

# 2023.9.1 情報処理技術セミナー Jupyter Notebookとは

NII クラウド担当 谷沢

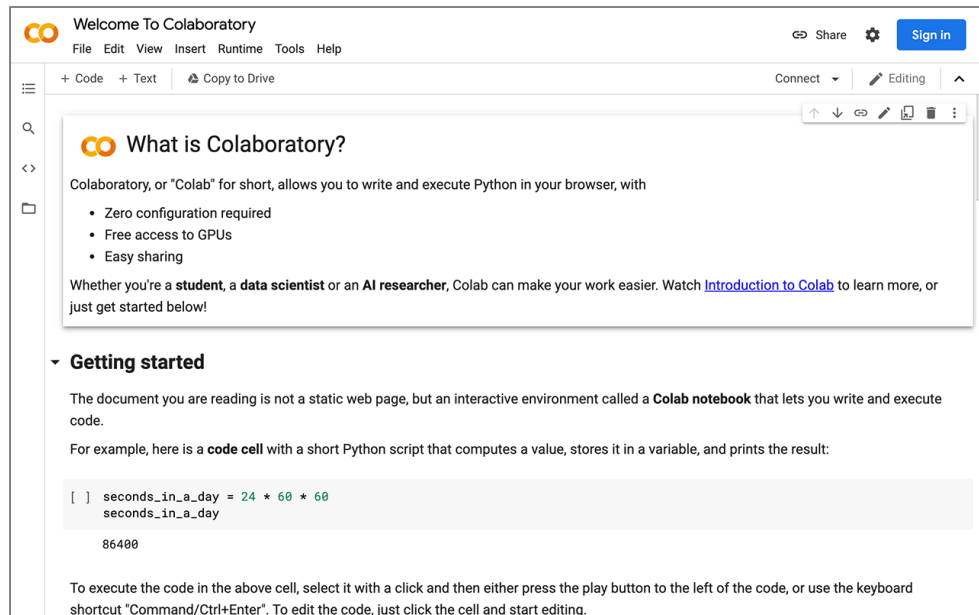
# Jupyter Notebookとは

- ▶ Jupyter Notebook ご存知ですか？/使ったことがありますか？
- ▶ ジュピター あるいは ジュパイター(**J**ulia + **P**ython + **R**) ノートブック
- ▶ Webベースのインタラクティブなユーザインタフェース + スクリプト実行環境
- ▶ 多種のスクリプトを利用可能
  - ▶ Python
  - ▶ R
  - ▶ Julia
  - ▶ Perl
  - ▶ PHP
  - ▶ 他にも: <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>



# オープンソースソフトウェア

- ▶ 多くの開発者により活発に開発が進められている
- ▶ Jupyter Notebookを利用したサービスも多く存在
  - ▶ Google Colab(Google Colaboratory)
  - ▶ Microsoft Azure Machine Learning
  - ▶ ...



Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

+ Code + Text Copy to Drive

Connect Editing

## What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

### Getting started

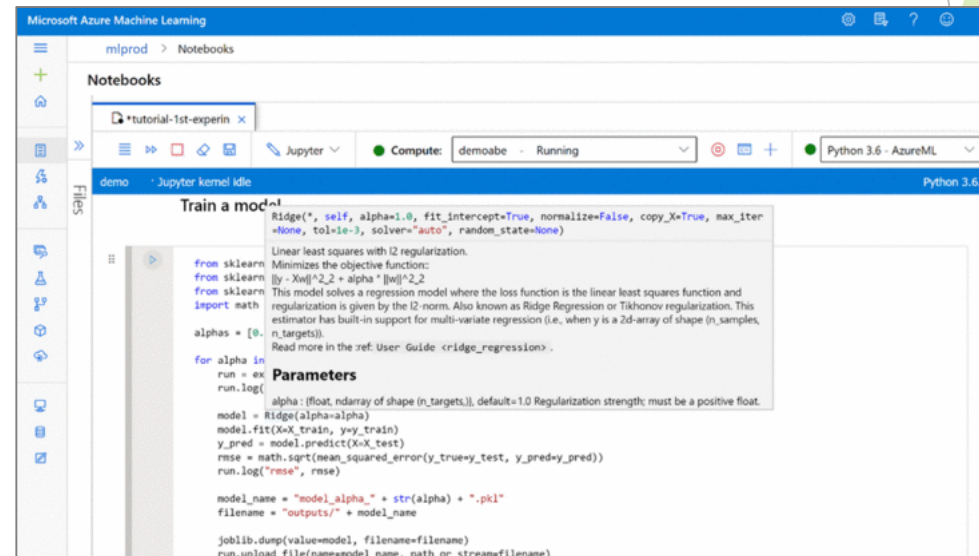
The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

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

86400
```

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.



Microsoft Azure Machine Learning

miproduct > Notebooks

## Notebooks

\*tutorial-1st-experin x

Jupyter Compute: demoabe - Running Python 3.6 - AzureML

demo - Jupyter kernel idle Python 3.6.9

### Train a model

```
Ridge(*, self, alpha=1.0, fit_intercept=True, normalize=False, copy_X=True, max_iter=None, tol=1e-3, solver="auto", random_state=None)
Linear least squares with L2 regularization.
Minimizes the objective function:
||y - Xw||^2_2 + alpha * ||w||^2_2
This model solves a regression model where the loss function is the linear least squares function and regularization is given by the L2-norm. Also known as Ridge Regression or Tikhonov regularization. This estimator has built-in support for multi-variate regression (i.e. when y is a 2d-array of shape (n_samples, n_targets)).
Read more in the ref: User Guide <ridge_regression>..
Parameters
alpha : float, ndarray of shape (n_targets,), default=1.0 Regularization strength; must be a positive float.
model = Ridge(alpha=alpha)
model.fit(X=X_train, y=y_train)
y_pred = model.predict(X=X_test)
mse = math.sqrt(mean_squared_error(y_true=y_test, y_pred=y_pred))
run.log("mse", mse)

model_name = "model_alpha_" + str(alpha) + ".pkl"
filename = "outputs/" + model_name

joblib.dump(value=model, filename=filename)
run.upload_file(filename=model_name, path=os.getcwd()+filename)
```

# ↩ Notebookを試してみたいならばココ

The image is a collage of several JupyterLab notebook screenshots. The top-left screenshot shows the 'Intro.ipynb' notebook with a file browser on the left and introductory text about Jupyter Notebooks. The middle-left screenshot shows a code cell with Python code for generating random data and plotting it as a scatter plot with bubble sizes. Below the code is a scatter plot titled 'Some random data, created with JupyterLab!' showing a cluster of points with varying bubble sizes and colors. The middle-right screenshot shows a code cell for solving the Lorenz system of differential equations, followed by a text prompt to explore the system and a plot of the Lorenz attractor. The bottom-right screenshot shows a code cell with a Vega chart configuration and a resulting chart titled 'Charting Using Vega' that displays a table of Level vs Hitpoints data as a scatter plot.

```
[1]: from matplotlib import pyplot as plt
import numpy as np

# Generate 100 random data points along 3 dimensions
x, y, scale = np.random.randn(3, 100)
fig, ax = plt.subplots()

# Map each onto a scatterplot we'll create with Matplotlib
ax.scatter(x=x, y=y, c=scale, s=np.abs(scale)*500)
ax.set(title="Some random data, created with JupyterLab!")
plt.show()
```

Some random data, created with JupyterLab!

```
[4]: w=interactive(solve_lorenz,sigma=(0.0,50.0),rho=(0.0,50.0))
w
```

[4]: sigma 10.00  
beta 2.63  
rho 28.00

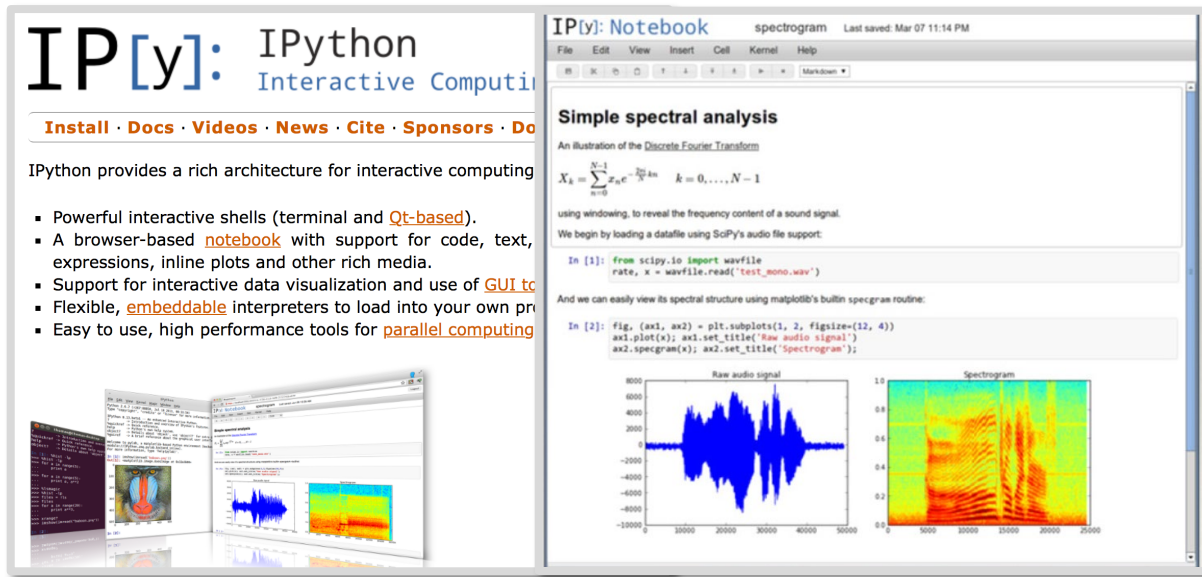
Charting Using Vega

```
[12]: %VEGA_PLOT
X_FIELD Level
Y_FIELD Hitpoints
MARK circle
WIDTH 100
HEIGHT 200
SELECT Level, Hitpoints FROM players
```

Level	Hitpoints
3	40
2	20
1	10



# 端緒は IPython Notebook



1991: Python  
1995: Numeric → NumPy (~2006)  
2001: SciPy  
2001: IPython  
2003: Matplotlib  
...  
2011: IPython Notebook  
  
2015: Project Jupyter  
2015: GitHub rendering  
2017: BinderHub 2.0 reboot  
2019: JupyterHub, JupyterLab  
2019: BinderHub is out of beta  
Voilà is a Jupyter subproject  
2021: JupyterLab 3.0  
Jupyter Lite

- ▶ Python の Rich Web client な対話インターフェース
- ▶ 実行Code, 実行結果 (計算の結果や, 結果を埋め込んだ図表), 自然言語による手順や経緯の説明をひとつにまとめて記述
- ▶ “The purpose of Computing is insight, not numbers” - Hamming [Numerical Methods for Scientists and Engineers (1962) Preface]
- ▶ 数理系研究者のツールとして発展

アイデアの探索課程の記録, 共同開発, 再現可能な成果の公開, 教育

# Jupyter is Everywhere

## Jupyter Notebookで始めるプログラミング

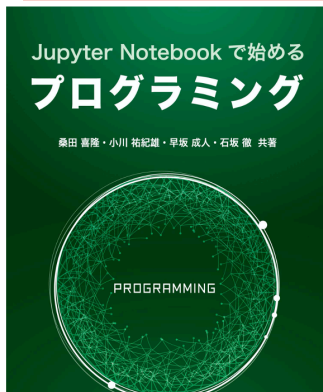
はじめてプログラミングを学習する人のための入門書。

Jupyter Notebook上でプログラミング言語「Python」を使って対話的に学ぶことができる。

桑田 喜隆, 小川 祐紀雄, 早坂 成人, 石坂 徹

ISBN-10 : 4780608589  
出版社 : 学術図書出版社  
ISBN-13 : 978-4780608588

## 室蘭工業大学



## CoursewareHubを利用した授業: Python入門

群馬大学・数理データ科学教育研究センター  
井上 仁

- ✓ 講義演習環境
- ✓ 予復習
- ✓ 小テスト
- ✓ 対話型教材 など

幅広く利用が進んでいる

## 松尾研究室

Matsuo Lab., the University of Tokyo ABOUT RESEARCH EDUCATION PUBLICATIONS JOIN

## Deep Learning基礎講座演習コンテンツ公開ページ

### 演習中心で効率よく学習

本プログラムの最大の特徴の一つは、全てのトピックについて、演習を中心に構成されている点です。実際に手を動かしながら理解を進めることで、効率よく学習することができます。

### モデルの学習

実際にモデルを学習させながら技術を習得する本格的な演習内容となっています。Deep Learningは、モデルが実際に学習する様子を観測し、パラメータを調整することでアプリケーションに応じたパフォーマンス最大化を行うことが非常に重要な技術ですが、この一連の流れを全ての演習で経験しながら重要な要素を身につけることが可能です。

### 研究者向けコンテンツ

本コンテンツは、Deep Learning基礎講座で利用している演習コンテンツです。線形代数や機械学習を前提知識として要しますが、Deep Learningの新しいモデルを構築したり、高度な研究・開発を行うために重要な知識が基礎から学べるように設計されています。

### Jupyter notebook形式

全てのコンテンツはJupyter Notebook形式で作成されています。Jupyter環境があれば、ブラウザからコードを実行できるだけでなく、自由に試行錯誤できるので、効率よく学習することが可能です。

## Pythonプログラミング入門

Open in Colab

### Navigation

- 1-0. Colaboratoryによるノートブックの使い方
- 1-1. 数値演算
- 1-2. 変数と関数の基礎
- 1-3. 論理・比較演算と条件分岐の基礎
- 1-4. デバッグ

## Pythonプログラミング入門

▲で始まる項目は授業では扱いません。興味にしがって学習してください。

ノートブック全体に▲が付いているものもありますので注意してください。

- 1-0. Colaboratoryによるノートブックの使い方
  - Colaboratoryの立ち上げ
  - ノートブックのアップロード
  - 教材のオープン
  - ノートブックのダウンロード
  - ノートブックのアップロード (再び)
  - ノートブックの作成
  - ノートブックの操作
  - セル
- 1-1. 数値演算
  - 簡単な算術計算

<https://utokyo-ipp.github.io/index.html>

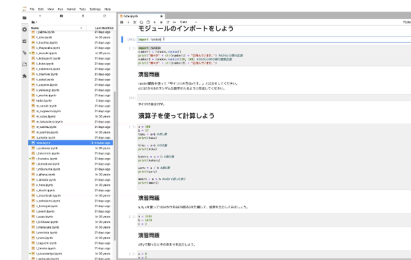


田浦 健次郎 東京大学情報基盤センター長/情報理工学系研究科/工学部電子情報工学科 教授  
【第39回】大学等におけるオンライン教育とデジタル変革に関するサイバーシンポジウム

## STEM教育プロジェクト

### 2020年8月21日 高校生対象のオンラインPython入門講座 (Python)

全国の高校生を対象とし、オンライン上でのプログラミング入門講座を実施しました。Zoomを通してJupyterを用いたPythonの基礎を学習しました。遠くは四国に居住する高校生も参加し、約50名の受講者、13名の講師が参加しました。

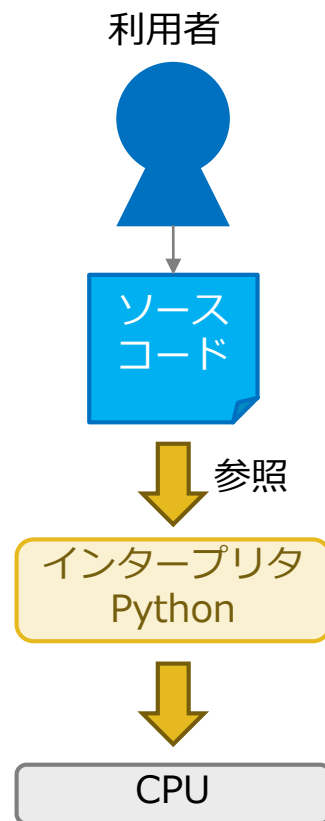




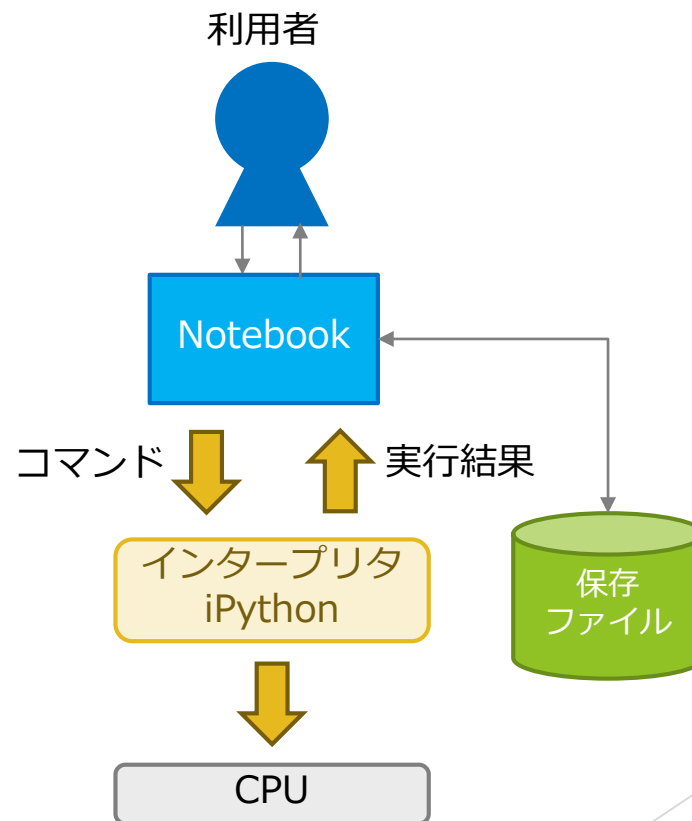
# スクリプトの実行モデル

- ▶ Jupyter Notebookにおいてスクリプトがどのように実行されるか？
- ▶ Jupyter Notebookにおける逐次実行
  - ▶ インタープリタのことを Jupyter Notebookでは カーネル(Kernel) と呼ぶ

## スクリプトの一括実行



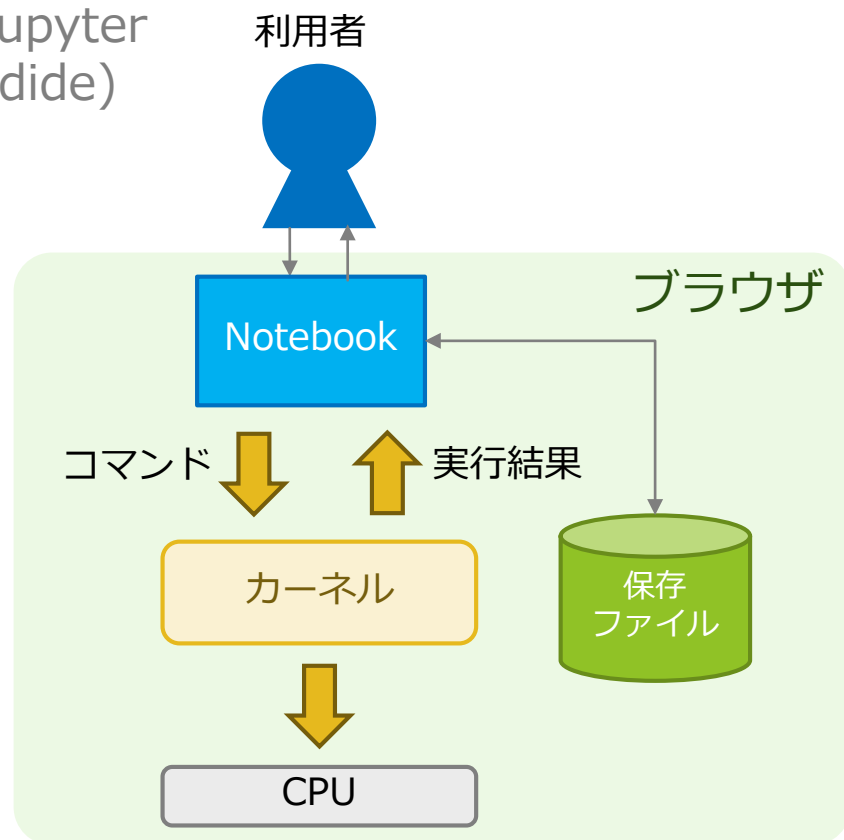
## Jupyter Notebookの逐次実行



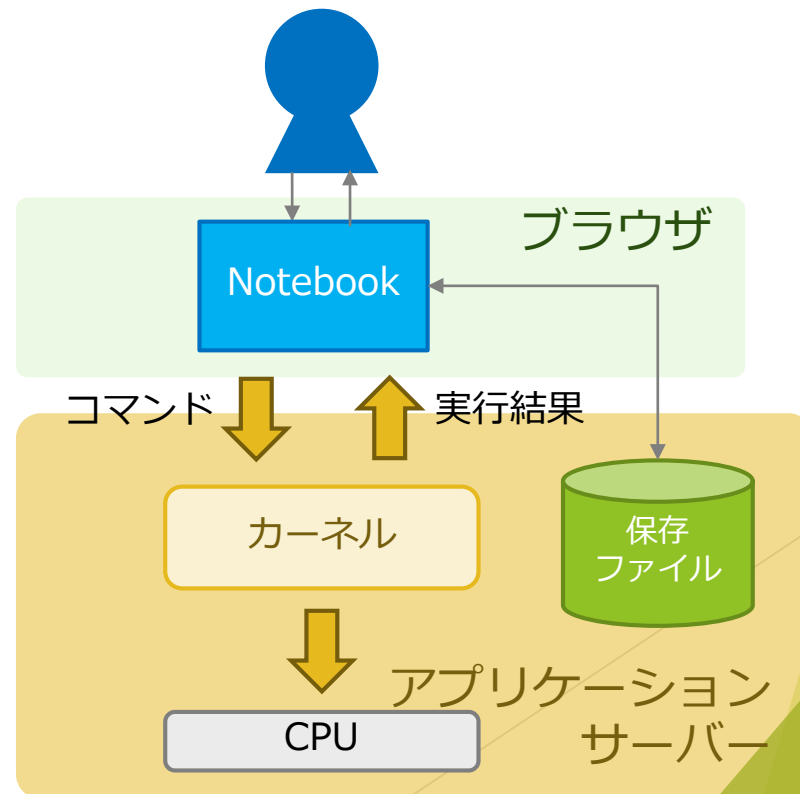
# スクリプトの実行環境

- ▶ どこでカーネルが動作？
- ▶ <https://jupyter.org/try-jupyter/lab> はブラウザで実行される
  - ▶ Pyodide: a port of CPython to WebAssembly/Emscripten (!!)
- ▶ 基本モデル: カーネルおよびNotebookファイルの管理はアプリケーションサーバーで実現

try-jupyter  
(Pyodide)



基本モデル 利用者





# mybinder.org

- ▶ 従来型モデルで試せる環境
- ▶ <https://github.com/binder-examples/requirements>
- ▶ GitHubリポジトリに launch binder というボタンがあるので、クリックしてみましょう

☰ README.md

## Python environment with a requirements.txt

 launch binder

A Binder-compatible repo with a `requirements.txt` file.

Access this Binder at the following URL

<http://mybinder.org/v2/gh/binder-examples/requirements/HEAD>

### Notes

# mybinder.org

- ▶ 簡単に、無償でJupyter Notebook(他関連プロダクト)を試せる環境
- ▶ GitHub等の**公開**リポジトリに配備されたコードに基づいて、Jupyter Notebook環境を起動することができる
  - ▶ 先の図のアプリケーションサーバー側がmybinder.orgにセットアップされる
- ▶ **時間制限あり** - 10分限定
- ▶ ソフトウェアを公開して、簡単に試してもらうには非常に便利

# 他にもさまざまな環境に対応

- ▶ Sample Binder Repositories
  - ▶ [https://mybinder.readthedocs.io/en/latest/examples/sample\\_repos.html](https://mybinder.readthedocs.io/en/latest/examples/sample_repos.html)
  - ▶ 他にも、GitHubを見ると、さまざまなコードが公開されている
  - ▶ 時間制限があるが、**ブラウザだけで容易に/様々なデータ分析等を試すことができる**世界
- ▶ mybinder.orgもオープンソース
  - ▶ <https://github.com/jupyterhub/binderhub>
- ▶ **GakuNin RDMデータ解析機能**はmybinder.orgと同じくbinderhubを配備したもの
  - ▶ mybinder.org: 公開データのみ → GakuNin RDMデータ解析機能: **プライベートデータをサポート**
  - ▶ mybinder.org: 時間制限あり → GakuNin RDMデータ解析機能: **データの永続化をサポート**