

数理生物学演習

第1回 Hello, World!

野下 浩司 (Noshita, Koji)

✉ noshita@morphometrics.jp

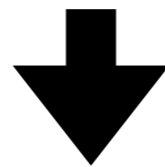
🏠 <https://koji.noshita.net>

理学研究院 数理生物学研究室

本演習の進め方

演習の基本方針

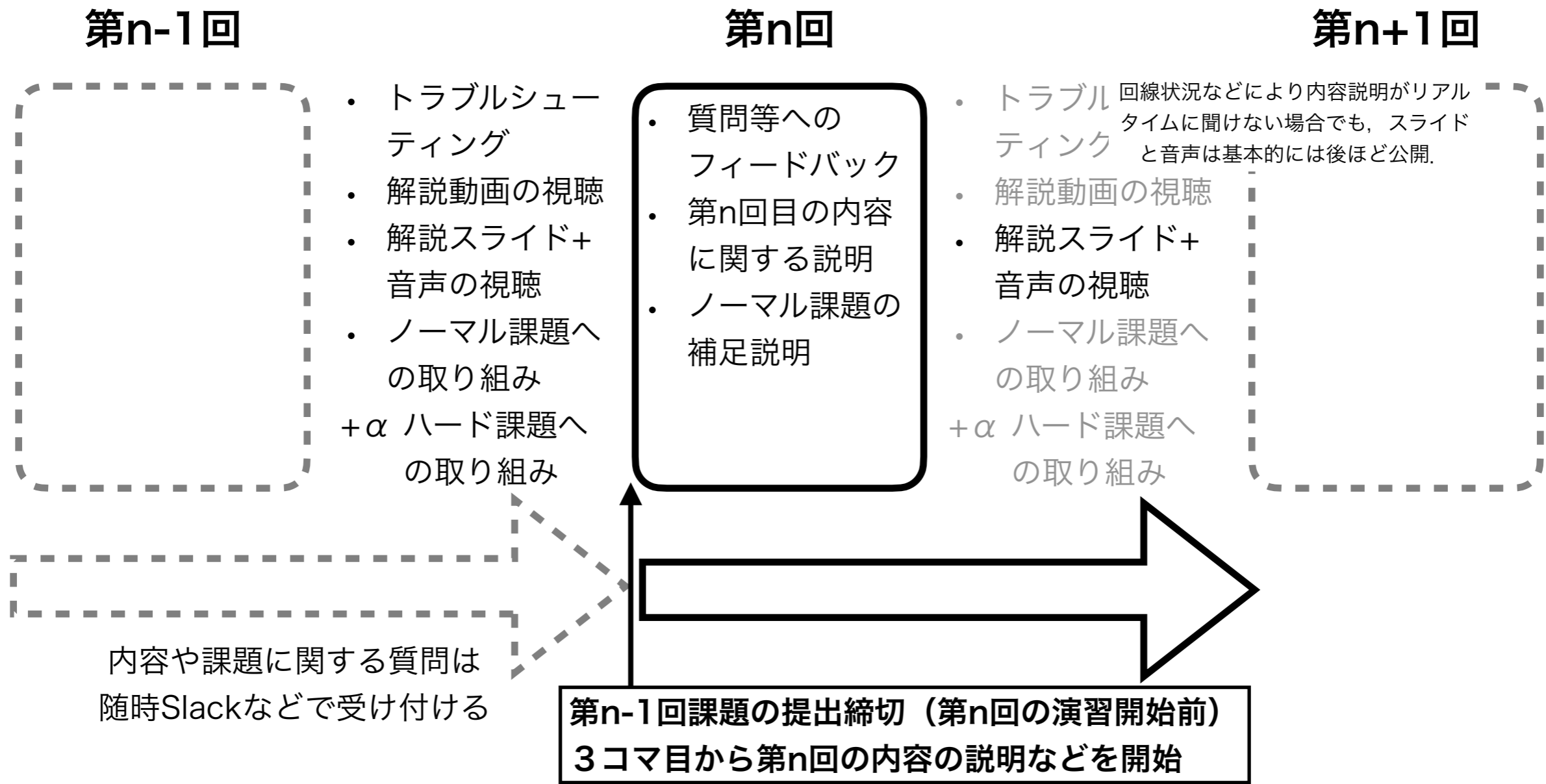
- ★★★ 数理生物学的トピック：数理生物学に興味を持ってもらう
例：ロトカ-ボルテラモデルによる被食者-捕食者の個体群動態の記述
- ★★ 研究への挑戦：研究を進める上で必要な、または役立つ経験をする
例：文献の探し方、使うと便利なツールの紹介
- ★ 数値計算的アルゴリズム：計算機で問題を解く方法を学ぶ
例：ニュートン法により方程式を数値的に解く
- ★★★ プログラミングのスキル：実際に具体的な問題を解いてみる
例：if文, forループ, 配列



自分の知りたいことへアプローチする方法を学ぶ

演習の進め方：基本的な取り組み

好きなタイミングで好きなだけ取り組む。締め切りまでに課題を提出すればOK。



変更があればWebページにて連絡。

今後の予定

原則対面実施（一部オンライン実施）。

Webページ（後ほど紹介）を随時確認すること！

1. Hello, World!
2. Pythonの基本的な使い方
3. 個体群動態の数理モデル（1）：離散ロジスティック成長モデル
4. 個体群動態の数理モデル（2）：指数増殖モデル, ロジスティック成長モデル
5. 個体群動態の数理モデル（3）：ロトカ-ヴォルテラ モデル
6. ランダムな現象：突然変異固定までの待ち時間
7. 空間構造の数理モデル（1）：理論形態学, Raupのモデル
8. 数理生物学は役に立つのか？（1）：疫学モデル, 研究紹介
9. 研究をはじめめるために：論文の検索, 管理, 引用
10. 空間構造の数理モデル（2）：人工生命, セルオートマトン, （ライフゲーム）
11. 空間構造の数理モデル（3）：パターン形成, （拡散反応モデル）
12. 数理生物学は役に立つのか？（2）：機械学習, 画像解析, 研究紹介
13. 最終課題発表会
14. 数理生物学でのプログラミング：まとめ, 最終課題へ向けた調整

現時点での予定なので, 今後変更される可能性あり.

成績

GPA目安

- S Excellent 90以上
- A Good 89-80
- B Satisfactory 79-70
- C Pass 69-60
- F Fail 59以下

理論的最大点

- 120

- **課題提出**：基本毎回提出。出席を兼ねる。Moodleにて提出。
 - ノーマル：操作説明と内容説明を視聴して、ある程度の試行錯誤と調べ物を行えばできるであろう応用的な課題
 - ハード：自分で積極的に調べたり試行錯誤を繰り返したりしなければならないであろう発展的な課題
 - ハードは希望者のみ（取り組みの量や難易度に成績が比例しない）。
 - ノーマル：60点（平均目安：45点），ハード：15点（平均目安：10点）
- **最終課題発表会**（第？回（第12回前後）に実施予定，チームでの実施)
 - 最終課題への方針についてプレゼンテーション
 - 10点（平均目安：8点）
- **最終課題**：Moodleにて提出。
 - 数理生物学的なテーマを設定し，計算機を使ったアプローチで取り組み，レポートにまとめる
 - 35点（平均目安：23点）
- ボーナス **バグ報告**：Slackの「#バグ報告」チャンネルで報告。
 - 単純なバグ指摘～修正の提案などに応じて1つの報告につき最大2点
 - 同じバグについて基本は先着1名。誰かが報告したバグに修正案を出してもOK。

TA紹介



工藤 秀一



中谷 優介



染野 大輝



松原 壮吾



浅井 駿平

手を挙げて or Slackなどで積極的に質問をしよう。きっとTAが助けてくれる。

本日の課題 ノーマル

1. 今回の授業でよくわからなかった点を述べる。特になければ、より知りたいと思った点を教えて。
2. 今後扱って欲しい内容を挙げる（複数可）。
3. 「Hello, World!」プログラムを作成せよ。
4. Slackの「#自己紹介」チャンネルに自分の自己紹介を投稿する。（例. ～～です. 趣味は～～です. 最近～～にハマっています. など）
5. その他質問, 感想, 要望をどうぞ。

！注意：第1回課題提出の締め切りは、第2回の開始前まで。

課題をノートブック (.ipynbファイル) にまとめて、Moodleにて提出すること

ファイル名は[回数, 01~15]_[難易度, ノーマル nかハード h].ipynb. 例. 01_n.ipynb 8

連絡

- Webページ

<https://koji.noshita.net/courses/compbio/compbio2024/>

- email

野下 浩司 noshita@morphometrics.jp

- Slack (後ほど説明します)

ku-compbio-2024.slack.com

- 訪問 (事前連絡してください)

伊都キャンパス ウェスト1号館 W1-C-906

研究室の様子を覗きに来るのも可

その場合はW1-C-909付近で誰か見つけて話しかけて！

第1回：Hello, World!

本日の目標

- プログラミング言語とは？
- 本演習で使うプログラミング言語の紹介
- プログラミング環境の構築

プログラミング言語って？



4人で3人組を
作ると余るのは？



```
0000110101110111110110  
1011010110010110101011  
0101011010100101010011  
1001101011100010101111  
0000100101100110100010  
10101.....
```

このままでは分かり合えない

→ 互いの言語の“翻訳”が必要

プログラミング言語って？

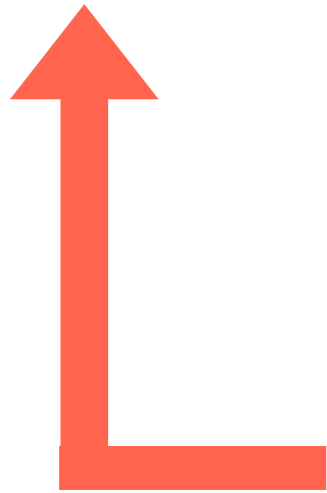


>> 4%3

“翻訳”：人 → コンピュータ

```
#include <stdio.h>
int main(){
  int m,n;
  m=4;
  n=3;
  printf(“%d人”, m%n);
  return 0;
}
```

RandomChoice[{"I", "you", "he", "she"}]

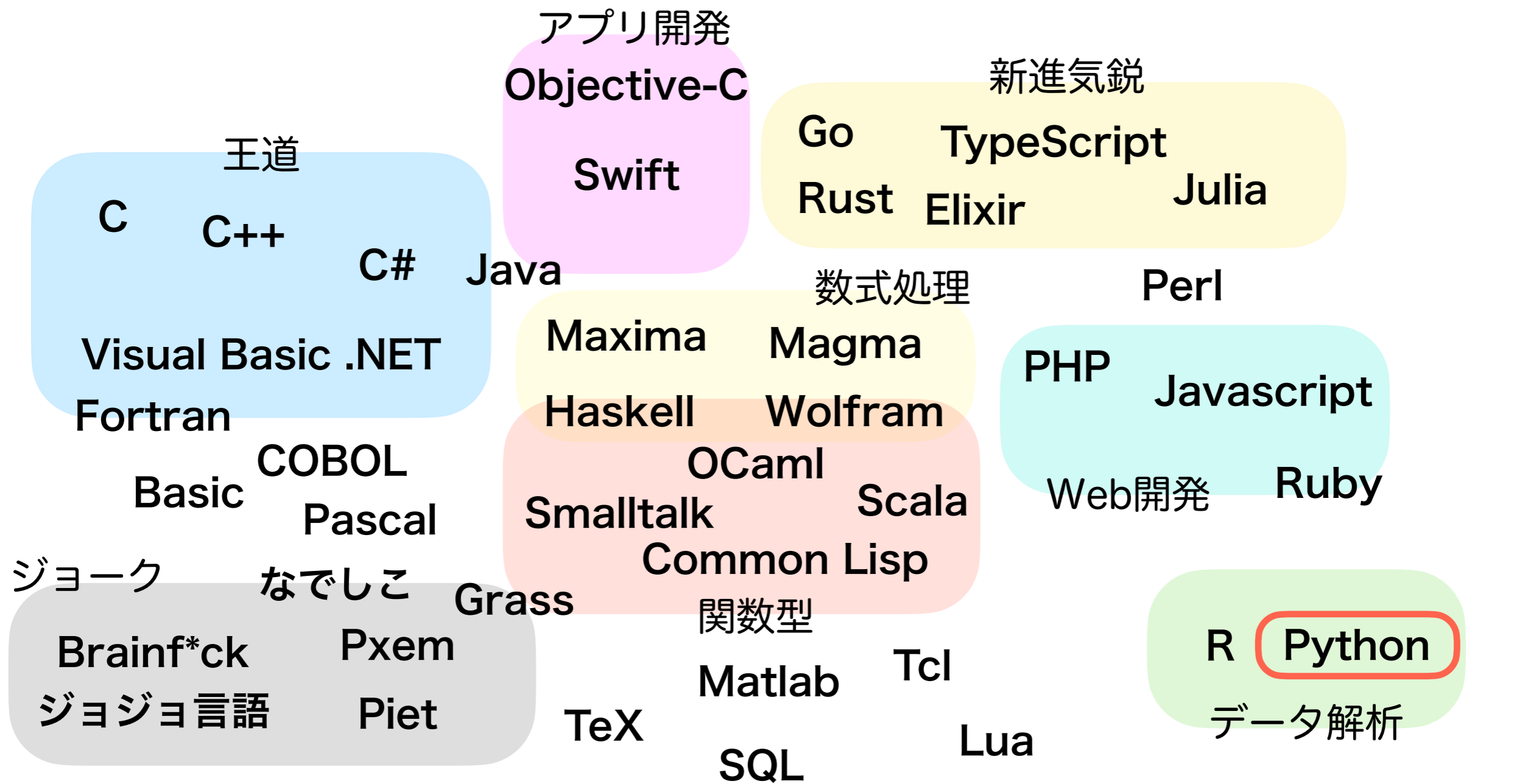


1
1人
you



“翻訳”：コンピュータ → 人

プログラミング言語いろいろ



コンパイラ
静的型付け
速い

※野下の主観がかなり入っているので目安程度に

インタプリタ
動的型付け
扱いやすい

Python

<https://www.python.org/>



この演習で利用するプログラミング言語

インタプリタ型, 動的型付け

用途: 数値シミュレーション, 可視化, データ解析

Pythonは数理生物学的なシミュレーションだけでなく, データ解析や画像解析などにも利用されている

→ 生物の分野で扱うデータやその解析にも利用できる. 様々な環境 (ラボとか職場とか個人開発とか) へ応用しよう.

Hello, World!

特に理由は無いけれどプログラミング言語を
学ぶと最初に出てくる練習用プログラム

目的

「Hello, World!」と出力（画面に表示）する

- きちんと環境の設定が出来ているか
 - Pythonはインストールできている？
 - パスは通っている？
 - 開発環境の準備は？
- …などをチェックする。

プログラミング環境を準備しよう！

演習 Webページ

<https://koji.noshita.net/courses/compbio/compbio2024/>

本演習の情報は基本的にここに公開される。ブックマーク推奨。

九州大学Moodle

<https://moodle.s.kyushu-u.ac.jp/>



オープンソースeラーニングプラットフォーム「Moodle」を利用した
九州大学のオンライン学習システム（の一部）

演習での活用方法

- 課題の提出
 - 超重要！提出方法については最後に説明します。

登録方法

1. 九州大学Moodleへログイン（SSO-KIDとパスワードが必要）
2. コースを検索する → 「数理生物学演習」をキーワードに検索
3. 「2024年度前期・月3月4・数理生物学演習」を選択
4. 「私を登録する」をクリック

Slack

<https://slack.com>



チャット系のコミュニケーションツール

演習での活用方法

- 質問する 「#質問」 チャンネル
 - 例) インストールが~で止まる, ~の図がきれいに表示されない
- 他の人の質問に答える 「#質問」 チャンネル
 - 例) ~にチェックを入れるといいよ, 途中の~の計算ができてないんじゃない?
- 情報を共有する 「#general」 「#雑談」 チャンネル
 - 例) ~の参考になるページを見つけた, ~は~からダウンロードできる
- バグを報告する 「#バグ報告」 チャンネル ボーナス点あり
 - コードの誤りやスライド・資料の間違いを指摘する
 - 例) ~を実行しても~にならない. ~ではなく~ではないか?
- その他にも趣味の話など様々な活用してください

登録・利用方法

- 大学メールアカウント (~@s.kyushu-u.ac.jp) へ招待を送るので, そこからアカウントの作成をおこなう
- PCやスマートフォン用のアプリがあるのでインストールすることを薦める

Slack デスクトップアプリの導入

- Slackへはブラウザや各プラットフォーム向けのアプリからアクセスできる
- 演習では演習に用いるPCにデスクトップアプリをインストールすることを推奨
- それに加えて、スマホのアプリなどもインストールしても良い

ダウンロードページ (<https://slack.com/intl/ja-jp/download>)

**Mac 版
Slack**

Slack アプリを使えば、ワンクリックでチームと連携できます。

ダウンロード

[Mac App Store からダウンロードする](#)

好きな方から導入してください

**Windows 版
Slack**

Slack アプリを使えば、ワンクリックでチームと連携できます。

ダウンロード (64 ビット)

[Microsoft Store からダウンロードする](#)

好きな方から導入してください

Slack登録後にやること

- 氏名 (full name) の設定

- 漢字, ひらがな, アルファベットのいずれでも構わないが, 本名で設定すること
- 例: 野下 浩司, のした こうじ, Koji Noshita

- 表示名 (display name) の設定

- 名字_学籍番号 (名字はアルファベット, すべて半角)
↑アンダーバー
- 例: noshita_1SC20****

- プロフィール写真の設定 (次回までに)

- 絵でも, 写真でも, 文字でも良いが,
他と見分けやすく, 混乱が生じにくいものを作成・採用する。

- 目安

- 画像サイズ 500px x 500 px 以上 (四角形がおすすめ)
- ファイルサイズ 1MB以下
- フォーマット png (透明部分は無いほうが良い)

プロフィールを編集する

氏名
野下 浩司

表示名
noshita

これは名字や名前、ニックネームなど、好きに設定できます。Slackでメンバーから呼ばれたい名前にしましょう。

役職・担当
役職・担当

数理生物学演習2020年度 でのあなたの役割を説明しましょう。

電話番号
(123) 555-5555

電話番号を入力してください。

タイムゾーン
(UTC+09:00) 大阪、札幌、東京

現在のタイムゾーン。サマリーや通知メールの送信、アクティビティフィードやリマインダーの時間に使われます。

フィールドを追加、編集、または順序変更する

キャンセル 変更を保存する

Google アカウント

<https://myaccount.google.com>



Googleに関連するサービスを利用するためのアカウント。
既にGmailやGoogleドライブを利用している人は、利用しているアカウントがこれに当たる。

Google アカウントの作り方

(既に持っている人は新たに作る必要はありません。それを使ってください)

- Googleアカウントのページ (<https://myaccount.google.com/>) へアクセス

Google アカウント

概要

プライバシー ツール



アカウントを作成する

Google アカウントに移動

- 「Googleアカウントの作成」をクリック
- アカウント情報を入力
 - 名前, ユーザー名 (これがGmailのアドレスになる), パスワード
- 電話番号を入力
 - SMSで確認コードが送られてくるので, それを入力
- 生年月日, 性別を入力
 - 再設定用メールアドレスを入力しておくとおとパスワードを忘れた際に助かる
- プライバシーポリシーへ同意
 - 問題がなければ同意する

Google ドライブ

<https://www.google.com/drive/>



Googleのオンラインストレージサービス。
本演習では、コードやその実行結果などを保存する。
また、Colab（後述）の外部ストレージとして利用する。

演習での活用方法

- コード等の保存
 - サンプルコード，自分が作ったコード
 - 出力した図，ファイル など
- Colabの外部ストレージ
 - Colabにマウントして，結果の出力
 - Colabに読み込ませるデータをGoogleドライブにアップロード など

最低150MB以上の空き容量を確保しておくこと。

もし不足している場合は，必要のないファイルの削除等の対応をしてください。

Googleドライブ内に演習用フォルダを作る

1. 「マイドライブ」の直下に「CompBio2024」というフォルダを作る



2. 「CompBio2024」の中に、「01」というフォルダを作る



今回は、この「01」の中にコードを配置していきます。

次回以降は新たに02, 03, …と回数を名前にしたフォルダを「CompBio2024」の直下に作って、毎回その中にコードを配置するようにしてください

皆さんへのお願い

- わからないところがあればすかさずググろう！
調べる習慣をつける。
- 質問や回答をSlackへ投稿しよう。
情報が共有できる。一人の質問が皆の質問に！
- 困ったら（手を挙げて/Slack上で）助けを呼ぼう
（特に，TAがサポートしてくれる）。困っている人がいれば助けてあげよう。
- 演習中の休憩は自由。疲れ果てる前に休もう。

ColabによるPythonの実行

Pythonの実行方法いろいろ

- 対話型評価環境

式やコードを入力し，それをインタプリタが解釈・実行し，結果を表示する．ちょっとした計算や確認に用いられる．

- ソースコード

プログラミング言語で書かれたテキストファイル（=ソースコード）を読み込んで，上から順に解釈・実行していく．大規模な計算や繰り返し用いられるコード（パッケージ）を利用する場合に用いられることが多い．

- ノートブック

セルと呼ばれる入力・出力エリアからなる実行環境であり，コードやその出力結果，注釈などのテキストを記録できるファイル．データ解析を行う際の環境としてよく利用される．

本演習では主にColab上でノートブックを利用して進めていきます

Google Colab

<https://colab.research.google.com>



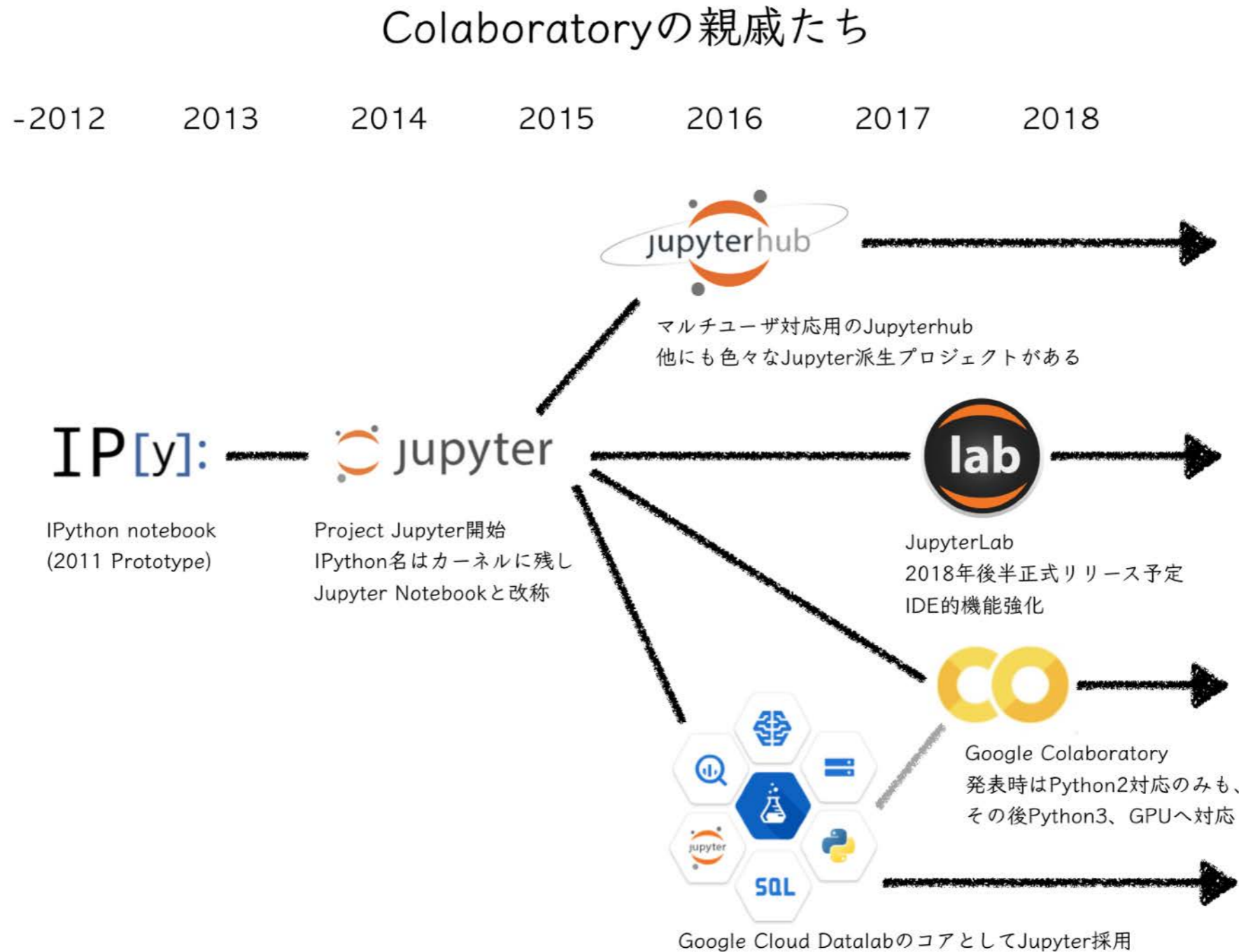
Googleが提供しているJupyter notebookのフォーク・計算環境
本演習では、Colabを使って、Pythonのコードを実行していく

The screenshot shows the Google Colaboratory interface. On the left is a table of contents with items like 'はじめに', 'データサイエンス', '機械学習', and 'その他のリソース'. The main area displays a notebook titled 'Colaboratory とは' with a list of features: '構成が不要', 'GPUへの無料アクセス', and '簡単に共有'. Below this is a code cell with the Python code `seconds_in_a_day = 24 * 60 * 60` and its output `seconds_in_a_day` resulting in `86400`. The interface includes a top navigation bar with '共有', 'ログイン', and 'ヘルプ' options, and a bottom status bar.

- クラウド（環境設定が必要ない）
- （基本的に）無料
- 演習に必要なライブラリが導入済み

- Googleドライブをマウントできる
- GPU環境も利用可能
- など便利な機能満載

Google Colabとその周辺



【秒速で無料GPUを使う】 深層学習実践Tips on Colaboratory | Qiita
https://qiita.com/tomo_makes/items/b3c60b10f7b25a0a5935

Colabの使い方

Colabへのアクセス

<https://colab.research.google.com/>

1. Google アカウントでログイン

Colaboratory へようこそ
ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

共有 接続 編集 ログイン

Colaboratory とは

Colaboratory (略称: Colab) では、ブラウザから Python を記述し実行できるほか、次の特長を備えています。

- 構成が不要
- GPU への無料アクセス
- 簡単に共有

Colab は、学生、データサイエンティスト、AI リサーチャーの皆さんの作業を効率化します。詳しくは、[Colab のご紹介](#)をご覧ください。下からすぐに試してみることもできます。

はじめに

ご覧になっているドキュメントは静的なウェブページではなく、**Colab ノートブック**という、コードを記述して実行できるインタラクティブな環境です。

たとえば次の**コードセル**には、値を計算して変数に保存し、結果を出力する短い Python スクリプトが含まれています。

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

86400

上記のセルのコードを実行するには、セルをクリックして選択し、コードの左側にある実行ボタンをクリックするか、キーボードショートカット「command+return」または「Ctrl+Enter」を使用します。コードはセルをクリックしてそのまま編集できます。

1つのセルで定義した変数は、後で他のセルで使用できます。

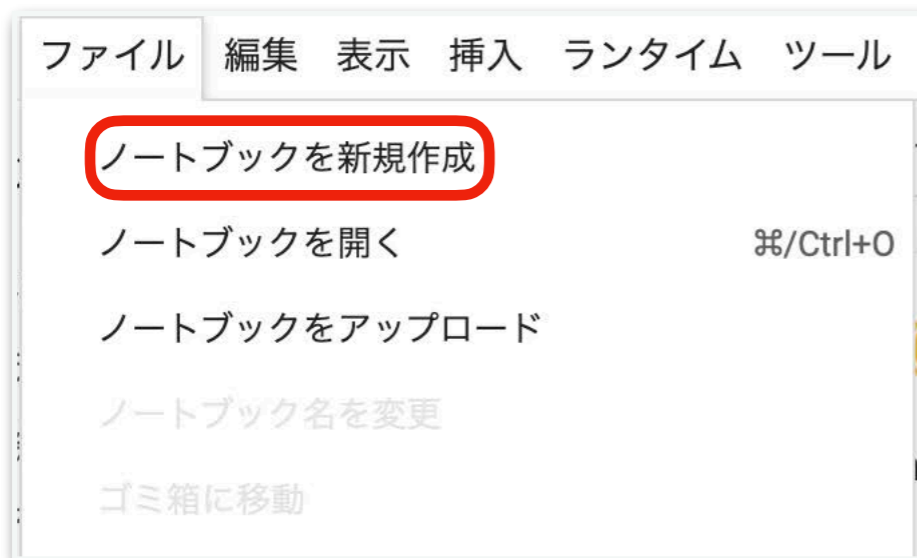
ノートブックを新規作成する (1)

Colab内で新規作成

1. ファイル



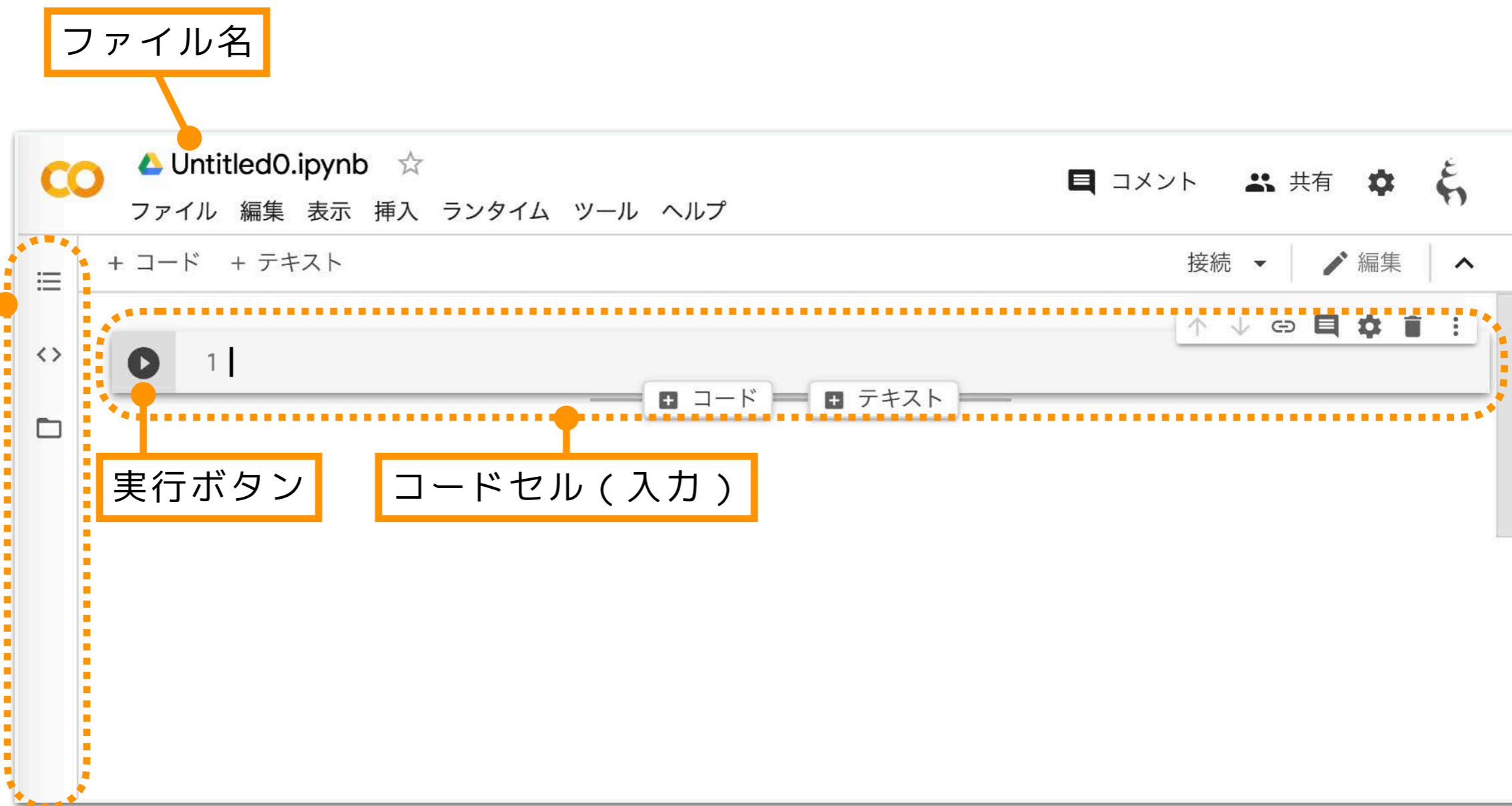
2. ノートブックを新規作成



3. Colabの操作画面が表示される



Colabの操作画面の概要



- セル (cell) : コードなどを入力する, あるいはその結果が出力される領域.
 - コードセル : プログラムのコードを入力するセル. 実行ボタンを押すと, 対象のセルのコードが上から順に実行される.
 - テキストセル : 文章を入力・出力するためのセル.

Hello, World!

プログラミング言語を学ぶと最初に出てくる練習用プログラム

```
# 01-01. Hello, World!ノートブック  
print("Hello, World!")
```

出力
Hello, World!

コードの実行：▶をクリック

- コメントアウト（#）
Pythonでは#から文末までが
（実行時に）無視される
- print(オブジェクト)
オブジェクトを出力



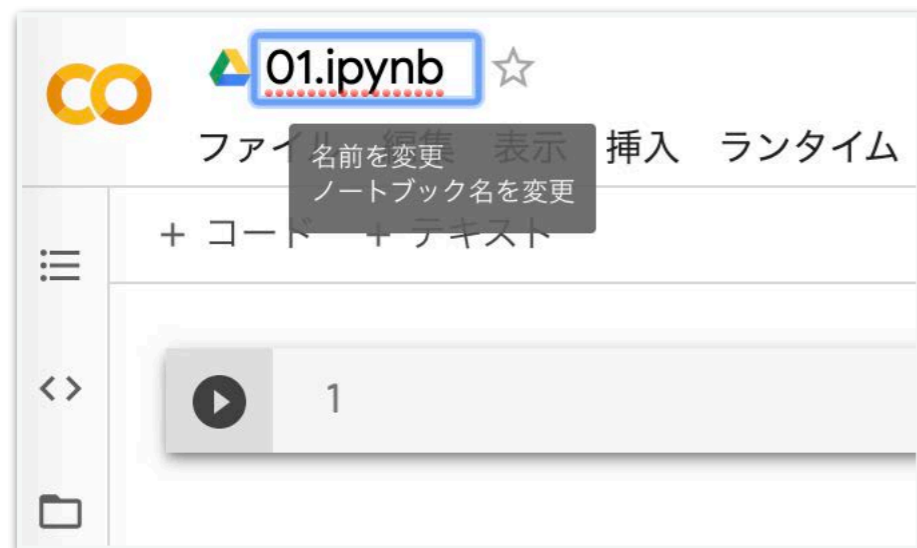
ノートブックに名前をつけて保存しよう！

ノートブックに名前をつける

1. デフォルトでは「Untitled[番号].ipynb」のような名前がついている



2. 名前の部分をクリックすると編集できるので、適切な名前をつける



演習でのノートブックの名前の付け方
サンプルコードには[数字2桁]-[数字2桁]の番号を割り振っている（例. 01-01）。前半の2桁はノートブック番号、後半の2桁はコード番号を想定している。
ノートブックのファイル名は、ノートブック番号と同じ（例. 01）あるいはノートブック番号 内容（例. 01 hello world）とするのがおすすめ。
その他の名前でも構わないが、自分でどのノートブックでどのような作業をしたかが把握できるようにしておくこと。

ノートブックの保存

1. Colabの「ファイル」
2. 「保存」をクリック



ノートブックを閉じる

1. ブラウザのタブ (もしくはウィンドウ) を閉じる



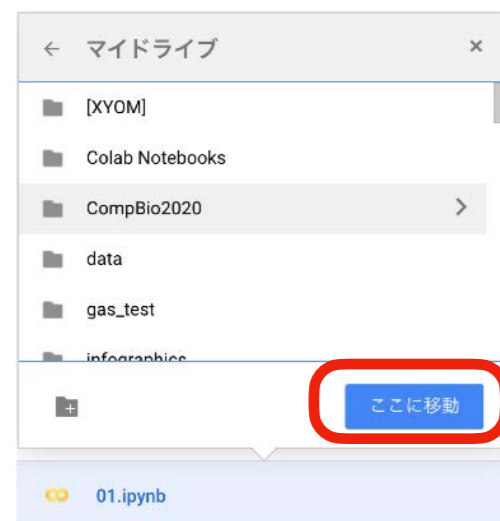
Google ドライブ内でのファイルの移動

右クリックから指定の場所へ移動

1. 移動させたいファイルを右クリック

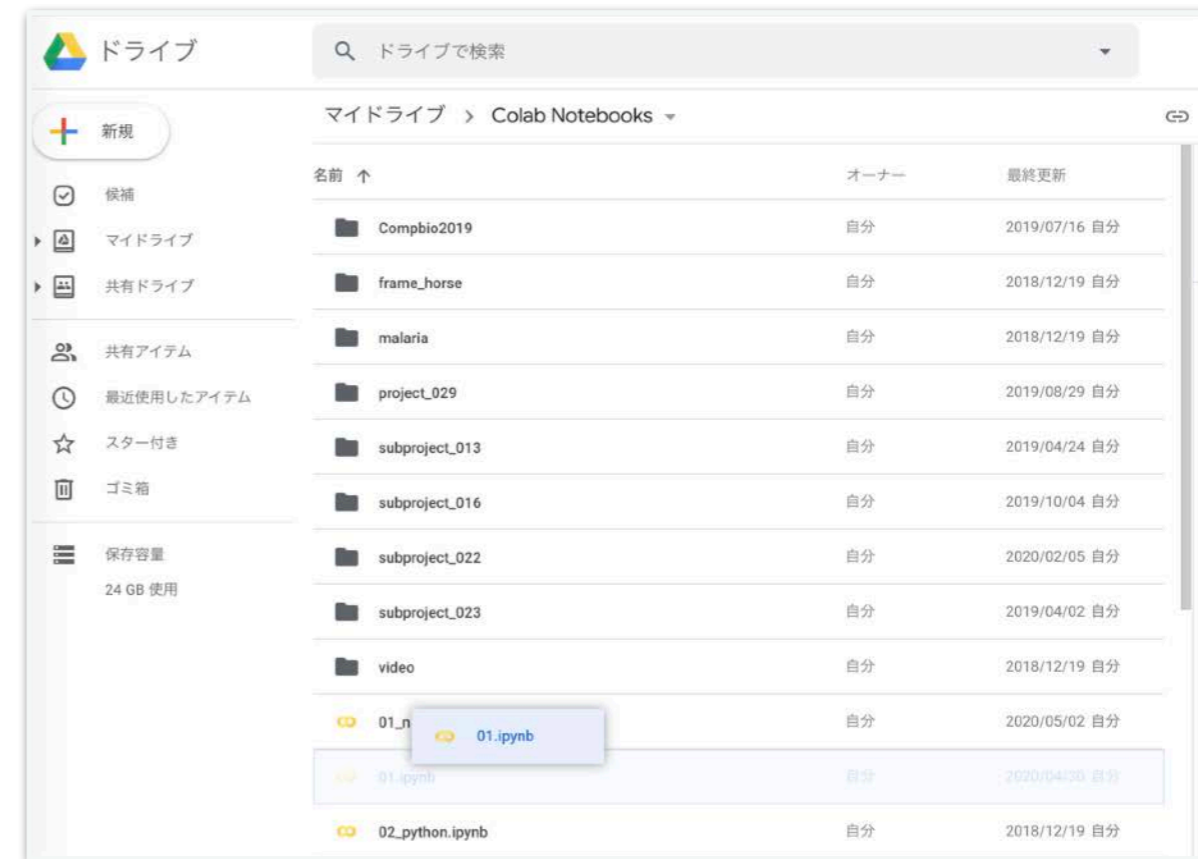


2. 「指定の場所へ移動」
3. 移動先を選び「ここに移動」



ドラッグ&ドロップ

1. 移動させたいファイルをドラッグ&ドロップして、移動先まで持っていく。



課題用フォルダ（今回だと「CompBio2023/01」）へ
忘れずにノートブックを移動させよう

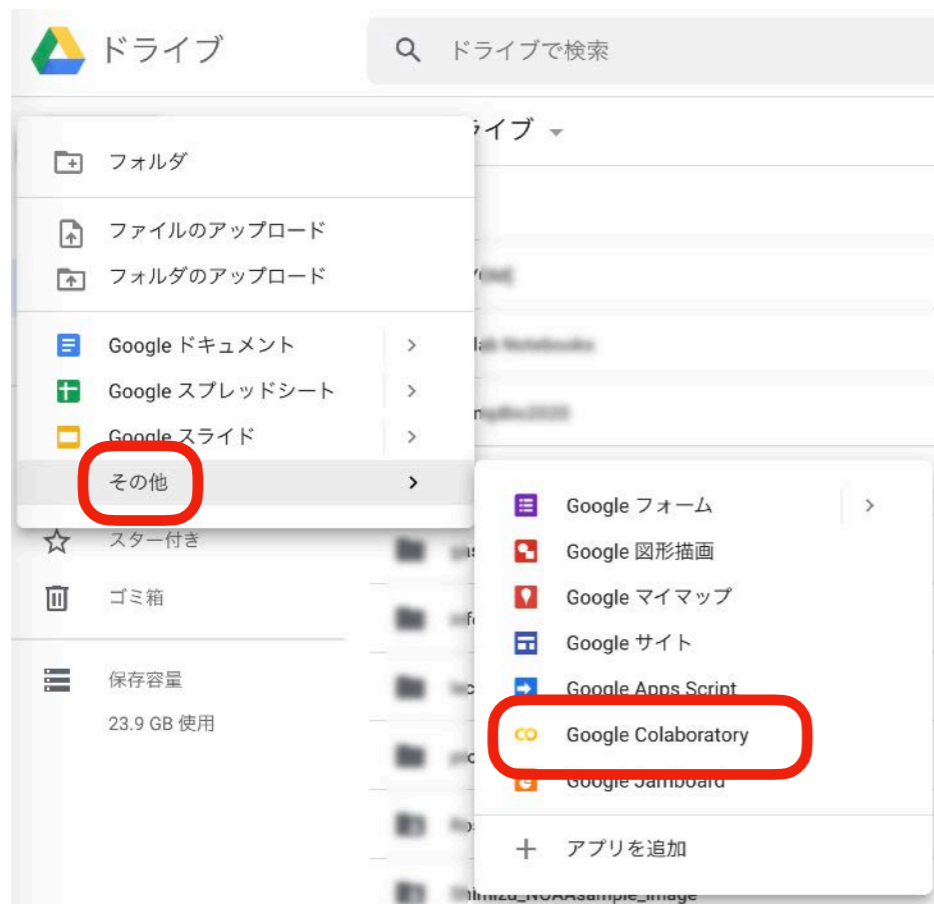
ノートブックを新規作成する (2)

Googleドライブから新規作成

1. ノートブックを保存したいフォルダへ移動

例. マイドライブ → CompBio2024 → 01

2. 「新規作成」 (もしくは右クリック) から「その他」 → 「Google Colaboratory」



好きな場所にノートブックを作成できる

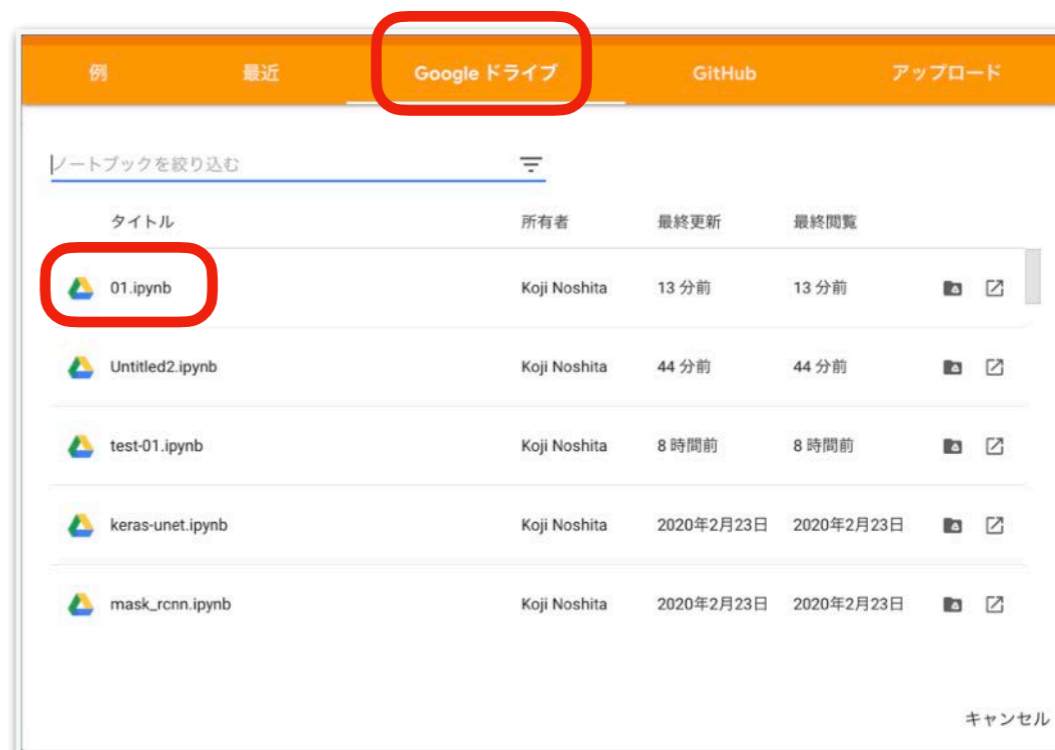
(既存の) ノートブックを開く (1)

Googleドライブから開く (Colab内)

1. Colabの「ファイル」
2. ノートブックを開く



3. Googleドライブのタブを選択



4. 開きたいファイルを選択

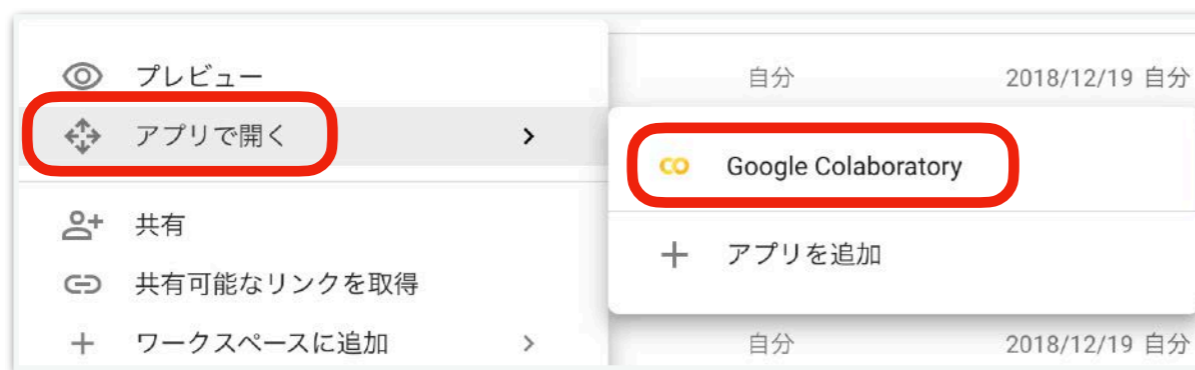
(既存の) ノートブックを開く (2)

Googleドライブから開く (直接)

1. Googleドライブで開きたいファイルのあるフォルダへ移動
例) CompBio2024/01

2. 以下のいずれか

- 右クリック→アプリで開く
→Google Colaboratory



- 開きたいファイルをダブルクリック → Google Colaboratoryで開く



Colabを使うときの注意点

- ノートブックを何も操作しない（アイドル）状態だとセッションが切れる
 - セッションが切れたただけであれば再接続できる
- ノートブックのセッションが切れて90分経過するとインスタンスが落ちる
- インスタンスは起動後12時間立つと（セッションの有無に関係なく）落ちる
- 同時起動できるインスタンスは3つ（CPUのみ, GPU, TPU）
 - 複数のノートブックをただ単に開いても同じインスタンス上で動いている
 - 共有したノートブックの閲覧だけならインスタンスへの接続は必要ない

より詳しい情報

- こんにちはColaboratory
<https://colab.research.google.com/notebooks/welcome.ipynb>
- Colaboratory よくある質問
<https://research.google.com/colaboratory/faq.html>
- 【秒速で無料GPUを使う】 深層学習実践Tips on Colaboratory | Qiita
https://qiita.com/tomo_makes/items/b3c60b10f7b25a0a5935
- 【はやわかり+保存版】 Google Colaboratory チートシート
https://qiita.com/tomo_makes/items/ec12dd61e4ced0a3dd66

あまり気にしなくても良い補足

- セッション：ここではローカルマシン（皆さんが使っているPC）とインスタンスとの接続のこと。
- インスタンス：ここではGoogle Cloud上の仮想マシンのこと。通常はColabはこのインスタンス上で動作している。つまり、Pythonのコードは（皆さんのPC上では実行されておらず）クラウドにあるインスタンスで実行され、その結果をブラウザを介して見ている。

テキストの記述

- テキストセル (text cell) に文章や数式を入力することができる

- 文章はMarkdown記法が使える
- 数式はLaTeXが使える

- 見出し (#) : Markdownでは#をつけることで見出しにできる。##や###など#を重ねることで小見出しにできる。

コードセルの追加 テキストセルの追加

The screenshot displays a Jupyter Notebook interface for a file named '01_hello_world.ipynb'. The top navigation bar includes 'ファイル', '編集', '表示', '挿入', 'ランタイム', 'ツール', and 'ヘルプ'. Below this, there are buttons for '+ コード' (highlighted with a red box) and '+ テキスト' (highlighted with a blue box). The notebook content is divided into two cells. The first cell is a text cell containing the text '# Hello, World!' and '「Hello, World!」と出力する.' The second cell is a code cell containing the code '1 # 01-01. Hello, World!ノートブック' and '2 print("Hello, World!")'. Below the code cell, the output 'Hello, World!' is displayed. A red dashed line connects the '+ コード' button in the top bar to the code cell, and a blue dashed line connects the '+ テキスト' button to the text cell. On the left side, there are two labels: 'テキストセル' (Text Cell) and 'コードセル' (Code Cell), both written vertically.

ノートブックを保存する (2)

直接自分のPCにダウンロードする

1. ファイル → .ipynbをダウンロード



2. ローカルPCのダウンロードフォルダにファイルが保存されているはず。

デフォルトだと

Windows：マイドキュメント → ダウンロード

macOS：ホームディレクトリ（ユーザー名のフォルダ） → ダウンロード（もしくは Downloads）

課題提出の際に利用することになる。

課題提出用ノートブックが完成したら、自分のPCにダウンロードし、それをMoodleから提出する。詳しくは後述。

課題への取り組み・提出

本日の課題 ノーマル

1. 今回の授業でよくわからなかった点を述べる。特になければ、より知りたいと思った点を教えて。
2. 今後扱って欲しい内容を挙げる（複数可）。
3. 「Hello, World!」プログラムを作成せよ。
4. Slackの「#自己紹介」チャンネルに自分の自己紹介を投稿する。（例. ～～です. 趣味は～～です. 最近～～にハマっています. など）
5. その他質問, 感想, 要望をどうぞ。

！注意：第1回課題提出の締め切りは、第2回の開始前まで。

課題をノートブック (.ipynbファイル) にまとめて、Moodleにて提出すること

ファイル名は[回数, 01~15]_[難易度, ノーマル nかハード h].ipynb. 例. 01_n.ipynb 45

課題提出用ノートブックの作成 (1)

課題をノートブック (.ipynbファイル) にまとめて, Moodleにて提出する

ファイル名は

[回数, 01~15]_[難易度, ノーマル nかハード h].ipynb.

例. 01_n.ipynb

はじめに, 大見出し (#) を作る.
ここに氏名, 所属, 学籍番号を書く.

課題それぞれは1つのサブセクション (##) にする

課題回答の文章は, 別のテキストセルへ

01_n ☆

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ [すべての変更を保存しました](#)

+ コード + テキスト

接続 編集

第1回 課題
野下 浩司,理学部,1SC18****

第1回 課題
野下 浩司, 理学部, 1SC18****

1. 今回の授業でよくわからなかった点を述べる. 特になければ, より知りたいと思った点を教えて.

~を~してみたが, ~の操作がわからない. ~が~となる理由がわからない.

課題提出用ノートブックの作成 (2)

課題をノートブック (.ipynbファイル) にまとめて, Moodleにて提出する

The screenshot shows a Jupyter Notebook window with the following content:

- Header: 01_n ☆, コメント, 共有, 接続, 編集
- Menu: ファイル, 編集, 表示, 挿入, ランタイム, ツール, ヘルプ, すべての変更を保存しました
- Buttons: + コード, + テキスト
- Section 3: 「Hello, World!」プログラムを作成せよ.
- Section 3.1: Hello, World!
 - Text: 「Hello, World!」と出力する.
 - Code cell:

```
[ ] 1 # 01-01. Hello, World!ノートブック
2 print("Hello, World!")
```
 - Output: Hello, World!
- Section 4: Slackの「#自己紹介」チャンネルに自分の自己紹介を投稿する. (例. ～～です. 趣味は～～です. 最近～～にハマってます. など)
- Text: (Slack参照)

コードの説明があるとなお良い (その場合はサブサブセクション (###) などにする)

コードと出力結果

ノートブック内に回答がない場合はその旨書く.
(例. Slack参照)

質問・トラブルシューティング

Googleなどで調べる

- 状況にあったキーワードを2～5個程度を考え、検索する
 - 例) Python Colab ノートブック ダウンロード.
- エラーメッセージを読む、検索する
 - 例) TypeError～～.

Slackの「#質問」チャンネルを使う

- 質問する
 - 例) インストールが～で止まる. ～の図がきれいに表示されない. ～をやってみたけど, ～にならない.
- 他の人の質問に答える
 - 例) ～にチェックを入れるといいよ. 途中の～の計算ができてないんじゃない?

(そもそも何で詰まってるのか説明できないと, 答えようがないので) まずは自分で調べる・考える. それでも無理そうだったら野下やTAにどんどん気軽に質問する (特に最初はしょうもないミスに時間を取られがち. 無駄に数時間悩むより質問する.). 自分ができていたら困っている人を助けよう. 48

バグ報告

サンプルコード，スライド，資料などに間違いや不正確な内容がある可能性がある。
これらを本演習では総じて「バグ」と呼ぶ。もし，バグを見つけたら報告しよう。

Slackの「#バグ報告」チャンネルを使う

- バグの指摘
 - 例) サンプルコードの～を実行するけど，（予想される結果）～にならない。
- バグの指摘と原因の特定
 - 例) ～を実行するけど，（予想される結果）～にならない。～のせいだ。
- 既に指摘されているバグの修正案を提案
 - 例) ～のバグだけど，～を～に修正すると正しい結果が得られるよ。
- バグの指摘と修正案の提案
 - 例) ～を実行するけど，（予想される結果）～にならない。～のせいだ。～を～に修正すると正しい結果が得られる。

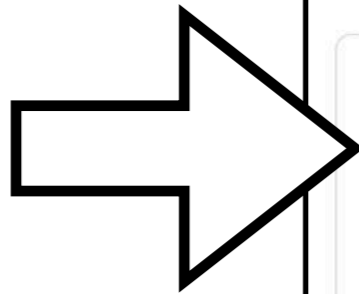
バグ報告をしてもらえるとコードや資料の改善に繋がり非常に助かる。
バグ報告にボーナスポイントあり。

課題提出

Moodleから課題を提出

1. Moodleへのログイン
2. 「2024年度前期・月3月4・数理生物学演習」コースへ移動
3. 各回の課題モジュールから提出

ココ！



▼ 第1回 Hello, World!

- プログラミング言語とは？
- 本演習で使うプログラミング言語の紹介
- プログラミング環境の構築



第1回 課題

- 今回の授業でよくわからなかった点を述べる。特になければ、より知りたいと思った点を教えて。
- 今後扱って欲しい内容を挙げる（複数可）。
- 「Hello, World!」プログラムを作成せよ。
- Slackの「#自己紹介」チャンネルに自分の自己紹介を投稿する。（例. ～～です。趣味は～～です。最近～～にハマってます。など）
- その他質問、感想、要望をどうぞ。

JupyterLab Desktop

<https://github.com/jupyterlab/jupyterlab-desktop>

Jupyterノートブックファイル (.ipynb) の確認・編集

自分の環境に合わせて
macOSかWindowsの
いずれかをダウンロード
してインストールする

Installation

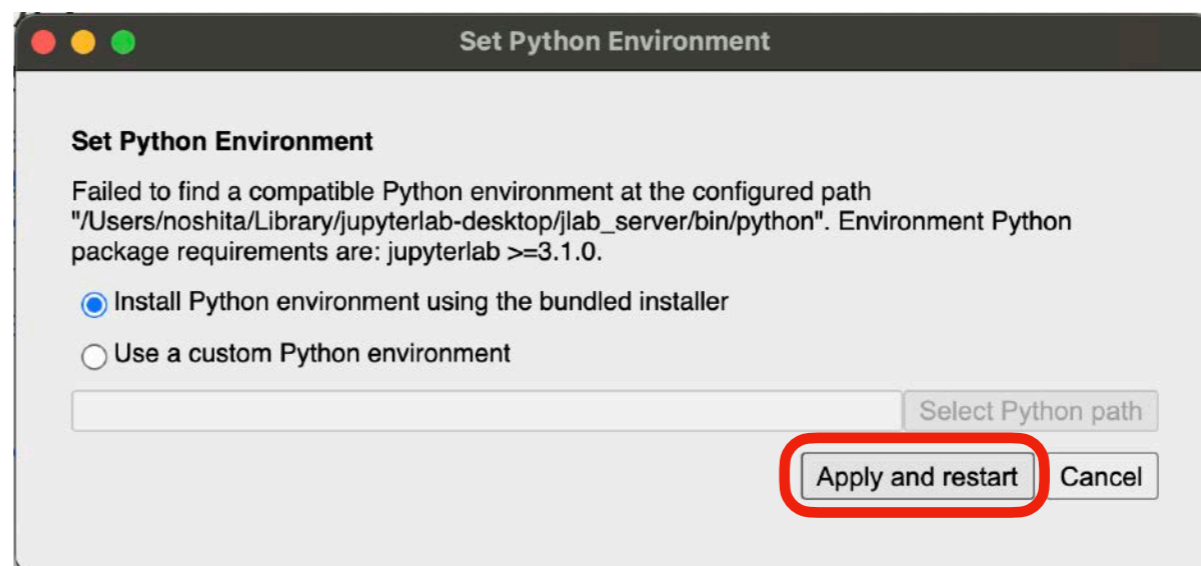
Install JupyterLab Desktop using one of the following methods:

Windows (10, 11)	Mac (macOS 10.15+)	Linux
x64 Installer	arm64 Installer (Apple silicon)	Snap Store [recommended]
	x64 Installer (Intel chip)	.deb x64 Installer (Debian, Ubuntu)
Windows		.rpm x64 Installer (Red Hat, Fedora, SUSE)

Additionally, JupyterLab Desktop can be installed via winget: `winget install jupyterlab`.

If you need to remove a previous JupyterLab Desktop installation, please follow the [uninstall instructions](#).

初回起動時にPython環境のインストールを指示されたら、インストールし再起動する。



本日の課題 ノーマル

1. 今回の授業でよくわからなかった点を述べる。特になければ、より知りたいと思った点を教えて。
2. 今後扱って欲しい内容を挙げる（複数可）。
3. 「Hello, World!」プログラムを作成せよ。
4. Slackの「#自己紹介」チャンネルに自分の自己紹介を投稿する。（例. ～～です. 趣味は～～です. 最近～～にハマっています. など）
5. その他質問, 感想, 要望をどうぞ。

！注意：第1回課題提出の締め切りは、第2回の開始前まで。

課題をノートブック (.ipynbファイル) にまとめて、Moodleにて提出すること

ファイル名は[回数, 01~15]_[難易度, ノーマル nかハード h].ipynb. 例. 01_n.ipynb 52

次回予告

第2回：Pythonの基本的な使い方と
数理生物学演習で使う数学の復習

4月22日

復習推奨

- ベクトル，行列の演算
- 固有値・固有ベクトルの計算
- 微分方程式の変数分離による解法