

プログラミング未経験人文社会系学生向けプログラミング講座

安藤 昇¹ 大川内 隆朗² 谷 聖一²

概要：日本大学文理学部は人文系6学科・社会系6学科・理学系6学科の計18学科からなり、「文」「社」「理」の融合を特色とした教育と研究を行っている。しかし、プログラミングを学べる科目は、2020年度より全学生が履修可能なものが用意されるものの、現行カリキュラムでは学科専門科目の中にのみ存在し、学科によってはプログラミングに触れられる学科専門科目を用意していない。そこで、プログラミング経験のない人文系・社会系学生を対象に、課外授業として、全12回のプログラミング講座を実施した。ビジュアル型言語での導入から始まり、micro:bitを利用した通信プログラミング体験などを経て、Google Colaboratory を利用した Python 入門と機械学習ツール利用体験などを行った。本報告では、この実施概要と受講者のアンケート結果を紹介する。

キーワード：初学者向けプログラミング教育、一般情報教育、課外授業

Informal programming course for humanities and social sciences students

Noboru Ando^{†1} Takaaki Ohkawauchi^{†2}
Seiichi Tani^{†2}

1. はじめに

プログラミングをすべての人が学ぶ機運が日本でも高まっている。プログラミングをすべての人が学ぶべき理由・目的については、様々な議論がなされている。例えば、[1]では、『プログラミングでなければ教えられないことは唯一「コンピュータとは何か」ということで・・・「プログラムを書くことによってしか得られないコンピュータ観」としよう。それが定義できたとして、そのコンピュータ観は義務教育の貴重な時間を奪い合ってまでみな知る必要のあるものかどうか、そこが問われる。』という問題提起がなされている。2017年・2018年に改訂され、小学校では2020年度に全面実施、中学校では2021年度に全面実施、高等学校では2022年度より年次進行で実施される学習指導要領では、小・中・高等学校を通じてプログラミング教育が充実される([2])。高等学校における共通教科「情報」では、「社会と情報」と「情報の科学」のいずれかを選択して履修するという構成から「情報Ⅰ」を全員が履修し、発展的な科目として「情報Ⅱ」を選択できるようになる。「情報の科学」の中でもプログラミングが扱われるが、「この科目を履修する生徒は2016年度の教科書需要数から推定すると全体の2割程度科目」([3])である。「情報Ⅰ」で、高等学校で学ぶすべての生徒がプログラミングを学ぶようになる。

大学における一般情報教育においても、プログラミング教育が実施されている。情報処理学会が公表している「カリキュラム標準一般情報処理教育」の「一般情報教育の知

識体系(GEBOK2017.1)」にも「アルゴリズムとプログラミング」として含まれている([4])。

日本大学文理学部([5])は、人文系6学科・社会系6学科・理学系6学科の計18学科からなり、「文」「社」「理」の融合を特色とした教育と研究を行っている。2020年度から年次進行で実施するカリキュラム改訂では、すべての学生がプログラミングを学べるように、「アルゴリズムとプログラミング基礎」と「アルゴリズムとプログラミング応用」という科目を一般情報教育の枠組みの中で設置する。しかし、2019年度までに入学した学生は、学科専門科目の中でプログラミングを学ぶ機会があるものの、学科によってはプログラミングに触れられる学科専門科目を用意しておらず、残念ながらすべての学生がプログラミングを学べる状況にはなかった。高等学校で「社会と情報」を選択した学生は、高等学校でも大学でもプログラミングを学ぶ機会を得ずに社会にでることになる。そこで、急遽2018年度末に検討をし、2019年度に在学するプログラミング未経験学生、特に、人文系・社会系の学生を対象に、プログラミングを体験する課外授業を実施することとなった。本報告では、この実施概要と受講者のアンケート結果を紹介する。

2. 講座実施

2.1 実施体制・募集

第一著者が、本講座の教案・教材作成及び主講師を担った。第二著者は受講者アンケート作成及び集計を、第三著

¹ 青山学院高等部情報科
Aoyama Gakuin Senior High School

² 日本大学文理学部
Nihon University

者は補助講師を担当した。最初の数回は第二著者も補助講師として授業のサポートをしたが、基本的に2名での講座運営のため、定員は30名とし、最大35名まで受け入れることにした。2019年5月8日より2019年7月24日までの毎週水曜日18:00~19:00(各回1時間)に12回実施することとし、2019年3月27日に広報を開始すると同時に募集を開始した。この広報・募集及び講座開始後のコンピュータ室サポートは学部事務局が担った。4月25日頃まで募集を継続することになると想定していた(そのため講座開始を4月ではなく5月とした)が、募集開始日の午前10時過ぎには申込者が35名に達し、募集を終了した。このことから、人文系・社会系学生のプログラミングへの関心の高さが窺える。応募した学生の所属学科は、哲学、史学、国文学、英文学、ドイツ文学、社会学、教育学、心理学、地理学の9学科であった。一方、実際に5月8日の第1回の講座に出席したのは25名であった。10名が応募しながら受講しなかった原因として、前期授業開始前の募集であったことや応募から講座開始まで一ヶ月以上間があったことなどが推察されるが、実際の原因は追跡できていない。なお、最終週の講座に参加したのは7名であった。

2.2 講座内容

12週の講座内容を表1に示す。本講座の対象は、プログラミング経験のない人文系・社会系学生であるため、導入をビジュアル型言語で行い、テキスト記述型言語に移行することとした。プログラミングを体験することでプログラミングとはどのようなことかを体感することが本講座の最大の目的であるが、それに加えてIoTや機械学習技術にも触れるものとした。また、講座受講時以外にも自習できるような環境を選定した。

第1週から第4週は、まずScratch([6])でプログラミングを体験した。図1のようなマップ上を、番号をたどりながらスタート地点まで自動的に戻るプログラムを作成することで、条件分岐や繰り返しといった逐次処理の基本を学んだ。また、第3週と第4週では、第2週の成果をベースに受講生各自が考えた物語を追加することで、学んだことを活かし創造活動をしてもらうことを意図した。

表1. 講座概要

第1週	Scratch 第1回 1. 導入と体験
第2週	Scratch 第2回 課題 18名提出/25名 1. サイコロをふると数字を追って移動しスタート地点にもどろう
第3週	Scratch 第3回 課題 17名提出/25名 1. メッセージ機能を使って物語を作ろう 2. 配列変数を使って会話を作ろう (1/2)
第4週	Scratch 第4回 課題 12名提出/25名 1. 変数を使ってコース番号を数えよう 2. 配列変数を使って会話を作ろう (2/2)
第5週	micro:bit と MakeCode で IoT 入門 第1回 1. micro:bit で1人ジャンケン 2. micro:bit で2人ジャンケン (1/2)
第6週	micro:bit と MakeCode で IoT 入門 第2回 1. micro:bit で2人ジャンケン (2/2) 2. ジャンケンの勝敗判定
第7週	micro:bit と MakeCode で IoT 入門 第3回 1. micro:bit で神経衰弱 1: センサーを使って、micro:bit を裏返しと数字を表示 2. micro:bit で神経衰弱 2: センサーを使って、micro:bit を裏返す向きによって1~4の数字を表示 3. micro:bit で神経衰弱 3: 2台の micro:bit 間で通信して、神経衰弱
第8週	micro:bit と MakeCode で IoT 入門 第4回 1. MakeCode で JavaScript 入門 2. micro:bit で温度計 3. Google Cloud Platform 紹介: 機械学習 API 利用体験 - Vision API と Translation API
第9週	Google Colaboratory で Python 入門 第1回
第10週	Google Colaboratory で Python 入門 第2回
第11週	Google Colaboratory で Python 入門 第3回 1. ディープラーニングを用いた画像認識 (OpenCV)
第12週	手書き文字認識する LINE アプリを作ってみよう

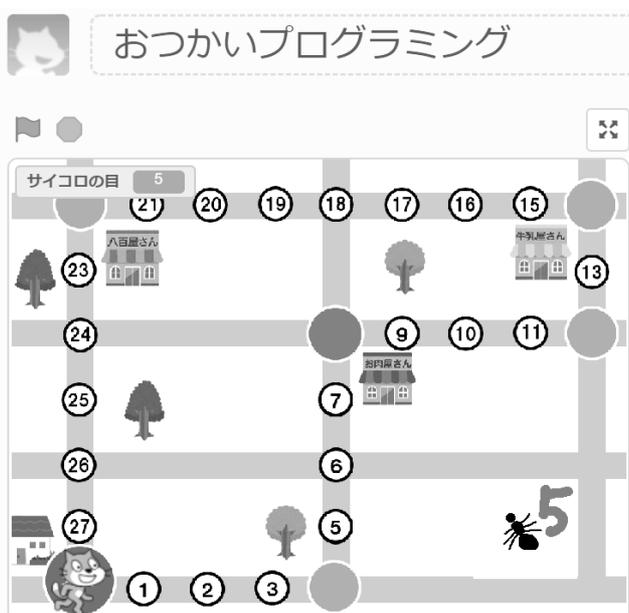


図1. 第2週で使用したステージ

第5週から第8週は、MakeCode ([7]) で micro:bit ([8]) のセンサー機能や通信機能を活用するプログラムを作成することで IoT を体験した (図 2)。micro:bit の活用は MakeCode 上のブロックエディタで始め、第8週には、MakeCode 上のテキストエディタで JavaScript を用いたプログラミングも体験した。これは、第9週からの Python でのプログラミングへの橋渡しでもある。



図 2. micro:bit 利用例

第9週から第11週は、機械学習技術に触れることも目的の一部として、Python を用いたプログラミングを学び、OpenCV の利用法を確認した。Python の開発環境は、開発環境を構築する必要なくウェブブラウザ上で学習できる Google Colaboratory ([9]) を利用した。また、Google Platform ([10]) の API を利用して機械学習や画像処理の機能を活用した。第12週は、普段使っているサービスがどのように実現されているかに興味を持ってもらうことを意図し、画像認識 API などを活用する LINE アプリを作成した。

教材の提示や受講者とのコミュニケーションなどには、Google Classroom ([11]) を利用した。第4週から第8週は、各自のペースで学べるよう、また、復習したい受講者や欠席した受講者が自習できるように第一著者が解説動画を YouTube で用意をした。希望する受講者には、いつでも学習できるように micro:bit を貸し出し、自宅に持ち帰らせた。また、第9週から第11週は、Google Colaboratory のノートブックで自習できるように教材を用意した。第11週の OpenCV も Colaboratory から利用をした。

3. アンケート

表2は「質問 ID1: 学校教育において、プログラミングはいつ頃から必修科目として学ぶべきと考えますか」という質問項目についての、本講座受講生の回答結果である。初回の講座開始時に取ったアンケートと最終回の講座終了時に取ったアンケートで、回答者が25名から7名に減っている。これは最終週に偶然欠席してしまった学生もいるが、講座を途中でドロップアウトした学生が多くいたことが主な原因である。

講座を通して受講者数が大きく減少したため、定量的な分析は困難であるが、受講前にはいなかった「高校で必修化すべき」という意見が実施後には2人に増えている。回答者の2名はいずれも、実施前には「中学で必修化すべき」と回答していた。変化した要因としては、講座を受けて、想像していたよりも内容が困難であったためと考えられる。その一方で、「中学で必修化すべき」から「小学校で必修化すべき」と実施前後で逆方向に回答を変えた受講者も1名いた。本講座では、ブロック型ビジュアル言語 Scratch よるプログラミングや micro:bit を用いたハードウェアと連携したプログラミンから Python による機械学習 API の活用まで、初心者にとってはイメージや扱いの異なる複数のテーマを通してプログラミングを学習したため、印象に差が生まれたと推察される。実施前に「必修化の必要は無い」と答えた2名は、実施後のアンケートには不参加であるが、最後まで受講した学生から必修化は不要という意見が出てこなかった。これはプログラミングを必修化として学習してこなかった学生たちが、受講した感想として、小学校から高校までのいずれかのタイミングでプログラミングを学習した方が良いと回答したことを意味しており、本報告のような講座の重要性が認識できる。

表 2. プログラミングをいつ必修として学ぶべきか

質問 ID 1	小学校	中学校	高等学校	大学	必修化の必要は無い
実施前 (n=25)	10	13	0	0	2
実施後 (n=7)	4	1	2	0	0

表3は「質問 ID5: プログラミングの学習で身につく知識や技能は、多くの人にとって、仕事の役に立つと思いますか」、表4は「質問 ID6: プログラミングの学習で身につく知識や技能は、日常生活の役に立つと思いますか」の結果をそれぞれ示したものである。

表 3. プログラミングは仕事の役に立つか

質問 ID 5	強く思う	思う	どちらとも言えない	あまり思わない	全く思わない
実施前 (n=25)	13	10	2	0	0
実施後 (n=7)	5	2	0	0	0

表 4. プログラミングは日常生活の役に立つか

質問 ID 6	強く思う	思う	どちらとも言えない	あまり思わない	全く思わない
実施前 (n=25)	8	8	6	2	0
実施後 (n=7)	5	2	0	0	0

特に質問 ID 6 (表 4) で問われている、日常生活の役に立つという意見の比率が全体的に上がっていることが確認できる。もちろん、講座に対する印象の良い受講生が残るとことは当然であるが、1名の受講生については、受講前は「あまり思わない」と回答していたにも関わらず、受講後には「強く思う」とポジティブな回答へと変化している。このような変化について、プログラミングが技術者やエンジニアの専門スキルではなく、論理的思考力や想像力・創造力などの日常生活により近いスキルとして、講座を通して認識されたことが窺える。また、IoT や機械学習の一端に触れたことも効果があったかもしれない。

以上のように、人文系・社会系の学生についても本報告のようなプログラミング講座を実際に受講することにより、プログラミングや取り巻く技術に対する意識や認識に変化が生じたといえる。

4. おわりに

本稿では、プログラミング経験のない人文系・社会系学生を対象に実施した課外プログラミング講座を報告した。前節で述べたように一定の成果は認められたが、講座参加者数が、第1週の25名から第2週は18名に、第3週の17名から第4週は12名と減ったことなど課題も多い。各回の難易度設定や進度が60分という1回あたりの講座時間に比して適切ではなかったと思われる。IoT や機械学習といったこれから社会や生活の中でますます活用されるであろう技術に触れてもらうことも目的の一部として本講座を実施し、その効果も見られたが、このようなプログラムの活用に触れてもらうことと、プログラミング自体を学ぶこ

と・楽しむことやプログラミングを通して自ら創造することとのバランスをどのように設定するのも課題である。2020年度から開講する「アルゴリズムとプログラミング基礎」「アルゴリズムとプログラミング応用」などの授業設計に活かしたい。

参考文献

- [1] 原田康徳, 小学生に分かるコンピュータサイエンスとしてのプログラミング教育—ビスケットを用いて—. 情報処理, 2016, Vol.57, No.4, p.344-348.
- [2] 「平成 29・30 年改訂 学習指導要領、解説等」.
https://www.mext.go.jp/a_menu/shotou/new-cs/1384661.htm (参照 2020-02-20).
- [3] 鹿野利春, 学習指導要領の改訂と共通教科情報科. 情報処理, 2017, Vol.58, No.7, p.626-629.
- [4] 「カリキュラム標準一般情報処理教育 (GE)」.
https://www.ipsj.or.jp/annai/committee/education/j07/ed_j17-GE.html (参照 2020-02-20).
- [5] 「日本大学文理学部」. <https://www.chs.nihon-u.ac.jp> (参照 2020-02-20).
- [6] “Scratch”. <https://scratch.mit.edu> (参照 2020-02-20).
- [7] “MakeCode”. <https://www.microsoft.com/ja-jp/makecode> (参照 2020-02-20).
- [8] “micro:bit”. <https://microbit.org/ja/> (参照 2020-02-20).
- [9] 「Colaboratory へようこそ」.
<https://colab.research.google.com/notebooks/intro.ipynb?hl=ja> (参照 2020-02-20).
- [10] 「Google Cloud Platform の概要」.
<https://cloud.google.com/docs/overview?hl=ja> (参照 2020-02-20).
- [11] “Google Classroom”.
<https://edu.google.com/intl/ja/products/classroom/> (参照 2020-02-20).