

一般情報教育におけるプログラミング教育 のあり方について

河村一樹[†]

本稿では、大学の一般情報教育に関するプログラミング教育がどうあるべきかについて論じる。「手順的な自動処理」を体験する場としてのプログラミング教育という視点から、JavaScriptを用いてどのように実現するか、留意点や教育事例を交えて述べる。

About the way of Programming Education in General Education

Kazuki Kawamura[†]

In this paper, I discuss about the way of programming education regarding the general education of the university. From the view point called programming education as the place of experience “procedure automatic processing”, I express about the method of programming education using JavaScript, including the consideration point and the educational case study.

1. はじめに

情報専門向けのプログラミング教育については、コンピュータサイエンス領域を基盤にした形でカリキュラムがすでに確立されており、良質な教科書も数多く出版されている。一方、非情報向けのプログラミング教育については、そもそもその必要性の有無から始まり、過去から紆余曲折してきており、確固たる指針が出されているとは言い難い。そこで、本稿では、一般情報[処理]教育におけるプログラミング教育がどうあるべきかについて、筆者の教育実践を踏まえて考察する。なお、処理に括弧をつけているのは、一般情報処理教育（処理を付けた場合）と一般情報教育（処理を付けない場合）の二つの言い方があることを明示するための表現である。

2. 一般情報[処理]教育の変遷

ここでは、一般情報[処理]教育に関する調査研究活動が、どのように進められてきたのか、その変遷について取り上げる。

2.1 ICT 環境の変容

情報に関する教育では、学校における ICT 環境がその基盤となることから、ICT における技術革新に大きな影響を受けることになる。その技術革新の流れを概観すると、コンピュータ産業の誕生→メインフレームの時代→パソコン時代の幕開け→インターネットの時代→ソフトサービス産業の広がりとなる¹⁾。この潮流をいくつかの観点から見直すと、表 1 のようになる。

このような ICT の潮流に応じて、一般情報[処理]教育の環境についても変化が生じてきた。それは、メインフレームによる集中処理から、サーバー/クライアントによる分散処理への移行である。それとともに、教育内容については、プログラミングだけでなく、アプリケーションソフトウェアや Web ソフトウェアの利用と多様化が進んできた。

2.2 情報[処理]教育研究集会の活動

1982 年に、国立大学情報処理教育センター協議会が発足した。これは、1972 年から 1981 年にかけて、10 大学（北海道大、室蘭工業大、東北大、東京大、名古屋大、名古屋工業大、京都大、大阪大、九州大、九州工業大）に情報処理教育センターが設置され、それらをメンバーにして組織化されるに至った。この時代は、情報処理教育センターに設置されたメインフレームによる ICT 環境が整備されていたことから、一般情

[†] 東京国際大学 商学部
Tokyo International University, Department of Commercial Science

表 1. ICTにおける技術革新の変容

変容	その概要：旧から新へ
コンピュータシステムのダウンサイジング化	○メインフレームコンピュータから、各種コンピュータ(ミニコン, オフコン, WS, さらには, PC) へ
ネットワークによる分散化	○メインフレームコンピュータを中核としたバッチ処理やTSSといった集中処理から、ネットワークインフラのもとでのCSSといった分散処理へ
プログラム言語の多様化	○低級言語から高級言語へ ○手続き言語から非手続き言語(オブジェクト指向化)へ ○プログラム言語から、マークアップ/スクリプト言語へ
ソフトウェア利用の拡大化	○自作プログラムから、出来合いのアプリケーションソフトウェアの利用へ ○有償ソフトウェアから、無償ソフトウェア(フリーソフトウェア) へ ○仕様未公開から、公開(オープンソース)へ

報処理教育においてもプログラミングに関する授業が中心だったといえる。

1988年には、九州工大において、文部省主催(ただし、2003年まで)という形で、第1回情報処理研究集会在開催された。このときの参加者の範囲については、

『情報教育(情報を専門とする学科の専門科目の授業を除く)を担当する教職員』と記載されている²⁾。これより、情報系の専門学部・学科以外のすべての学生に対する情報処理教育(つまり、一般情報処理教育のこと)が重視されるきっかけとなったといえる。そして、これ以降毎年、各メンバー校を会場にして研究集会在開催されており、2010年には京都大学で第23回が開催された。

なお、2006年には、「情報処理教育研究集会」から「情報教育研究集会」に改称された。このことは、一般教育においても、それまでの「処理」を主体としたプログラミング教育だけではなく、コンピュータリテラシー教育や情報に関連した教育(インターネット教育、マルチメディア教育、情報倫理教育、情報学教育など)などまで範囲を広げたことを意味している。

2.3 情報処理学会における委員会活動

情報の専門系学科における情報処理教育については、以前から調査・研究が行われてきた³⁾。情報処理学会でも、文部省から委託を受けて「大学等における情報処理教育の改善のための調査研究」による報告書も発刊した⁴⁾。

その報告書の中で、一般情報処理教育の現状と問題点が指摘された。情報処理とそれによる問題解決の原理を理解することは、一般教育の段階においても極めて重要であるとした上で、その教育内容については、プログラミング言語によるプログラミン

グが、利用可能な非プログラミングソフトをただ使っているだけであり、いずれの場合も実際の操作手順を修得させることだけに追われていると指摘した。これより、一般情報処理教育として「何を」「どうやって」教えるべきかについてのしっかりとした背景がないことを言及した。

1990年には、情報処理学会の大学等における情報処理教育検討委員会が発足し、コンピュータサイエンス教育を検討するCSWG(Computer Science Working Group)、情報システム教育を検討するISWG(Information System Working Group)、一般情報処理教育を検討する一般WGがそれぞれ設置され、報告書を発刊した⁵⁾。ただし、この中ではCSWGの検討結果が中心に記載され、ISWGと一般WGについてはそれぞれ検討課題だけが記載された。

1991年には一般情報処理教育の実態に関する調査研究委員会が発足し、1992年からは大学等における一般情報処理教育の在り方に関する調査研究委員会に継承され、一般情報処理教育に関する調査研究が行われた⁸⁾⁹⁾。ここでは、初めて一般情報処理教育のためのカリキュラムが策定され公開された。

2001年には、情報処理教育委員会の下部組織として、一般情報処理教育委員会が常置されるとともに、その委員会から報告書⁷⁾⁸⁾が発刊された。ここでは、全国の高等教育機関に対して一般情報処理教育の実態調査を行うとともに、一般情報処理教育のカリキュラムを策定した。そのカリキュラムについては、必修としての「情報とコンピューティング」(半期、2単位)、および、必修に準じる「情報とコミュニケーション」(半期、2単位)を設置することを提案した。これは、実態調査の結果、ほとんどの大学では、一般情報処理教育に相当する共通科目を1科目しか設置していなかったことによる。このため、2科目分開講することが時間的に難しいという実状から、「情報とコンピューティング」だけを必修扱いとするとともに、「情報とコンピューティング」には「情報とコミュニケーション」の一部分を含むこととした。

2007年には、J07プロジェクトの一つとして、一般情報教育の知識体系(GEBOK: General Education Body of Knowledge)を発表した¹⁰⁾。ここでは、GEBOKの教育時間数を、4単位分(90分×15回×2半期)に設定するとともに、コンピュータリテラシーについては必要な場合だけ補講として実施することとした。なお、2008年には、2.2と同様に、一般情報処理教育委員会は一般情報教育委員会に改称した。

2.4 出版活動

2001/2年に策定した一般情報処理教育カリキュラムを、全国の大学において認知してもらい利用してもらうためには、報告書の発刊だけでは不十分と言わざるを得ない。そこで、一般情報処理教育カリキュラムに準拠した教科書を発刊することで、できるだけ多くの人々にカリキュラムの内容を理解してもらうこととし、情報処理学会教科書編集委員会からITテキスト一般教育シリーズとして発刊する許諾を得た。その結果、

「情報とコンピューティング」¹¹⁾と「情報と社会」¹²⁾(科目「情報とコミュニケーション」)に対応)を発売するに至った。

その後、発刊後数年が経過したことから、現在改訂作業に入っている。改訂新版とするとともに、書名をそれぞれ「情報とコンピュータ」「情報とネットワーク社会」と改称する。これによって、旧版より新しい内容を網羅していること、重複部分を削除して技術分野系と社会分野系にそれぞれ統廃合すること、GEBOK の授業時間数(44時間分)を踏まえて2科目4単位開講を推奨することを提案する。

3. プログラミング教育の変容

ここでは、一般情報処理教育から一般情報教育への変遷の中で、プログラミング教育がどのように変容してきたかについて取り上げる。

3.1 旧来のプログラミング教育

一般情報処理教育委員会が発足する以前におけるプログラミング教育では、メインフレームコンピュータによる実用言語(たとえば、FORTRAN や BASIC など)でのプログラミングか、徐々に普及が始まったパーソナルコンピュータに搭載された BASIC インプリタによるプログラミングが主流であった。この頃は、何ならかの目的でコンピュータを利用するためには自分でプログラムを作成して実行する手立てしかなかったため、情報を専門としない利用者にとってもプログラミングの技が必要とされた。

各大学では、学内に情報処理教育センターを設置し、一般教養課程の自然科学系列の科目の一つにプログラミング関連の科目を選択として開講するところも多かった。その教育内容については、実用言語の構文や文法の習得を目的としていたといえる。

3.2 その後のプログラミング教育

1992/3年における一般情報処理教育カリキュラムの策定と公開は、一般情報処理教育の在り方について新たな潮流を生んだといえる。それは、特定の実用言語の習得だけを目的とする旧来のプログラミング教育に対する批判とともに、新しい「プログラミング」教育(かぎ括弧付き)の在り方として、一般情報処理教育においてもコンピュータサイエンスの基礎(例えば、頻出概念など)を理解させること、プログラミングを通してコンピュータの動作原理や仕組み(逐次制御、プログラム内蔵)といったことについても学ばせること、コンピュータ利用の可能性と限界を把握させることなどが提言されたことである。

この頃になると、各大学にパーソナルコンピュータが相当数導入されるとともに、学内 LAN による運営も始まりつつあった。それとともに、パソコンの OS が CUI か

ら GUI に移行し、さまざまなアプリケーションソフトウェアが利用できるようになってきたことから、一般情報処理教育で扱う内容も多様化しつつあった。それとともに、プログラミング教育では、実用言語によるプログラミングの技を習得することが目的ではなく、問題解決能力や論理的思考力の育成を目指すといった視点も組み込まれるようになった。

扱うプログラミング言語については、実用言語でなくてもよいとしたが、その当時では一般情報処理教育に適した教育用言語がなかったことから、情報専門系学科で使う Pascal や C 言語などが流用された。ただし、あくまでも一般情報処理教育として必要となる基礎的な言語仕様に限定し、ポインタや構造体といったデータ型や再帰呼出しなどは除外するという使い方であった。この具体的な取り組みの一つに、Basic C の提案がある¹³⁾。

なお、C 言語については、パソコンで稼働するフリーのコンパイラ(LSI C-86 など)やその開発環境(CPad for LSI C-86)が自由にダウンロードして使えるようになったことから、各大学での利用が増加したといえる。

3.3 現在のプログラミング教育

2005年に情報処理学会情報処理教育委員会から提言2005が発表され、その中で、小中高等学校および大学それぞれの発達段階に応じて、適切な「手順的な自動処理」を体験させることを提言した¹⁴⁾。これに合わせて、2006年には「教育用プログラミング言語に関するワークショップ2006」が開催された¹⁵⁾。また、この頃、アルゴリズムの教育に重点を置いた学習支援ツールもいくつか開発された。これらは、いずれも現在のプログラミング教育の実状を表しているといえる。

(1) 手順的な自動処理

手順的な自動処理とは、問題を定式化し、その解決方法をアルゴリズムとしてくみ上げ、何らかの方法(プログラミングだけでなくスクリプトやマクロの利用なども含む)によってコンピュータ上で実行し、その結果を検証し、必要ならばもとに戻り再実行するという一連の流れを示している。この手順的な自動処理を、初等中等高等教育機関それぞれにおいて体験させることを提言している。その中で、大学については、

『すべての学生に高等教育の水準にふさわしい情報・情報処理の理解を持たせ、わが国全体としての ICT に対する理解水準を底上げすること、および、その中でも ICT に関する適性と関心を持つ学生に対してはその適性を伸ばし、各分野における ICT 人材となれるように育成するために必要である。』
としている。

以上より、コンピュータの本質が手順的な自動処理であることを、学生に身をもって体験させるためには、コンピュータ上で実際に実行させることが必要であるとともに、大学では一般情報教育における実践が最も適しているといえる。

(2) 教育用プログラミング言語

現在、大学の一般情報教育でも使えるようないくつかの言語が登場してきている。

簡易的な言語としては、スクリプト言語があげられる。これは、通常の実用言語よりも簡単にプログラミングができるようになっており、JavaScript, Perl, Ruby, PHPなどがあげられる。また、ビジュアルプログラミングを指向し、BASICの流れを継承した Visual BASIC などもある。

教育用の言語として開発されてきたものに、Squeak eToy, ドリトル, なでしこなどがあげられる。これらの言語では、オブジェクト指向の概念を取り入れたり、日本語による記述も可能であり、いずれも初学者のための入門用言語といえる。

(3) PEN (Programming Environment for Novices)

PENでは、手続き的なプログラミングが容易に習得できるようにした学習環境を提供する¹⁶⁾。もともとは、大学入試センターの「情報関係基礎」で用いられている手順記述言語 DNCL¹⁷⁾、および、DNCLのサブセットである TUATLE¹⁸⁾に着目し、それらを用いてプログラミングが体験できる学習環境を目指したといえる。初学者を前提としているため、入力支援機能や実行状態表示機能なども備えている。

(4) JPADet (Japanese Problem Analysis Diagram editor and interpreter)

JPADetは、日本語(変数名や関数名など)と構造化チャートの一つであるPADを用いてアルゴリズムを記述するとともに、インタプリタにより実行し、結果を画面に表示するという一連の学習を支援するためのシステムである¹⁹⁾²⁰⁾。このため、プログラミング言語に依存することなくプログラムの実行ができることから、アルゴリズムの学習に専念することができる。

4. 一般情報教育におけるプログラミング教育

3で取り上げたプログラミング教育の変容を踏まえた上で、これからの一般情報教育におけるプログラミング教育のあり方について論じる。

4.1 前提となる方針

これからのプログラミング教育を考える上での基本方針として、3.2で取り上げた「プログラミング」教育(かぎ括弧付き)の考え方を継承することを前提とする。その上で、3.3で取り上げた「手順的な自動処理」を体験する場としてのプログラミング教育のあり方について考察する。

4.2 教育目標の設定

一般教育という観点からみて、プログラムに関する専門的な理論や知識、あるいは、

高度なプログラミングテクニックなどを習得することではなく、あくまでも「手順的な自動処理」の体験から得られる学習効果を会得することを目標に設定すべきである。

その学習効果としては、プログラミングを通してアルゴリズムをデザインするための論理的な思考力が育まれること、解決すべき問題をアルゴリズムに置き換え定式化することによって抽象化能力が身につけられること、プログラムを実行することによって自分で作り上げたアルゴリズムの良し悪しが判断できること、プログラミングを実体験することによってコンピュータサイエンスの基礎的な概念(頻出概念)を直感的に把握できること、などがあげられる。

4.3 プログラミング言語の選択

4.2の教育目標を実現するためのプログラミング言語として、いくつかの言語について検討する。

(1) 手続き型言語

教育用に使われてきた手続き型言語には、情報専門系向けにPascalやC、あるいは非情報系向けにBASICがあげられる。

BASICは、データ型が実数と文字列だけ、予約語の大文字と小文字を区別せず、算術演算子以外の記号は極力使用せず、代入文に「LET(何々せよ)」をつけることで意味がわかりやすい(左辺の値を右辺の値と同じにせよ)、命令文は改行で区切る、対話型の編集・実行環境、などといった特徴があることから、初学者向けのプログラミング言語といえる。このため、高等学校数学の教科書や大学センター試験に取り上げられたり、高等学校の授業で十進BASICが使われたりした。現在では、ビジュアル環境に対応したVB(Visual BASIC)やVBA(Visual BASIC for Applications)、完全にオブジェクト指向化したVisual BASIC.NET, REALbasic, ActiveBasicなどが登場している。

これらの中でVBは、Windows環境を施設した大学で部分的に使われた実績はあるが普及に至ってない。手続き型言語はいずれも実用言語に直結していることから、一般情報教育にはあまり向いていないといえる。

(2) オブジェクト指向言語

オブジェクト指向言語は、Smalltalkがその元祖といえるが、プラットフォームに依存しないJavaの登場によって急速に普及したといえる。いずれも、ローエンド(組込みシステム)からハイエンド(大規模情報システム)まで幅広く使われている実用言語である。

また、最近では、初学者に対してオブジェクト指向ベースのプログラミング教育を実施する試みが報告されている²¹⁾。教育用のオブジェクト指向言語(たとえば、ドリトルなど)では、画面上のオブジェクトを操作する動作を、オブジェクトのメソッドとして記述することで、クラス定義を意識せずに、直感的にプログラミングができるような工夫がなされている。

これより、大学の一般情報教育においてオブジェクト指向言語を使うことも考えられるが、現在でもそれほど普及しているとは言い難い。その理由として、そもそもなぜオブジェクト指向プログラミングである必要があるのかがはっきりしてない、教員がオブジェクト指向の概念を理解できてない、手続き型言語のような定番的なプログラミング教育法が確立されていない、などといった課題があげられる。また、オブジェクト指向では、情報隠蔽やデータ抽象といった概念が導入されるが、これらがコンピュータのブラックボックス化を生み出し「手順的な自動処理」をイメージしにくくさせることもある。

(3) スクリプト言語

スクリプト言語は、既存のプログラミング言語のように詳細に処理手順を記述するのではなく、台本 (script) のように比較的簡単に記述できることから、簡易的なプログラミング言語として位置づけられている。

もともとは、IBM 社のメインフレームコンピュータのバッチ処理で使われていた JCL (Job Control Language) がその原型といえる。現在では、Web ブラウザ用のスクリプト言語として、JavaScript、Internet Explorer のみで動作する VBScript、Firefox のみで動作する XUL (XML User Interface Language) などがある。また、Web サーバ用のスクリプト言語として、PHP、JSP、ASP、Ruby on Rails などがある。これらのスクリプト言語は、その簡便さという点において一般情報教育のプログラミング教育に適していると言える。つまり、学習者は、短期間に、一連の「手順的な自動処理」を実験することが可能になる。

4.4 教育用言語としての JavaScript

筆者は、スクリプト言語である JavaScript に着目した。JavaScript は、プロトタイプベースのオブジェクト指向スクリプト言語であるが、手続き型言語 (C 言語に似た) の記述スタイルも備えている。Web クライアント上で実装され、RIA (Rich Internet Application) によって Web ページの表現に動的な振る舞いを与えることができる。

もともとは、ネットスケープコミュニケーションズ社の Brendan Eich により開発され、Netscape Navigator2.0 に実装された。その後、マイクロソフト社の Internet Explorer3.0 にも移植され、Web ブラウザに表示される仕組みができたことから、JavaScript は急速に普及することになった。ただし、当初は、Web ブラウザ毎の機能に差があり、その互換性が問題になった。このため、現在では、JavaScript の基本部分について Ecma International により、ECMAScript として標準化されている。

(1) JavaScript の特徴

JavaScript は次のような特徴をもつことで、一般情報教育のプログラミング教育に適していると言える。

- OS に依存せず

Web ブラウザが OS の違い (Windows, MacOS, Unix など) を吸収することから、どのコンピュータでも稼働することができる。多くの大学では、コンピュータ演習室にいろいろな OS を搭載していることから、実習においても使い勝手がよい。

- 特別な開発環境を必要とせず

JavaScript のソースコードを編集するためのテキストエディタ (メモ帳や TeraPad など) と Web ブラウザがあれば、動作環境が揃うことになる。このため、特定のソフトウェアプロダクト製品やアドインソフトを用意する必要がなく、大学のコンピュータ演習室における諸作業 (ソフトウェアのインストールやセッティングなど) が不要となる。また、購入コストもかからず、ライセンス契約なども不要である。

- スクリプトの動作確認が簡単

Web ブラウザに JavaScript の処理系 (インタプリタ) が組み込まれていることから、テキストエディタで修正後、Web ブラウザを起動することによって、すぐにスクリプトの動作確認ができる。また、Web ブラウザにはスクリプトのデバッグ機能も標準で搭載されていることから、簡単に構文エラーなどを検出できる。これより、各大学において、JavaScript を用いた実習授業が実施し易いといえる。

- 学習者が興味を持つ題材を提供

情報を専門としない学生に対して一連のプログラミング教育 (変数・配列の宣言、基本制御構造、関数、探索・整列アルゴリズム) を行うと、プログラマになるわけでもないのに何のために学習するのかわからずやる気がでないという意見が出てくる。これに対して、JavaScript では、上記の内容だけでなく、Web ページの見栄えをよくする動的な表現もできるので、学生にとってもおもしろく関心が高くなる。

(2) 初等中等教育での利用

JavaScript は、初等中等教育におけるプログラミング言語としても注目されている。

- 教育用プログラミング言語に関するワークショップ 2006

ここでは、JavaScript について、

『Java については、オブジェクト指向を教育できる。JavaScript については、特定のブラウザに依存しない、という利点がある。どちらも、GUI・インターネット時代に登場した言語であり、初等中等教育への適用という視点で興味深い。』としている²⁾。

- 新・試作教科書

情報処理教育委員会の初等中等教育委員会が作成した高校生向けの新・試作教科書では、「情報Ⅱ」の第5章「コンピュータとプログラミング」において、「5.2 JavaScript によるプログラミング」を扱っている。この中で、JavaScript については、

『JavaScript は、Web ページに組み込んで実行するために作られた言語であり、一般的に使われているブラウザには JavaScript の言語処理系が組み込まれている。』

としている²³⁾。また、別の文献では、

『2章で扱うアルゴリズム的な題材には JavaScript のような「伝統的な」構文の方が向いている可能性があること、また早い段階から複数のプログラム言語（ドリトル）に接することは有益であるとの考えから、JavaScript を採用した。』

『Web ブラウザの中で動作するプログラムの形態を維持することで、ネットワークを通して動作するプログラムの学習に発展させたい。学習する言語が JavaScript であっても、Ajax 的な学習用のライブラリを用意することで、サーバとの通信を行い、データベースへのアクセスなどを実現することが可能である。このような体験を通じて、単に画面をマウスでクリックするだけであった Web 画面の仕組みを理解でき、自発的にインターネットの世界を理解することが可能になる。』

としている²⁴⁾。

● 高等学校検定教科書

高等学校教科情報の教科書でも、JavaScript を使ったプログラミングが記載されており、それらを内容分析した結果、

『19年度の教科書Cは、アルゴリズムの章で JavaScript のプログラムを10個掲載している。(中略) WEB ブラウザ上の GUI 部品の制御など、JavaScript が得意とする機能は使われていない。JavaScript を採用した理由は、おそらく「WEB ブラウザさえあれば、ほとんどの環境で動作する言語だから」ということであると推測される。教科書Bも、アルゴリズムを実現手段として、JavaScript によるプログラミングを実習させるようになっている。』

と述べている²⁵⁾。

以上のように、高等学校教科情報において JavaScript によるプログラミングが導入されていることにより、大学の一般情報教育においてもそのまま継承しやすいといえる。なお、2.4 で取り上げた「情報とコンピュータ」では、第5章「アルゴリズムとデータ構造」で取り上げたアルゴリズムを JavaScript でプログラミングしたものを付録として掲載している。

4.5 実施する上での留意点

一般情報教育において、JavaScript によるプログラミングを行う上での留意点もいくつかある。

(1) 半角と全角の相違

JavaScript のソースコードは、すべて半角英数字にしないと認識されない。全角は、ダブルクォテーションで囲んだ日本語によるテキスト文だけとなる。このため、キーワードだけでなく、括弧「()」や中括弧「{}」あるいはセミコロン「;」などは気をつけて入力しなければならない。これらのことが、初学者にとって面倒くさいという印象を与える。

それだけでなく、間違えて全角のスペースを文中に入れてしまうとエディタで見つけ出すのが困難となり、本来の「手順的な自動処理」を学ぶ以前の構文エラー探しに時間がとられてしまい、結果として学習意欲が低下することがある。このため、事前に注意するように指導する必要がある。ただし、このことを実体験することによって、自然言語とは異なる人工言語の特性を理解させることが可能になる。

(2) フォーマットは自由

JavaScript では、キーワードや変数や文字列などを区切るスペースやタブは、すべて無視する。このため、ソースコードは自由に記述できてしまう。このことから、見にくいプログラミングを量産する学生もいる。

そこで、ソースコードの記述において、字下げを徹底すること、1行には1文だけとする、などの注意をあらかじめしておく必要がある。あるいは、最初から専用のエディタ（たとえば、TeraPad など）を使う方法もある。

(3) 大文字と小文字は別もの

プログラミング言語によっては、大文字と小文字を区別しないものもあるが、JavaScript では、それぞれ別のもので認識する。たとえば、変数名については、アルファベットの並びが同じでも大文字と小文字では別の変数とみなす。また、スクリプトのキーワードは、小文字で記述しなければならない。これらについても、あらかじめ周知徹底させる必要がある。

(4) コメントの扱い

コメントは、ソースコードに行単位 (//) あるいは複数行 (/...*/) に付記するが、プログラムの実行には全く関係ない注釈文である。このため、無視する学生も多いが、スクリプトの見やすさを意識づけるためにコメントを使うように指導することも有益である。

(5) アルゴリズム指向か？ 実用指向か？

JavaScript は実用言語のような「伝統的」な構文 (if 文, if else 文, for 文, while 文, do while 文, 関数と再帰) が用意されていることから、手続き的なアルゴリズムを記述することに適している。このため、さまざまな「伝統的」なアルゴリズム (整列, 探索, 文字列処理, 簡単な数値計算など) を学習させるプログラミング教育ができる。このことは、「手順的な自動処理」を体験できることを意味していると言ってよい。

一方、JavaScript は、Web ページに動的な表現を組み込むことができる。具体的には、ウィンドウのスクロールやフェードインや揺れ、警告や確認のサブウィンドウの表示、文字の装飾 (フェードイン/アウト, グラデーション, スクロールなど) や編集, 背景色の変更, 画像の入れ替えや動的な表示, (オプション/ツリー/プルダウン/ポップアップ) メニューの表示などがあげられる。これらの機能を取り入れることで、見栄えの良い Web ページを作成するためのプログラミング教育ができる。学習者にとってもその実用性がよく理解できることから、学習の動機づけが高まると言ってよい。しかし、

JavaScript 特有のプログラミングテクニックに集中することは、アルゴリズムの学習よりも言語の学習になってしまうことから、4.2 の教育目標を達成することが難しくなると言える。

以上のことから、アルゴリズム指向を中心に一部実用指向を取り入れたプログラミング教育が妥当といえる。

4.6 実施例

本学科では、1年次の必修科目として「演習(1)」(通年1コマ、4単位)を開講している。これは、チュートリアル形式の演習であり、担当者毎に実施する内容も異なる。筆者は、当初、C言語によるプログラミング演習を行っていたが、ほとんどの学生がC言語を実用的に使いこなせるまで至らず基本的なプログラミングの習得で終わっていた。その後、JPADetによるアルゴリズム演習に全面的に切り替えたが、プログラミング言語に依存しないアルゴリズムの学習に対して、(とくにプログラマを指向する)学生があまり興味を示さないことが明らかになった。

そこで、昨年度から、JavaScriptを用いたプログラミング教育を実施している。前期はHTMLとCSSを用いてWebページのデザインを、後期はHTMLに組み込む形でJavaScriptによるプログラミングを扱っている。また、これに合わせて教科書を執筆した²⁶⁾。後期についてのシラバスは、表2のようになる。基本的には、伝統的なアルゴリズムを中心に、一部実用的なアルゴリズムを取り上げている。この中で特徴的な内容は、次の通りとなる。

第2回目では、プログラミングをする上で事前に知ってほしい内容について講義する。変数については、メインメモリ上のある領域にデータ(数値か文字など)を格納するための箱であること、変数名はその箱の先頭番地を修飾したものであることを説明する。代入文については、等価演算子と同じ「=」で表すがあくまでの右辺の値を左辺に代入することであること、メモリ上の制約として左辺の値は右辺の値に上書きされることから変数同士の値の入れ換えの場合は別の一時的な記憶用の変数を用意する必要があること、計算式の場合は右辺のすべての変数に値が設定されていなければ実行時に異常終了することを説明する。

また、繰り返し文については、プログラミング言語の表現上の制約として繰り返しを制御するための変数の初期値・繰り返し/脱出条件・増減値しか指定できず、時間の経過とともに変数の中身がどのように推移するかまでは表現できないので、自分でトレースするしかないと説明する。関数については、あらかじめ処理系が提供するもの(ビルトイン関数)と自分が作成するもの(ユーザ定義関数)があること、関連する一まとまりの処理を関数としてまとめることによってプログラムが見やすくなることを説明する。

表2. 後期のシラバス

授業回数	概要
第1回目	科目ガイダンス：プログラミング言語の歴史・種類、アルゴリズムとプログラミング言語、授業の進め方、評価方法
第2回目	アルゴリズム：変数の意味、代入文、繰り返し文、関数の扱い
第3回目	スクリプト言語：JavaScriptの特徴、HTMLとJavaScript、JavaScriptによるプログラミング
第4回目	変数と画面入出力のプログラミング：変数の宣言、prompt文、document.write文
第5回目	選択のプログラミング：if文、if else文
第6回目	多岐選択のプログラミング：switch文
第7回目	定数回繰り返し：のプログラミングfor文、break文、continue文
第8回目	前/後判定繰り返し：のプログラミング：while文、do while文
第9回目	配列のプログラミング：配列の宣言、初期化、値の参照、連想配列
第10回目	関数のプログラミング：関数の宣言、関数の呼出し、値参照、汎用関数
第11回目	探索のアルゴリズム：線形探索、2分探索
第12回目	整列のアルゴリズム：バブルソート、選択ソート
第13回目	実用的なアルゴリズム：ウィンドウ・サブウィンドウの表示
第14回目	実用的なアルゴリズム：文字の表示と装飾、背景色(画像)の変更
第15回目	アルゴリズムとプログラミング、コンピュータの動作原理(逐次制御、プログラム

第15回目では、4.1および4.2を実現することを目指して、アルゴリズムを記述したプログラムがコンピュータの中でどのように動作しているのかという仕組みを明らかにするためにプログラム内蔵や逐次制御について説明する。また、JavaScriptのプログラミングを体験して、コンピュータができることとできないことについても説明する。

5. おわりに

以上、一般情報教育におけるプログラミング教育のあり方についていくつかの視点から論じてきた。「手順的な自動処理」を体験させるという趣旨に基づき、プログラミング言語はJavaScriptを採用した。アルゴリズムについては、高等学校教科情報でも取り上げている探索や整列を扱うことにしている。ただし、JavaScript特有のWebページの動的な表現を取り込もうとすると、アルゴリズムよりも言語の構文の方に学習

の比重が移ることになるので、どの程度まで扱うべきか思案中である。

筆者が所属している情報処理学会一般情報教育委員会でも、プログラミング教育についてはいろいろと議論を続けてきた。その結果、我々が策定した GEBOK にも、改訂中の教科書にも、「手順的な自動処理」という意味でのプログラミングに関する内容を取り込んでいる。

今後については、一般情報教育委員会の Web サイト²⁷⁾において、さまざまな一般情報教育におけるプログラミング教育の事例を公開したいと考えている。

謝辞 日頃から情報処理学会の一般情報教育委員会活動に協力を頂いている委員の皆様には、謹んで感謝の意を表します。

参考文献

- 1) 情報処理学会歴史特別委員会編：日本のコンピュータ史，オーム社，2010年
- 2) 大阪大学担当校編：
<http://conf2008.isc.kyutech.ac.jp/wiki.cgi?page=%BE%F0%CA%F3%B6%B5%B0%E9%A5%BB%A5%F3%A5%BF%A1%BC%B6%A8%B5%C4%B2%F1>
- 3) 坂井利之，他編：情報工学の教育・研究，共立出版，1980年
- 4) 情報処理教育の改善のための委員会編：大学等における情報処理教育の改善のための調査研究，平成 63 年度教育改革の推進に関する研究委託 最終事業報告書，1989年
- 5) 大学等における情報処理教育検討委員会編：大学等における情報処理教育のための調査研究報告書（文部省委嘱調査），情報処理学会，1991年
- 6) 一般情報処理教育の実態に関する調査研究委員会編：一般情報処理教育の実態に関する調査研究（文部省委嘱調査研究），情報処理学会，1992年
- 7) 大学等における一般情報処理教育の在り方に関する調査研究委員会編：大学等における一般情報処理教育の在り方に関する調査研究（文部省委嘱調査研究），情報処理学会，1993年
- 8) 大学等における一般情報処理教育の在り方に関する調査研究委員会編：大学等における一般情報処理教育の在り方に関する調査研究（文部科学省委嘱調査研究），情報処理学会，2001年
- 9) 大学等における一般情報処理教育の在り方に関する調査研究委員会編：大学等における一般情報処理教育の在り方に関する調査研究（文部科学省委嘱調査研究），情報処理学会，2002年
- 10) J07 プロジェクト編：学部段階における情報専門教育カリキュラムの策定に関する調査研究（平成 19 年度文部科学省「先導的制度改革推進委託事業」報告書），情報処理学会，2008年
- 11) 川合慧監修，河村一樹編著：情報とコンピューティング，オーム社，2004年
- 12) 川合慧監修，駒谷昇一編著：情報と社会，オーム社，2004年
- 13) 阿部圭一著：C 言語によるプログラミング教育についての省察，情報処理学会研究報告，2009-CE-98(30)，pp.205-212，2009年
- 14) 情報処理学会情報処理教育委員会編：
<http://www.ipsj.or.jp/12kyoiku/proposal-20051029.html>
- 15) 兼宗進編：教育用プログラミング言語と授業利用，情報処理，VOL.48，No.6，pp.587-615，

2007年

- 16) 中村亮太，西田知博，松浦敏雄：プログラミング入門教育用学習環境 PEN，情報処理学会研究報告，2005-CE-81(10)，pp.65-71，2005年
- 17) 大学入試センター：センター試験用手順記述機能言語-DNCL-，平成 15 年度センター試験試験問題評価委員会報告書，pp.258-259，2003年
- 18) 中森眞理雄，中條拓伯，小谷善行，辰己丈夫，金子敬一，並木美太郎，品野勇治：平成 18 年度入試に向けての「情報」試行試験の実施報告(2)，情報処理学会第 46 回プログラミングシンポジウム報告集，pp.173-180，2005年
- 19) 斐品正照，徳岡健一，河村一樹：構造化チャートを用いたアルゴリズム学習支援システム，情報処理学会論文誌，Vol.45，No.10，pp.2454-2467，2004年
- 20) 河村一樹，斐品正照：よくわかる日本語 PAD によるアルゴリズム演習，日刊工業新聞社，2003年
- 21) 青木浩幸，崔淑敬，金顯哲，李元揆：オブジェクト指向プログラミングの一般情報教育における意義，情報処理学会研究報告，2008-CE-95(13)，pp.79-86，2008年
- 22) 並木美太郎，辰己丈夫，兼宗進，長慎也，久野靖，中野由章，西田知博：「教育用プログラミング言語に関するワークショップ 2006」の報告，情報処理学会研究報告，2006-CE-85(3)，pp.17-24，2006年
- 23) 情報処理学会情報処理教育委員会初等中等教育委員会編「新・試作教科書」：
<http://www.ipsj.or.jp/12kyoiku/teigen/v83joho-text0612.pdf>
- 24) 兼宗進，久野靖：「新・試作教科書」におけるプログラミングの扱い，情報処理学会研究報告，2007-CE-90(5)，pp.23-28，2007年
- 25) 長慎也，兼宗進，並木美太郎，西田知博，小井戸土政範：「情報 B」の教科書比較－「手順的な自動処理」の視点から，情報処理学会研究報告，2006-CE-85(8)，pp.53-59，2006年
- 26) 河村一樹：JavaScript よる情報教育入門，大学教育出版，2011年発刊予定
- 27) 情報処理学会一般情報教育委員会編：
<http://lecture.ecc.u-tokyo.ac.jp/~yamaguch/tmp/sigge/>