

## 07

慶應義塾大学 SFC における  
初年次プログラミング教育

服部隆志(慶應義塾大学環境情報学部)

## SFC における情報教育

慶應義塾大学湘南藤沢キャンパス (SFC と略す) には総合政策学部と環境情報学部があり、1990 年に設立された<sup>☆1</sup>。両学部ともに文理融合学部であり、開設当初から一貫して、外国語と情報技術がカリキュラムの重要な柱である。

SFC の特徴として、教育内容が大変幅広く、政治、経済、社会、技術、環境、芸術などありとあらゆる分野に対する志向を持った学生が入学してくるということが挙げられる。情報教育は、そのような多種多様な学生に共通の知的基盤として位置付けられている。

初年次の情報科目の内容は、コンピュータ・リテラシーとプログラミングである。リテラシーについては、初期のころはキーボードとマウスの使い方から始まって、テキストエディタや電子メールなど、「文房具として使える」ことが目的であったが、パソコンが家庭に普及するにつれ、情報科学の基礎(アナログとデジタル、情報量、コンピュータ・アーキテクチャ、ネットワークなど)と、HTML、CSS を使用した Web ページの制作が大部分を占めるようになった。

PC 教室の定員の制約から、1 クラスあたり 40 名未満のクラス制をとっており、春入学が 26 クラス、秋入学が 3 クラス程度である。共通教材を用意しており、原則として全クラスで同じ内容の授業を行うが、クラス担当者によっては独自教材を使用する場合もある。共通の期末試験を実施していた時期もあるが、現在は小テストと課題のみで、成績はクラ

スごとに決定する。

各クラスには学生アシスタント 2～3 名を付けている。以前は希望者が多く、ほとんどのクラスに 3 名配置できたが、最近は希望者が少なくなり、3 名のクラスもあれば 1 名しかいないクラスもあるという状況である。開設のころはコンピュータ教育に非日常的な魅力があり、積極的に参加しようという雰囲気があったが、パソコンが日用品になるにつれてそのような雰囲気は薄れてきた。

## 初年次科目の目標

技術者志望ではない学生が多い状況で、何を目標としてプログラミングを教えるかは難しい問題である。SFC では、プログラミングは分野を問わない共通の基盤であるという理念に基づき、すべての学生がプログラムを書けるようになること(具体的には、初年次科目では繰り返し、条件分岐、関数を含んだプログラムが書けること)を目標として授業を行っている。

しかし、実際問題として 1 年間の授業だけで全員が実用的なプログラムを書けるようになるわけではなく、また、多くの学生は必修科目を修了すればプログラミングの学習をやめてしまうのが現実である。それでも選択科目ではなく必修科目として学生全員にプログラミングを学習させることの意味は、次のようなものであると考えている。

- 論理的思考力の訓練を行う。一論理的に議論を展開する能力の訓練のためには、曖昧さの許されない形式言語を使うことが有効であると考えられる。
- 情報システムの原理的な動作を体験する。一あ

<sup>☆1</sup> 2001 年に看護医療学部も設置されたが、カリキュラムは独立しているため、本稿では触れない。



総合政策	環境情報	理工	経済	商
22.1%	30.3%	27.4%	12.6%	14.3%

表-1 2014年度就職者の中で情報通信業が占める学部ごとの割合

科目名	単位数	時間数	内容
情報基礎 1	2	週 180 分・半年	情報科学の基礎, HTML, CSS, JavaScript
情報基礎 2	2	週 180 分・半年	ネットワークの基礎, JavaScript

表-2 初年次プログラミング科目

らゆる分野で情報技術が応用されるので、どのような分野でも「情報技術を使えばこういうことができそうだ」という想像力が重要になる。そのためには、ブラックボックスとして捉えるよりも、ある程度中身が分かった上で発想する方が有利である。また、システムを発注する側に立った時、技術者とのコミュニケーションが容易になることも期待できる。

- プログラミングの才能を発掘する。一大学で「必修だから」という理由で初めてプログラミングを学んで興味を持ち、それから技術者や研究者の道へ進んだ例が数多くある。

上のような目的が本当に達成されているのかという判断は難しいが、1つの指標として、就職者の中で情報通信業の占める割合を慶應義塾の他学部と比較すると表-1 のようになる。もちろんプログラミング教育だけで左右されるわけではないが、どちらかという文系色の強い総合政策学部でも理工学部に近い数字になっているので、特に才能発掘という点ではある程度貢献できているのではないかと考えている。

## ..... プログラミングの動機付け .....

SFC では通年科目が存在せず、すべての科目は半年単位なので、初年次の情報科目も前半と後半の2科目になっている。開設当初からずっと前半科目でコンピュータ・リテラシー、後半科目でプログラミングという構成であったが、前回2014年度カリキュラム改定からは、前半と後半のそれぞれの科目でリテラシーとプログラミングを並行

して教えるように変更した(表-2)。それは、次のような考えに基づいている。

コンピュータが苦手な学生の授業に対する反応として、「これが何の役に立つのか分からない」というものがある。たとえば、HTMLとCSSの学習の際に、なぜマークアップ言語を使うのか、WYSIWYGで見た目そのままに書けばいいのではないかと思う学生が多く、その「何を意図しているか理解できない」状態が学習意欲を阻害していると思われる。マークアップ言語の利点を文章で説明することはできるが、それでもやはり実感としては伝わらないので、HTML、CSSとプログラミングを別々に学習するのではなく、1つの題材の中で関連付けて学習するようにした。CSSの属性を学んだ後に、JavaScript(JS)で操作して文字の色や大きさを変える例題をやってみると、マークアップという記号を使う必然性(プログラムで操作可能)が理解できるのではないかと期待している(図-1)。

逆にプログラミング学習の側から見ると、Web技術の理解という大目標があることが重要である。初心者向けのプログラミングの授業では、とにかくプログラムが書けるようになることが目標であり、それをどう使うかということはそれほど強調されないことが多く、そのことが学習意欲の低下につながっている可能性がある。

## ..... Web と プログラミング .....

初年次科目のテーマとしてWeb技術を取り上げたのは、次のような理由からである。

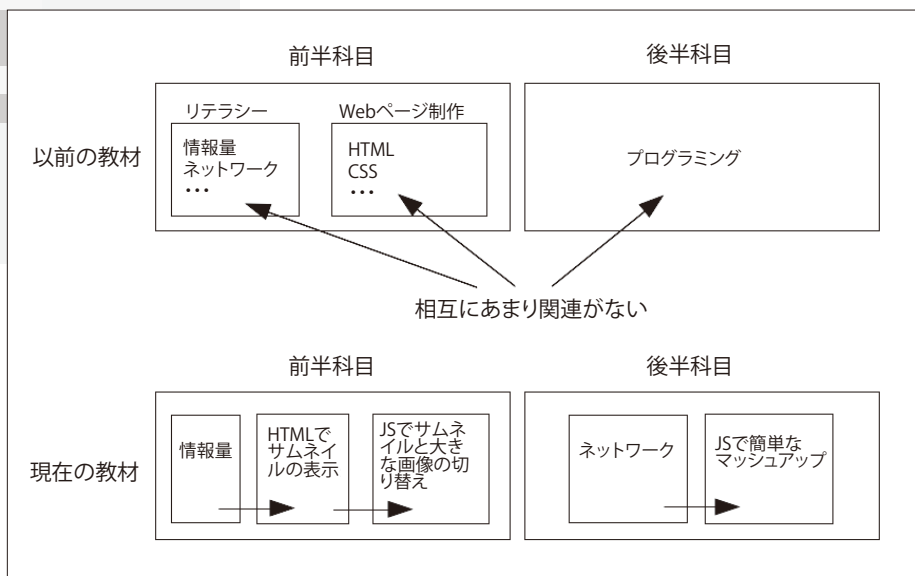


図-1 リテラシーとプログラミングの関連付け

- インターネットと Web は、伝統的に SFC が力を入れている分野である。
- 日常生活で使う機会が多く、学生にとって身近な題材である。
- さまざまな要素技術から成り立っており、広い範囲の基礎知識を教える初年次科目の題材として都合がよい。  
具体的には、プログラミングの単元では次のような例題を用いて導入を行っている<sup>☆2</sup>。
- 関数— Web ページのボタンを押した時の処理単位として関数を定義する。
- 場合分け— なぞなぞのページを作り、入力した答が正解かどうか判定する。
- 繰り返し— 画像の大きさを変化させたり、文字列を追加したりする関数を作り、ボタンを何回も押して表示がしだいに変わることを確認する。次に、while を使うとボタンを何回も押さなくても繰り返し実行できることをやってみる。  
「情報基礎 1」（初年次前半の科目）の学期末課題は「動きのある Web ページを作る」というものであり、学生が自分で選んだテーマの Web ページを作る際に、HTML と CSS だけでは実現できない動

的な表現を JavaScript プログラムで書かなければならない。これによって何のためにプログラムを書くかということが明確にできると考えている。

## ..... プログラミング環境 .....

プログラミングの授業において、教える内容だけでなく、教えるための環境も考慮すべき非常に重要なポイントである。

初心者を対象とした授業でどのような環境を使用するかは、情報系教員の間でも意見が分かれるところである。あくまで筆者の私見であるが、次のような分類ができるのではないだろうか。

- 原理主義— 基本的な概念の理解を目標とし、そのためにできるだけシンプルな開発環境、あるいは教育用に設計された言語を用いる。これはさらに伝統的なプログラミング教育を踏襲する正統派と、Scratch などのビジュアルプログラミング言語や、アンプラグドの手法を使う改革派に分かれる。
- 世俗主義— 実用的なプログラムが書けるようになることを目標とし、実際のソフトウェア開発で使われているものにできるだけ近い開発環境を用いる。なるべく早い段階で実用的な題材を

<sup>☆2</sup> <https://itclass.sfc.keio.ac.jp/text/info1-2015-9/>, <https://ipl.sfc.keio.ac.jp/text/info2-2015-9/>



扱うために、詳しい説明抜きでライブラリやフレームワークを「おまじない」として使うことが多い。

現在の教材は、題材については世俗主義寄りとなっている。ただし IDE は使用せず、テキストエディタ<sup>☆3</sup>でプログラムを書き、ブラウザで実行するというスタイルである。この点については原理主義的で、プログラミングに限らず、まずプレーンテキストによる記述に慣れてもらうことを考えている。また、ブラウザに開発ツールが備わっているため、デバッグに関してはわざわざ別に IDE を教えなくても同等のことができるという理由もある。

## ..... 今後の課題 .....

初年次科目に Web 技術を理解するという大目標を設定し、その中でプログラミング教育を行うという方式は、まだ 2 年目であるが、次のような問題が起きている。

第 1 に、たくさんの新しい概念が次々と出てきて、それを組み合わせなければならないので、認知的な負荷が大きくなる。特に、Web ページを書くために HTML, CSS, JavaScript という 3 種類の言語を習得しなければならないところが難しく、混乱してしまう学生が多い。

第 2 に、前半科目の最終課題は「動きのある Web ページを作る」というものであるが、Web ペ

☆3 mi, emacs, atom が使用可能である。共通教材はまだ mi を標準としているが、今後は atom への移行を考えている。

ージが作ればよいという発想で、プログラムは既製のものをコピーして済ませる学生がいる。コピーはあらゆる科目で問題になっているので、この方式にしたことが原因とは限らないが、プログラムだけの課題のときには完全コピーはほとんどいなかったことを考えると、プログラミングは道具であるという位置付けで授業を行うことが、コピーでよいという意識を助長している可能性があるため、検討が必要である。

カリキュラム改定前後で小テスト得点を比較した教員からは、中間層が減って二極化したという指摘があった。動機付けが成功した学生は、通年でプログラミングを学習することにより理解度が増したが、第 1 の問題で早々に落ちこぼれた学生も増えていると推測される。対策として、後半科目の最初の部分で、前半科目の復習を強化している。

将来、初等・中等教育でプログラミング教育が普及するとすれば、大学におけるプログラミングの授業は、言語の文法や基本的なアルゴリズムを教える必要性が減るので、より目的指向を強めることになると思われる。SFC では今は Web 技術という情報科学の枠内での目標設定であるが、ほかの分野と関連付けた授業（たとえば、ビッグデータの解析という目標設定で、統計学とプログラミングを並行して教える授業）も検討する価値があると思われる。

(2015 年 12 月 29 日受付)

服部隆志（正会員） ■ hattori@sfc.keio.ac.jp

1990 年京都大学大学院理学研究科博士課程研究指導認定退学。1993 年理学博士号を取得。1994 年より慶應義塾大学に勤務。ビジュアルプログラミング、実演によるプログラミングなどに興味を持つ。

.....