

プログラミング応用 ガイダンス

清水 伸高 (塩浦研 助教)

- **講義**: アルゴリズムの理論的側面と様々な最適化アルゴリズムを学ぶ.
 - **演習**: 講義で紹介したアルゴリズムを応用して出題した問題をPythonで解いてもらう.
-
- 評価基準
 - 演習課題 (前日に課題ファイルを提示するので, 1週間後までに提出)
 - Pythonでの実装が中心
 - 最終回の課題(期末課題)は多め(比重が高い)
 - 期末試験はなし

受講に必要な環境

講義の受講にあたっては以下の環境が**必須**:

- 大学のSlackアカウント
 - 講義の連絡や課題ファイルの共有に利用する (LMSでも課題ファイルをDLできる)
 - 手続きについてはこちらを参照
 - この手続きが済んでいれば, 私の方でチャンネルに追加します.
 - 講義に関する私への連絡は**必ずSlackのDM**で行ってください
- LMSへのログイン
 - 課題の提出に利用する
- Pythonの実行環境
 - 個人のGoogleアカウントがあれば, Google Colabを利用できる (特に何もインストールしなくてよい).

(個人的に)オススメな環境

将来的にプログラミングに携わる可能性があるならば, 以下のソフトやスキルを身につけておくとよい:

- **VSCode** (Visual Studio Code)
 - エディタ(プログラムを書くソフト)のデファクトスタンダード
 - 拡張機能が豊富で, Pythonの実行環境も整えやすい
 - 生成AI (GitHub Copilot) を用いたコーディングも可能
 - 最近は生成AIを用いたコーディングに特化したエディタ**Cursor**も人気 (私はこれを使っています)
- **GitHub** (バージョン管理システム)
 - ソースコードの「セーブデータ」の履歴を管理できる (例えば学位論文の執筆で便利)
 - GitHub Pagesを使ってウェブサイトを作成できる
 - VSCodeにはGitHub用の拡張機能がある
 - 踏み込んだ使い方をしようとすると学習コストが高い

(個人的に)オススメな環境

■ Markdown形式

- テキストファイルの書き方の一つで, 拡張子は `.md`
- 簡単に言うと「htmlの簡易版」(可読性が高い)
- Wordで書くよりもAIに学習させやすい
- このスライドもマークダウン形式で書いている

■ NotionまたはObsidian

- Markdown形式でドキュメントの作成・管理に便利 (勉強ノートや研究ノートの作成に便利)
 - 去年はNotionで講義資料を作っています
- Notionは初心者でも直感的で使いやすく, Obsidianは拡張機能が豊富で細かいカスタマイズが可能

生成AIの利用について

生成AIを使ったCodingは推奨

- 例えば, GitHub Copilotを使うと, コードの自動補完やコードの説明をしてくれる
 - Google ColabではGeminiが使える
- ただし, 間違える可能性があるので, 出力されたコードをそのまま使うのではなく, 自分で確認すること
 - この講義の課題は主に数学の問題であり, AIでは解けないことが多い
- もちろん, 自分の力だけで書くのもOK

Google Colabについて

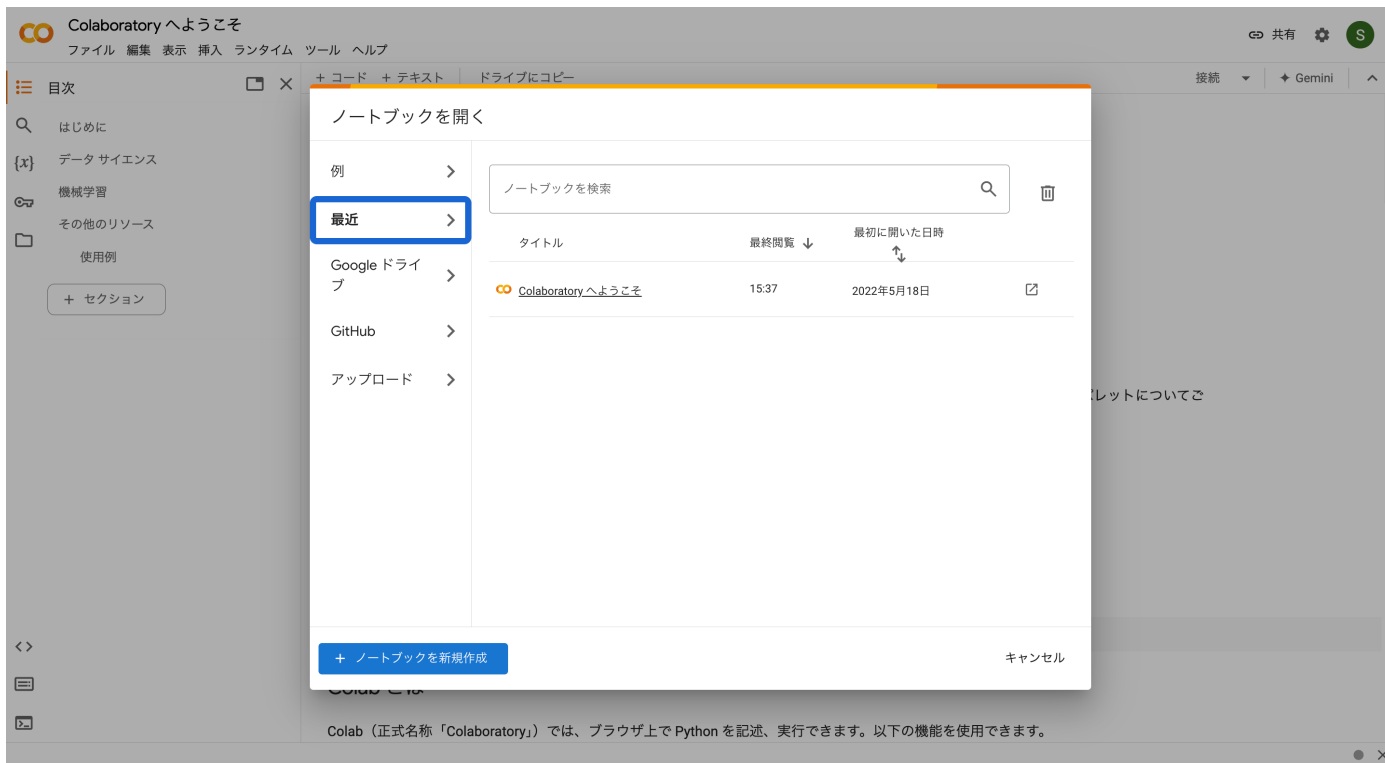
注意

以降はGoogle Colabの説明なのでPythonの実行環境が整っている方は聞かなくても構いません。

- 正式名称は**Google Colaboratory** (略称: Colab).
- Googleが提供する**Jupyter Notebook**のクラウド上での実行環境です.
 - Jupyter Notebook: 端的に言うと, プログラミング実行環境と文書をまとめたもの.
 - コードと文章を同じファイルに書けるので, 課題の提出に適している.
 - 拡張子は `.ipynb`
 - 三つの言語(**Julia**, **Python**, **R**)に対応している.
 - ブラウザでGoogle Colabのページにアクセスするだけで利用できる.

Google Colabの使い方

アクセスすると以下のような画面が表示される:



Google Colabの使い方

右下の「キャンセル」をクリックするとポップアップ画面が消え, 以下のような画面になる:



The screenshot displays the Google Colaboratory (Colab) interface. At the top, the header reads "Colaboratory へようこそ" (Welcome to Colaboratory) with navigation links for "ファイル" (Files), "編集" (Edit), "表示" (View), "挿入" (Insert), "ランタイム" (Runtime), "ツール" (Tools), and "ヘルプ" (Help). On the right, there are icons for "共有" (Share), settings, and a profile picture.

The left sidebar contains a "目次" (Table of Contents) section with icons and links for "はじめに" (Getting started), "データサイエンス" (Data science), "機械学習" (Machine learning), "その他のリソース" (Other resources), and "使用例" (Use cases). A "+ セクション" (Add section) button is at the bottom of the sidebar.

The main content area is titled "Colab へようこそ" (Welcome to Colab) and features a heading "(新規) Gemini API をお試しください" (Try the new Gemini API). Below this, there is a list of links: "Generate a Gemini API key", "Talk to Gemini with the Speech-to-Text API", "Gemini API: Quickstart with Python", "Gemini API code sample", "Compare Gemini with ChatGPT", and "More notebooks".

A paragraph follows, stating: "すでに Colab をよくご存じの場合は、この動画でインタラクティブなテーブル、実行されたコードの履歴表示、コマンドパレットについてご覧ください。" (If you already know Colab well, please watch this video about interactive tables, execution history, and the command palette). Below the text is a video thumbnail titled "3 Cool Google Colab Features" showing a man's face and a play button icon.

At the bottom of the main area, there is a code editor with a single line of code: "[] 1 コーディングを開始するか、AI で生成します。" (Start coding or generate with AI). Below the code editor, the heading "Colab とは" (What is Colab) is followed by a paragraph: "Colab (正式名称「Colaboratory」) では、ブラウザ上で Python を記述、実行できます。以下の機能を使用できます。" (In Colab (official name "Colaboratory"), you can write and execute Python in the browser. You can use the following features).

The bottom right corner of the interface shows a close button (X) and a page indicator "9 of 13".

Google Colabの使い方

左上の「ファイル → ノートブックをアップロード」でダウンロードしたipynbファイルをアップロードして開ける:

Google Colabの使い方

ダウンロードした課題ファイルを開くと, 以下のような画面になる:

▼ 問題例

二つの正方行列 A, B が与えられた時, その積 AB を出力する関数 `matrixmult(A,B)` を書け.

```
[ ] 1  def matrixmult(A,B):
2      n = len(A)
3      C = [[0 for _ in range(n)] for _ in range(n)]
4      for i in range(n):
5          for j in range(n):
6              for k in range(n):
7                  C[i][j] += A[i][k]*B[k][j]
8      return C
```

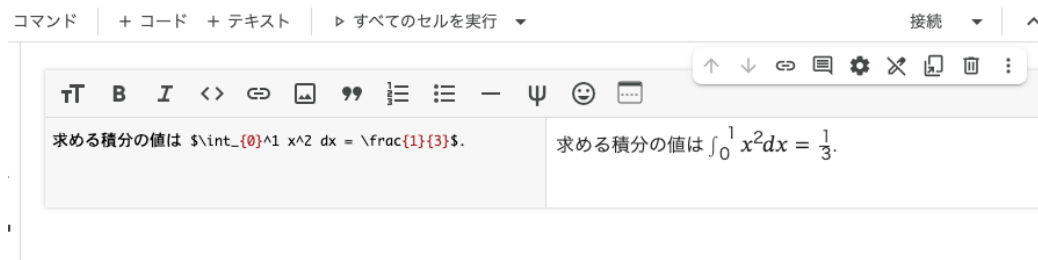
コードが書いてある欄(セルと呼ばれる)をクリックして, その中にコードを書いていく. セル内のコードの実行は, セルを選択してから上部の「▶」ボタンをクリック.

Google Colabの使い方

書き終わったら, 上部の「ファイル → ダウンロード → .ipynb」を選択して, ipynbファイルをダウンロードし, これをLMSを通じて提出すればよい.

記述問題について

- 課題の中には, コードを書くだけでなく, 数学的な証明や考察を求める記述問題もある.
- その場合は, 解答欄のテキストセルに回答を書いてください (ダブルクリックすると編集できるようになる)
- 数式の入力は, **LaTeX形式でなければならない**
 - セル内に `$数式$` と書くと, 数式が表示される
 - 例えば, `$\int_0^1 x^2 dx = \frac{1}{3}$` と書くと以下のようになります:



- 数式の入力方法については, 様々なサイトで解説されているので, そちらを参照してください (私に質問しても構いません).