JavaScriptプログラミングと情報教養教育

JavaScriptプログラミングと情報教養教育

江澤　義典*

要　旨

高等学校で教科「情報」が必修科目として開始されたのは平成15年度からであった．平成18年度以降には，その教科「情報」を学んだ高校生が大学へ進学することになるので，大学における情報教育や情報処理教育への影響を検討する必要がある．文部省が作成している指導要領によれば，高校における教科情報においては，情報や情報処理の「科学的な理解」が重要な項目になっていることに着目すべきである．情報系の大学学部教育においても，情報や情報処理に関する常識を涵養することが「科学的理解」の本質であると考えると判り易いかもしれない．常識というものは様々な体験を通して獲得できるものであるから，情報や情報処理を科学的に理解するためには，コンピュータによる情報処理の原理を，具体的な操作で，身をもって体験することが必須になる．

本論文では，高等学校の教科「情報」を学習する段階で，ある程度のプログラミング教育が必要であることを示す．そして，そのようなプログラミングの基礎を学んで大学に進学した学生には，大学における情報教育や情報処理教育として，情報教養のカリキュラムがどのように設計可能になるかを考察する．

JavaScript Programming and Computer Literacy

Yoshinori EZAWA

Abstract

From fiscal year 2003, a prerequisite class named "JOHO(informatics)" was included in the curricula of every high school in Japan. Therefore, we must reconsider the contents of computer literacy classes for freshmen in universities starting in 2006. The Ministry of Education in Japan designed the curricula to provide every high school student with a kind of 'scientific sense of Informatics'. To acquire a scientific sense of informatics, it is considered essential to perform practical computer programming in the laboratory sessions.

In this paper, we discuss how to design the practical contents of a university's curriculum on computer literacy when almost all freshmen have already experienced some kind of practical programming course.

*関西大学総合情報学部

1 はじめに

高等学校で教科「情報」が必修科目として開始されたのは平成15年度からであった．したがって，平成18年度以降には，その教科「情報」を学んだ高校生が大学へ進学することになるので，大学における情報教育や情報処理教育への影響を検討する必要がある[1]．

文部省が作成している指導要領によれば，高校における教科情報においては，情報や情報処理の「科学的な理解」が重要な項目になっていることに着目すべきである[2]．さて，ここでいう「科学的」とはどのような理解構造を意図しているのであろうか．

一般的な科学のイメージは，『その恩恵は期待できるが近づき難い』というのが，いわゆる文系の人々の本音かもしれない．しかし，科学の考え方は常識から「離れた」ものでもなく，常識に「反している」わけでもない．また，『科学的思考の秘密は常識を「鍛え上げる」という創意工夫なのである』という見解もある[3]．結局のところ，情報系の大学学部教育においても，情報や情報処理に関する常識を涵養することが「科学的理解」の本質であると考えると判り易いかもしれない．常識というものは様々な体験を通して獲得できるものであるから，情報や情報処理を科学的に理解するためには，コンピュータによる情報処理の原理を，具体的な操作で，身をもって体験することが必須になる．

近年は，さまざまな学術的思考の色々な局面において，情報科学の基礎的な素養（情報教養）が要求されるようになってきている．たとえば，社会科学系のデータ処理を行う場合には，その統計的な性質に関する注意が肝要である．また，人文科学系のデータベース活用においては，データベースシステムに対する信頼度や保存データの属性について実際的に考察することが要請される．

本論文では，高等学校の教科「情報」を学習する段階で，ある程度のプログラミング教育が可能であることを示す．そして，そのようなプログラミングの基礎を学んで大学に進学した学生には，大学における情報教育や情報処理教育として，情報教養のカリキュラムがどのように設計可能になるかを考察する．

2 情報教養とプログラミング

関西大学が，平成５年に総合情報学部新設を申請したときの設置理念は「情報教養人の養成」であった．そのとき，『情報学』は「情報科学・コンピュータ科学・コミュニケーション科学を包括するもの」と考えられていた．また，道具としてのコンピュータや多くの情報機器を使いこなす「コンピュータリテラシ教育」が必須だと考えられていた．その後，インターネットや携帯電話・パーソナルコンピュータなどの技術発展と爆発的な普及にともなって，情報教養に含まれる実際的な技術項目は変容し続けている．今や，小学生も教室のパソコンでインターネットにアクセスしているし，個別に所有している携帯電話を使った電子メールで友人や家族と連絡を取り合うことも日常的である．もう，テレビやDVDのリモコン操作とかディジタル

カメラのシャッター操作に困難を感じる学生はいないだろう．しかし，英字や数式の書き方を覚えただけで英語や数学をマスターしたことにはならないのと同様，情報機器の操作ができるだけではリテラシ教育には不十分である．

一般的な大学における教養教育については，歴史年表を暗記したり，化学構造式を暗記するなどの，個別知識を教条的に学習することが重要なのではなく，教育担当者がそれぞれの学問を通した教養教育が求められているのである．ここでは，中央教育審議会において阿部謹也が平成13年6月に述べた意見が参考になるので，以下にその一部を引用する[4]．

> 『あらかじめ模範解答をつくって，それを皆が読み上げるという形での教養教育はだめである．つまり，それぞれの担当者が自分の学問にかけて，独自の答えを出していくという形で，専門科目を通した教養教育が大学では可能であり，それがすべての教養教育問題の一番の解答になる．』

情報を扱う技術は人類の文化発展とともに改良されてきたものであり，コンピュータの発明によって突発的に出現したものではない[5]．現代社会における情報教養は，コンピュータ技術と無関係ではないが，いわゆる情報機器の操作技能を身につけることのみに腐心していたのでは，情報教養の涵養には至らない．コンピュータに関わる技術の進展は，他の技術に比べて，格段に目覚しいのであるが，その動向に左右されない基本的な考え方や方法を習得する必要がある．

実際，情報教養を単なる情報機器の操作習熟だけに限定することが不適切であることは論をまたない．タッチタイピングの練習をすることはキーボード操作の初期訓練としても，人間工学的な情報機器設計に関する考察の基礎としても，実際的に有用である．また，文書ファイルをプリント出力する場合でも，ワードプロセッシング機能をもつアプリケーションソフトウェアを操作する体験を通して，多様なフォントが設計された文化的背景を考察することも現実的になる．しかし，漢字の明朝体とか英数字のCenturyフォント指定に必要となる操作を暗記することが，情報教養として不必要であることは明白である．駅前のパソコン教室や市民講座で教えているような，「ワード・エクセル・パワーポイント・ネットショッピングの手引き」などの皮相的な技術習得の流行に拘泥していては，ことの本質を見失うことになる．

一方，コンピュータにおける情報の表し方や処理の原理を理解しておれば，コンピュータを活用するための判断の根拠として生かすことが可能になる．一般に，ディジタル電子回路技術に基づいて開発されたコンピュータで扱うデータをディジタルデータと呼ぶのは間違ってはいないが，電磁的な媒体で表現されるバイナリデータだけがディジタルデータの実態であるとの誤解は解消されなければならない．また，コンピュータの可能性と課題とは，ソフトウェアを開発したり利用したりする，人間が工夫し解決していくべき問題であるということを認識させることが重要なのである．コンピュータのハードウェアやソフトウェアの欠陥を報道する新聞

記事とかテレビ番組に散見される技術に無理解なコメントにも批判的に対処できることが望ましい.

そのためには，学校教育においても，簡単なアルゴリズムを理解し，「学生自身に工夫させることができる」実習を行うことが有用である．とくに，コンピュータによるデータ操作が人間によるデータ操作に比べて，極めて高速である例を体感できる必要がある．そのとき，エディタでソースプログラムファイルを作成するプログラミングと，プログラムの実行との違いを体験できることは重要なポイントである.

情報処理学会では，2005年10月に「日本の情報教育・情報処理教育に関する提言2005」を公表し，手順的な自動処理の体験学習の重要性を訴えている．ここでは提言の一部[6]を引用しておく.

『十分多くの人に，「課題を分析し，系統的に解決策を考え，コンピュータに実行可能な形で明示的に表現し，実行結果を検討し必要なら反復改良する」プロセス（以下「手順的な自動処理」の構築と呼ぶ）を体験的に理解してもらう必要がある.』

その後2006年2月に，情報処理学会は，「2005年後半から2006年初頭にかけての事件と情報教育の関連に関するコメント」を公表し，情報システムの原理に対する理解の欠如に警鐘をならしている．そのコメントの一部[7]を引用しておく.

『わが国の情報技術に関する問題の多くは，企業や組織のトップを含む大多数の国民が「情報技術のことは専門家に任せておこう．自分は情報技術の素人だが，もっと重要な本業の内容について考え，それを専門家に発注するのが仕事だから，情報技術には我関せずでよい」という態度でいることに起因している．これと同等に問題なのは，そのように「我関せず」的であるにも関わらず，「コンピュータの処理結果だから信用できる」という根拠のない信頼感をシステムに抱くことである（そしてそれが覆された時，上に述べたような不信感を抱くことになる).

今日では情報システムはすべての組織における活動の基盤となっており，その高度さと複雑さは単に仕様書やマニュアルの品質を改善すれば済むという段階を超えたものとなっている．このため，「我関せず」的態度では，組織が必要とする高度な情報システムを構築／維持することも，それらを駆使した社会活動を続けることも，到底できない．技術的細部に立ち入ることまでは必要ないとしても，自己の活動の根幹に関わる原理的部分については，それぞれが自分の責任ないし問題として引き受けることが必要である．そのために，すべての国民の情報水準を底上げし，また，高等教育を受けてわが国の将来を担う人材であれば誰もが情報および情報技術に関する一定の原理理解を身につけるようにさせなければ，21世紀の情報社会に対応してゆけない，というのが提言の主旨である.』

結局は，具体的なプログラミング言語を用いての実習や演習が望まれるのであり，例えばJavaScriptプログラミング[8]は，実習教室での演習が極めて手軽に可能だという利点がある[9]．実際，情報処理教育研究会による試作教科書ではJavaScriptプログラミングを含んだ具体的な教材案を提示している[10]．

高度な情報技術の専門家を育成する多くのプロジェクトがあるが，なかでも中学生や高校生を対象とした「情報オリンピック」では，2006年には日本の高校生が金メダルを獲得して脚光を浴びている[11]．また，大学生を対象とした「ACMプログラミングコンテスト」では，毎年のように国内予選，アジア地域予選を経て，優秀な大学生が選抜されて世界大会に出場している[12]．

しかし，本論文で考察している情報教養とは，一般的な高校生や大学生の「情報水準」の底上げを目指すものであり，その結果として高度な情報技術の専門家が育成可能になるという点に主眼がある．情報オリンピックとかプログラミングコンテストにおいては，情報科学的な思考自体が基盤になるのであり，小手先のプログラミング技法暗記だけでは成果は望めないものである．

3 JavaScriptとブラウザ

多くのブラウザにはJavaScriptの処理系が組み込まれている(1)．しかも，インタプリタ方式であるから，コンパイラなどの実行系を準備する必要もない．エディタは，Windowsならアクセサリに含まれているアプリケーションソフトの「メモ帳」で十分であるし，UnixならEmacsとかviやexなどの編集ソフトウェアを使うことになる．つまり，いつでもどこでもプログラミングを楽しめるということになる．簡単なプログラムから始めて，みずから動かせる経験を積めば，コンピュータソフトウェアのふるまいが自然に理解でき，コンピュータによる情報処理の科学的理解が楽しくなる．

プログラミングは「ソースプログラムを白紙に書き下ろしていく」のではなく，現代では「用意された部品プログラムをうまく活用して，やりたいことをこなす」ことが推奨される．さまざまな「モノ」をプログラムの上で「オブジェクト」として扱える，いわゆるオブジェクト指向の機能が必須になってきている．なかでも，JavaScriptは基本メカニズムが簡単であり，導入教育には適した言語といえる[14]．とくに，基礎的なオブジェクトを組み合わせるプロトタイプ方式のプログラミングが適用できる利点は見逃せない．

ブラウザで処理させるHTMLファイルにはプログラムのソースファイル名を記述するだけでよい．例えば，ファイル名がpl.jsの場合には図1のようになる．

---

(1) Netscapeに組み込まれた1994年以来，MSIEにもほぼ互換性のあるJScriptが搭載された．その後，ヨーロッパの情報通信システム標準化機構(ECMA)において標準化が実現されている[13]．

```
<html>
<head>
  <title>Programming Lab. session @ Kansai University</title>
</head>
<body>
  <script type="text/javascript" src="p1.js"></script>
</body>
</html>
```

図1 JavaScriptプログラムを参照するHTMLファイルの例

プログラミングの学習で必要となる多くの機能がJavaScriptには備わっているが，ネットワークを経由したさまざまなアプリケーションを開発するという設計方針であるので，ローカルサーバのファイルを直接的に操作することはできない．これは，ネットワーク経由でのアクセス制御（セキュリティ確保）の問題であり，止むを得ない制限事項といえる．したがって，プログラミングの学習としては，JavaScript学習の次の段階として，C言語などの学習を通してファイル操作プログラミングを体験することが望ましい．

4 JavaScriptによる実習の例

プログラミングの対象はさまざまであるが，ここでは，プログラミングの学習として「変数と代入」・「順次実行」・「制御構造（if-then-else, for, break, exit)」・「関数と引数」などの機能に関する知識を前提として，パソコンによる実習の事例を示すことにする．

4.1 利子計算プログラム

まずは，複利計算による元利合計の計算プログラム（図2）を考えてみる．

```
var r, n;
 r = parseFloat(prompt("年利率？")), n;
 document.write("<br>年間の利率 "+r+"のとき<br>");
 for(n = 1; n < 365*24; n=n*2){
  document.write(n+"回複利の利率は "+r/n+
  "となり，その元利合計は "+Math.pow(1+r/n, n)+"<br>");
  }
 document.write("<br> 究極の複利における利子計算 "+Math.exp(r));
```

図2 複利計算のプログラム

利率データとして出資法[2]の上限金利0.292を入力したときの実行例の一部を図3に示す.

```
年間の利率 0.292 のとき
1 回複利の利率は 0.292 となり，その元利合計は 1.292
2 回複利の利率は 0.146 となり，その元利合計は 1.3133159999999997
4 回複利の利率は 0.073 となり，その元利合計は 1.3255584662409998
8 回複利の利率は 0.0365 となり，その元利合計は 1.332154056081999
16 回複利の利率は 0.01825 となり，その元利合計は 1.335582439185596
 ・・・
究極の複利における利子計算　　1.3391030176392937
```

図3　複利計算プログラムの実行例

このプログラムでは，1時間ごとに利子を支払う時間複利計算までを計算しており，近似的な月利計算の場合（年16回複利）1.336に近い値が得られる．複利期限の極限を数式で表現するならば，次式のようになるので，e^rに代入すると$2.71828^{0.292} \fallingdotseq 1.339\cdots$となる．つまり，一瞬ごとに利子を計算する場合でも元利合計は1.34倍が限度になる．

$$\lim_{n\to\infty}(1+\frac{r}{n})^n = e^r \tag{1}$$

利子計算の途中結果が小数第16位まで表示されているのは，JavaScriptでの計算精度限界を示している．実際，JavaScriptではすべての演算が倍精度実数演算[3]になっており，10進約16桁の精度になる．したがって，整数の場合には2^{54}なども誤差なく表示できる．他方，0.01 + 0.05とか0.01 + 0.06などの小数演算（浮動小数点演算）においては，2進10進変換誤差および丸め誤差が影響するので，応用分野によっては注意が必要である．

4.2 最大約数の計算

世界最初のプログラム記憶方式のコンピュータSSEM（Baby）で，最初に動いた[4]テストプログラム，「任意の正の整数の最大約数（自分自身は含めない）を求める」のC言語版を大駒が紹介している[15].そのJavaScript版は図4のようになる．当時のコンピュータには除算命令がなかったので，減算の反復で除算に代替したということであり，262144（$=2^{18}$）の最大約数（131072）を求めるのに52分を要したそうである．このプログラムの場合には，減算と判定をそれぞれ約13万回も反復することになるので，JavaScriptでの実行時間は手元のノー

(2) 出資の受入れ，預り金及び金利等の取締りに関する法律.
(3) 標準的な倍精度規格IEEE754に準拠している[13].
(4) 1948年6月21日.

トパソコン[5]で0.5秒ほどであった.

JavaScriptの整除演算を用いる場合には次のプログラム（図5）のようになる．このプログラムでは素数$2^{31}-1$（=2147483647）の判定も可能であり，約3万回の反復になるのであるが，その場合の実行時間は1秒もかからない．なお，除算を減算の反復で代行する先ほどのプログラム（図4）で，この素数判定を行うにはプログラム第1行目にあるparseIntの引数に数値(2147483647）を指定することになるが，その計算には30分以上の時間が必要になると推量できる．このような計算量に関する考察の原理を理解しておくことは，計算機科学の重要な項目でもあり，情報教養の一端になるのである．

```
var n = parseInt("262144"),j,k;
document.write("世界最初のテストプログラムと同じ方法で");
document.write(n+"の最大約数を計算<br>");
for (j = n-1;j > 0;j--){
  for (k = n;k > 0;k -= j){;}
  if (k == 0) {
    document.write("GD of "+n+" = "+j);break;}
}
if(j == 0){document.write(n+" is prime");}
```

図4　最大約数を減算の反復で求めるプログラム

```
var n = parseInt(prompt("自然数？")),j,jmax = Math.sqrt(n);
document.write("整除算で"+n+"の最大約数を計算<br>");
if (n%2 == 0)  {
 document.write(n+"= "+2+"x"+n/2);}
else {
 for (j = 3;j <= jmax;j = j+2){
   if (n % j == 0) {
   document.write(n+"= "+j+"x"+n/j);break}
 }
 if (j > jmax) {
  document.write(n+" is prime");}
}
```

図5　最大約数を整除算の反復で求めるプログラム

特定の自然数に対する最大約数を求めるプログラムを作成しただけでは発展性が無いので，

(5) Pentium M, 2.66GHz, 1.99GB RAM, Windows XP.

色々な自然数の場合に応用できるように，ブラウザのフォーム入力を使うことにするには次のHTMLファイルが利用できる（図6）．ここでは，p4.js（図7）なるプログラムファイルを指定している．

実際，最大約数を計算する手順は前述のプログラム（図5）と全く同一であるが，整数値の初期設定部分でオブジェクトのValue属性を利用している．

```
<html>
<head>
  <title>Form input program </title>
  <script type="text/javascript" src="p4.js"></script>
</head>
<body>
<h1> 色々な数の最大約数を求めてみよう </h1>
<p><form name = "Mado">
   <input type="text" size=18 name="seisu">
   <input type="button" value="計算" onclick="calc('Mado')">
</form></p>
</body>
</html>
```

図6　Formを使ってデータを入力するHTMLファイル

```
function calc(Mado){
 var n, j, jmax;
 n = parseInt(Mado.seisu.value); jmax = Math.sqrt(n);
 if (n%2 == 0) {
   window.alert(n+"=2x"+n/2); }
 else {
 for (j = 3; j <= jmax; j = j+2){
   if (n%j ==0) {
     window.alert(n+"="+j+"x"+n/j); break;}
    }
 if(j > jmax){window.alert(n+" is prime");}
  }
 }
```

図7　オブジェクト形式データを使うプログラム

5 むすび

ディジタルデータとして表現され伝達される情報は量的に増大する一方である．しかも，これらを読み解くための訓練には従来の手法に無い工夫が必要である．一般的には，情報スキルを教師が解説したり，例題に使われているような規模のデータで情報処理の訓練をして，学生達が最新の情報機材を使いこなせるようになっただけでは意味がない．

情報ツールの効果を科学的に理解し，データを適切に加工したり編集する訓練を通して，情報を読み解く基礎になる情報教養を涵養することが本質的なのである．その結果，いわゆる〈ほんもの〉という倫理[16]について，科学的に考える情報教養が涵養されることを期待したい．

参考文献

[1] 江澤：「高校教科「情報」と情報スキル」，情報処理学会，情報処理教育委員会シンポジウム，高校教科「情報」の現状と将来，http://sigps.tt.tuat.ac.jp/index.php, 2005.

[2] 文部省：「高等学校学習指導要領解説　情報編」，2000.

[3] 瀬戸：「科学的思考とは何だろうか」，ちくま新書461, 2004.

[4] 阿部：「大学における教養教育について」，http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo1/gijiroku/001/010601a.htm, 2001.

[5] 江澤：「IT革命と情報倫理」，システム／情報／制御，Vol. 45, No.9, pp. 523 -527, 2001.

[6] 情報処理学会 情報教育委員会，日本の情報教育・情報処理教育に関する提言2005, http://www.ipsj.or.jp/12kyoiku/proposal-20051029.html.

[7] 情報処理学会：「2005年後半から2006年初頭にかけての事件と情報教育の関連に関するコメント」，http://www.ipsj.or.jp/12kyoiku/statement2006.html, 2006.

[8] 久野：「入門　JavaScript」，ASCII, 2001.

[9] 江澤：「JavaScriptを導入した授業の展開—明日から実践可能なプログラミング—」，ICTE関西大学，2006.

[10] 情報処理学会 情報教育委員会：「試作教科書　教科書案2」，http://sigps.tt.tuat.ac.jp/, 2006.

[11] 「情報オリンピック」，http://www.ipsj.or.jp/01kyotsu/topics/olympic.html, http://olympiads.win.tue.nl/ioi/, http://olympiads.win.tue.nl/ioi/ioi2006/contest/results.html+, 2006.

[12] 「ACMプログラミングコンテスト」，http://icpc.baylor.edu/icpc/，http://www.acm-japan.org/Welcome-j.html, 2006.

[13] 「ECMA Script Language Specification Edition 3 Final」，http://www.mozilla.org/js/language/E262-3.pdf, 2000.

[14] プロジェクトA：「標準HTML，CSS＆JavaScript辞典」，インプレス，2004.

[15] 大駒：「コンピュータ開発史」，共立出版，2005.

[16] ティラー（田中訳）：「〈ほんもの〉という倫理」，産業図書，2004.