

Web ブラウザベースのプログラミング実行環境

鳥居 隆司*1・杵淵 信*2・安藤 明伸*3・田村 謙次*4・川崎 直哉*5
 ・大岩 幸太郎*6・中野 健秀*7・藤尾 聡子*8・古金谷 博*8

Email: torii@sugiyama-u.ac.jp

- | | |
|--------------------|---------------------|
| *1: 相山女学園大学 文化情報学部 | *5: 上越教育大学 学校教育学部 |
| *2: 北海道教育大学 教育学部 | *6: 大分大学 教育福祉科学部 |
| *3: 宮城教育大学 教育学部 | *7: 愛知学院大学 商学部 |
| *4: 中央学院大学 商学部 | *8: システムズ・クリッパー株式会社 |

◎Key Words プログラミング, Web, 描画機能

1. はじめに

情報社会に対応し、さらに未来の情報社会を支えられるために、情報そのものに関する考え方や情報に関連する応用分野の教育、そして、情報技術の教育が重要な課題であるが、うまく機能しているとは考えにくい。情報社会における我々の豊かな生活は、情報技術によってもたらされている部分が非常に大きい。たとえば、日々の情報収集には検索エンジンを用い、日常のコミュニケーションや仕事の連絡には、電子メールや SNS などを利用し、ファイルの保存にクラウドを利用する。また、日常生活における家電製品から様々な産業を支えている電子部品にはコンピュータが組み込まれており、それらはソフトウェアで動作している。

しかし、このような情報社会でありながら、多くの人々は、コンピュータの仕組みや動作原理、プログラミングやソフトウェア開発に関する内容、コンピュータネットワークにおけるプロトコル等について基本的な部分さえも理解されていないと考えられる。コンピュータが電子計算機とされていた時代には、大学などの高等教育機関において情報教育として、プログラミング言語とプログラミング、アルゴリズムが教えられていたが、インターネットが普及しコンピュータが各家庭に普及し始めた頃からこれまでの情報教育はアプリケーションの使い方を学ぶ方向へ進んできたように思われる。

高等学校では、2003 年度より普通科において教科「情報」として情報 A、情報 B、情報 C の 3 つの教科が新設されたが、全国的にも「コンピュータや情報通信ネットワークなどの活用を通して、情報を適切に収集・処理・発信するための基礎的な知識と技能を習得させるとともに、情報を主体的に活用しようとする態度を育てる」との目標を掲げた情報 A を採用する高等学校が 8 割程度⁽¹⁰⁾もある。教科「情報」を教えることのできる教員は、主に情報とは異なる教科の現職教員を対象とした短期間の研修によって養成されたことや受験科目に該当しないこと、教科「情報」の新設された意図ではない方向のカリキュラムを行っていた高等学校も少なくないとも考えられる。2013 年度からは、共通教科「情報」として、「情報社会に参画する態度」や「情報の科学的な理解」を主として科目が構成され、

「情報の科学」では、「問題解決を行うために情報と情報技術を効果的に活用する学習活動やそのために必要となる科学的な考え方を身に付ける学習活動をより一層重視」とする方向⁽¹⁾へと変化している。

学校教育とは別に最近では、スマートフォンなどのコンピュータを誰もが使いこなすようになってきており、いわゆるアプリの需要をはじめとして、安定した良質な情報環境を供給する必要がある。そのためには、ソフトウェアを開発できる人材の養成が急務であることなどからプログラミング教育^{(3)~(7)}を重視すべきであるとの声が高まってきている。

たとえば、情報科学やプログラミングを誰もが学習できることを目標にプログラミングを学ぶ^{(11)~(25)}ことのできるコンテンツへの橋渡しを行う「Code.org⁽²⁶⁾」には、「アメリカではだれもがコンピュータにどのようにプログラムするかを学ぶべきだ、なぜならば、そのことはどのように考えるかを教えてくれるからだ」とのステーブ・ジョブズの言葉やビル・ゲイツ、Facebook の創業者であるマーク・ザッカーバーグ、ザッポス(Zappos.com)の創業者のトニー・シェイ、Dropbox の創業者のドリュー・ヒューストンらが、プログラミングの魅力をそれぞれの言葉で語っているインタビューがある。また、今後、コンピュータに関する仕事はますます増加し、情報科学を専門に学ぶ学生の数と比較して、圧倒的に多いことも示している。

2. 情報科学に関する教育

現在の情報通信技術が社会基盤を支えており、ネットワークの高速化や大容量化、ワイヤレスネットワークの整備、クラウドコンピュータの活用、スマートフォンやタブレット端末の普及、社会基盤のスマート化などが進んでいるが、コンピュータや情報処理および情報システムの仕組みなどの理解、ソフトウェアを開発できる人材、これらの教育ができる人材や教育機関の不足など情報通信技術に関する基礎から応用、活用に至るまで多くの問題点がある。デジタル・ネイティブという言葉も古くなってしまった感もあるが、最近では、物心のつく子どもの頃からコンピュータに接し利用しており、また、ソフトウェアの GUI、ネットワーク接続の容易さ、各種センサーの充実などからコン

コンピュータの仕組みや動作について、ほとんど考えることなく活用できるため、その仕組みはブラックボックス化してしまっており、中身を詳しく知る必要もない。

現在の学習指導要領では、小学校の「総則」「道徳」「総合的な学習の時間」や中学校の「総則」「道徳」にいくつかの関連する記述を見ることができるが、いずれも、コンピュータや情報通信ネットワークなどの適切な利用・活用と情報モラルに関する内容にとどまる。コンピュータや情報処理に関する内容に踏み込んだものは、中学校、技術・家庭の技術分野⁽²⁾に「コンピュータを利用した計測・制御の基本的な仕組みを知ること」と「情報処理の手順を考え、簡単なプログラムが作成できること」のみと言ってもよい。高等学校においては、前述のように「情報の科学」でその内容についてやや踏み込んで学習できるが、「社会と情報」を選択する高校が多くなれば、当然、コンピュータや情報処理に関する内容をほとんど理解しないまま進学し、社会に出ることとなる。

3. プログラミング学習の環境

このような背景から、プログラミングに少しでも興味を持ってもらい、わかりやすく学習できるように工夫された様々なプログラミング言語や学習環境がいくつもある。アニメーションに特化しオブジェクトに対してその特性を設定する **Viscuit**、条件判断・繰り返しなどのプログラムの基本構造をタイルでつなぐことでプログラミングができる **Scratch**、タートルグラフィックスが容易にできる **LOGO**、ドリトルの他、日本語でプログラミングができる **PEN** やなでしこなどである。また、特にこのようなプログラミング学習に特化したものでなくても、**Web** ブラウザによる **JavaScript** や **Excel** で **VBA**、**OS** 標準で用意されているシェルなどを利用することもできるし、**Visual Studio Express** や **Java** など利用することで、業務で利用されているプログラム学習も可能である。

しかしながら、前者の学習環境では、プログラムはどのようなものかということやコンピュータが手順に従って自動的に処理を行う仕組みやアルゴリズムの理解はある程度可能であるが、通常の命令文や文法に触れることは難しい。また、後者では、プログラミングを行うための環境を構築することが初学者にとっては極めて困難であり、学習者にしても教える側にしてもプログラミング言語ではなく、アプリケーション開発環境に関する知識も必要となる。

そこで、我々は、これまでにプラットフォームに依存せず **Web** ブラウザ上で編集、コンパイル、実行ができるプログラム学習環境を構築⁽⁸⁾⁽⁹⁾しているが、最近、様々な場面で活用されているスマートフォンでも容易に利用できる **C** 言語のプログラム実行環境を **Fig.1** に示すようにクラウド上に構築した。現実には **C** 言語などの現在も業務で実際によく活用されているコンパイラ型の言語を学習しようとする場合には、プログラムを編集するエディターの環境とコンパイラのインストールの他、エディターのコマンド実行機能を利用したコンパイラの実行設定やディレクトリパスの設定などを行う必要がある。プログラミング初学者にとっては、こ

れらの作業が非常に負担であり、多くの学習者が初期段階においてプログラミング環境を構築することができず、プログラミング学習をあきらめてしまう結果となる。また、教える側にとっては、最近のアプリケーション開発環境は、その準備や更新作業の他、非常に多機能であるため、学習者が使い切れない機能を誤って操作した場合の挙動について常に注意を払う必要があり、多くの時間を費やすこととなる。

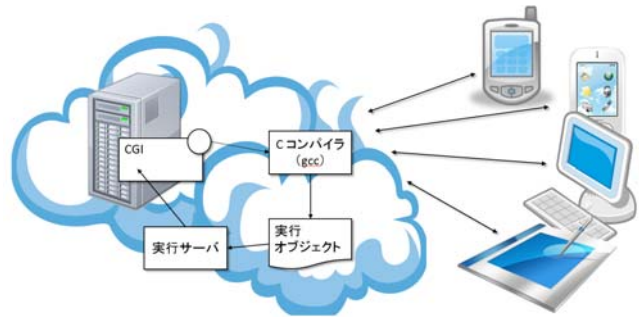


Fig.1 本環境のシステム構成

本環境では、**Fig.2** に示すように学習環境にログイン後、プログラムの記述ができるテキストエリアとコンパイル・実行ボタンによって、記述したプログラムを実行させることができるほか、記述したソースプログラムをクラウド上に保存することができる。



Fig.2 Android OS のスマートフォンでのログインとプログラミングおよび実行環境へのメニュー

プログラムの記述ができるテキストエリアで記述したソースは、「コンパイル」ボタンをクリックすることでデバッグされ、エラー等がなければ、コンパイルされた実行オブジェクトを実行できる「実行」ボタンが表示され、実行できる。このようにログイン後、すぐにプログラム言語の学習を行うことができ、コンパイラには **gcc** を利用しているため、言語仕様の制限がほとんどなく、一般的な市販の教科書や **Web** 上に多数に存在するテキストの例題などを学習教材として活用することも可能である。また、サーバとやりとりされるデータもテキスト程度のものでありスマートフォンのような携帯情報端末と基地局とも通信速度が上がらない

場合のような環境下でも十分使用することができる。

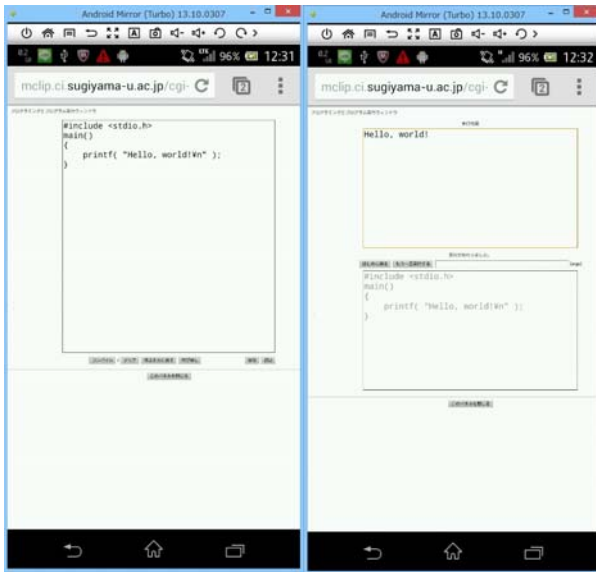


Fig.3 Android OS のスマートフォンのテキストエリアでの C 言語の記述とデバッグや実行の様子



Fig.4 文法エラーのあるプログラムのエラー出力の様子

4. 描画機能

タートルグラフィックスでよく知られている LOGO はその容易な図形描画機能を用いて学習者の論理的な思考や図形の理解などを学習者自らが気づく過程を通して学ぶことができる。また、再帰プログラムの理解のためにフラクタルの描画を用いることはよく行われている。プログラミングによる図形描画は、その結果が図示されるので、プログラミングの誤りについて視覚的に気づくことができる。

しかし、初学者がグラフィック出力を行うためには、単にプログラム言語の編集・実行環境を整えるより、さらにそのインストールや設定に時間と労力を費やす必要があり、肝心のグラフィックプログラムの学習にたどり着くことができないという結果になる。たとえば、描画ウィンドウへのハンドルやどんなペンで描画させればよいのか、ペンの線種や色、太さなどをどのように設定するかを理解している必要がある。しかし、初学者がこの設定を行うために、オブジェクトや構造体を用いる必要があるとしても、プログラム言語の学習以前にこれらの設定を行うことはほとんど不可能であろう。

さらに、グラフィック出力のための関数を用いるためには、ヘッダファイルの参照を行う必要があるが、この場合にも、関数のプロトタイプ宣言の他、描画の

指定をわかりやすくするための様々な設定も初学者が行うことが極めて困難な内容である。

このことは当然教える側にとっても多くの時間と労力を費やすことになる。本実行環境では、学習者は、描画機能に関する手順をほとんど必要としない環境とした。本機能の仕組みを Fig.5 に示す。

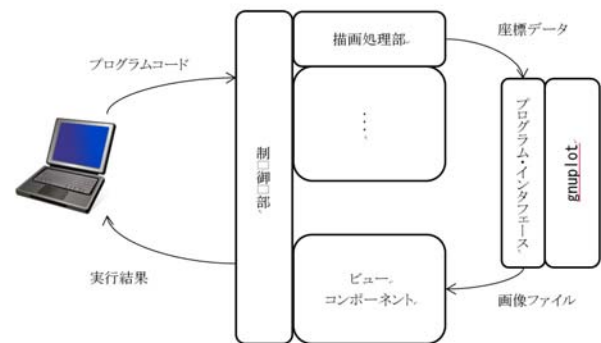


Fig.5 描画機能の仕組み, gnuplot に伝達されて生成された画像ファイルは、ビューコンポーネントに表示される

プログラム上で描画関数を呼び出すことによって、描画処理部で座標データが生成され、その座標データは、プログラム・インタフェースを通じて gnuplot から画像ファイルが生成される。この画像ファイルを Web ブラウザに表示させることで実行結果としている。

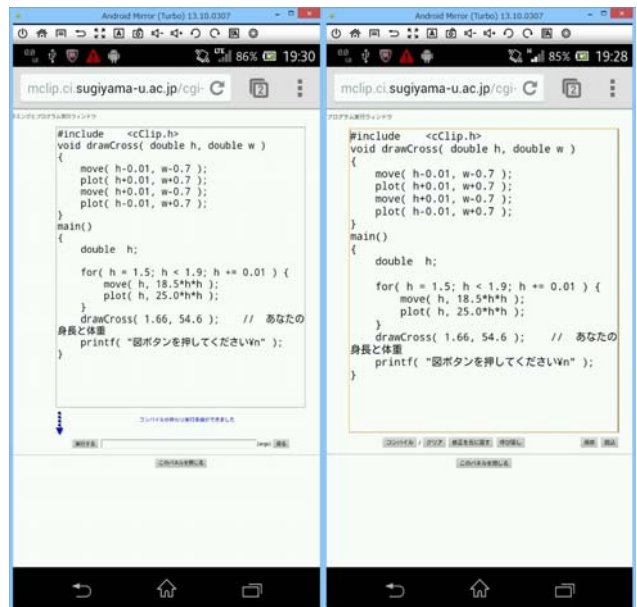


Fig.6 BMI による標準体型の範囲を描くプログラムのソースコード

表 1 絶対座標指定 (プロッター・タイプ)

関数	機能
move(x, y)	ペンアップで(x, y)に移動する
plot(x, y)	ペンドアウンで(x, y)に移動する

描画機能では、表 1 に示すように xy 座標を指定してペンを移動させ、絶対座標を用いるタイプと、表 2 に

示すようにペンの現在位置と進行予定方向を基準とし、そこからの回転方位角と距離を指定して移動するタイプを用意し、その軌跡が図を構成することができる。

表2 相対指定 (タートルグラフィック・タイプ)

関数	機能
fd(s)	距離 s だけ前進する
bk(s)	距離 s だけ後退する
lt(d)	進行方向を反時計回りに d 度回転する
rt(d)	進行方向を時計回りに d 度回転する

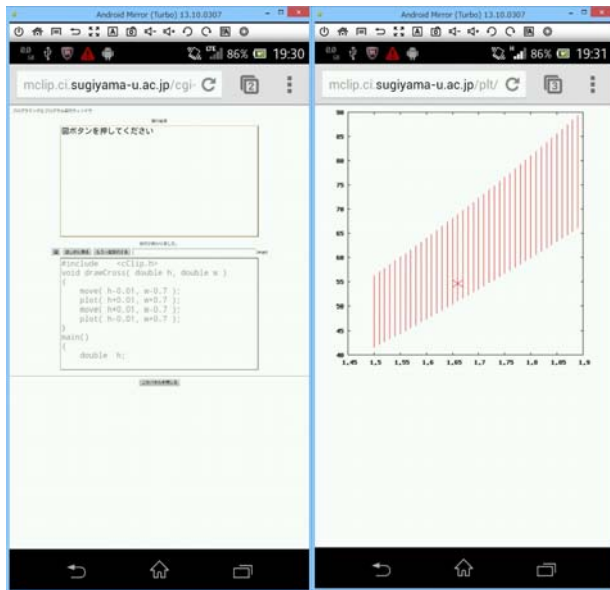


Fig.7 BMI で見た標準体型の範囲を描く例

5. 考察

コンパイラ型のプログラミング言語の学習を初めて行う場合、少なくともそのプログラミング言語がコンパイル・実行できる環境を整える必要があり、初心者の場合には、困難を伴い多くの時間が費やされることも多い。最近では GUI が優れユーザにコンピュータのファイルシステムや OS の存在を知ることなく利用することが可能であることもその原因の一つである。いつでも、どこからでも同じ実習環境が使える。そこで本研究では、プログラミング環境の構築準備が不要。プログラム入力から実行までの操作が簡単。CPU パワーのない情報端末であっても、ネットワークの速度が遅い環境でも、動作や反応が早く、安定している。コンピュータのメモリなどハードウェアを容易に直接操作できるコンパイラ型の言語であってもセキュリティ上問題がないなどの環境を目指して、PC の Web ブラウザからの利用だけでなく、様々な携帯情報端末の Web ブラウザから利用可能なプログラム学習環境をクラウドコンピュータ上に構築し、サーバ上において、任意の C 言語のプログラムコードを入力でき、さらにコンパイルの上、実行できる環境を整えることができたと考えられる。このことは、学習者が容易にコンピュータの基本的な特徴を体験できるということが可能となり、簡単な手続きによって機能する描画機能によりソフトウェアの動きを直感的に知ることが可能であり、たとえば、ソフトウェアのバグによる社会システムの

障害がなぜ起きるのかを学ぶこともできるということにつながる。その結果、学習者は、我々の社会システムを構成する情報システムの挙動についての知見を得ることができるのではないかと期待している。

6. おわりに

本研究において構築した仕組みがプログラミングを行う体験を通してコンピュータに関する幅広い関心を見出す機会を与え、アプリケーションや ICT を使いこなす情報教育から、少しでもソフトウェアを創造しアプリケーションをデザインできる人材を育成する方向へ向けた情報教育が重視されるようになるために、さらに検討していきたいと考えている。

謝辞

本研究の一部は、JSPS および MEXT 科研費 2450169 および 25381242 の助成を受けたものである。

参考文献

- (1) 文部科学省, 高等学校学習指導要領 平成 21 年 3 月.
- (2) 文部科学省, 中学校学習指導要領 平成 20 年 3 月(平成 22 年 11 月一部改正).
- (3) ACM K-12 Task Force Curriculum Committee, "A Model Curriculum for K-12 Computer Science 2nd Edition", 2003.
- (4) <http://csta.acm.org/Curriculum/sub/CurrFiles/K-12ModelCurr2ndEd.pdf>
- (5) UNESCO, IFIP Curriculum, Information and Communication Technology in Secondary Education, D. ICT Specialisation (1994, Update 2000),
- (6) <http://www.wedu.ge.ch/cptic/prospective/projets/unesco/en/>
- (7) 情報処理学会情報処理教育委員会, 日本の情報教育・情報処理教育に関する提言 2005 (2005).
- (8) 古金谷博, 鳥居隆司, 井上明, 杵淵信, 田淵哲明, 田村謙次, 中野健秀, プラットホームに依存しない C 言語学習環境の構築, PC カンファレンス論文集, PC カンファレンス実行委員会 (2005).
- (9) 古金谷博, 藤尾聡子, 鳥居隆司, プログラム言語学習の位置づけと目的, コンピュータ&エデュケーション, Vol.22, 2007.
- (10) 2003 年度高校教科採択状況, 内外教育, 2002.
- (11) Codecademy, <http://www.codecademy.com/>, (2013/05/19).
- (12) ドットインストール, <http://dotinstall.com/>, (2013/05/19).
- (13) CodeStudy, <http://jeek.jp/study/exp/>, (2013/05/19).
- (14) code.9leap.net, <http://code.9leap.net/>, (2013/05/19).
- (15) ミニツク - Ruby の e-ラーニング研修システム, <http://www.minituku.net/>, (2013/05/19).
- (16) プログラミン, <http://www.mext.go.jp/programin/>, (2013/05/19).
- (17) Google Blockly, <http://code.google.com/p/blockly/>, (2013/05/19).
- (18) Khan Academy, <http://www.khanacademy.org/>, (2013/05/19).
- (19) Codeschool, <http://www.codeschool.com/>, (2013/05/19).
- (20) Code Avengers, <http://www.codeavengers.com/>, (2013/05/19).
- (21) LearningJar, <https://learningjar.com/about.html>, (2013/05/19).
- (22) TheCodePlayer, <http://thecodeplayer.com/>, (2013/05/19).
- (23) (un)classroom, <http://www.unclassroom.com/>, (2013/05/19).
- (24) Treehouse, <http://teamtreehouse.com/>, (2013/05/19).
- (25) Learn jQuery & JAVASCRIPT for free|appendTo Developer Learning Center, <http://learn.appendto.com/>, (2013/05/19).
- (26) Code.org, <http://www.code.org/>, (2013/05/19).