

情報科学研究 2

データ解析特論

第1回データの理解

情報科学領域

平居 悠（ひらい ゆたか）

自己紹介

平居 悠 （ひらい ゆたか）

メディア情報コース 講師

居室：H-2研究室

E-mail: yutaka.hirai@koeki-u.ac.jp

出身：埼玉県

専門：天文学、数値シミュレーション

学位：博士（理学）東京大学 2018年

職歴：理化学研究所、東北大学、

アメリカ・ノートルダム大学

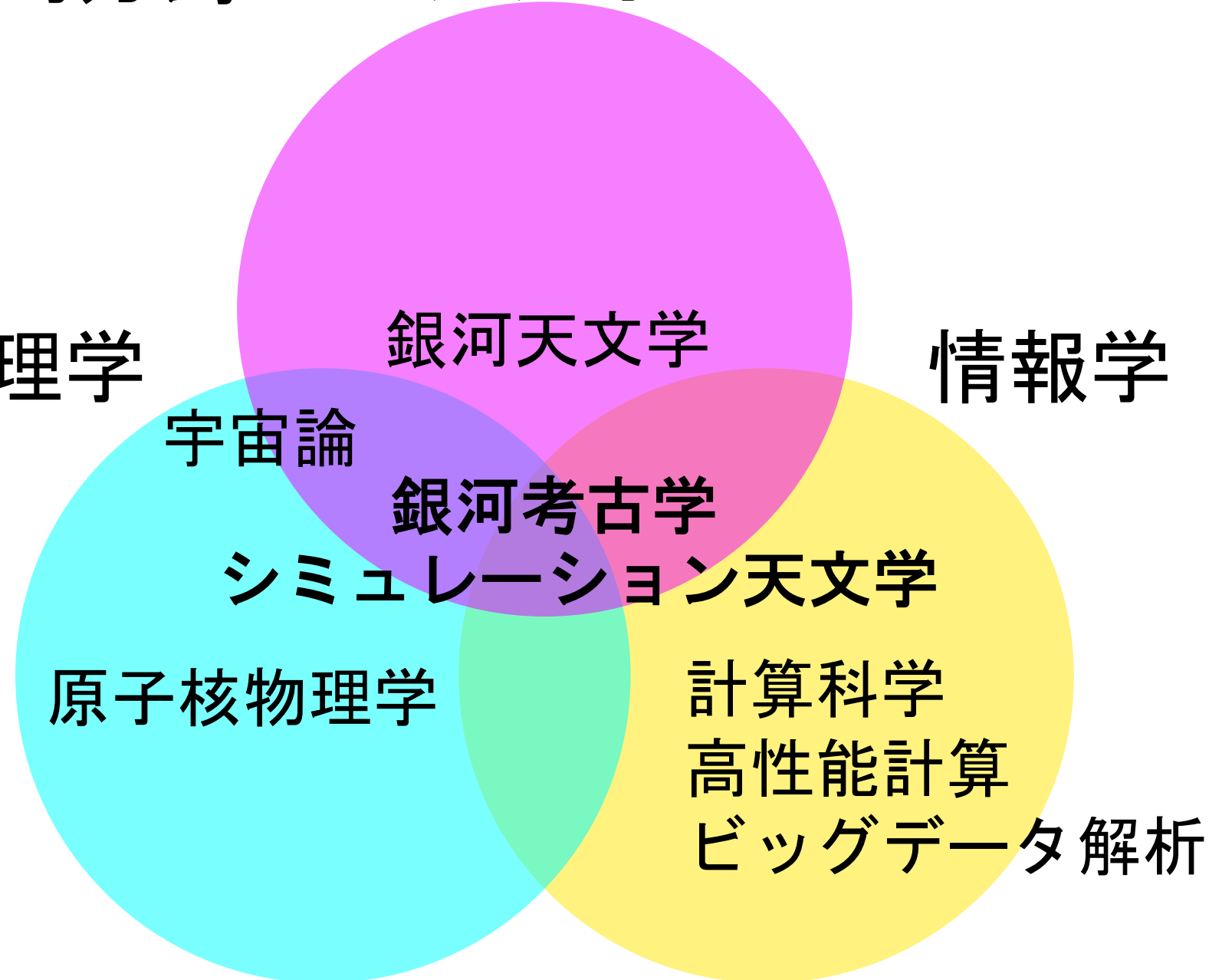


専門分野

天文学

物理学

情報学



授業概要

- 統計モデルの基本的考え方と観測データへの応用方法を学ぶ。
- 統計モデリング基礎理論の講義と統計ソフトウェアを用いた演習により、データ解析の手法を修得する。

到達目標

- 基本的な統計モデルについて説明できる。
- 観測データに適切な統計モデルを適用し、解析できる。
- 定量的なデータ解析結果を示すことができる。

授業計画

第1回	データの理解	第9回	マルコフ連鎖モンテカルロ法
第2回	確率分布と統計モデル	第10回	ベイズ統計モデル
第3回	確率分布と統計モデルの最尤推定	第11回	一般化線形モデルのベイズモデル化
第4回	一般化線形モデル	第12回	事後分布の推定
第5回	統計モデルの選択	第13回	階層ベイズモデル
第6回	尤度比検定と検定の非対称性	第14回	空間構造のある階層ベイズモデル
第7回	一般化線形モデルの応用	第15回	まとめ
第8回	一般化線形混合モデル		

成績評価

平常課題

60%

レポート課題

40%

教科書

確率と情報の科学

データ解析のための 統計モデリング入門

一般化線形モデル・階層ベイズモデル・MCMC

久保拓弥

岩波書店

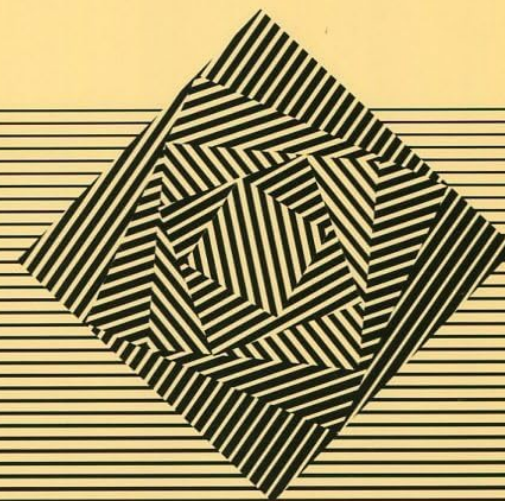
参考書

基礎統計学 I

統計学入門

東京大学教養学部統計学教室 編

東京大学出版会



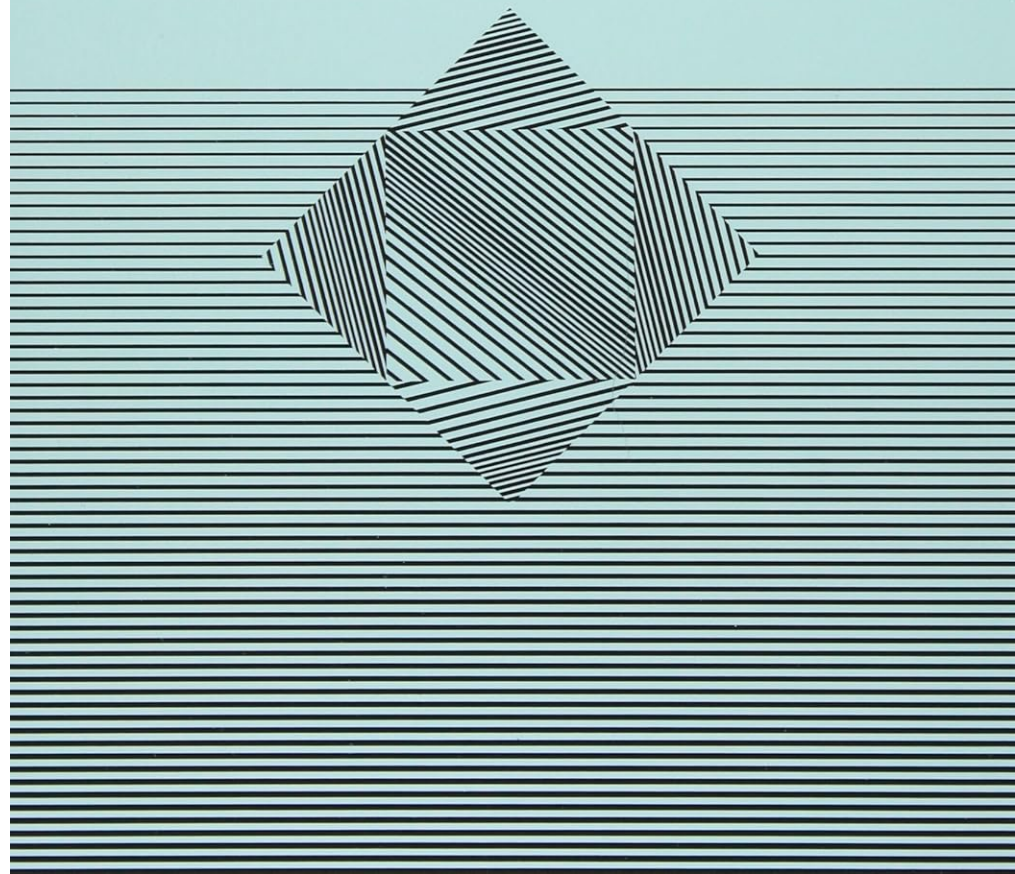
参考書

基礎統計学Ⅲ

自然科学の統計学

東京大学教養学部統計学教室 編

東京大学出版会



授業ページ

学内：<http://roy.e.koeki-u.ac.jp/>

学外：<https://www.koeki-prj.org/roy/>

東北公益文科大学 学生専用サーバ

重要連絡

[s4 アカウントの設定を](#) / [世界大学ランキングアンケート2025のお願い](#) [オンライン学習環境について](#) / [個人PCのネットワーク接続について](#) / [教育研究棟での無線LAN接続](#) / [教科書・参考書の購入](#) / [パスワード管理をしっかりと](#) / [情報PC端末使用上の注意](#) / [s4](#) / [設定変更・障害情報](#)

教員のページ

[神田](#) / [広瀬](#) / [西村](#) / [山本](#) / [植田](#) / [平居](#) / [廣瀬美](#) / [保科](#) / [宝賀](#) / [神田和](#) / [カロール](#) / [ヤグナ](#)

1年次(編入生)

- [各種マニュアル](#) (まずここ!)
- [Webページの書き方](#)
- [情報教室授業環境のショートカットキー](#)

学生専用サーバ→平居→大学院情報科学研究2 (データ解析特論)

今回

第1回 データの理解

第2回 確率分布と統計モデル

第3回 確率分布と統計モデルの最尤推定

第4回 一般化線形モデル

第5回 統計モデルの選択

第6回 尤度比検定と検定の非対称性

第7回 一般化線形モデルの応用

第8回 一般化線形混合モデル

第9回 マルコフ連鎖モンテカルロ法

第10回 ベイズ統計モデル

第11回 一般化線形モデルのベイズモデル化

第12回 事後分布の推定

第13回 階層ベイズモデル

第14回 空間構造のある階層ベイズモデル

第15回 まとめ

今回の目標

統計モデルとは何か
説明できる。

今回の内容

統計解析の必要性

統計モデル

プログラミング言語Python

今回の内容

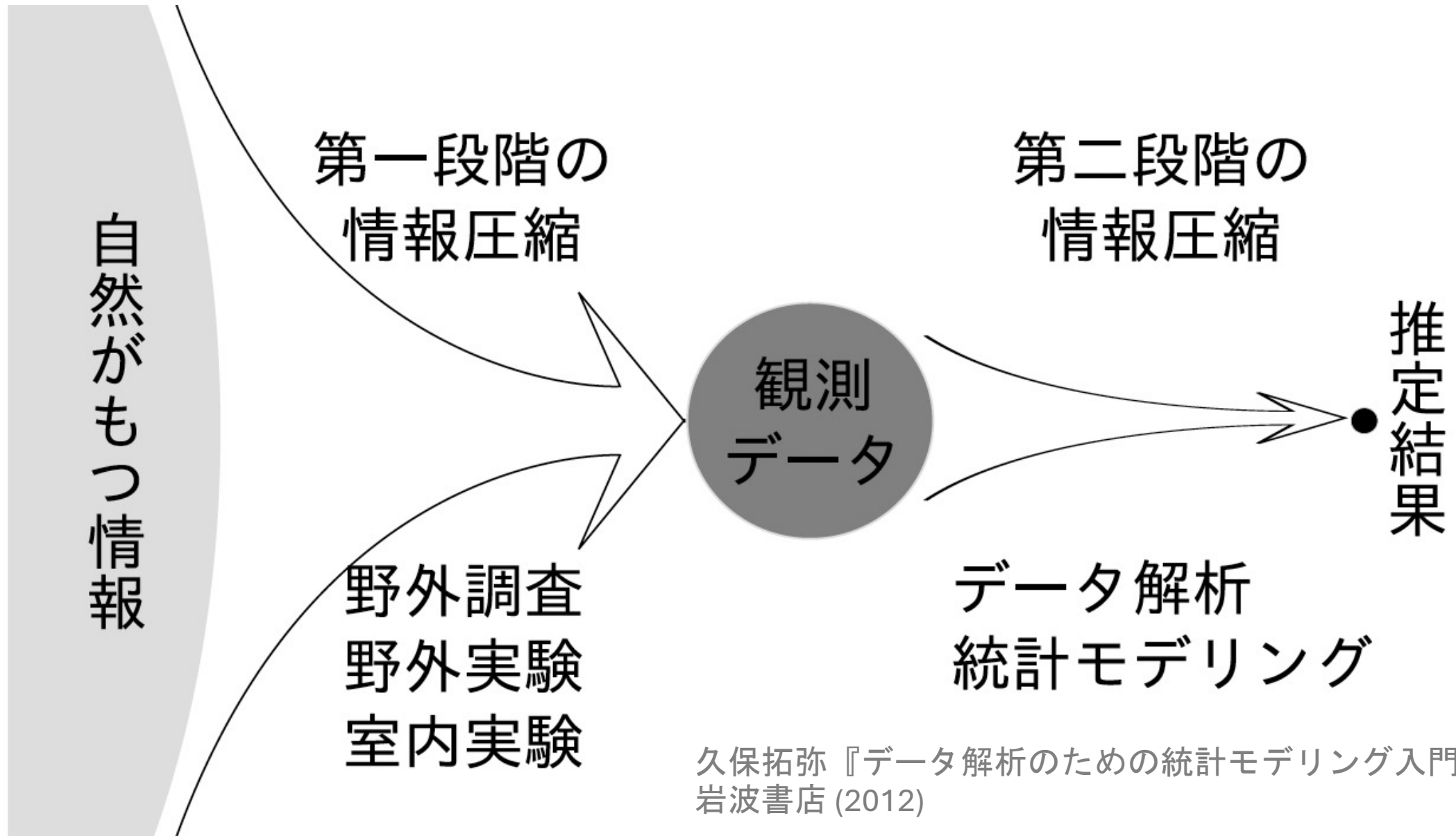
統計解析の必要性

統計モデル

プログラミング言語Python

なぜ統計学・統計モデリングが必要か？

情報消失・情報圧縮



久保拓弥『データ解析のための統計モデリング入門』
岩波書店 (2012)

情報損失 ・ 情報圧縮

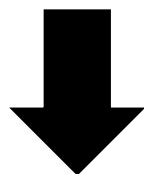
- 第一段階：自然現象から観測・実験などで情報を取り出してデータに変えるとき
- 第二段階：データを統計解析するとき

第一段階の情報消失・情報圧縮

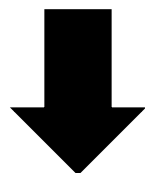
様々な観測機器を駆使しても自然が持つ情報のごく一部しかデータ化できない。

第二段階の情報損失・情報圧縮

我々は数字・記号の羅列をうまく把握できない。



統計解析



情報が失われる。

自然現象、社会現象を理解するには？

情報を捨てることで人間でも理解できるようにする必要がある。

統計解析の必要性

データ化した情報は統計的な方法で扱うしかない。

統計解析は客観的で再現性がある。

今回の内容

統計解析の必要性

統計モデル

プログラミング言語Python

データ解析とは

収集されたデータを分析した結果から、一定の法則性や共通点を見出し、問題の解決に役立てること

データ解析＝統計モデリング

統計モデル

観測データやそのデータ
に見られる様々なパター
ンをうまく説明できる確
率論的な数理モデル

統計モデルの特徴1

観測データのパターン
をうまく説明できる。

統計モデルの特徴2

統計モデルの部品は確率分布

確率分布の形はパラメーターによって決まる。

統計モデルの特徴3

観測データをうまく説明できるようにパラメータの値を決める。

＝「統計モデルのあてはめ」

または「統計モデルによる推定」

統計モデルへのあてはまりの良さを定量的に評価できる。

今回の内容

統計解析の必要性

統計モデル

プログラミング言語Python

データ解析とコンピュータ

大規模なデータ解析にはコンピュータが必要



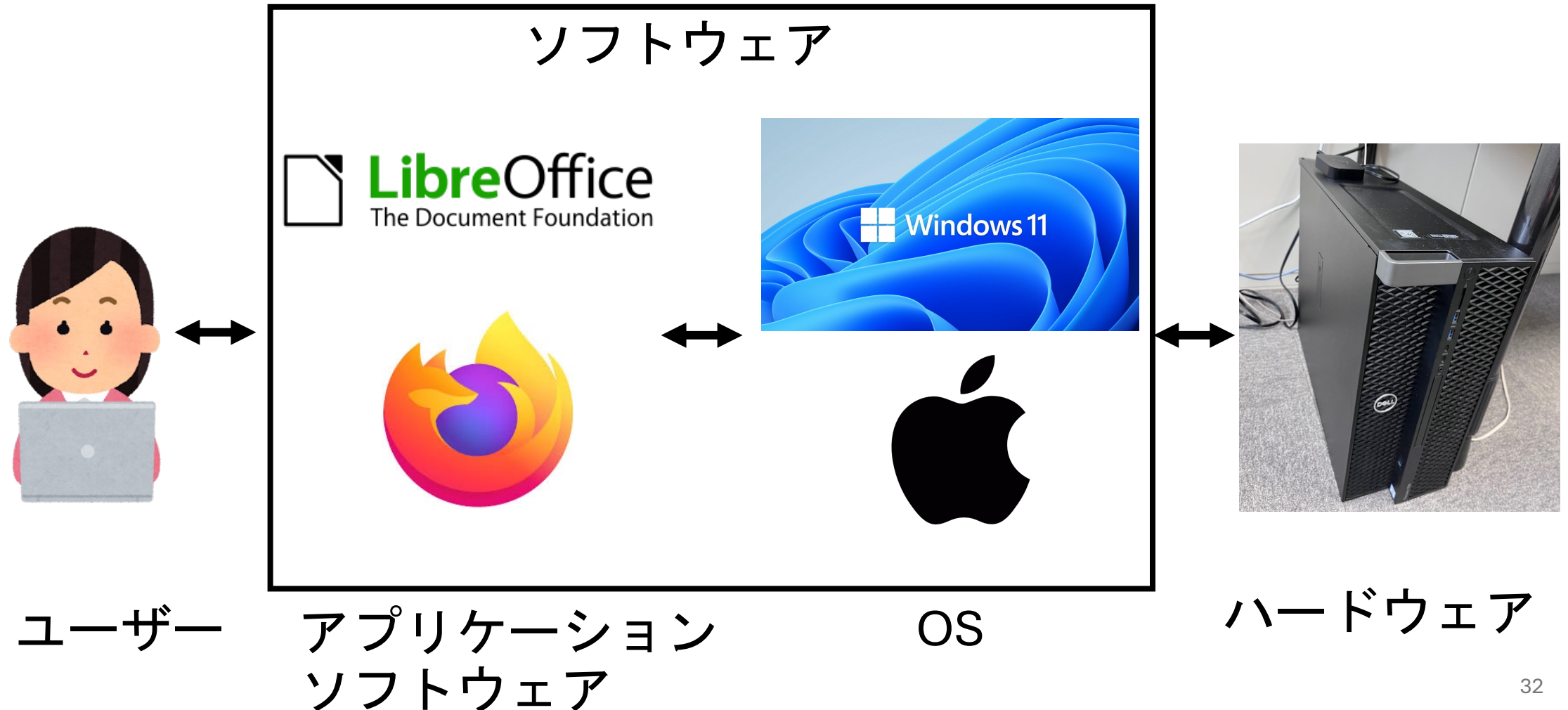
<https://www.dsk-cloud.com/blog/solution-big-data-for-the-future>



©NAOJ

計算機の仕組み（ソフトウェア）

コンピュータを動かすにはプログラミングが必要



プログラミングとは？

プログラミング言語の
文法に従って処理手順
を書く工程

コンピュータは日本語を読めない

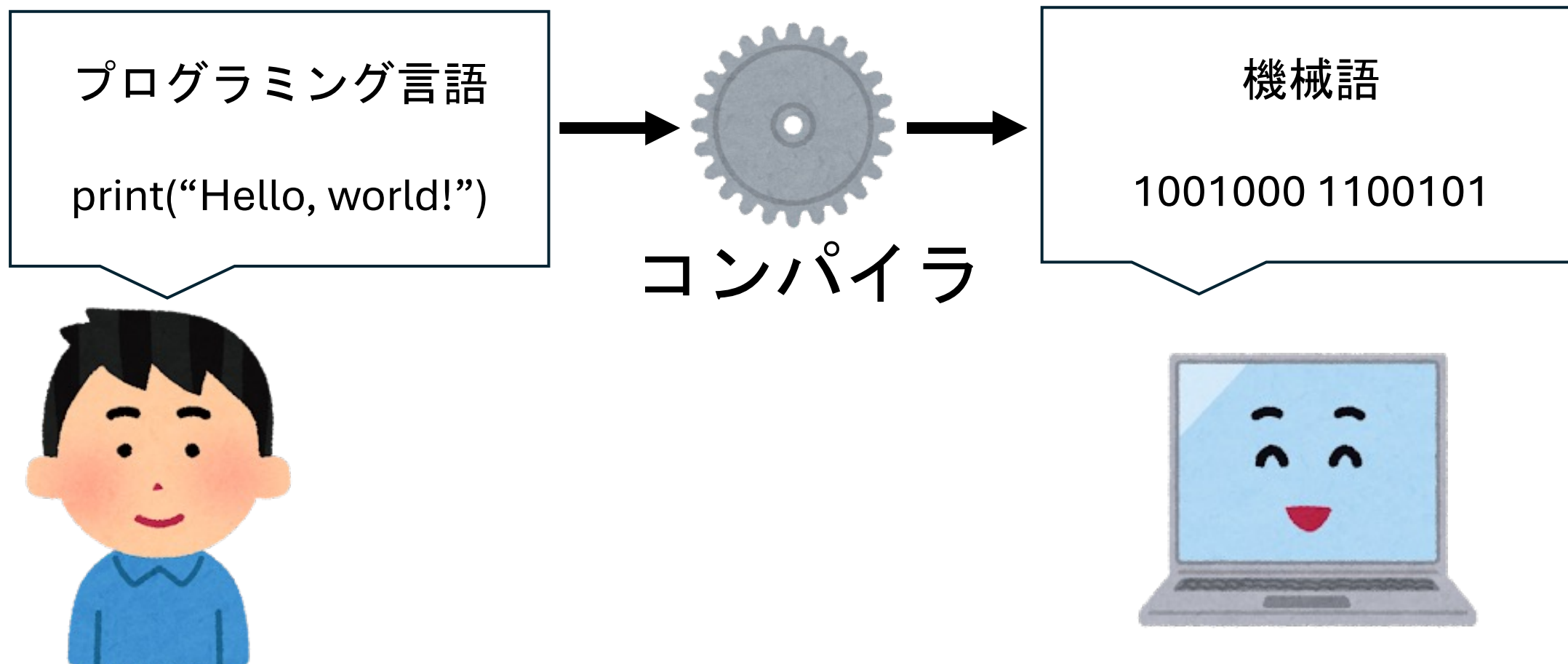


1 + 1 を
計算して



コンパイル

プログラミング言語をコンピュータが読める機械語に翻訳



プログラミングの用語

ソースコード：

人間が読みやすいプログラミング言語で書かれたプログラム

コンパイラ：

人間が書いたプログラムを機械語に変換する機能

機械語：

コンピュータが読むことができる0と1で表された言語

主なプログラミング言語

Python
(パイソン)



開発者：

グイド・ヴァンロッサム
(1991年)

用途：

データ解析、人工知能
など

Ruby
(ルビー)



まつもとゆきひろ
(1995年)

Web開発など

C言語
(シーげんご)



デニス・リッチー
(1972年)

科学技術計算、OSなど

プログラミング言語Python python™

1991年にオランダ出身のプログラマ、グイド・ヴァン・ロッサムにより開発



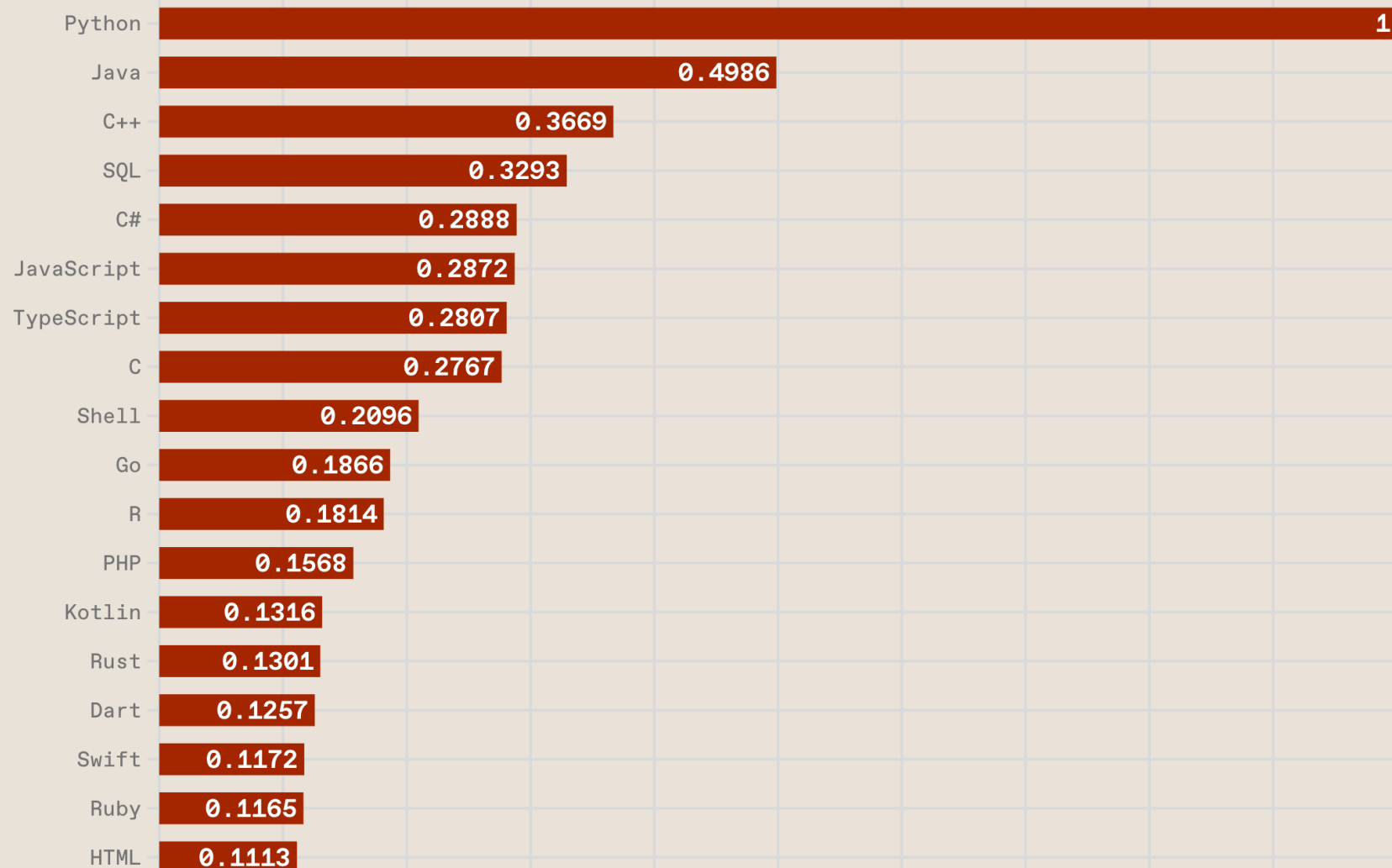
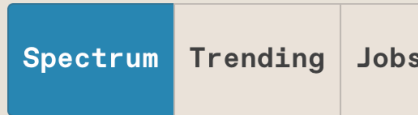
なぜPythonを学ぶのか？

IEEEプログラミング言語ランキング9年連続1位
(2017–2025年)

Top Programming Languages 2025

<https://spectrum.ieee.org/top-programming-languages-2025>

Click a button to see a differently weighted ranking



なぜPythonを学ぶのか？

多数のアプリケーションで開発言語として使用



なぜPythonを学ぶのか？

データ解析や人工知能が得意

充実したライブラリ（再利用可能なコードの集合体）

データ分析：Numpy, Pandas, Matplotlibなど

機械学習：Scikit-learn, TensorFlowなど

Web開発：Flask, Djangoなど

なぜPythonを学ぶのか？

文法（書き方）がシンプル

例：“Hello, world!”と出力するプログラム



```
print("Hello, world!")
```



```
#include <stdio.h>

int main (void)
{
    printf("Hello, world!\n");
    return 0;
}
```

なぜPythonを学ぶのか？

コードをすぐ実行できる。



```
print("Hello, world!")
```

コードを1行
ずつ翻訳



```
#include <stdio.h>

int main (void)
{
    printf("Hello, world!\n");
    return 0;
}
```

コード全体を
翻訳



Pythonインストール

Pythonのダウンロード

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

The screenshot shows the Python.org website. At the top is a navigation bar with links: Python, PSF, Docs, PyPI, Jobs, and Community. Below this is the Python logo and a search bar with a 'GO' button and a 'Socialize' link. A secondary navigation bar contains links: About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area features a heading 'Download the latest version for macOS' and a yellow button labeled 'Download Python 3.14.4'. Below the button, there is text providing links for other operating systems: 'Looking for Python with a different OS? Python for [Windows](#), [Linux/Unix](#), [macOS](#), [Android](#), [other](#)'. Further down, it says 'Want to help test development versions of Python 3.15? [Pre-releases](#), [Docker images](#)'. The background of the main content area features an illustration of two parachutes with cargo boxes hanging from them, set against a blue sky with clouds.

Pythonインストール

ダウンロードしたファイル
(python-manager-26.1)を
ダブルクリックして実行

「Pythonのインストール」をクリック

Python Install Manager をインストールしますか？

発行元: Python Software Foundation

バージョン: 26.1.240.0

ソース: www.python.org



機能:

- すべてのシステム リソースを使用する
- アンインストール時にクリーンアップされていないレジストリ エントリとレジストリ ファイルを書き込みます

☒ 準備ができたなら起動

Python のインストール

キャンセル

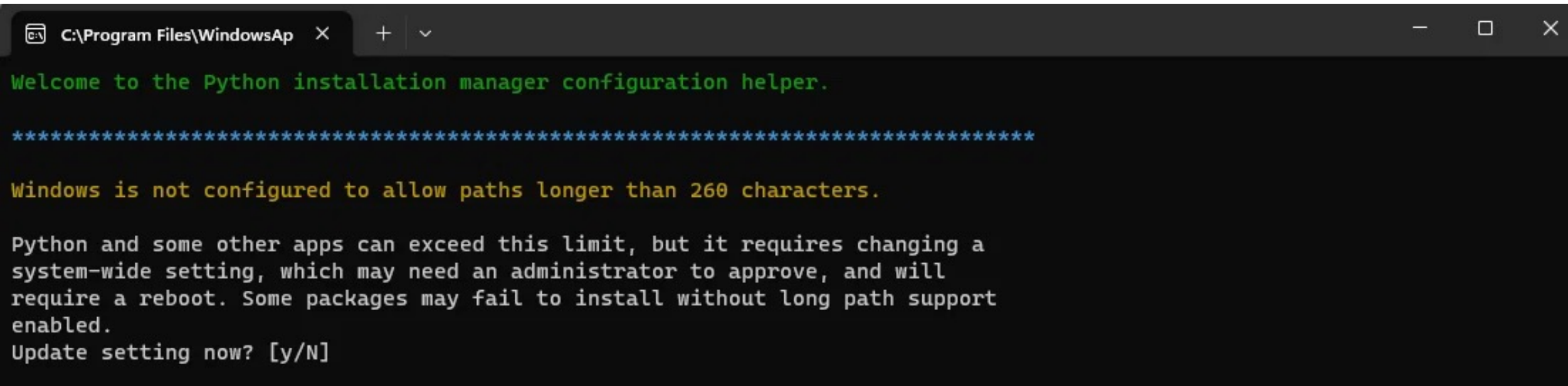
[Python website](#)

[Get help with this installer](#)



インターネット アプリケーションは、コンピューターに問題を起こす可能性があります。ソースが信頼できない場合は、このソフトウェアをインストールしないでください。 [詳細情報](#)

Pythonインストール



```
C:\Program Files\WindowsAp
Welcome to the Python installation manager configuration helper.

*****

Windows is not configured to allow paths longer than 260 characters.

Python and some other apps can exceed this limit, but it requires changing a
system-wide setting, which may need an administrator to approve, and will
require a reboot. Some packages may fail to install without long path support
enabled.
Update setting now? [y/N]
```

yを選択してEnter

https://note.com/maru_5251/n/n173a1c9623cb

Pythonインストール

```
Welcome to the Python installation manager configuration helper.

*****

Windows is not configured to allow paths longer than 260 characters.

Python and some other apps can exceed this limit, but it requires changing a
system-wide setting, which may need an administrator to approve, and will
require a reboot. Some packages may fail to install without long path support
enabled.
Update setting now? [y/N] y
The setting has been successfully updated, and will take effect after the next reboot.

*****

The global shortcuts directory is not configured.

Configuring this enables commands like python3.14.exe to run from your
terminal, but is not needed for the python or py commands (for example, py
-V:3.14).

We can add the directory (C:\Users\astorp25\AppData\Local\Python\bin) to PATH
now, but you will need to restart your terminal to use it. The entry will be
removed if you run py uninstall --purge, or else you can remove it manually when
uninstalling Python.
Add commands directory to your PATH now? [y/N] . .
```

yを選択してEnter

Pythonインストール

```
You do not have the latest Python runtime.
```

```
Install the current latest version of CPython? If not, you can use 'py install  
default' later to install.
```

```
Install CPython now? [Y/n] ☐
```

Yを選択してEnter

https://note.com/maru_5251/n/n173a1c9623cb

Pythonバージョン確認

PowerShellに

```
py list
```

と入力し、インストール済みの
Pythonバージョンを確認する。

作業用フォルダの作成

PowerShellに

```
mkdir C:\projects\data_analysis
```

と入力

作業用フォルダへの移動

PowerShellに

```
cd C:\projects\data_analysis
```

と入力

仮想環境 (venv) を作成

PowerShellに

```
py -m venv .venv
```

と入力

仮想環境をアクティベート

PowerShellに

```
.\.venv\Scripts\Activate.ps1
```

と入力

パッケージのインストール

pipコマンド

パッケージ（プログラムをまとめたもの）をインストールするためのコマンド

パッケージ公開サイト:

PyPI – The Python Package Index

<https://pypi.org/>

pipの更新

PowerShellに“**py -m pip
install -U pip**”と入力して
Enterキーを押す

Numpy

数値計算に特化したライブラリ

大規模データを効率良く扱える

公式サイト：

<https://numpy.org/ja/>



Numpyインストール

PowerShellに“**pip install numpy**”
と入力してEnterキーを押す

```
PS C:\Users\hirai> pip install numpy
Collecting numpy
  Downloading numpy-2.2.5-cp313-cp313-win_amd64.whl.metadata (60 kB)
  Downloading numpy-2.2.5-cp313-cp313-win_amd64.whl (12.6 MB)
    12.6/12.6 MB 16.6 MB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-2.2.5
```

Pandas

データ整形用ライブラリ

機能：

- ・ ファイルの読み書き
- ・ 列・行の追加や削除
- ・ データの抽出 など



公式サイト：

<https://pandas.pydata.org/>

Pandasインストール

PowerShellに“**pip install pandas**”と入力してEnterキーを押す

```
PS C:\Users\hirai> pip install pandas
Collecting pandas
  Downloading pandas-2.2.3-cp313-cp313-win_amd64.whl.metadata (19 kB)
Requirement already satisfied: numpy>=1.26.0 in c:\users\hirai\appdata\local\programs\python\python313\lib\
(2.2.5)
Collecting python-dateutil>=2.8.2 (from pandas)
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting six>=1.5 (from python-dateutil>=2.8.2->pandas)
  Downloading six-1.17.0-py2.py3-none-any.whl.metadata (1.7 kB)
Downloading pandas-2.2.3-cp313-cp313-win_amd64.whl (11.5 MB)
  11.5/11.5 MB 21.3 MB/s eta 0:00:00
Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Downloading pytz-2025.2-py2.py3-none-any.whl (509 kB)
Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Downloading tzdata-2025.2-py2.py3-none-any.whl (347 kB)
Installing collected packages: pytz, tzdata, six, python-dateutil, pandas
Successfully installed pandas-2.2.3 python-dateutil-2.9.0.post0 pytz-2025.2 six-1.17.0 tzdata-2025.2
```

Matplotlib



グラフ描画用ライブラリ

データの可視化に便利

公式サイト：

<https://matplotlib.org/>

Matplotlibインストール

PowerShellに“**pip install matplotlib**”と入力してEnter
キーを押す

```
PS C:\Users\hirai> pip install matplotlib
```

```
Successfully installed contourpy-1.3.2 cyclor-0.12.1 fonttools-4.57.0 kiwisolver-1.4.8 matplotlib-3.10.1 packaging-25.0 pillow-11.2.1  
pyparsing-3.2.3  
PS C:\Users\hirai> |
```

rdataインストール

PowerShellに“**pip install rdata**”
と入力してEnterキーを押す

JupyterLab

対話型のプログラム実行環境

Webブラウザ上で各種プログラムの実行
と結果の参照、ドキュメント作成などが
行える

他の言語 (Julia, R, Rubyなど)にも対応

公式サイト：

<https://jupyter.org/>



JupyterLabインストール

PowerShellに“**pip install jupyterlab**”と入力してEnter
キーを押す

```
PS C:\Users\hirai> pip install jupyterlab
```

JupyterLabの実行

PowerShellに

“**py -m jupyter lab**”

と入力してEnterキーを
押す

JupyterLab

localhost:8888/lab

File Edit View Run Kernel Tabs Settings Help

Launcher

Notebook

Python 3 (ipykernel)

Console

Python 3 (ipykernel)

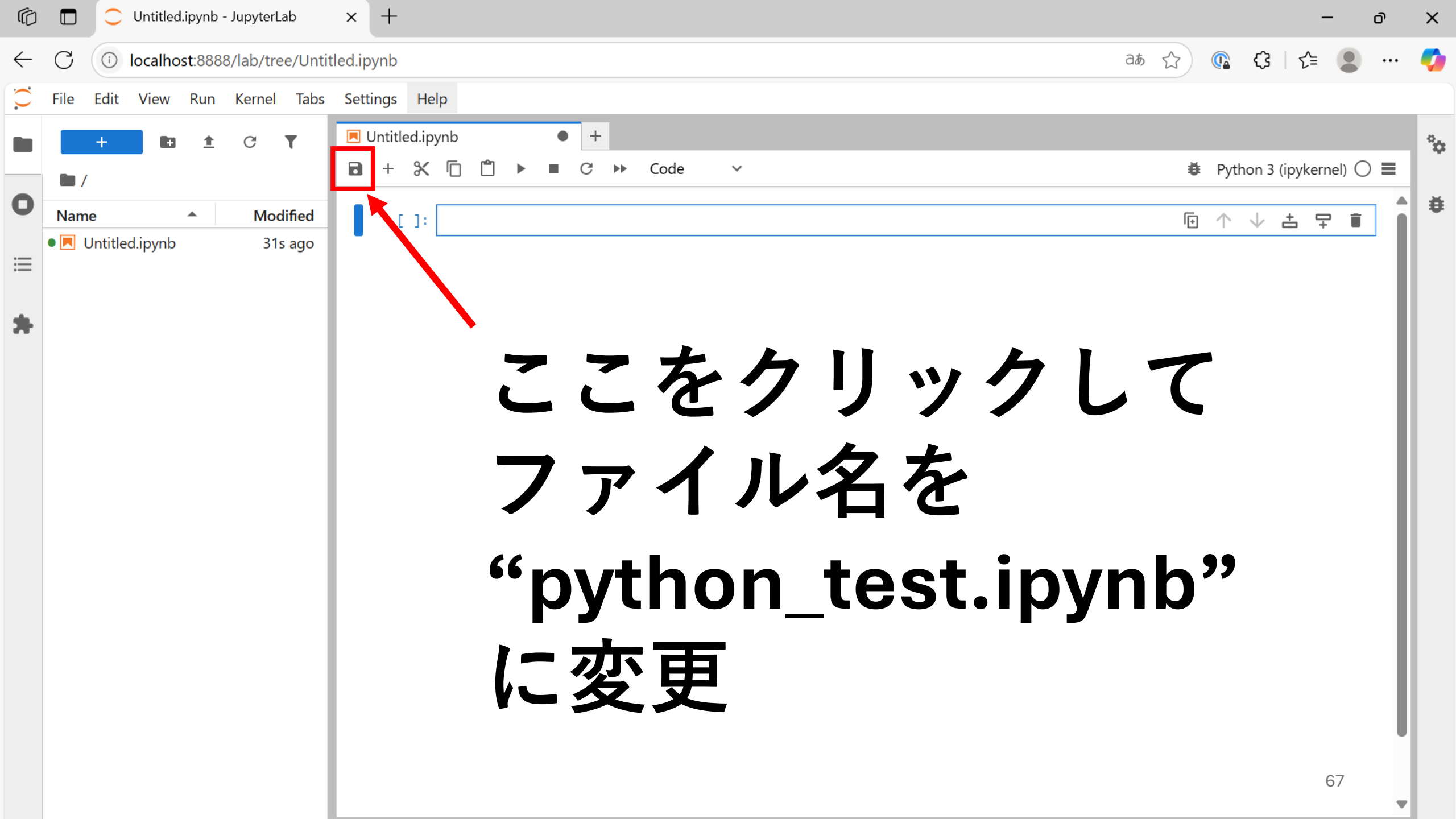
Other

Terminal Text File Markdown File Python File Show Contextual Help

Simple 0

Launcher 0

ここをクリックして
ノートブックを作成

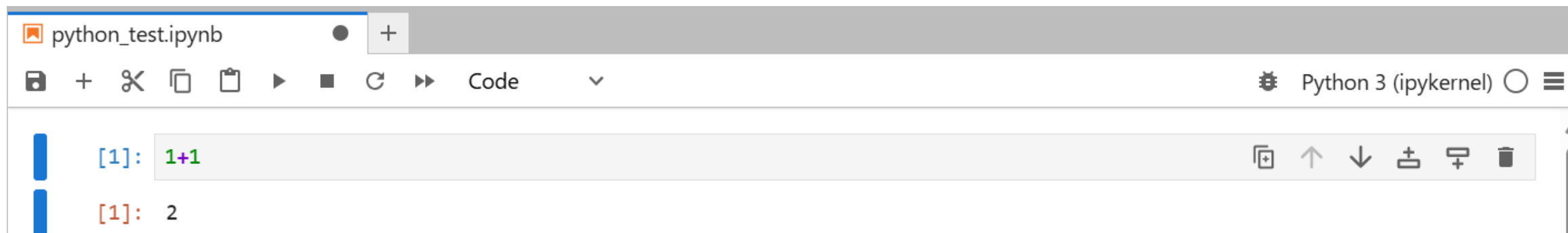


ここをクリックして
ファイル名を
“python_test.ipynb”
に変更

Pythonの実行

1. セルに “1+1” と入力

2. Shift+Enter か ▶ をクリック



The screenshot shows a Jupyter Notebook window titled 'python_test.ipynb'. The interface includes a toolbar with icons for saving, adding, deleting, copying, pasting, and running code. The 'Code' tab is selected. A code cell contains the text '[1]: 1+1'. Below the code cell, the output is displayed as '[1]: 2'. The notebook is running on 'Python 3 (ipykernel)'.

エラーが出たら

[2]: 1+1

Cell In[2], line 1

1+1

^

SyntaxError: invalid character '1' (U+FF11)

日本語入力はoffにする
エラーメッセージを見てコードを修正
正しく実行できるまで試す

様々な計算

```
[3]: 50 - 5 * 6
```

```
[3]: 20
```

```
[4]: (50 - 5 * 6) / 4
```

```
[4]: 5.0
```

```
[5]: 17 / 3
```

```
[5]: 5.666666666666667
```

```
[6]: 17 // 3 #整数除算 ("#"の後はコメント)
```

```
[6]: 5
```

```
[8]: 17 % 3 #余り
```

```
[8]: 2
```

```
[9]: 5 ** 2 #5の2乗
```

```
[9]: 25
```

文字の出力

`print()` という関数を使う

```
[10]: print("Hello, world!")
```

```
Hello, world!
```

```
[11]: s="Hello, world!"
```

```
[12]: print(s)
```

```
Hello, world!
```

リスト

```
[13]: squares = [1, 4, 9, 16, 25]
```

```
[14]: squares
```

```
[14]: [1, 4, 9, 16, 25]
```

```
[15]: squares[0] #最初の数字
```

```
[15]: 1
```

```
[16]: squares[2] #3番目の数字
```

```
[16]: 9
```

```
[18]: squares[1:4] #listの2番目から4番目までを取得
```

```
[18]: [4, 9, 16]
```

```
[19]: squares[2:] #listの3番目から最後までを取得
```

```
[19]: [9, 16, 25]
```

簡単なプログラムーフィボナッチ数列

$$\begin{cases} F_0 = 0 \\ F_1 = 1 \\ F_n = F_{n-1} + F_{n-2} \quad (n \geq 2) \end{cases}$$

```
#フィボナッチ数列
a, b = 0, 1          # aに1, bに0を代入
while a < 10:        # aが10より小さい間実行
    print(a)         # aを出力
    a, b = b, a + b  # aにbを代入、bに代入前のa + bを代入
```

0
1
1
2
3
5
8

今回の内容

統計解析の必要性

統計モデル

プログラミング言語Python

今回の目標

統計モデルとは何か
説明できる。

課題

Pythonのインストールを完了させ、授業資料
p.61-66のプログラムを実行する。

提出方法：第2回授業冒頭で実行結果を確認

次回

第1回	データの理解	第9回	マルコフ連鎖モンテカルロ法
第2回	確率分布と統計モデル	第10回	ベイズ統計モデル
第3回	確率分布と統計モデルの最尤推定	第11回	一般化線形モデルのベイズモデル化
第4回	一般化線形モデル	第12回	事後分布の推定
第5回	統計モデルの選択	第13回	階層ベイズモデル
第6回	尤度比検定と検定の非対称性	第14回	空間構造のある階層ベイズモデル
第7回	一般化線形モデルの応用	第15回	まとめ
第8回	一般化線形混合モデル		