

Pythonプログラミング入門

2019年度A1 初回ガイダンス

数理・情報教育研究センター

# 本日の流れ

- 授業について
  - 日程・場所
  - 目標/計画/進めかた
- 教材
  - 講義ページ
  - Jupyter Notebook / G-Suite ドライブ
  - 予習動画 / YouTube
  - 課題提出 / ITC LMS
- 成績評価方法
- 参考書・教科書
- オフィスアワー
- Python 環境

# 日程・場所

- ガイダンス・第一回
  - 9月24日（火）5限 本郷キャンパス理学部1号館285教室
  - 9月26日（木）5限 駒場キャンパス情報教育棟E31
  - 9月27日（金）5限 柏キャンパス新領域基盤棟大講義室
- 第二回以降も上記曜日・教室で開講する
  - 火5限：10/1, 10/8, 10/15, 10/29, 11/5, 11/12 本郷キャンパス理学部1号館285教室
  - 木5限：10/3, 10/10, 10/17, 10/24, 10/31, 11/14 駒場キャンパス情報教育棟E31
  - 金5限：10/4, 10/11, 10/18, 10/25, 11/1, 11/8(17:00より) 柏キャンパス新領域基盤棟大講義室

# 学部・プログラム・期間・単位

- 理学部
  - 学科共通科目
- 数理・データサイエンス教育プログラム
  - 数理・情報教育研究センター
  - [www.mi.u-tokyo.ac.jp/mds-oudan](http://www.mi.u-tokyo.ac.jp/mds-oudan)
- 期間
  - S1ターム / 夏期集中 / A1ターム
- 単位
  - 1単位

# 対象

- 後期課程学生
  - 学部を問いません
- 大学院生も聴講可能
  - 単位も取得可能ですが、修了要件に含められるかどうかは、研究科・専攻によります
- 1,2年生も聴講可能
  - 単位を取得することはできません

# 履修登録

UTAS と ITC-LMS で登録番号が異なることに注意してください。

- UTAS
  - 0590102-2 Python プログラミング入門 (A1 共通)

UTASとは別に、履修を希望するいずれかの曜日においてITC-LMSで登録を行って下さい。 (行わなければ、課題が提出できません。)

- ITC-LMS
  - 0590102-21 Python プログラミング入門 (A1火5)
  - 0590102-22 Python プログラミング入門 (A1木5)
  - 0590102-23 Python プログラミング入門 (A1金5)

# 担当

- 講師
  - 本郷
    - 萩谷 昌己 教授
    - 森 純一郎 准教授
    - 山肩 洋子 准教授
    - 小林 克志 特任准教授
  - 駒場
    - 山口 和紀 教授
    - 小林 浩二 講師
  - 柏
    - 佐藤 重幸 助教
- TA

# 授業の目標

- データサイエンスや計算科学等に必要なプログラミングの基礎をPython言語を通して修得を目指す
- データ構造、制御構造、オブジェクト指向等、プログラミング言語の基礎概念について学ぶとともに、計算量等のアルゴリズムの基本原理について理解を目指す
- 最終的に、計算の手続きを自分である程度自由にPythonのプログラムとして表現できるようになることをを目指す

# 授業計画

- 授業は7回の演習から成り、各回の演習までにオンライン教材をあらかじめ自分で学んでおくことを想定している
- 各回の演習では、プログラミング実習を行いながら質問等を受け付け、出欠も兼ねて各自が作成したコードの提出を行う
- 最後にミニプロジェクトを行い、その成果をレポートとして報告する

# 授業計画

- 第1回 Jupyter Notebookの使い方 数値演算 変数と関数の基礎 論理・比較演算と条件分岐の基礎 デバッグ
- 第2回 文字列 リスト 条件分岐
- 第3回 辞書 繰り返し 内包表記 関数
- 第4回 ファイル入出力の基本 コンピュータにおけるファイルやディレクトリの配置 モジュールの使い方 モジュールの作り方
- 第5回 Pythonスクリプトとコマンドライン実行 Numpy
- 第6回 関数プログラミング オブジェクト指向プログラミング
- 第7回 pandasライブラリ scikit-learnライブラリ ミニプロジェクト

# 授業の進め方

授業は、オンライン教材を各自が事前に自学し、当日は演習を行い理解を深める反転授業方式で進めます

- 授業前
  - 次回授業のオンライン教材（Jupyterノートブック）を自学
  - 予習課題の演習とITC-LMSへの提出
- 授業
  - 前回の本課題の解説
  - 予習課題の解説
  - **本課題の演習とITC-LMSへの提出**

講義全体の最後にはミニプロジェクトを行い、その成果をレポートとして提出

# 授業の進め方

- 第1回
  - 本ガイダンス・講義（教材1-0, 1-1, 1-2, 1-3, 1-4）
  - 時間があれば予習課題（pre1）の演習と提出（オプション）
- 第2回まで
  - Jupyterノートブック教材2-1, 2-2, 2-3を予習
  - 予習課題（pre2）の演習と提出
- 第2回
  - 予習課題（pre2）の解説
  - 本課題（ex1, ex2）の演習と提出
- 第3回まで
  - Jupyterノートブック教材3-1, 3-2, 3-3, 3-4を予習
  - 予習課題（pre3）の演習と提出
- 第3回
  - 本課題（ex1, ex2）と予習課題（pre3）の解説
  - 本課題（ex3）の演習と提出

# 授業の進め方

- 第4回まで
  - Jupyterノートブック教材4-1, 4-2, 4-3, 4-4を予習
  - 予習課題 (pre4) の演習と提出
- 第4回
  - 本課題 (ex3) と予習課題 (pre4) の解説
  - 本課題 (ex4) の演習と提出
- 第5回まで
  - Jupyterノートブック教材5-1, 5-2を予習
  - 予習課題 (pre5) の演習と提出
- 第5回
  - 本課題 (ex4) と予習課題 (pre5) の解説
  - 本課題 (ex5) の演習と提出

# 授業の進め方

- 第6回まで
  - Jupyterノートブックと説明動画で教材6-1, 6-2を予習
  - 予習課題 (pre6) の演習と提出
- 第6回
  - 本課題 (ex5) と予習課題 (pre6) の解説
  - 本課題 (ex6) の演習と提出
- 第7回まで
  - Jupyterノートブック教材7-1, 7-2を予習
  - 予習課題 (pre7) の演習と提出
- 第7回
  - 本課題 (ex6) と予習課題 (pre7) の解説
  - ミニプロジェクトの解説と演習

# 本日の流れ

- 授業について
  - 日程・場所
  - 目標/計画/進めかた
- 教材
  - 講義ページ
  - Jupyter Notebook / G-Suite ドライブ
  - 予習動画 / YouTube
  - 課題提出 / ITC LMS
- 成績評価方法
- 参考書・教科書
- オフィスアワー
- Python 環境

# 教材

- 教材は各回の学習項目ごとのJupyterノートブックと説明動画からなります
- 教材は授業ホームページのリンクからアクセスできます
  - <https://sites.google.com/view/ut-python>  
(<https://sites.google.com/view/ut-python>)
  - 教材 ⇒ 教材のフォルダ（すべての回のノートブックと関連ファイル）
  - intro2python.pdf に課題を除くすべての教材がまとめられています  
(索引付き)
- 教材にアクセスするにはECCSクラウドメール（G Suite）アカウントが必要です
  - Gmailまたは以下のページからECCSクラウドメールアカウントでログイン後に教材にアクセスしてください  
<http://mail.google.com/a/g.ecc.u-tokyo.ac.jp>  
(<http://mail.google.com/a/g.ecc.u-tokyo.ac.jp>)

# ECCS アカウントの取得方法

大学院生などで、ECCS アカウントがない場合は取得してください。

- ECCSをご利用になるには「オンライン新規利用者講習」・・・を受講していただく必要があります。
  - **未受講の方が何人かいます！**
- 受講していただくことで、お持ちの UTokyo Account でECCSの利用権が有効になります。
  - UTokyo Account の通知書を受け取っていない場合は、ご所属の学部・大学院研究科の教務担当までお問い合わせください。

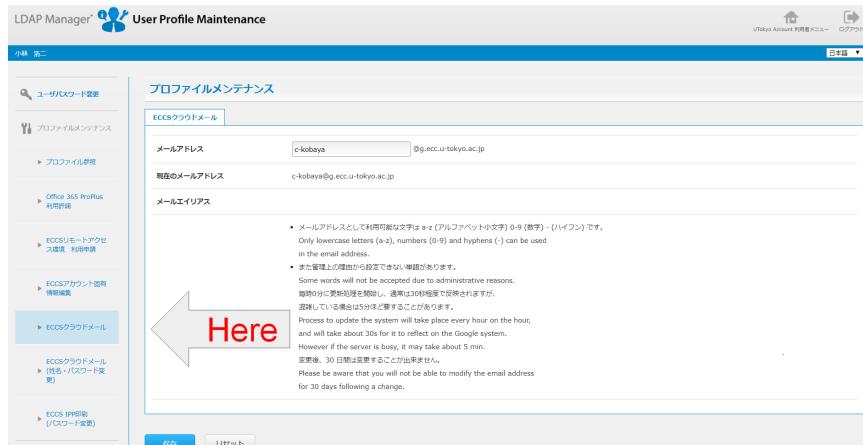
# ECCSクラウドメールアカウントの申請と取得

- UTokyo Accountの利用者メニューから申請してください

U-Tokyo Account



U-Tokyo Profile



ノートブックはG Suiteのドライブにあります



ドライブを検索



J

新規

共有アイテム &gt; mi-kiban &gt; notebook &gt; 1 &lt;



マイドライブ

ファイル

名前 个

共有アイテム

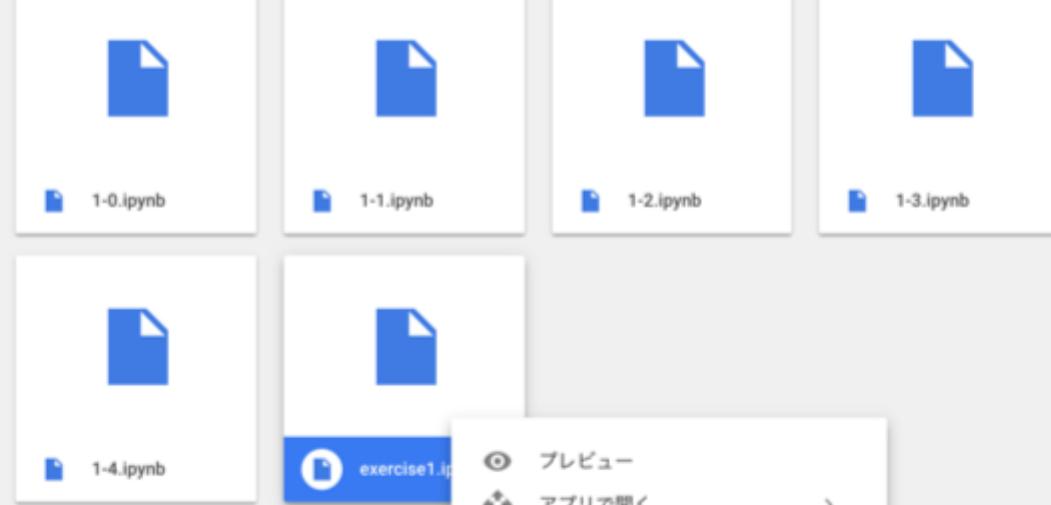
最近使用したアイテム

Google フォト

スター付き

ゴミ箱

246 KB 使用



- プレビュー
  - アプリで開く >
  - 共有...
  - 共有可能なリンクを取得
  - マイドライブに追加
  - スターを付ける
  - 名前を変更...
- 
- 詳細を表示
  - 版を管理...
  - コピーを作成
  - ダウンロード
- 
- 削除

共有ファイルや重要なイベントの通知をパソコンで受け取り  
ましょう。

オンにする X

# Jupyter Notebook



File

Edit

View

Insert

Cell

Kernel

Widgets

Help

Not Trusted

Python 3

New Notebook

Open...

Make a Copy...

Rename...

Save and Checkpoint

Revert to Checkpoint

構造のうち条件分岐について説明します

Print Preview

Download as

Trust Notebook

Close and Halt

# thonプログラミング入門 第3回

## インデントによる構文

分岐の前に、Python のインデント（行頭の空白、字下げ）について説明します。Python のインデントは実行文をグループまとめる機能をもちます。

実行文はインデントレベル（深さ）の違いによって異なるグループとして扱われます。細かく言えば、インデントレベルが戻る（浅くなる）まで実行文は同じグループに属し、レベルが深くなると実行文はもとのグループに加え、別のグループに属するものとして扱われます。

### インデントによる構文(続き)

具体例として、第1回で定義した関数 bmax を使って説明します:

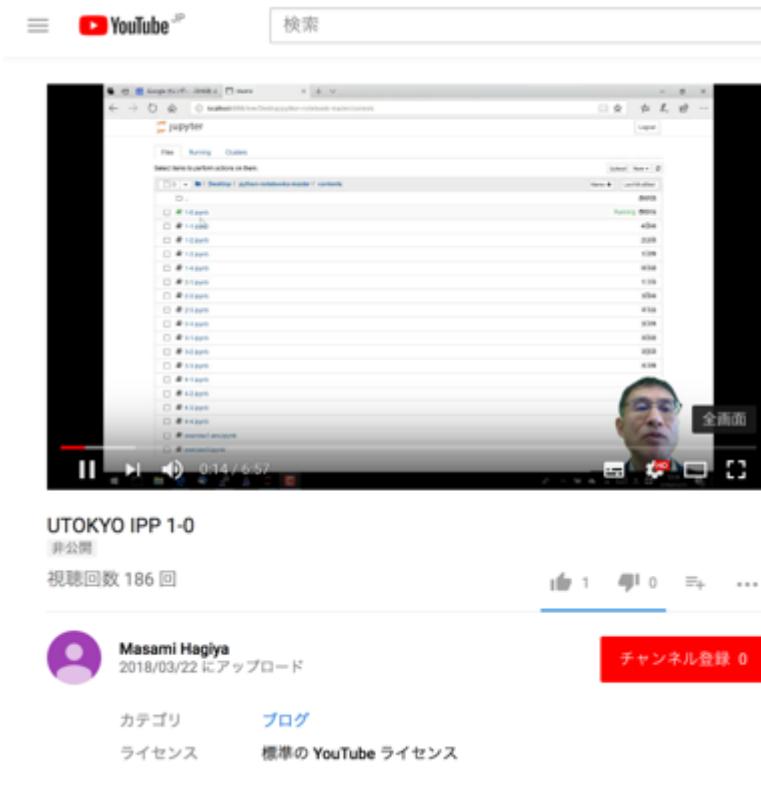
In [ ]:

```
def bmax(a,b):
    if a > b:
        return a
    else:
        return b

print("Hello World")
```

# 説明動画はYouTubeで公開されています

<https://sites.google.com/view/ut-python/resource/%E5%8B%95%E7%94%BB>  
[\(https://sites.google.com/view/ut-python/resource/%E5%8B%95%E7%94%BB\)](https://sites.google.com/view/ut-python/resource/%E5%8B%95%E7%94%BB)



# 課題提出

- 方法

- 予習課題・本課題は、解答を入力したJupyterノートブックのファイルを、ITC-LMSにログインして提出してください
  - 曜日ごとにITC-LMSのコースが作られています
  - それぞれ自己登録可となっています
- 予習課題・本課題にはテスト例が含まれていますので、必ずテスト例で動作を確認してから提出してください
  - 与えられたテスト例以外のテスト例で試すことも推奨します
- 予習課題の締め切り：その回の前日22時
- 本課題の締め切り：その日の22時
  - 期限をすぎての提出も認めますが、減点されます

# ITC-LMS の画面

理学部 0590102-11 Python プログラミング入門（月5）

担当教員 小林 浩二, 山肩 洋子, 小林 克志, 山口 和紀, 萩谷 昌己, 森 純一郎, 佐藤 重幸  
開講学期 S1/月曜日/5限  
コースの概要説明/Course Summary

お知らせ	お知らせタイトル	掲載期間	操作

アンケート	アンケートタイトル	アンケート期間・期限	結果公開期間	操作
	情報分野アンケート	2019/03/29 10:50 ~ 2019/06/30 22:00		
	どうやってこの授業を知りましたか？	2019/03/29 10:08 ~ 2019/06/30 22:00		

掲示板	テーマタイトル	投稿期間・期限	操作
	質問用掲示板	2019/04/01 00:00 ~ 2019/06/30 22:00	

# 本日の流れ

- 授業について
  - 日程・場所
  - 目標/計画/進めかた
- 教材
  - 講義ページ
  - Jupyter Notebook / G-Suite ドライブ
  - 予習動画 / YouTube
  - 課題提出 / ITC LMS
- 成績評価方法
- 参考書・教科書
- オフィスアワー
- Python 環境

# 成績評価方法

以下の内容で評価します

- 各回の予習課題の提出状況と結果
- 各回の本課題の提出状況と結果
- ミニプロジェクトのレポート

不正行為に対しては厳正に対処します

# オフィスアワー

一般的な質問については、以下で受け付けます

- ITC-LMSの掲示板
- 授業前後に口頭

個別の相談・質問があれば、学年・学部・学籍番号・氏名を明記して、萩谷までメールをください

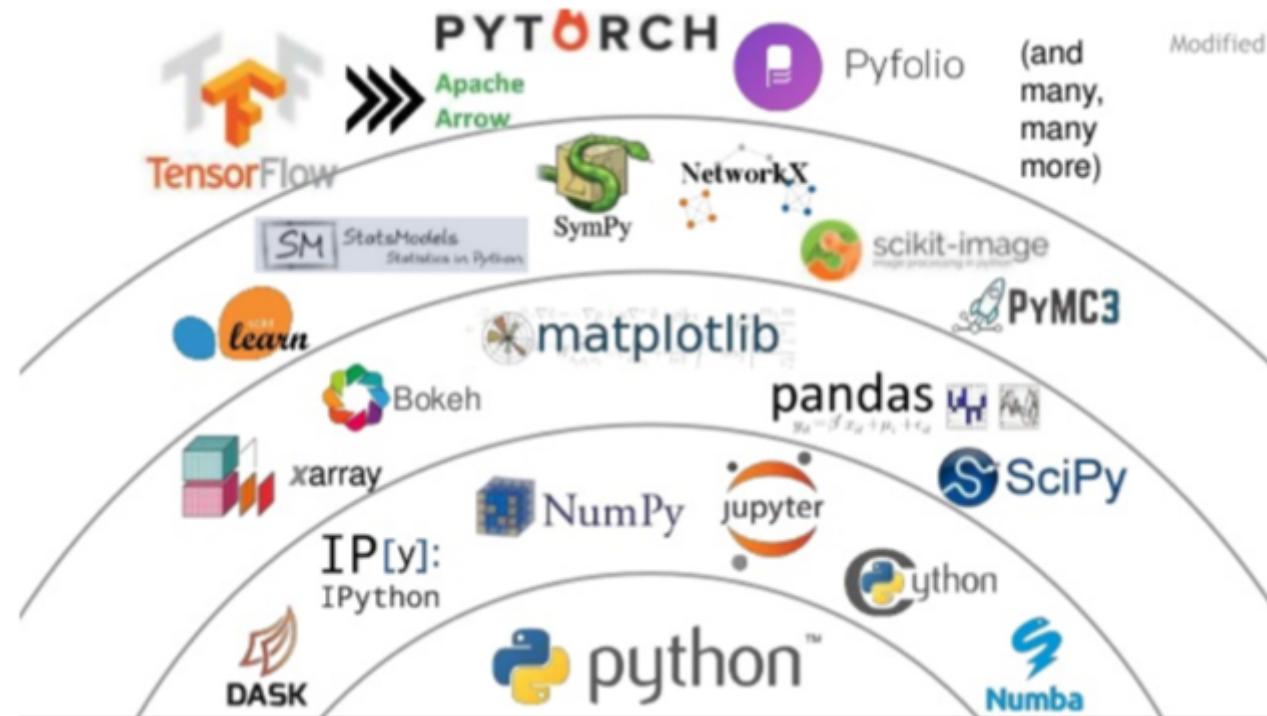
- Masami Hagiya <hagiya@is.s.u-tokyo.ac.jp>

# 教科書・参考書

- 教科書
  - Jupyterノートブックを配布します。
- 参考サイト・参考書
  - [Pythonチュートリアル](https://docs.python.org/ja/3/tutorial/) (<https://docs.python.org/ja/3/tutorial/>)
  - [Python ドキュメント](https://docs.python.org/ja/3/) (<https://docs.python.org/ja/3/>)
  - [Allen B. Downey\(原著\) Think Python:How To Think Like a Computer Scientist](https://greenteapress.com/wp/think-python-2e/) (<https://greenteapress.com/wp/think-python-2e/>)
    - [相川利樹\(翻訳\) Think Python:コンピュータサイエンティストの  
みよう\(第2版\)](http://www.cauldron.sakura.ne.jp/thinkpython/thinkpython/Thinking%20Like%20a%20Computer%20Scientist.pdf)  
(<http://www.cauldron.sakura.ne.jp/thinkpython/thinkpython/Thinking%20Like%20a%20Computer%20Scientist.pdf>)
  - [Introduction to Computer Science and Programming in Python \(MIT\)](https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/introduction-to-computer-science-and-programming-in-python-fall-2016/)  
(<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/introduction-to-computer-science-and-programming-in-python-fall-2016/>)
  - John V. Guttag (原著), 久保 幹雄 (翻訳) Python言語によるプログラミング・  
ーション第2版 近代科学社 4764905183
  - Bill Lubanovic (著), 斎藤 康毅 (監修), 長尾 高弘 (翻訳) 入門 Python 3 オライ  
4873117380
  - [Online Python Tutor](http://pythontutor.com/) (<http://pythontutor.com/>) : Pythonプログラム実行の

# Python

- フリー・オープンソースのインタプリタ・スクリプト言語
- 科学技術計算に利活用
  - データ分析、機械学習、数値計算、最適化、画像・信号処理、可視化など
- 設計思想「シンプルで読みやすいコード」
  - The Zen of Python
- Pythonエコシステム



# Anaconda / Jupyter Notebook

- Anaconda
  - <https://anaconda.org/> (<https://anaconda.org/>)
  - Pythonと主要なライブラリ（モジュール）を一括でインストール可能にしたディストリビューション
    - 以下のJupyter Notebookも含む
    - ECCSにはすでにインストールされています
  - Python2系とPython3系があり、講義ではPython3系を使用
- Jupyter Notebook
  - <http://jupyter.org/> (<http://jupyter.org/>)
  - Pythonコードをブラウザ上で対話的に実行可能な環境
  - ノートブックドキュメントを通して、ブラウザ上でコードの記述と実行結果の保存と共有が可能
  - ECCSではAnaconda Navigatorから起動できます

# Python環境のセットアップ

いくつかの方法について説明します

- 個人PC（Windows）にAnacondaをインストールする場合
- 個人PCにAnacondaをインストールしない・できない場合
- 教育用計算機システム（駒場のみ）

# Python環境のセットアップ（Windows向け）

個人PC（Windows 10, youユーザ）にAnacondaをインストールする場合

## 1. Anacondaのダウンロードページ

(<https://www.anaconda.com/distribution/#windows>)からWindows用のPython 3.7 version 64-Bit Graphical Installer (Anaconda3-2018.12-Windows-x86\_64.exe) をダウンロード

2. インストーラを起動し、License Agreementに [I agree] を選択

3. Install forに [Just me] を選択

4. インストール先をデフォルト (C:\Users\you\Anaconda3) から変更したい場合は、インストール先を指定して [Next] を選択

5. Advanced Optionsの2つのチェックボックス (Add Anaconda to my PATH environment variableとRegister Anaconda as my default Python 3.7) のチェックを外したままインストール [install] を選択

- Python本体やJupyter Notebookをコマンドラインで起動したい場合は、コマンドプロンプトの代わりにAnaconda Command Promptを使用してください

6. Windowsのスタートメニューのプログラム一覧にAnaconda3 (64bit) というフォルダが新しく作られており、その中のJupyter Notebookを選択すると、既定のブラウザでJupyter Notebookが起動する

- Anaconda Navigatorから起動してもよい

# Python環境のセットアップ (Mac向け)

個人PC (MacOSX) にAnacondaをインストールする場合

## 1. Anacondaのダウンロードページ

(<https://www.anaconda.com/distribution/#macos>)からMac用のPython 3.7 versionインストーラ (Anaconda3-2018.12-MacOSX-x86\_64.pkg) をダウンロード

## 2. インストーラを起動しインストールしてください

## 3. インストール先をデフォルト (~/anaconda3) から変更したい場合は、インストール先を指定してください

## 4. アプリケーション一覧に"Anaconda Navigator"というアプリケーションがインストールされているので、Anaconda Navigatorを起動して、その中にある Jupyter Notebookから[Launch]するとJupyter Notebookが起動します

## 5. 本スライド中のECCSでのJupyter Notebookの起動を参考にしてください



The screenshot shows a web browser window with the URL [anaconda.com](https://anaconda.com) in the address bar. The page title is "Anaconda Python/R Distribution - Anaconda". Below the title, there are links for "Windows", "macOS", and "Linux". The main content features two large boxes: one for the "Python 3.7 version" and one for the "Python 2.7 version". Each box contains a "Download" button and links to graphical and command-line installers. Below these boxes, a section titled "Get Started with Anaconda Distribution" is shown, consisting of five vertical cards with partial text visible.

## Anaconda Python/R Distribution - Anaconda

Windows | macOS | Linux

### Anaconda 2018.12 for macOS Installer

#### Python 3.7 version

[Download](#)

64-Bit Graphical Installer (652.7 MB)  
64-Bit Command Line Installer (557 MB)

#### Python 2.7 version

[Download](#)

64-Bit Graphical Installer (640.7 MB)  
64-Bit Command Line Installer (547 MB)

### Get Started with Anaconda Distribution

Documentation Installation user guide for Anaconda Distributi on 5	Anaconda Blog News, software releases, and developer best practices	Community Support Solutions and knowledge from the community	Anaconda Webinars Industry trends and tutorials from Anaconda	Anaconda Training Learn Python for Data Science with DataCamp
--	---	--	---	---

# Python環境のセットアップ

個人PCにAnacondaをインストールしない・できない場合

1. ブラウザで以下にアクセス

<https://colab.research.google.com/notebooks/welcome.ipynb>  
[\(https://colab.research.google.com/notebooks/welcome.ipynb\)](https://colab.research.google.com/notebooks/welcome.ipynb)

2. Jupyterノートブックファイル (.ipynb) をアップロードして利用



**Hello, Colaboratory**

ファイル 編集 表示挿入 ランタイム ツール ヘルプ

Python 2 のノートブックを新規作成  
Python 3 のノートブックを新規作成

Welcome ドライブのノートブックを開く... ⌘/Ctrl+O  
Local run 最近のノートブックを開く... ⌘/Ctrl+Shift+L  
Python 3 ノートブックをアップロード... ⌘/Ctrl+U  
TensorFic 名前を変更...  
Visualizat ゴミ箱に移動  
Forms ドライブにコピーを保存...  
Exam GitHub にコピーを保存...  
For more 保存 ⌘/Ctrl+S  
セクション 変更内容を保存して固定 ⌘/Ctrl+M S  
変更履歴  
.ipynb をダウンロード  
.py をダウンロード  
ドライブ プレビューを更新  
印刷 ⌘/Ctrl+P

ブにコピー 変更を破棄

共有 J

**Welcome to Colaboratory!**

Colaboratory is a Google research project created to help disseminate machine learning education and research. It's a Jupyter notebook environment that requires no setup to use and runs entirely in the cloud.

Colaboratory notebooks are stored in [Google Drive](#) and can be shared just as you would with Google Docs or Sheets. Colaboratory is free to use.

For more information, see our [FAQ](#).

**runtime support**

Colaboratory supports connecting to a Jupyter runtime on your local machine. For more information, see our [documentation](#).

**Python 3**

Colaboratory supports both Python2 and Python3 for code execution. When creating a new notebook, you'll have the choice between Python 2 and Python 3. You can also change the language associated with a notebook; this information will be written into the .ipynb file itself, and thus will be preserved for future sessions.

```
import sys
int('Hello, Colaboratory from Python {}!'.format(sys.version_info[0]))
```

Hello, Colaboratory from Python 3!

**TensorFlow execution**

Colaboratory allows you to execute TensorFlow code in your browser with a single click. The example below adds two matrices.

$$\begin{bmatrix} 1. & 1. & 1. \\ 1. & 1. & 1. \end{bmatrix} + \begin{bmatrix} 1. & 2. & 3. \\ 4. & 5. & 6. \end{bmatrix} = \begin{bmatrix} 2. & 3. & 4. \\ 5. & 6. & 7. \end{bmatrix}$$

```
[ ] import tensorflow as tf
import numpy as np

with tf.Session():
    input1 = tf.constant(1.0, shape=[2, 3])
    input2 = tf.constant(np.reshape(np.arange(1.0, 7.0, dtype=np.float32), (2, 3)))
    output = tf.add(input1, input2)
    result = output.eval()

result
```

array([[ 2., 3., 4.],
 [ 5., 6., 7.]], dtype=float32)

**Visualization**

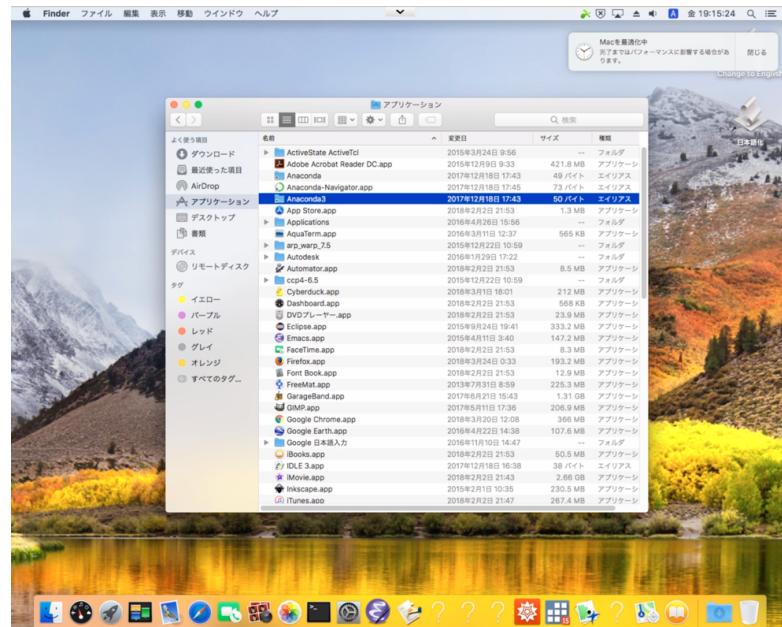
Colaboratory includes widely used libraries like [matplotlib](#), simplifying visualization.

# Python環境のセットアップ

教育用計算機システムを利用する場合

1. MacOS で起動すること
2. アプリケーションのAnaconda3フォルダを開きます
3. Anaconda-Navigatorを起動
4. Jupyter Notebookを起動

## Application Folder



## Start Anaconda Navigator

