるかどうかを他者に判断してもらうためである。項目は、教材についてわかりやすいか、見やすいかどうか。解答は分かりやすいか、見やすいかどうか。そして更に改善してほしい部分や解説してほしい点、そのほか気付いたところの感想をそれぞれ聞くことにした。

参考にしてもらうマニュアルは、できるならすべてのページに目を通して欲しかったのだが、それではかなりの時間がかかってしまうため、自らがプログラミング演習の授業を受けていた時にわかりづらかった部分を思い出して、そこを中心に読んでもらうことにした。

(2) 教材の評価

教材の全体については全員が「分かりやすい」「見やすい」の項目に丸をつけていた。具体的に良かったところは、以下の意見があった。

- ・ 演習プリント 6 で図を使っているところがよかった
- ・ 改行されていて文字が読みやすくなった

図解については、できればもう少したくさんのプリントに付け加えたかったのだが、作成に時間がかかるため、特に図で説明をしたほうが良いと感じる演習プリント 6 のみでの使用だったが、その内容での意見があったので良かった。文字が見やすくなったという意見は、1 度目のアンケートにおいて最も多かった意見だったので、改善できたという実感があった。

ところどころ脱字があるとの意見があったので、プログラムに脱字があってはいけないと思い、もう一度プリントを見直し確認した。プログラムのうち間違い自体は幸いなかったが、文章での誤字脱字はチェックを怠っていたことに気付いたため、即座に直すことにした。

(3) 解答の評価

解答の解説についての意見は、「分かりやすい」が全員であり、「見にくい」という意見が一人いた。まずは、良かった点から挙げていく。

- ・ 解答に解説が書かれているので見やすかった
- ・ すぐに答えを確認できる使い勝手の良さがよかった

それぞれの良かった点での意見は、「解答が欲しい」という意見から作成することを決めたので、その成果がうかがえた。今回は Microsoft Word で作成したものを印刷して配ったのではなく、public にある先生のフォルダに入れて、プリントと解答を同時に見られる、というような

形式にしたので、印刷時に見やすくすることも視野に入れなければいけないと思った。

見にくいという理由については、「演習プリント 6 の模範解答の表記の仕方は、内容の横に解説を書くよりも、内容の上に解説があったほうがいい」という意見があった。これはプログラムのところに日本語の解説を書く部分での感想なのだが、プリントを見返すと、確かにプログラムの横に日本語を書いているとやや見にくい印象を受けた。

プログラムの上に日本語の解説を書いてほしい、という意見があったのだが、覚えて欲しいのはプログラムそのものなので、日本語の解説はプログラムの下部に表記することにした。ここはプログラム自体の説明が長いため、少しでも見やすいように、空行を入れた。視覚的にはこれでだいぶ見やすくなったように思う。同じプリントの内容で、「引数を合計して結果を表示するプログラムという解説は、import sys の行ではなく gokei=0 の行に近づけるべきだと思う」という意見があったのでプリントを見直すと、確かに表記の場所が間違っていたため、すぐに修正を施した。

(4) そのほかの指摘

そのほか気付いた点はどこか、というアンケートの質問において、気になる意見を見つけた。それは、1 度目のアンケートにもあったはずの、「プログラムを実行するとどうなるかという結果が欲しい」という意見であった。計画ではその解説を作ることも考えていたはずなのに、作成していなかったのである。確かに、日本語の解説ばかりを載せても仕方が無い。私が必要としているものを忘れていたこともだが、何のために自分が作った解答をもう一度ターミナルで確かめたのか、呆れる結果に

なった。

私はすぐに、解答の解説に表示例を載せた。ここでターミナルで実行した結果を画像で作ることも考えたのだが、ターミナルは基本的に文字で反映されるため、画像を貼っても意味がないと思い、文章で作成することにした。それでも、説明の文章と混同しないようにフォントを太字にしたり、「質問文が表れるので、これを入力すれば、次はこれを入力すると、この答えになる」というような文章を追加した。

例外として、演習プリント 13 では、簡易掲示板を作るという内容であったので、実際にプログラムを動かして掲示板の画像を HTML にて反映し、スクリーンショットを撮って貼り付けることにした。

これは文章だけで理解するのは難しく、自分の目で完成形を確かめてもらうほうが良いと考えたためである。似た様な内容である、HTML を使った授業で使用する演習プリント 10 の内容でも、画像を貼り付けるべきか悩んだのだが、このプログラムの内容は、大学の人文情報学科のページを簡略化するという目的の内容であったため、画面は至ってシンプルなものにしかならないので、画像を貼ってもあまり意味がないのでは、と判断したので、解答のみの作成にした。

(5) 自己評価

今回、マニュアルの制作を決め、1 度目のアンケートを取ったが、私はどれぐらいその時の意見を取り入れることができたのだろうか。できるだけ見やすいものを作る、という点は達成できたと思われるし、解答を作成したことで、復習をする機会は増えるはずである。たとえ授業を受けた時点で曖昧にしか理解していなくても、もう一度作ったプログラムを動かせば、納得いかなかった部分の確認もできる。この点に関して

は、当初自分が描いていた理想に近づくことが出来たと思うので、自分自身も活用することができたと思う。

解答には日本語の解説が付け加えてあるので、もし Python のプログラムについて分からないことがあれば、そのプログラムの意味を確認することができる。しかし、そこでの文章表現をもう少し容易にする工夫ができたかもしれない。

そして、私自身がプログラミング演習の基礎を固め直したことで、自らの理解力が少しでも上がったと思ったのならば、例題や練習問題なども自分でももう少し考えることができたはずである。

しかし、やはり復習をしても、不安に思うことや、自分がこの答えを正しいと思い込んでるだけなのではないかという考えになり、もう一步踏み切ることができなかった。痛感するのは、明らかに自分の技術不足と理解力の乏しさである。

解答の作成も、もう少し構成をしっかり組み立てればよかったと思う。今回は要望を取り入れるために日本語での解説を重視して書いたのだが、ただ単に日本語の意味を並べ立てたに過ぎないかもしれないという思いがあった。だが、アンケートにはその解説が良かったとの意見があったので、自己の不安は少し解消されたように思える。

全体的に見て、やはり、教える側の立場になるのなら、勉強が必要だという結論になる。分からない人に手ほどきをする役目を果たすのは、その分からないことをちゃんと理解している者だからこそ、納得できる答えを教えることができる。それはアシスタントの経験をしたときも同じことだった。だが、このマニュアルを見て、私の制作したものが少しでも後で役に立つものであると考えてくれたならば、本望である。

5 課題

(1) 改良する部分

今回は時間が足りずに作成することができなかったが、解答を作成する上で、図解をつけて説明することは、やはり必要であると考えた。私は今回、最低限の箇所にしか図解をつけていなかった。演習プリント3のリスト作成において、リストの中のリストを取り出す、という部分は、文章では一見難解なもののような気がする。もう少し細かいところに気がつけば、よりいいものになったかもしれないと思うところがあった。

計画の時点で考えていた、HTMLでのオンライン教材も同様の理由で断念したが、これは是非制作するべきだと思った。練習問題を増やして解答フォームを作ったり、解答を間違えた時には、なぜ間違えたのかを画面上ですぐ確認できる要素などがあればとても便利になると思われる。HTML文法チェックなども行えば、HTMLについての知識も高めることができるだろう。

1度目のアンケートにあった、「国語力が必要である」という意見を思い出すと、プログラミング演習という授業には数学の知識はあまり必要がないことを改めて思う。制作したマニュアル自体にも、そういった、読解力を高める要素を含むべきではないだろうか。事前に、「プログラミングとは、文章を読解する力を要する」といったチュートリアルなどがあれば、私がそうだったように、情報技術という数学的なことを学ぶものではない、という考えにたどり着くのではないかと思う。

(2) 改善すべき点

Windowsでの使用法と、Macでの使用法の詳しい違いをもう少し記すべきだったように思う。Macでのプログラミング演習の授業は今

年から始まるものだったため、私自身も使い方に慣れておらず、苦労した。しかし、プログラミングを学ぶ大半の人が未経験であるのと同じく、Macを使用する人物もまた少ないため、作業に慣れるにはやはり時間がかかる。

Windowsを使用する場合のKnoppixの起動法も載せるべきか考えたのだが、Knoppixのデータが入ったCD-ROMが必要であることと、これから先はMacでのプログラミング演習の授業が続くであろうと考えたので、それは省くことにした。

解答の文字コードは、すべてWindows用のeuc-jpからMac用のutf-8に書き換えて改良をしたのだが、プログラムの解答は別のパターンも載せるべきであった。模範解答以外にもこんな答え方がある、という説明もあったほうが、より知識力を高める結果になったように思うからである。

(3) これからの課題

復習を行っている時点で、気になる部分があった。演習プリント9にある、『wc.pyのオプション引数処理』、『複数ファイル処理可能なwc.py』という項目の解答がないことである。私たちが3回生のときも、今年度のプログラミング演習の授業でもこの内容は省略されていたためなのだが、問題文をよく読むとこれは応用問題なので、授業で習っていない部分であっても解答の制作は行うべきであった。

教材にある練習問題についても、BMI値を使った計算の例が多く使われていたが、もう少し他に計算問題の例を増やしたほうが、プログラム自体に慣れるかもしれない。ただし、あくまでもこれは初心者向けの教材であるということを前提に作成したので、あまりにもその意思から

かけ離れたものにならないよう、注意が必要である。

そして、マニュアルを見てアンケートを書いてもらった人物は、プログラミング演習の授業をある程度進めた3回生であったため、本当に初めから習う者の意見が聞けなかったことがとても残念である。しかし制作時期が時期なので、4月からこの教材を使ってもらうことは不可能であった。実際に、プログラミング演習が始まった時点でこの教材を使用してみて、その度に意見を聞くことができれば、もっと充実した教材を作成することが出来るかもしれない。

今回はマニュアルとその解説を中心に作成したが、問題集や参考書のようなものをセットで作ることができれば、力を付けることができるのではないだろうか。他のマニュアルの考察でも述べたが、参考にするためのマニュアルやサイトの内容は初心者を対象にしたものではないので、問題を解くことが困難なものが多い。

基礎の問題集を新たに作ったとして、たとえばこれから先、福田ゼミに入りたいと希望する2回生に、今回のマニュアルとセットで配布して、基礎力を十分に高めてから授業に挑んだりすることもできるのではないだろうか。これは、今後の自分にとっても良い内容であると思うので、制作の構成を練ってみようと思う。

(4) 最後に

プログラミング演習ではとにかく、「習うより慣れろ」という言葉を元に、授業を受けて欲しい。問題を解けば解くほどに力がつく授業なので、ひとつ分からないところがあれば、その先もずっと分からないままになってしまう。そのためにも、今回制作した教材を参考にしてほしい。この制作物は、本当にプログラミングで扱う Python の基礎の部分

——プログラミング演習用教材の作成について——

のみをピックアップして作成したものなので、使わない、という機会はないはずである。

もし、この先、先生の授業でこの教材を使用するという機会があれば、この制作物がプログラミング演習を受ける人のために役立つ、という目的を果たす。完璧な出来とは言えないかもしれないが、自身のスキルアップや読解力を高める結果に繋がれば、教材を制作した意味がそこにあると、私は考えている。

注

- (1) <http://www.python.jp/Zope/intro>
- (2) <http://www.python.jp/doc/release/tut/>
- (3) http://www.python.jp/Zope/intro/instant_hacking.jp
- (4) <http://www.tohoho-web.com/>

文献表

アラン・ゴールド 著 松葉泰子 訳

2001 『Python で学ぶプログラミング作法』ピアソン・エデュケーション
シヨン