

# ノートブック形式のプログラミング環境サービス Google Colaboratory を利用した C 言語入門授業の試み

館野 浩司<sup>1</sup>

**概要：**近年はデータサイエンスおよび機械学習のチュートリアルなどの用途で Google Colaboratory というノートブック形式のプログラミング Web サービスが利用されることが多く、大学などにおけるプログラミング授業でも利用される。このノートブック形式によるプログラミングは、処理の単位ごとに出力を確認しながら作業を進めつつ、その手順の記録も残すことができ、またさまざまな実行パターンを試した記録も残すことができるという利点があり、教育現場になじみやすい。Google Colaboratory 上ではプログラミング言語として Python が用いられることが多いが、本実践では C 言語を用い、入門講義において利用した。本報告では、C 言語入門講義における Colaboratory 利用上の工夫と、利点、欠点などを合わせて紹介する。

**キーワード：**プログラミング授業、Google Colaboratory、C 言語入門

## 1. はじめに

大学工学部初年度の C 言語プログラミング入門授業における実践報告である。プログラミング環境は Google Colaboratory (以下、Colaboratory と略す)、およびパソコン上の TCC (Tiny C Compiler、作者サイト [1]) を用いた。本稿では Colaboratory について述べる。なお、今回の授業形態はリアルタイム、オンデマンド併用のオンライン形式であった。実施期は、2021 年後期 (2021 年 9 月下旬から 2022 年 1 月下旬) である。

近年はデータサイエンスおよび機械学習のチュートリアルなどの用途で Colaboratory が利用されることが多く、大学などのプログラミング授業でも利用される。このノートブック形式によるプログラミングは、処理の単位ごとに出力を確認しながら作業を進めつつ、その手順の記録も残すことができ、またさまざまな実行パターンを試した記録も残すことができるという利点があり、教育現場になじみやすい。

通常、Colaboratory 上ではプログラミング言語として Python が用いられることが多いが、事前の調査で同サービスは C 言語でも有用であることがわかり、今回の授業での利用を決めた。その利用上の工夫と、このようなノートブック形式の C 言語プログラミング入門講義における利点、欠点などを合わせて紹介する。

## 1.1 Google Colaboratory と Jupyter Notebook

Colaboratory は Jupyter Notebook をベースにしたツールである (GitHub における初期の開発リポジトリ参照 [2])。Jupyter Notebook では、Jupyter カーネルと呼ばれるプロセスが動作しており、コードセルに記述されたコード片の実行、シンタックスチェックなどを司る。Jupyter Notebook 自体には数十タイプのカーネルがあり、さまざまなプログラミング言語に対応している (Jupyter Project の Wiki 参照 [3])。

Colaboratory はデフォルトでは 2022 年 1 月現在、Python にしか対応していないが、若干の設定、インストール作業をすることにより Python 以外の言語カーネルで動作させることもできる。

今回の授業実践では、デフォルトの Python カーネルは変更せずに C 言語プログラミングを行った。

## 2. 授業科目対象学生と授業の目的および概要、授業形態など

### 2.1 対象学生と授業の目的および概要

対象は大阪工業大学工学部 1 年生であり、科目名は「基礎情報処理 II」である。内容はプログラミング入門である。学科ごとに使用する言語が異なっており、C 言語を利用するのは電子情報システム工学科、機械工学科である。これら 2 学科の筆者担当クラスで実践した。担当クラスの受講者人数は、それぞれの学科について 37 人、50 人、合計 87

<sup>1</sup> 大阪工業大学、同志社大学

人である。

授業の目的は工学部生の基本スキルとしてプログラミングの基礎を学ぶということである。変数の扱い、分岐処理、繰り返し処理、ユーザ関数の基本までを学習する（詳細は付録 A.1）。

## 2.2 授業形態

リアルタイム、オンデマンド併用のオンライン授業である。学生はどちらの形式かを自由に選択できる。

## 2.3 学生のスキルレベル

ほぼ全員がプログラミング初心者である。一部の学生のみ工業高校などで C 言語プログラミングの経験がある。また、電子情報システム工学科の学生は、同時期に開講される別の PBL 授業において Arduino を用いた演習を行っており、そこでも C 言語に触れている。

タイピングについても、まだ慣れていない段階の学生も多い。記号類の入力について丁寧に指導する必要がある。

さらには、フォルダの扱いやファイル名についての基本的なルール（拡張子など）についても不慣れた学生も後期の時点でまだ存在する。ターミナルアプリからのコマンド入力についても未経験の場合がほとんどである。

## 3. 従来の方法における問題点

ここで、従来の方法とは、テキストエディタでソースファイルを作成し、ローカルにインストールされたコンパイラを用い、ターミナルアプリ（コマンドプロンプト、Windows Terminal など）で、コンパイルおよび実行する方法のことを指すとす。この方法で授業を行う際、次のような問題点がある。

- (1) 使用するウィンドウが複数あり、学生が切り替えに手間取る
- (2) 以前の実行結果がスクロールして流れてしまい、入力値による処理結果の違いの比較などがしにくい
- (3) 通常、授業時に複数のプログラムを扱うが、それらの実行結果が単一のターミナルアプリに表示されるという運用が多く、どの結果がどのコードのものかが分かりにくい
- (4) 学生が記号の入力について不慣れたため、ソースコードのみならずコマンドラインの入力ミスでコンパイルできないことがある
- (5) エラーメッセージを読む余裕がない。読んでも修正作業しているうちにスクロールしてしまっ見えなくなり、記憶に残りにくい

このような環境でプログラミングできるようになることも必要なスキルではあるが、初心者にとって超えなければならないハードルは少なくはない。上記の問題点について、



図 1 A Tour of Go

教員側で軽減、分割などの対策を考慮しておくことは必要であろう。

## 4. 先行例、類似例

ノートブック形式のプログラミング環境ということであれば、かなり古くまで遡れるが、ここではウェブブラウザから手軽に利用でき別途アプリを導入する必要のないものに限定する。Colaboratory の利用方法のヒントとするため、あるいは、利点、欠点の考察のため、いくつか列挙する。

### 4.1 弘前大学情報連携統括本部情報基盤センター

[4] では、さまざまな言語のカーネルで Jupyter Notebook が利用できるようである（学内専用）。C 言語カーネルもあり、使い方が例示されている。

### 4.2 Eloquent JavaScript

Marijn Haverbeke 氏<sup>\*1</sup>によるウェブサイトおよび書籍 Eloquent JavaScript [5] は JavaScript ベースのプログラミングチュートリアルである。C 言語ではないが、HTML ドキュメント内のコードが実際にその場で編集、実行できる。コード部分も紙の本に印刷されたような見た目だが、クリックするとその場で編集、実行ができるようになるのが特徴である。10 年以上継続的に版を重ねている。紙に印刷されたものも出版されており、最近、日本語訳も出版された。

### 4.3 A Tour of Go

[6] は Go 言語の公式チュートリアルである。

これは説明文とコード部分がウェブページの左右半々のレイアウトとなっている（図 1）。また、決まった大きさのページ概念がありページ全体を一目で見渡すことができる。Colaboratory の場合、ノートブックは縦に長くなるのが常であり、全貌を見るにはスクロールしなければならないので、この点は大きく違う。A Tour of Go のような形式は、大学でのプログラミング授業でも参考になるであろう。

<sup>\*1</sup> CodeMirror などの開発者。https://marijnhaberbeke.nl/

#### 4.4 MDN Web Docs, JavaScript Primer

Mozilla 財団の各種技術ドキュメントにおいて、各項目でコード片が実行可能となっている。たとえば `if...else` 文の項目 [7] では、ページ冒頭でコードを書き換えて実行することができる。

ページ冒頭以外でもコード片の書き換え、実行を試せる例としては JavaScript Primer というサイトがある (書籍版もあり)。たとえば条件分岐の項目 [8] において、単一ページ内に複数の書き換え、実行可能なパーツが埋め込んである。こちらは、前掲の Eloquent JavaScript サイトにより近い形となっている。

### 5. Colaboratory における C 言語プログラミングの方法

本実践では新たに C 言語用の Jupyter カーネルをインストールするという方法はとらなかった。Colaboratory ノートブックを新規作成してからコードを書き始めるまでに、こうした操作が入るとコーディングが億劫になるという懸念があるからであり、また、プログラミング初心者にとって、より心理的ハードルが高くなると思われるからである。その代償として多少の不便は甘受する必要が出てくる。こうした点については後述する (8 節参照)。

以下で Colaboratory デフォルトである Python 用 Jupyter カーネルでの C プログラミングの方法をいくつか紹介する。

#### 5.1 ソースファイルを作成後、コンパイル、実行

まず Colaboratory が動作しているインスタンス上にソースファイルを作成し、ついで、それを別のコードセルからコンパイラを起動してコンパイルし、実行するというのが、素直な方法であろう。ソースファイルの作成には次の 2 つの方法が考えられる。

- (1) `%%file` マジックコマンドを利用 (図 2)
- (2) Colaboratory のファイラーから新規作成  
作成したファイルをダブルクリックすると、Colaboratory 内で編集できる。

```
[22] 1 %%file hello.c
2 #include <stdio.h>
3
4 int main(void)
5 {
6     printf("Hello");
7
8     return 0;
9 }
10

Writing hello.c

[23] 1 !clang hello.c && ./a.out

Hello
```

図 2 `%%file` マジックコマンドを利用したソースファイル作成

#### 5.2 ヒアドキュメントを利用

次のような方法もある。

`%%bash` または `%%shell` マジックコマンドにより、コードセルにシェルスクリプトを記述できる。そこで 図 3 のようにヒアドキュメントでコードを記述し、同じコードセルでコンパイル、実行まで行う。

```
1 %%bash
2 (clang -x c - && ./a.out) << CODE
3 #include <stdio.h>
4
5 int main(void)
6 {
7     printf("Hello");
8
9     return 0;
10 }
11 CODE
12
```

□ Hello

図 3 ヒアドキュメントによるコードの生成、コンパイル、実行

これは実行ボタンを一度押すだけで、ソースの生成から実行までできるので、手軽にワンライナーの感覚でプログラムが作成できる。

しかし、特殊な方法であり入門者には勧められない。授業時に配布する Colaboratory ノートブックで何か補助的な処理をするために利用するのがよさそうである。たとえば、データの統計処理を課題とする場合、その場で乱数を利用してデータを生成する場合などに利用できるであろう。学生はその生成過程を見る必要がないのであれば、この方法が取れる。

### 6. さまざまな工夫例

#### 6.1 各種ウェブアプリの埋め込み

コードを見せての説明だけでは不十分な場合は、Colaboratory に各種アプリを埋め込んで説明した。

- (1) 外部スライド、動画サービスの埋め込み (図 4)
- (2) Scratch (図 5)、micro:bit (MakeCode)
- (3) 数学アプリ GeoGebra (図 6)、Desmos
- (4) ステップ実行アプリ Python Tutor、C 言語版 (図 7)
- (5) 自作アプリ (`%%html` マジックコマンドを利用)

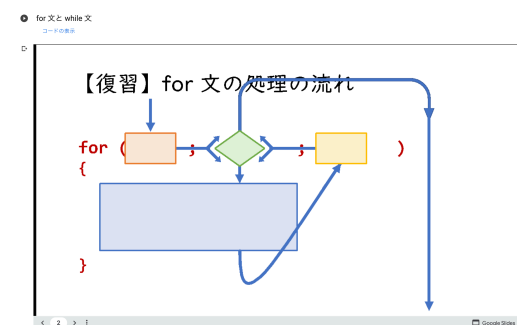


図 4 Google Slides を Colaboratory に埋め込んだ例



図 5 Scratch アプリを Colaboratory に埋め込んだ例

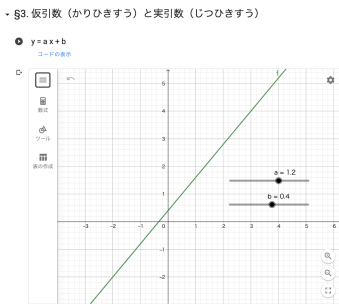


図 6 Google GeoGebra アプリを Colaboratory に埋め込んだ例

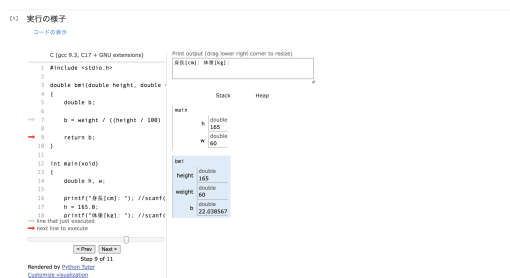


図 7 ステップ実行アプリ (Python Tutor、C 言語版) を Colaboratory に埋め込んだ例

## 6.2 コマンドラインツールの利用

Colaboratory は Linux マシン上で稼働しているので、Linux のコマンドが利用できる。インストールされていないコマンドもパッケージマネージャ (APT) で追加インストール可能である。入門者向けの授業においても次のようなツールは役立つ。

- (1) 16 進ダンプ
- (2) コード整形ツール clang-format など
- (3) シンタックスハイライタ

## 6.3 クリッカーの設置

学生の作業進行の様子をクリッカーを設置して把握した。クリッカーは Goola App Script を利用して Web API を作成し、Colaboratory の HTML ボタンを押すと呼び出されるようにした。

今回は前回オンライン授業であり、とくにクリッカーが役にたった。

## 6.4 その他の工夫

### 6.4.1 C で画像ファイルを生成し、ノートブック内で表示

たとえば、SVG ファイルを C 言語のコードで生成して Colaboratory 上で描画するということができる。ただし、生成された SVG ファイルの描画には、Python のコードを利用する。図 8 は実際の課題の例である。

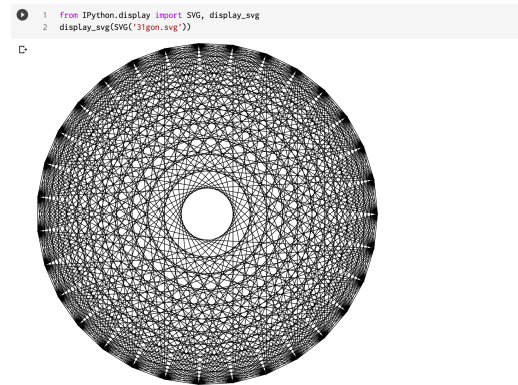


図 8 C で生成した SVG ファイルを Colaboratory 上で表示

SVG ファイルを生成する C 言語プログラムは、[10] 所収のユーティリティ関数群 (svgplot.c) を利用した。

### 6.4.2 飾り罫線の挿入

Colaboratory では、セクション間の境目が狭く、授業時に学生がノートブックの現在位置を見失う可能性がある。そこで、飾り罫線画像を適宜挿入し、セクションの区切りを見やすくした (図 9)。また、各セクションはあまり長くしないようにした。

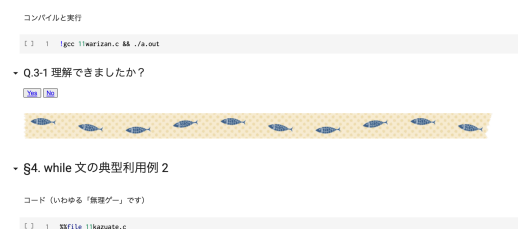


図 9 節の区切りに挿入した飾り罫線

### 6.4.3 コーディング用フォントの導入

C コンパイラとして、GCC や Clang を利用すると警告メッセージやエラーメッセージ内で、問題がある箇所を波線を引いて指し示されることがある。ブラウザのフォントがデフォルトのままであると、この表示がずれることがあるので、日本語対応の等幅フォントをインストールさせる。今回の担当クラスでは Myrica Monospace (配布サイト [9]) をインストール、設定させた。

### 6.4.4 課題に利用したノートブックを PDF 化して提出

学生が授業時間内に取り組んだノートブックは、ブラウザの PDF 化機能を利用して PDF 化させて提出させた。提出先は Google classroom とし、タブレット (iPad) でペ



図 10 授業開始時に課す復習問題

入れ後、Google classroom のシステムで返却した。

この際、次の 2 点の印刷設定をさせるべきであろう。

(1) 背景を表示

採点の際、コードセルの背景に色がついて採点対象が視認しやすくなる。

(2) ヘッダー、フッターの非表示

提出された PDF ファイルは、一般に複数ページなるので、ヘッダー、フッターは採点の際の邪魔になる。

## 7. 本実践における授業サイクルと授業用ノートブック例

今回の実践における授業の進め方について述べる。

### 7.1 授業サイクル

各回の授業は次のように進めた。1 コマ 100 分である。カッコ内の時間はおよその時間である。回によって変動する。

(1) 前回の復習 (20 分)

- (a) エラー修正 (図 10)
- (b) コード整形の練習
- (c) 穴埋め

(2) 新規事項説明 (30 分から 40 分)

学生の簡単な作業もあり (コードを一部変更、追加など)。

(3) 演習 (30 分から 40 分)

(4) 発展的内容の説明 (10 分)

(5) 出席アンケート (5 分)

アンケートの自由記述欄へリアルタイム時間内に書かれたコメントへは即返答

(6) Colaboratory ノートブックを PDF 化させ Google classroom へ提出させる

(7) 提出された PDF を添削して Google classroom 経由で返却

(8) 別途宿題あり

(9) 次回の授業のための予習課題



図 11 授業で使用したノートブックの例

## 7.2 授業時に配布するノートブックの例

筆者の GitHub リポジトリからサンプルを参照可能である (図 11)。

## 8. Colaboratory における C 言語入門授業の利点、欠点など

今回の実践を通して気づいた Colaboratory での C 言語入門授業の利点、欠点についてまとめておく。

### 8.1 利点、有利な点など

- ソースファイル作成、コンパイル、実行の流れを可視化できる
- 慣れていないうちは、コンパイルコマンドを授業資料のノートブックに事前入力しておくことができる
- 従来法でソースファイルを複数配布する場合でも、ノートブック 1 枚だけ配布すればよい

たとえば、間違いを含むソースコードを複数配布して、デバッグさせるという課題を行う場合、ノートブック形式なら 1 つのファイルを配布するだけでよいし、説明の際も、学生が現在どのファイルについて説明しているかを見失うことが少なくなる。

- コーディング前にテストケースを列挙させることが容易にできる
- オンライン授業で受講しやすい

今期は前回オンライン配信としたので、Colaboratory による授業は受講しやすかったのではないかと思われる。この形式は従来法に比較して開く画面が少なく、切り替える必要もない。

### 8.2 欠点、不利な点などとその対処

- コードセルの先頭行の 1 行目にマジックコマンドを入れる必要がある

これは本来のコードにはないものである。Colaboratory 上でプログラミングするために必要な行であるということを学生にはっきり認識させる必要がある。以下の項目でも述べているが、毎回の宿題はかならず

パソコン上ローカルにインストールされたプログラミング環境を使用するというルールにしていた。これでこの欠点に対処できていたようである。

- コンパイラのメッセージで指摘される行番号とコードセルの行番号が1つずれる。

1行目にマジックコマンドがあるせいである。しかし、これはあまり問題にはならなかったようである。学生は特に混乱しているようではなかった。これについては、さらに詳しく学生の様子を観察する必要がある。

- ファイルの扱いについて練習する機会が少なくなる

Colaboratory 上だけでプログラミングしていると、ファイルに適切に名前をつけて決まったフォルダ内に保存して管理するという習慣が身につかない。この点についての対処として、宿題はパソコンにローカルにインストールされたエディタとコンパイラを使用させた。提出物は、エディタとターミナルアプリのスクリーンショットとした。

## 9. 学生の反応

第13回目(全14回)の授業のアンケートで、Colaboratory についての感想、意見を自由記述形式で書いてみたところ、おおむね好評のようであった。そのいくつかのコメントを紹介する。

- とても使いやすかったし、しっかりエラー文も出て、自分がどこで間違えているかなどが分かった。プログラミングの処理の手順がわかりやすかった。
- 楽しかった。
- 自動的に保存されて間違えたときにすぐ元に戻せて良いと思います。
- 上から順番に入力するだけでC言語が構築されていくので、理解しやすかった。
- 使いやすかった。
- この授業で初めてプログラミングに取り組んだが、初めてでも間違えたところが分かりやすかったので良かった。これからも触れられるときに触れてみたいと思った
- 一つのページで完結できるのでいいと思った。
- いちいち Mery に一つずつコードを書く必要がないので理解しやすかった。
- Colaboratory で特に問題なくできました。
- 正直、よくわかっていない。
- 極めると使えそう
- プログラムから実行までの手順が少なく楽だと感じています。
- Mery と Terminal より使いやすい。
- ツール自体は良いものだと感じました。使いやすかったです。

## 10. おわりに

C 言語入門演習授業において、Colaboraty (および、これに類するノートブック形式のプログラミング環境) の利用はいくつかの欠点があり、他の手段の併用が必要であるが、それを補って余りある利点がある。学生の反応を見るに、事前に危惧された欠点も、それほど問題ではないようである。また、例年より実質的な内容に触れる時間の余裕が増えた。

こうした利用ケースについて、今後も継続的に注視および実践していきたい。

**謝辞** 本実践の事前調査、および実践時において大阪工業大学情報センター越智徹先生にアドバイスをいただき大変お世話になりました。厚くお礼申し上げます。

## 参考文献

- [1] Bellard, F.: *Tiny C Compiler* (online), 入手先 <https://bellard.org/tcc/> (2022.1.8).
- [2] Project Jupyter: *GitHub - jupyter/colaboratory* (online), 入手先 <https://github.com/jupyter/colaboratory> (2022.1.12).
- [3] Project Jupyter: *Jupyter kernels · jupyter/jupyter Wiki · GitHub* (online), 入手先 <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels> (2022.1.12).
- [4] 弘前大学情報連携統括本部情報基盤センター: *Jupyter Notebook の利用法 - HEROIC 2021* (online), 入手先 <https://home.hirosaki-u.ac.jp/heroic-2020/jupyter/> (2022.1.12).
- [5] Haverbeke, M.: *Eloquent JavaScript* (online), 入手先 <https://eloquentjavascript.net/> (2022.1.7).
- [6] Google: *A Tour of Go* (online), 入手先 <https://go.dev/tour/welcome/1> (2022.1.7).
- [7] Mozilla and individual contributors: *if...else - JavaScript* (online), 入手先 <https://developer.mozilla.org/ja/docs/Web/JavaScript/Reference/Statements/if...else> (2022.1.12).
- [8] azu, Suguru Inatomi: *条件分岐 · JavaScript Primer* (online), 入手先 <https://jsprimer.net/basic/condition/> (2022.1.12).
- [9] Estable: *プログラミングフォント Myrica* (online), 入手先 <https://myrica.estable.jp/> (2022.1.12).
- [10] 奥村 晴彦: [改訂新版] *C 言語による標準アルゴリズム事典*, 技術評論社 (2018).

## 付 録

### A.1 「基礎情報処理 II」 シラバスより

第 1 回	ガイダンス	ノート PC で C 言語を演習する廃棄環境の使い方を説明する。実際にコンパイルできることを確認する。
第 2 回	操作環境とコンパイルの仕方	C プログラムの作り方について学ぶ。
第 3 回	標準出力	データの標準出力について実習を行う。
第 4 回	変数、データ型、配列	変数、データ型、配列について実習を行う。
第 5 回	演算子	演算子・型変換について実習を行う。
第 6 回	データの入出力	データの入出力について実習を行う。
第 7 回	演習 (1)	これまでに内容について復習し、演習課題を行いながら理解を深める。
第 8 回	選択処理 (1)	if 文、if~else 文を利用した選択処理について実習を行う。
第 9 回	選択処理 (2)	選択処理における論理演算子について実習を行う。また前半の取組みについてフィードバックする。
第 10 回	繰り返し処理 (1)	for 文を利用した繰り返し処理について実習を行う。
第 11 回	繰り返し処理 (2)	while 文を利用した繰り返し処理について実習を行う。
第 12 回	関数	関数の仕組みについて実習を行う。
第 13 回	演習 (2)	これまでの内容について復習し、演習問題を行いながら理解を深める。
第 14 回	総復習	総復習を行う。