

マテリアルズ・ インフォマティクス の勉強法など

早稲田大学 応用化学科
畠山 歓

マテリアルズ・ インフォマティクス

Materials
Informatics

岩崎悠真 著
Yuma Iwasaki

材料開発のための
機械学習
超入門

この
機械学習
アルゴリズムは
どう使える？



日刊工業新聞社

12の開発事例で
わかりやすく解説！

おすすめサイト・書籍など

- マテリアルズ・インフォマティクス-材料開発のための機械学習超入門
 - MIの概要を知る
- 明治大学 金子先生のWebサイト
 - <https://datachemeng.com/>
 - MIのTipsなどが充実

東京大学の

中山浩太郎【監修】 松尾 豊【協力】
塚本邦尊、山田典一、大澤文孝【著】

データサイエンティスト 育成講座

Pythonで手を動かして
学ぶデータ分析



「これからデータサイエンティストとして活躍したい方だけでなく、研究や開発の現場で利用したい方、個人や組織の持つデータを解析する技術を身に付けたい方など、**幅広い方に対する入門書**として利用できるように構成しています」



— 東京大学 大学院工学系研究科 松尾研究室 リサーチディレクター —
中山 浩太郎

おすすめ書籍

Pythonで手を動かしながら、データサイエンスを学ぶ

MIのやり方

- プログラミング言語(例: Python)を使う
 - 敷居は少し高い
 - 拡張性に優れ、最先端のモデルなどを使えるので、最終的にはこれ一択
- ソフトウェアを使う
 - 雰囲気を知るのにお勧め。下記に例。
 - [KNIME](#)
 - 機械学習ツール
 - [Neural Network Console](#)
 - 深層学習ツール
 - [Microsoft Azure Machine learning](#)
 - 機械学習ツール

勉強の 基本姿勢

- 習うより慣れる (プログラミングの基本)
- 分からないことは、聞く or 検索
- まずは完成済みのコードを真似・改変してみる

マイルストーン

Pythonの基礎を習得

機械学習の基礎的な勉強

材料構造から物性を予測

Python入門

参考 MIに必要な 基本技能

1. Python 基本文法
2. 基本的なライブラリの操作習得 (numpy, pandas, matplotlib)
3. 機械学習ライブラリの操作習得(Scikit-learn)
 - 各種回帰モデル (Lasso, ランダムフォレスト)
 - データセットの作成(正規化, 訓練・試験データ分割)
 - 任意: 次元圧縮
4. 任意: ニューラルネットワークの操作(Keras)
 - 画像認識(MNIST, cifar10)、文章 (LSTM, seq2seq)、その他(VAE, GAN, ...)
5. 化学構造の取扱い (RDKit)
 - Fingerprintや記述子を使い、”化学構造を数値化”
 - 公開データベースを用い、融点などの予測

基本文法の習得

- **Webサイトの活用がおすすめ**

- プロゲート https://prog-8.com/users/sign_in

- Numpy, pandas, 線形回帰など 入門

- <https://www.codexa.net/>

- データプラーニング入門

- <https://aidemy.net/courses/5090>

- Paiza

- <https://paiza.jp/works/python3/primer>

- Aidemy

- <https://aidemy.net/courses/3010>

機械学習に最低限必要なレベル?

- 誰かが書いたコードを実行できる
- 基本ライブラリを少し使える (numpy, pandas, matplotlib)
- 機械学習ライブラリを少し使える (scikit-learn)

Pythonの導入

```
for object to mirror:
    mirror_mod.mirror_object = object

    operation == "MIRROR_X":
        mirror_mod.use_x = True
        mirror_mod.use_y = False
        mirror_mod.use_z = False
    operation == "MIRROR_Y":
        mirror_mod.use_x = False
        mirror_mod.use_y = True
        mirror_mod.use_z = False
    operation == "MIRROR_Z":
        mirror_mod.use_x = False
        mirror_mod.use_y = False
        mirror_mod.use_z = True

#selection at the end -add
mirror_ob.select= 1
modifier_ob.select=1
context.scene.objects.active = modifier_ob
context.scene.objects.active.select = 1
mirror_ob.select = 0
= bpy.context.selected_objects
data.objects[one.name].select

print("please select exactly one object")

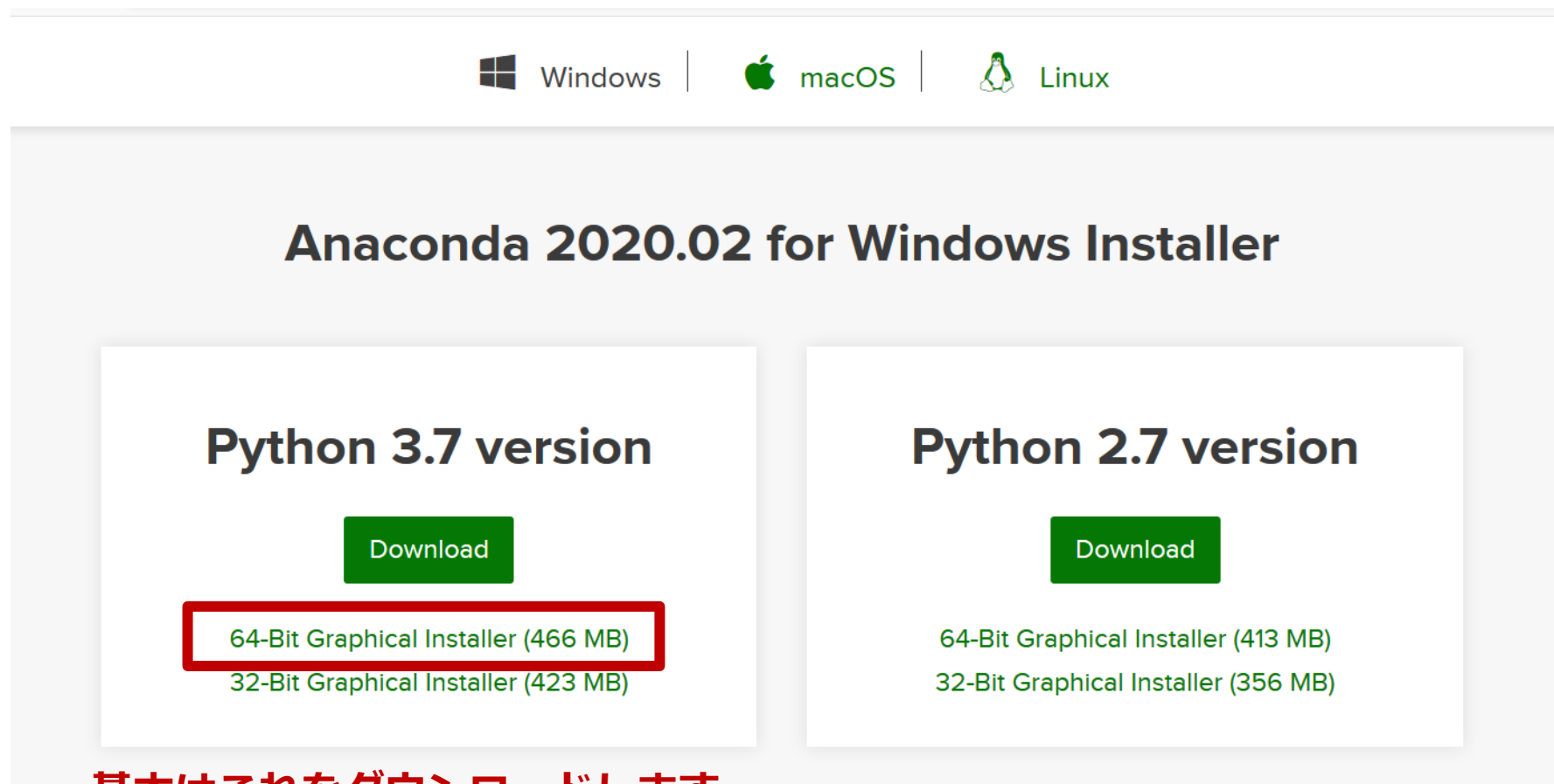
-- OPERATOR CLASSES -----

bpy.types.Operator):
    bl_name = "X mirror to the selected object.mirror_mirror_x"
    bl_label = "Mirror X"
```

Pythonのインストール

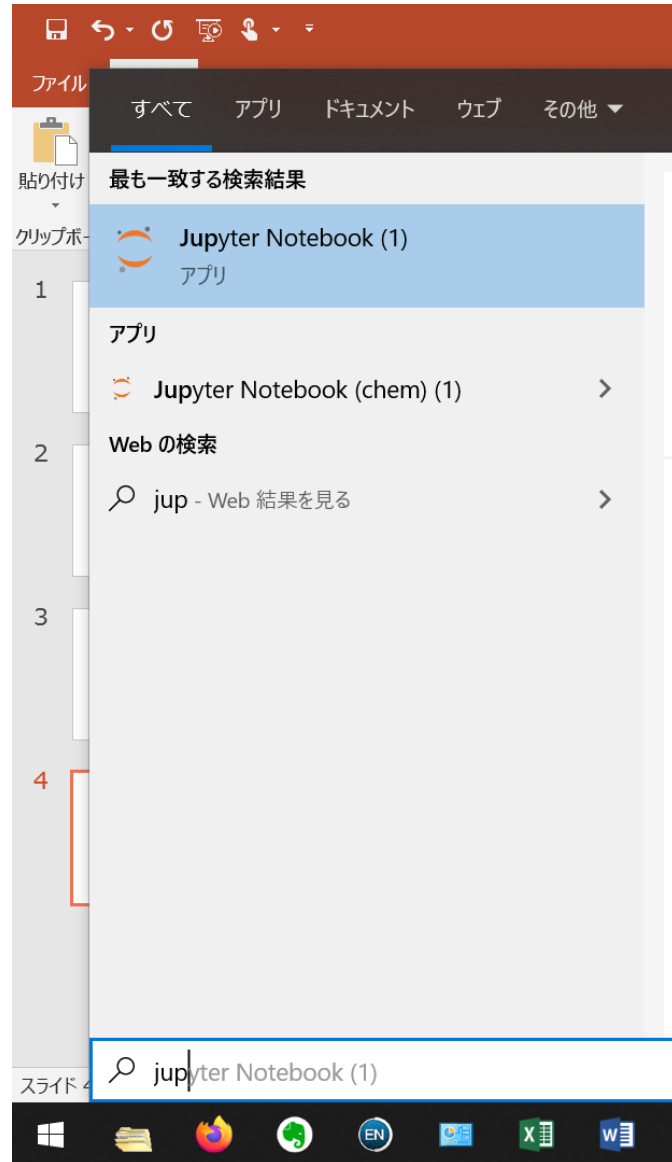
Anacondaというパッケージをダウンロードし、実行

<https://www.anaconda.com/distribution/#download-section>



基本はこれをダウンロードします
(python 3.7, windows, 64bit PCの場合)

起動方法



Windowsのスタートメニューから
“Jupyter notebook”と入力し、
このアプリを実行

ブラウザにこんな感じの画面が出てくる

The screenshot displays the JupyterLab web interface in a browser. The address bar shows the URL `localhost:8888/tree`. The Jupyter logo is on the left, and 'Quit' and 'Logout' buttons are on the right. Below the navigation bar, there are tabs for 'Files', 'Running', and 'Clusters', with 'Files' being the active tab. A message 'Select items to perform actions on them.' is followed by 'Upload', 'New', and a refresh icon. The main area shows a file browser with a table of files and folders. The table has columns for 'Name', 'Last Modified', and 'File size'. The files listed are '3D Objects', 'Anaconda3', 'Box Sync', 'build', 'chemaxon', 'Contacts', and 'Creative Cloud Files', each with a checkbox and a folder icon. The 'Last Modified' column shows dates like '19日前' and '7ヶ月前'.

localhost:8888/tree

jupyter

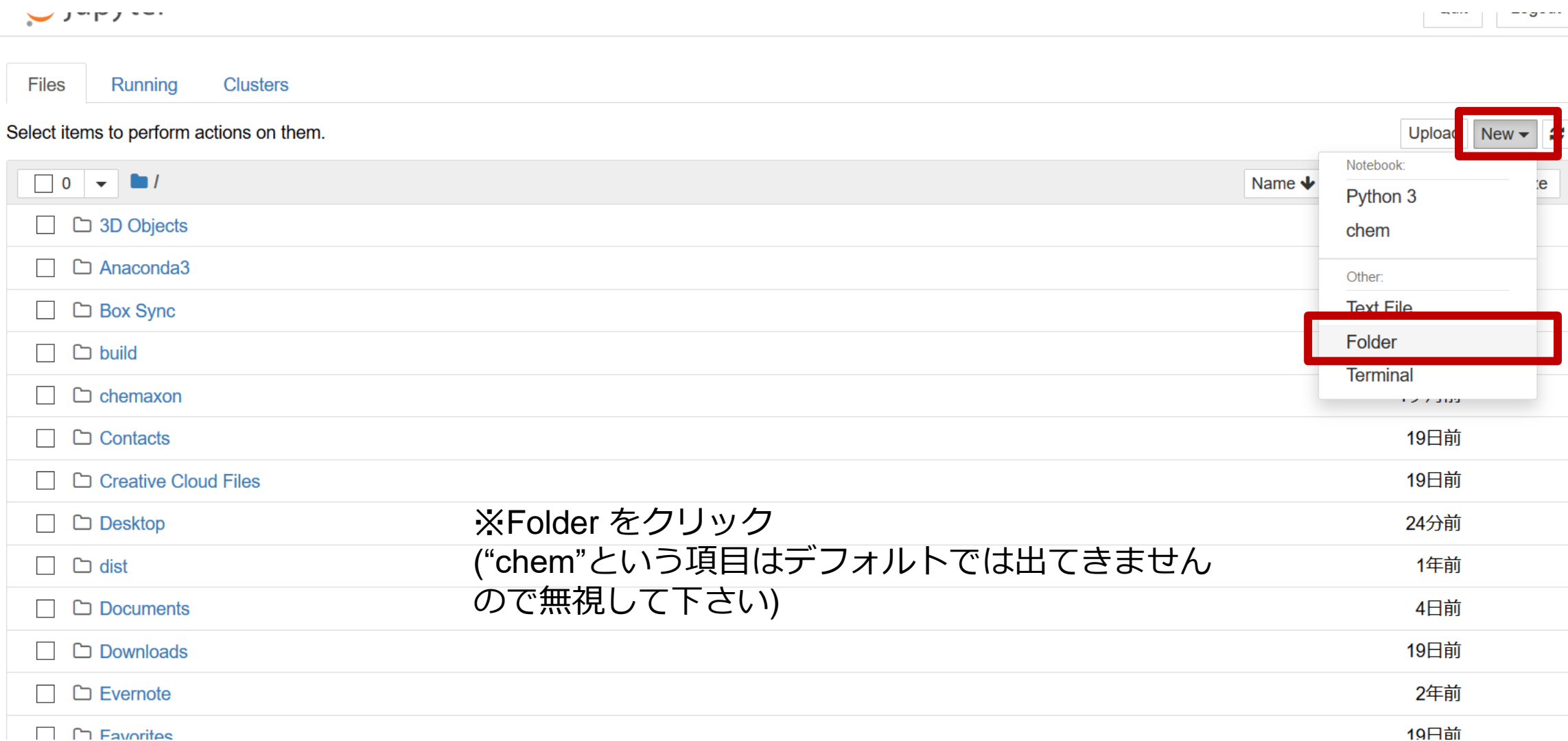
Quit Logout

Files Running Clusters

Select items to perform actions on them. Upload New ↺

<input type="checkbox"/> 0 ▾	📁 /	Name ▾	Last Modified	File size
<input type="checkbox"/>	📁 3D Objects		19日前	
<input type="checkbox"/>	📁 Anaconda3		7ヶ月前	
<input type="checkbox"/>	📁 Box Sync		1年前	
<input type="checkbox"/>	📁 build		1年前	
<input type="checkbox"/>	📁 chemaxon		1ヶ月前	
<input type="checkbox"/>	📁 Contacts		19日前	
<input type="checkbox"/>	📁 Creative Cloud Files		19日前	

好きな場所に作業用フォルダを作る



The screenshot shows a cloud storage interface with a top navigation bar containing 'Files', 'Running', and 'Clusters'. Below the navigation bar, there is a prompt 'Select items to perform actions on them.' and a 'New' button highlighted with a red box. A dropdown menu is open from the 'New' button, showing options: 'Notebook:', 'Python 3', 'chem', 'Other:', 'Text File', 'Folder' (highlighted with a red box), and 'Terminal'. The main area displays a list of folders with checkboxes and a 'Name' column header. The folders listed are: 3D Objects, Anaconda3, Box Sync, build, chemaxon, Contacts, Creative Cloud Files, Desktop, dist, Documents, Downloads, Evernote, and Favorites. A text overlay is present over the 'Desktop' and 'dist' rows.

Files Running Clusters

Select items to perform actions on them.

Upload New

Notebook:
Python 3
chem

Other:
Text File
Folder
Terminal

<input type="checkbox"/> 0	Name ↓	
<input type="checkbox"/>	3D Objects	
<input type="checkbox"/>	Anaconda3	
<input type="checkbox"/>	Box Sync	
<input type="checkbox"/>	build	
<input type="checkbox"/>	chemaxon	
<input type="checkbox"/>	Contacts	19日前
<input type="checkbox"/>	Creative Cloud Files	19日前
<input type="checkbox"/>	Desktop	24分前
<input type="checkbox"/>	dist	1年前
<input type="checkbox"/>	Documents	4日前
<input type="checkbox"/>	Downloads	19日前
<input type="checkbox"/>	Evernote	2年前
<input type="checkbox"/>	Favorites	19日前

※Folder をクリック
("chem"という項目はデフォルトでは出てきません
ので無視して下さい)

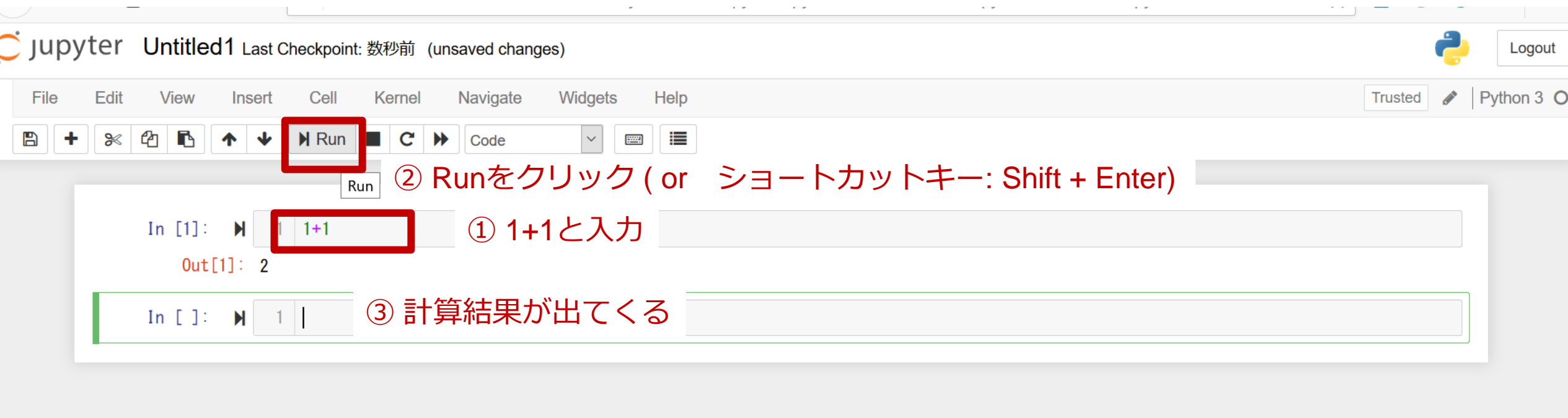
好きな場所にpython用のファイルを作る



The screenshot shows the JupyterLab web interface. At the top left is the Jupyter logo and the word 'jupyter'. On the top right are 'Quit' and 'Logout' buttons. Below the header is a navigation bar with 'Files', 'Running', and 'Clusters' tabs. The 'Files' tab is active, and below it is the text 'Select items to perform actions on them.' The main area shows a file browser with a breadcrumb path: '/ Box Sync / Personal / python / python練習会'. Below the path is a list of files: '..' (a folder), 'rdkit.ipynb', and 'Untitled.ipynb'. On the right side of the file browser, there are buttons for 'Upload' and 'New'. The 'New' button is highlighted with a red box, and its dropdown menu is open, showing options: 'Python 3', 'chem', 'Other:', 'Text File', 'Folder', and 'Terminal'. The 'Python 3' option is also highlighted with a red box.

※Python 3をクリック
(“chem”という項目はデフォルトでは出てきませんので無視して下さい)

スクリプトの入力画面が出てくる



The screenshot displays the Jupyter Notebook interface. At the top, the title bar shows 'jupyter' and 'Untitled1' with a status message 'Last Checkpoint: 数秒前 (unsaved changes)'. The top right corner includes a 'Logout' button and the Python logo. Below the title bar is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Navigate, Widgets, and Help. To the right of the menu bar are 'Trusted' and 'Python 3' indicators. A toolbar below the menu bar contains icons for saving, adding, deleting, and running cells. The 'Run' button, which has a play icon, is highlighted with a red rectangle. Below the toolbar, a code cell is shown with the prompt 'In [1]:' followed by a text input field containing '1+1'. This input field is also highlighted with a red rectangle. To the right of the input field, the text '② Runをクリック (or ショートカットキー: Shift + Enter)' is written in red. Below the code cell, the output 'Out[1]: 2' is displayed. At the bottom, another code cell is shown with the prompt 'In []:' followed by a text input field containing '1'. To the right of this input field, the text '③ 計算結果が出てくる' is written in red.

jupyter Untitled1 Last Checkpoint: 数秒前 (unsaved changes)

File Edit View Insert Cell Kernel Navigate Widgets Help Trusted Python 3

Run

② Runをクリック (or ショートカットキー: Shift + Enter)

In [1]: 1+1 ① 1+1と入力

Out[1]: 2

In []: 1 ③ 計算結果が出てくる

もっと複雑な計算をする

- “import numpy as np”

と入力 (計算用ライブラリnumpyをnpという略称で使いますという宣言)

色々と計算を試みる

例)

$\sin(2) + \cos(999)$

$\sin(\pi)$

$\sin(1+i)$

$\exp(1)$

```
In [2]: 1 import numpy as np
```

```
In [4]: 1 np.sin(2)+np.cos(222)
```

```
Out[4]: 0.41439901223627934
```

```
In [11]: 1 np.sin(np.pi)
```


```
Out[11]: 1.2246467991473532e-16
```

```
In [12]: 1 np.sin(1+1j)
```

```
Out[12]: (1.2984575814159773+0.6349639147847361j)
```

```
In [14]: 1 np.exp(1)
```

```
Out[14]: 2.718281828459045
```



やってみよう1

- Numpyを使って、 $e^{-i\pi}$ を計算

やってみよう2

(プログラミングの勉強が必要)

- 次のような九九の表を作る
- 多重ループを使うこと
- ヒント:改行しないでprintする方法は、“print 改行しない”でgoogle検索

1	2	3	4	5	6	7	8	9	10	11
2	4	6	8	10	12	14	16	18	20	22
3	6	9	12	15	18	21	24	27	30	33
4	8	12	16	20	24	28	32	36	40	44
5	10	15	20	25	30	35	40	45	50	55
6	12	18	24	30	36	42	48	54	60	66
7	14	21	28	35	42	49	56	63	70	77
8	16	24	32	40	48	56	64	72	80	88
9	18	27	36	45	54	63	72	81	90	99
10	20	30	40	50	60	70	80	90	100	110
11	22	33	44	55	66	77	88	99	110	121

今後、何をするか？

例

1) MIセミナー動画

(2020年度の講義(学内)用に作成した入門動画)

2) KNIMEによる材料探索(任意)

(プログラミング無しで、分子物性を予測する)

3) PythonによるMIの基本コードの動作確認

4) プラスα