

一般教育演習

プログラミングで問題を解く：

集計から人工知能まで

瀧川 一学

工学部 情報理工学コース

次回からPythonを掘り下げたい

宿題：

- USBメモリからJupyter Notebookを起動する手順を確認復習しておくこと
- 特に毎回必要となるコマンドプロンプト作業も確認復習しておくこと
- Jupyter Notebookにいろいろ打ち込んでエラー出してみたり遊んでみることに
- 余裕ある人はPythonの教材もチェック

自分のPCを使いたい人へ

演習ではNotebookをダウンロードして作業することがあるためネットワークの設定が必要です。下記を読んで自分のPCで北大内の無線LANが使えるよう設定をしてください。

<https://www.hines.hokudai.ac.jp/wlan/>

USBメモリは必要なく、下記の作業のみです。

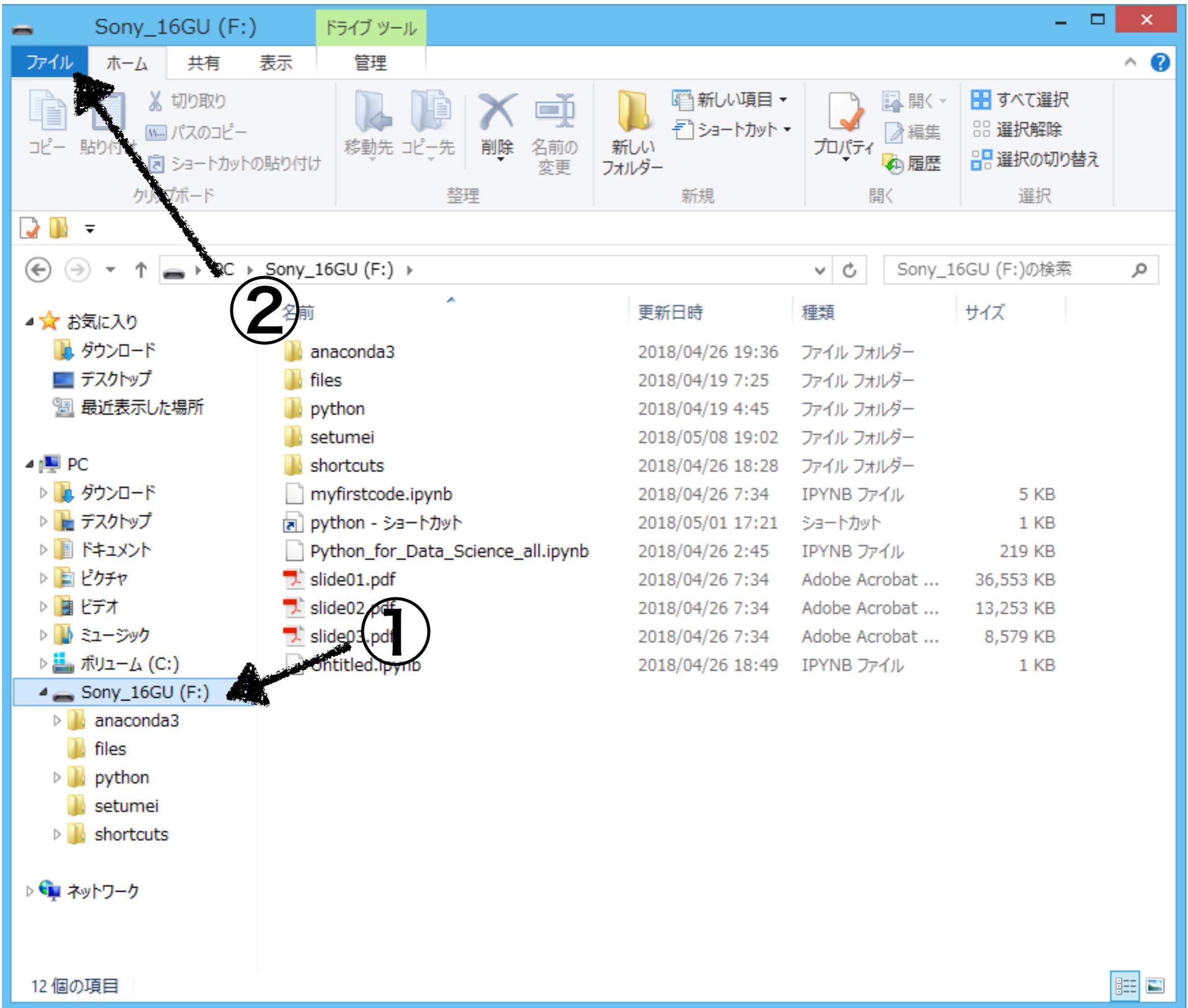
- 1) Anacondaをダウンロードしインストーラの指示通りに通常インストールします。
- 2) Jupyter Notebookの起動アイコンができるので普通にそこから起動して使うだけです。

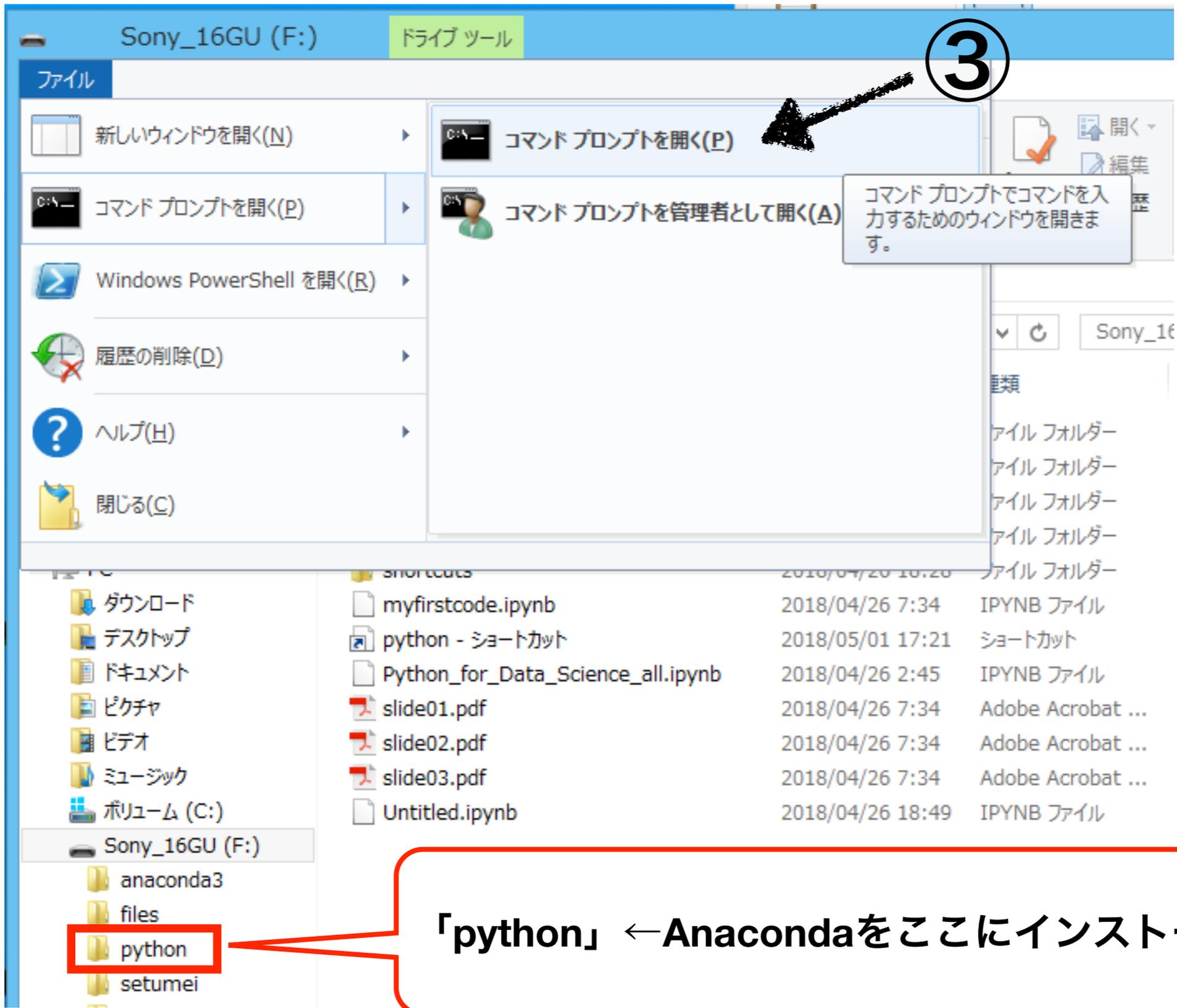
Macでも同じです。

USBメモリのPython環境を使う：

毎回演習開始時に以下を実行

1. USBのフォルダ「python」を選択
2. コマンドプロンプトを起動
3. 「python -m ipykernel install --user」を入力





「python」 ← Anacondaをここにインストールしたとする

```
C:\Windows\system32\cmd.exe

F:\>cd python
F:\python>python -m ipykernel install --user
Installed kernelspec python3 in C:\Temporary\AppData\Roaming\jupyter\kernels\python3
F:\python>
```

python -m ipykernel install --user

ここにどのフォルダを見ているかが表示されるので...

最初にAnacondaをインストールしたフォルダ(ここでは「python」という名前とする)を選択してから「コマンドプロンプト」を起動していたらここに「F:\python>」と出ているので、そのまま「python -m ipykernel install --user」と入力

「F:\>」など、別であれば「cd F:\python」として見ているフォルダをかえ表示が「F:\python>」となるか見る

ここまで毎回準備！

Jupyter Notebookの起動の仕方

先ほどの準備作業をした後

その1：一番直接的なやり方

1. USBメモリのAnacondaを入れたフォルダを開く
2. そのフォルダの中にある **「Scripts」** を開く
3. **「jupyter-notebook」** を起動

(ファイル名は「jupyter-notebook.exe」だが
たぶんexeは通常表示されない)

Jupyter Notebookの起動の仕方

その2：ショートカットを作っておき それから起動

<http://art.ist.hokudai.ac.jp/~takigawa/prog/>

講義ウェブサイトから「Jupyter_Notebook.lnk」をダウンロードし、USBに保存しておく

毎回講義開始時に先ほどの準備作業をした後、
このファイルをダブルクリックで起動

起動しない場合、色々修正する必要あり

Jupyter_Notebook.lnkのアイコンを右クリックし、プロパティを選択

ショートカットを選び

右のような画面が出たら

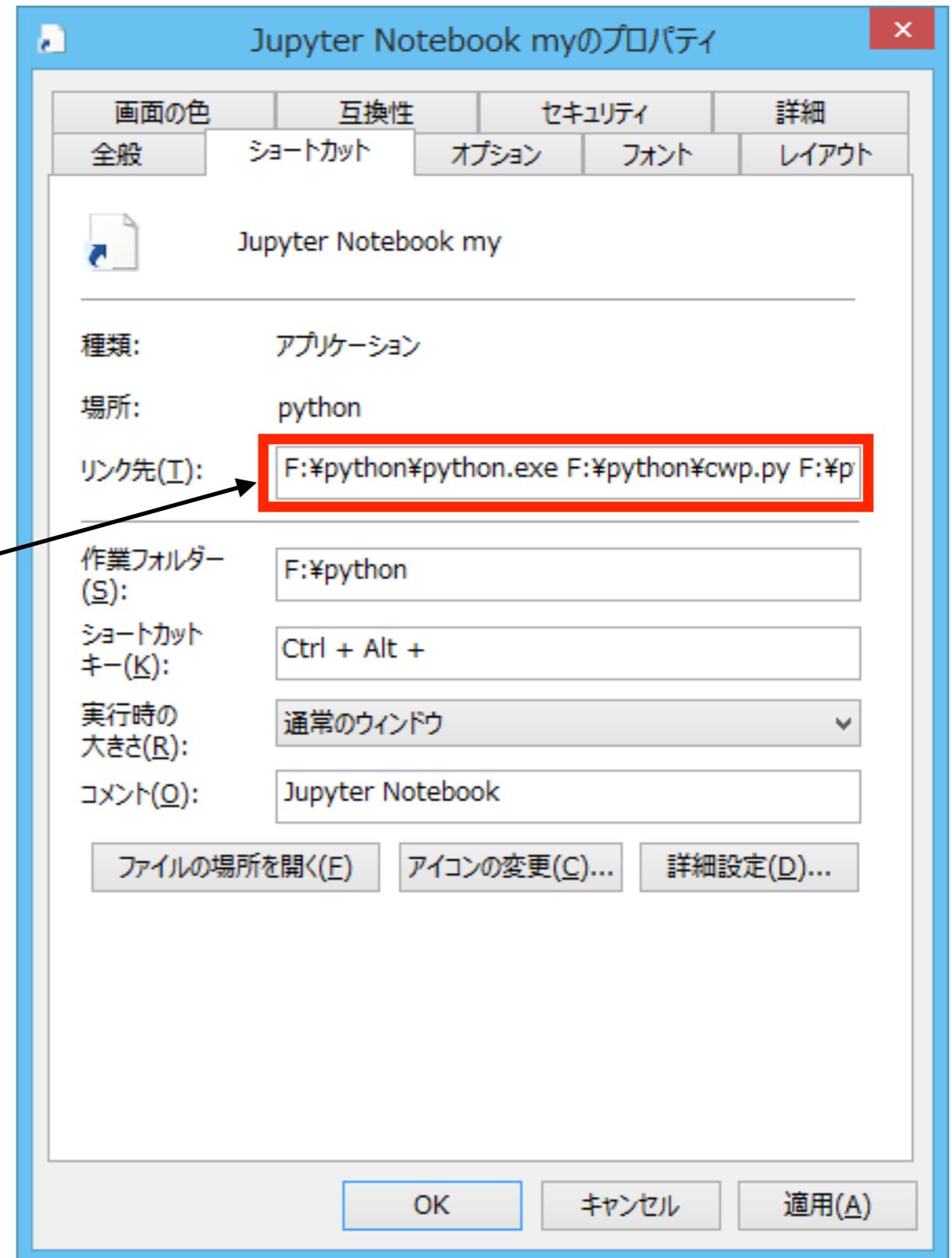
「リンク先」の部分の

「F:¥python」をすべて

自分の環境と同じに修正

「コマンドプロンプト」が

「F:¥」以外ならそれも





Logout

Files Running Clusters

Select items to perform actions on them.

Upload New ↕ ↻

<input type="checkbox"/> 0 ▾	📁 / Desktop / mypython	Name ↓	Last Modified
	📁 ..		数秒前
The notebook list is empty.			

Jupyter Notebookの終了の仕方

**起動時に開いた黒いウィンドウ
(コマンドプロンプト)を閉じる**

今日のお題：Pythonに慣れよう

- 前回の復習とふりかえり
- 演習室を使うための**毎回の準備作業**について
- Jupyter Notebookの動かし方
- Notebookをダウンロードして実行してみる
- Python言語の基本を学ぶ
 - print文、変数、数字の計算
- Markdownで文書を飾ってみよう
- ミニレポートタイム

先週のお題：Pythonを始めよう

- 前回の復習とふりかえり
- paiza.IOでPythonに触れてみる
- Python言語の基本を学ぶ + 最初はどうやって勉強すればいいか？
- Pythonインタプリタと実行環境について
- USBメモリのPython環境の使い方 (改)
- Jupyter Notebookの動かしかた
- (補足) なぜ前回動かなかったか？
- (補足) USBメモリについて
- ミニレポートタイム

大雑把なビジョン

- この演習でやりたいことは大きく2つ

1. 「プログラミング」とは何かを理解しそれを
どうやって学んでいけば良いかの指針を与える

2. 「Python」というプログラミング言語の基礎と
実習を実際にやってみる

まずはprintと電卓

1. 文字を出力

`print('出力したい文字')`

`print("Don't do it!")` ← 'があるときは"でくくる

2. 四則演算

足し算 `+` 引き算 `-` 掛け算 `*` 割り算 `/`

割った余り `%` べき乗 `**`

`round(1.2345, 3)` ← 小数点以下3桁に丸める

(四捨五入とちょっと違う)

print文をもう少し

```
print('答えは', 3, 'です')
```

```
print('答えは', 3, 'です', sep=":")
```

```
print('答えは', 3, 'です', sep=":", end=";")
```

```
print('答えは', 3, 'です', sep=":", end=";")
```

```
print()
```

```
print('答えは', 3, 'です', sep=" ", end="\n")
```

バックスラッシュは

日本語キーボードでは「¥」



変数と代入

```
a = 3
```

```
b = 4
```

```
hoge = 5
```

```
kotae = a * b + hoge
```

```
print('答えは', kotae, 'です', sep=" ")
```

バックスラッシュは 

日本語キーボードでは「¥」

演習環境：Jupyter Notebook

▼ 1 お絵かきしてみよう！

下のセルを選択して、CtrlキーとEnterキーを同時におしてください。

```
In [40]: import matplotlib.pyplot as plt
%matplotlib inline

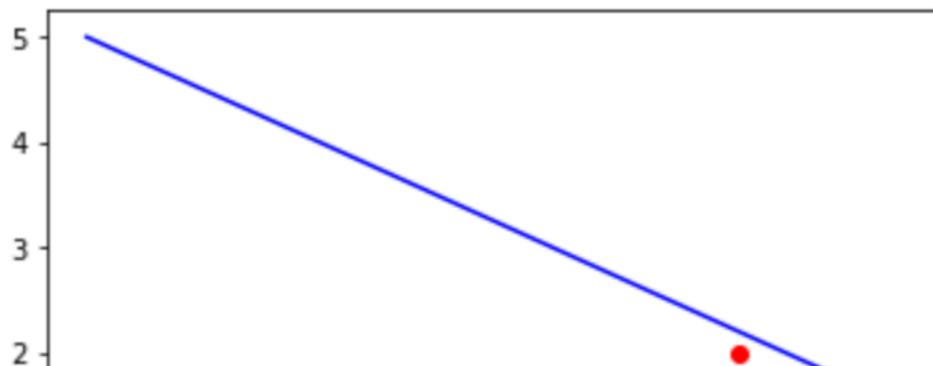
def line(p1, p2, c='blue'):
    plt.plot([p1[0], p2[0]], [p1[1], p2[1]], color=c)

def point(p, c='red'):
    plt.plot(p[0], p[1], 'o', color=c)
```

上はふたつの命令「line」と「point」を定義しました。これを使ってお絵かきしてみましょう。

下記のセルを自由に編集して命令を並べてください。実行はCtrlキーとEnterキーの同時押しか、上部のRunアイコンで！

```
In [45]: line([0, 0], [1, 1])
line([1, 1], [2, 1.5])
line([2, 1.5], [-3, 5])
point([0, 1])
point([1, 2])
```



<http://art.ist.hokudai.ac.jp/~takigawa/prog/>

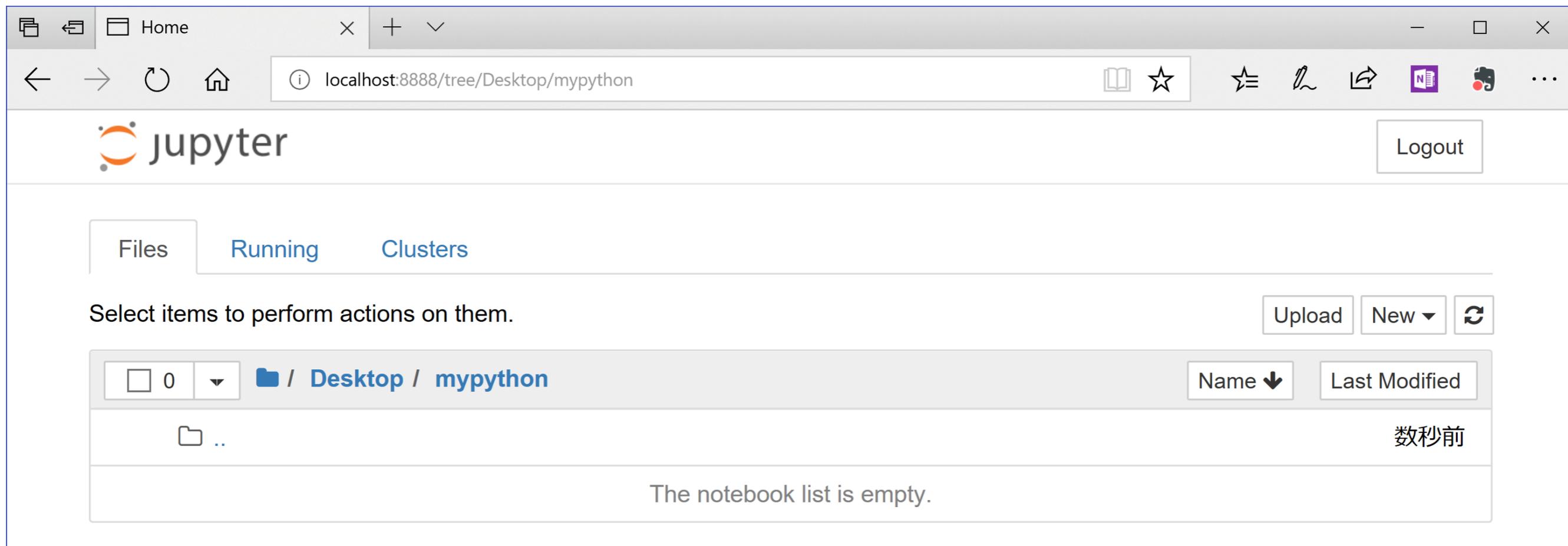
講義ウェブサイトから「Jupyter_Notebook.lnk」をダウンロードし、USBに保存しておく

毎回講義開始時に先ほどの作業をした後、USBからこのファイルをデスクトップにコピーしダブルクリックすれば「Jupyter Notebook」が起動する

ただし先ほどの表示が「F:」で始まっていること、USBにanacondaを入れたフォルダ名が「python」であること、は必要

演習環境：Jupyter Notebook

起動すると下記のような画面に



localhost:8888/tree/Desktop/mypython

jupyter Logout

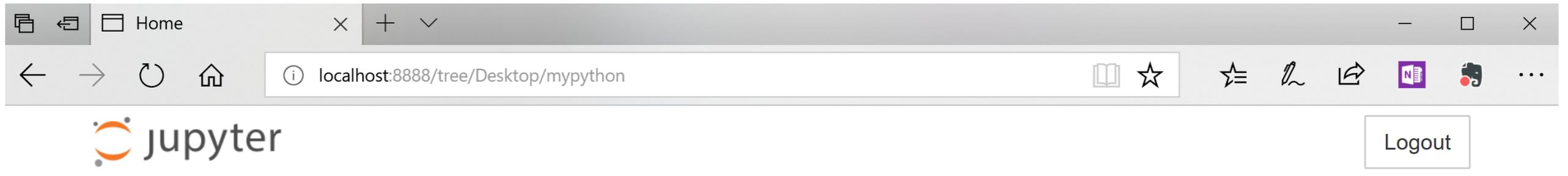
Files Running Clusters

Select items to perform actions on them. Upload New ↕ ↻

0 ▾ / Desktop / mypython Name ↓ Last Modified

数秒前

The notebook list is empty.



Files Running Clusters

Select items to perform actions on them.

Upload New

0 / Desktop / mypython

..

The notebook list is empty.

新規作成

Notebook:

Python 3

Other:

Text File

Folder

Terminal

フォルダなども作成できる

localhost:8888/notebooks/Desktop/mypython/Untitled.ipynb?kernel_name=python3

jupyter Untitled (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In []: |

ここにPythonコードを打ち込む
「Run」をクリックするか、ShiftとEnterの同時押しで実行
「緑の枠」が表示されているとき：コマンドモード

localhost:8888/notebooks/Desktop/mypython/Untitled.ipynb?kernel_name=python3

jupyter Untitled (unsaved changes) Python 3

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In []: |

「緑の枠」が表示されているとき：編集モード
ここにPythonコードを打ち込む
「Run」をクリックするか、
ShiftとEnterの同時押しで実行

クリック or Enter 欄外クリック or Esc

In []:

「緑の枠」が表示されているとき：コマンドモード

The image shows a Jupyter Notebook interface with several annotations in Japanese. The interface includes a browser window at the top, a Jupyter logo and title 'Untitled (unsaved changes)', a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help', and a toolbar with icons for file operations and execution. A code cell is visible with the prompt 'In []: |'. The annotations are as follows:

- ヘルプ** (Help) - points to the 'Help' menu item.
- ファイル名 (クリックして変更も可能)** (Filename (clickable to change)) - points to the 'Untitled' title.
- セルを実行** (Execute cell) - points to the 'Run' button in the toolbar, with the shortcut **Shift+Enter** below it.
- セルを上移動** (Move cell up) - points to the up arrow button in the toolbar, with the shortcut **K** to its right.
- セルを下移動** (Move cell down) - points to the down arrow button in the toolbar, with the shortcut **J** to its right.
- ショートカット一覧を表示 (or Help→Keyboard Shortcuts)** (Show keyboard shortcuts) - points to the keyboard icon in the toolbar, with the shortcut **H** to its left.
- セルを貼り付け** (Paste cell) - points to the paste icon in the toolbar.
- セルをコピー** (Copy cell) - points to the copy icon in the toolbar, with the shortcut **C** below it.
- セルを消す** (Delete cell) - points to the delete icon in the toolbar, with the shortcut **DD** below it.
- セルをたす** (Select cell) - points to the select icon in the toolbar, with the shortcut **A** (up) and **B** (down) below it.
- キーボードショートカット (ただし「コマンドモード」)** (Keyboard shortcuts (but in 'command mode')) - a general note in purple text.

変数の情報を表示

```
In [4]: z = 19
```

```
In [5]: z?
```

「?」を最後につける

```
Type:          int
String form: 19
Docstring:
int(x=0) -> integer
int(x, base=10) -> integer
```

```
In [4]: z = 19
```

```
In [5]: z
```

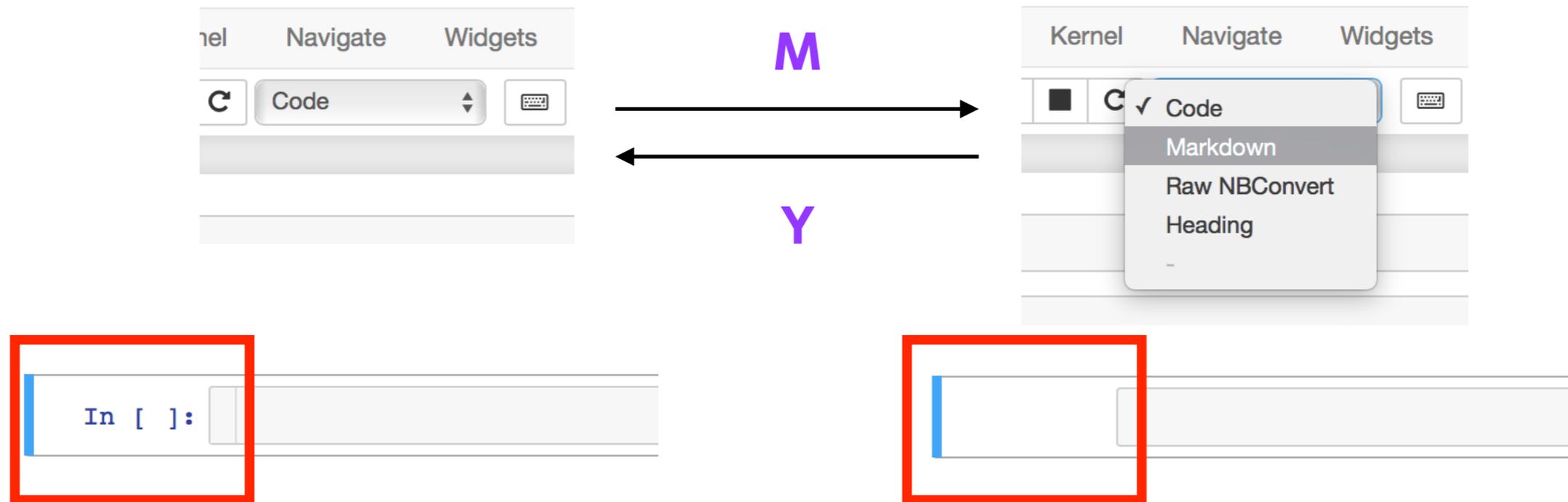
Shift+Tab

```
Type:          int
String form: 19
Docstring:
int(x=0) -> integer
```

^ + x

Markdown

注：これはpythonとは直接関係ありませんが...

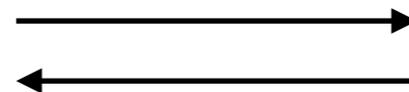


このように

色々書いてみると

- どういう感じに
- なるのでしょうか

Enter



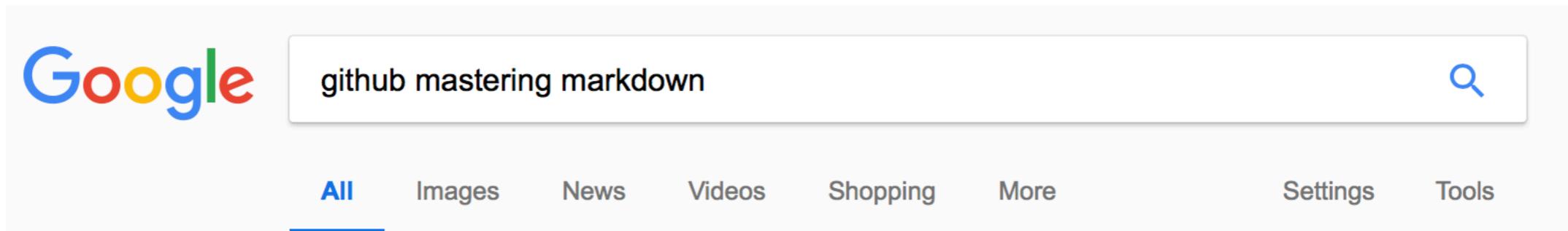
Shift+Enter

このように

色々**書いて**みると

- どういう感じに
- なるのでしょうか

Markdownのルール



About 29,500 results (0.54 seconds)

Mastering Markdown · GitHub Guides

<https://guides.github.com/features/mastering-markdown/> ▼

Jan 15, 2014 - **Mastering Markdown**. Markdown is a lightweight and easy-to-use syntax for styling all forms of writing on the **GitHub** platform.

GitHub Guides

Video Guides



Markdown記法

これはヘディング

これはサブヘディング

普通の文はこのように書いておく。
強調したければ**こう**という風にかく。

箇条書きは

- その1
- その2

などとハイフンを使って書いておく

Pythonコードもこのようにかける。

```
```python
print('Hello!')
```
```

例えば、表もこのようにしてかける。

| 札幌 | 旭川 |
|-------|-------|
| このような | 感じですか |
| 例えば | こう |

(ちなみに)LaTeXという数式表記法も使えるようになっている(理工系の人向け)。

```
$$ \sum_{i=1}^n \frac{1}{n} $$
```

フォーマット後

これはヘディング

これはサブヘディング

普通の文はこのように書いておく。 強調したければ**こう**いう風にかく。

箇条書きは

- その1
- その2 などとハイフンを使って書いておく

Pythonコードもこのようにかける。

```
print('Hello!')
```

例えば、表もこのようにしてかける。

| 札幌 | 旭川 |
|-------|-------|
| このような | 感じですか |
| 例えば | こう |

(ちなみに)LaTeXという数式表記法も使えるようになっている(理工系の人向け)。

$$\sum_{i=1}^n \frac{1}{n}$$

コードセルの説明などをMarkdownセルとして入れておくと実行もでき結果も含み説明もあるインタラクティブ文書となる！

作業：授業のホームページから「[myfirstcode.ipynb](#)」や「[ex1.ipynb](#)」「[ex2.ipynb](#)」をダウンロードして実行してみよう。

<http://art.ist.hokudai.ac.jp/~takigawa/prog/>

Pythonが起動しているフォルダにコピーし、クリックして開く！

 jupyter

Logout

Files

Running

Clusters

Nbextensions

Select items to perform actions on them.

Upload

New ▾

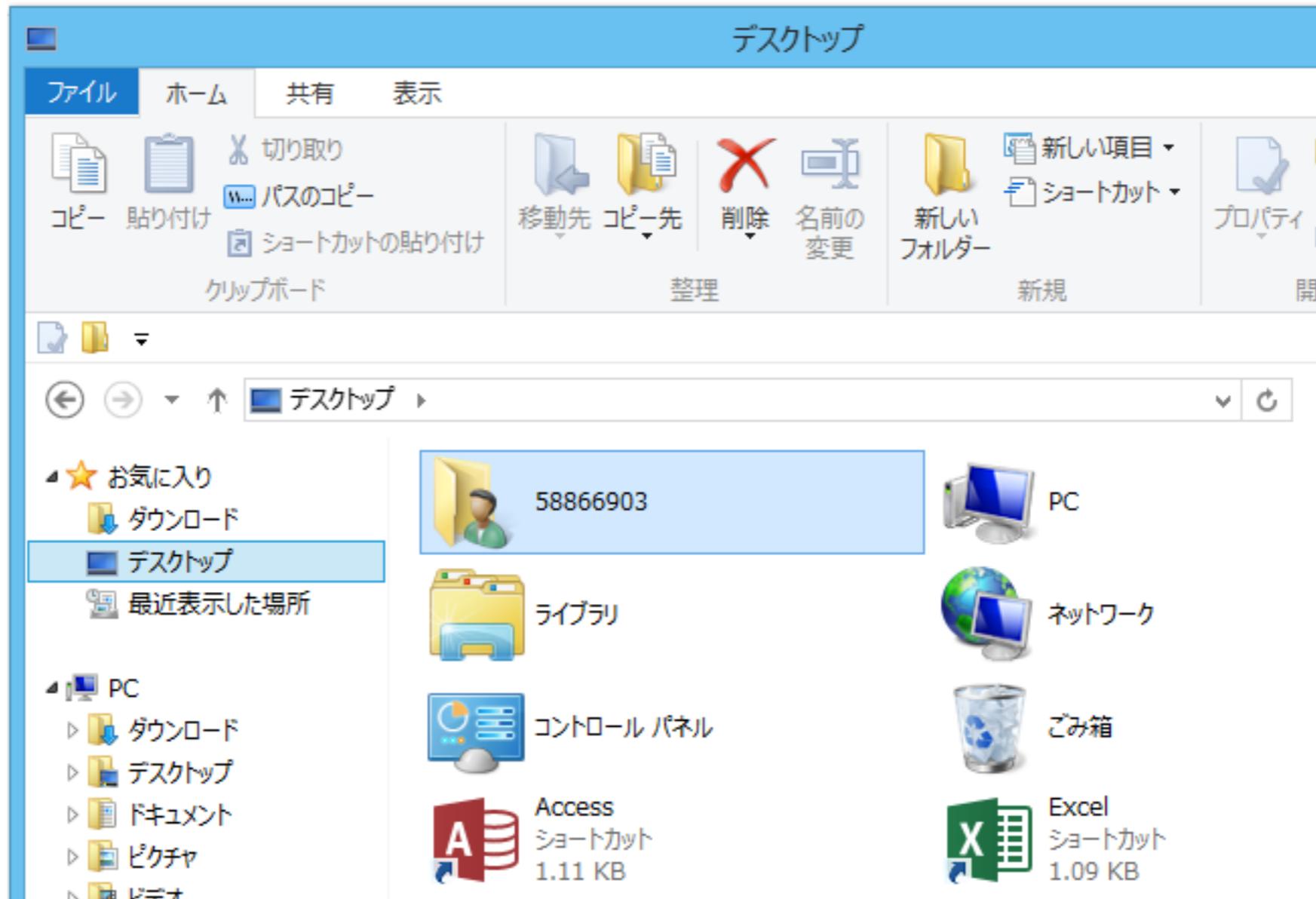


| <input type="checkbox"/> 0 ▾ | | Name ▾ | Last Modified |
|------------------------------|---------------------|--------|---------------------|
| <input type="checkbox"/> | 📁 files | | 7 days ago |
| <input type="checkbox"/> | 📁 python | | 7 days ago |
| <input type="checkbox"/> | 📄 myfirstcode.ipynb | | Running seconds ago |

「Pythonが起動しているフォルダ」

Anacondaを入れたフォルダのScriptsのjupyter-notebook
を実行して起動した場合 → そのScriptsフォルダ

「Jupyter_Notebook.lnk」を設定し起動した場合



「Pythonが起動しているフォルダ」

ファイルブラウザの画面でホーム()をクリックしたところに作ったPython3ファイル(.ipynb)で

Windowsの場合

コードセルで「!cd」を実行して表示されるフォルダ

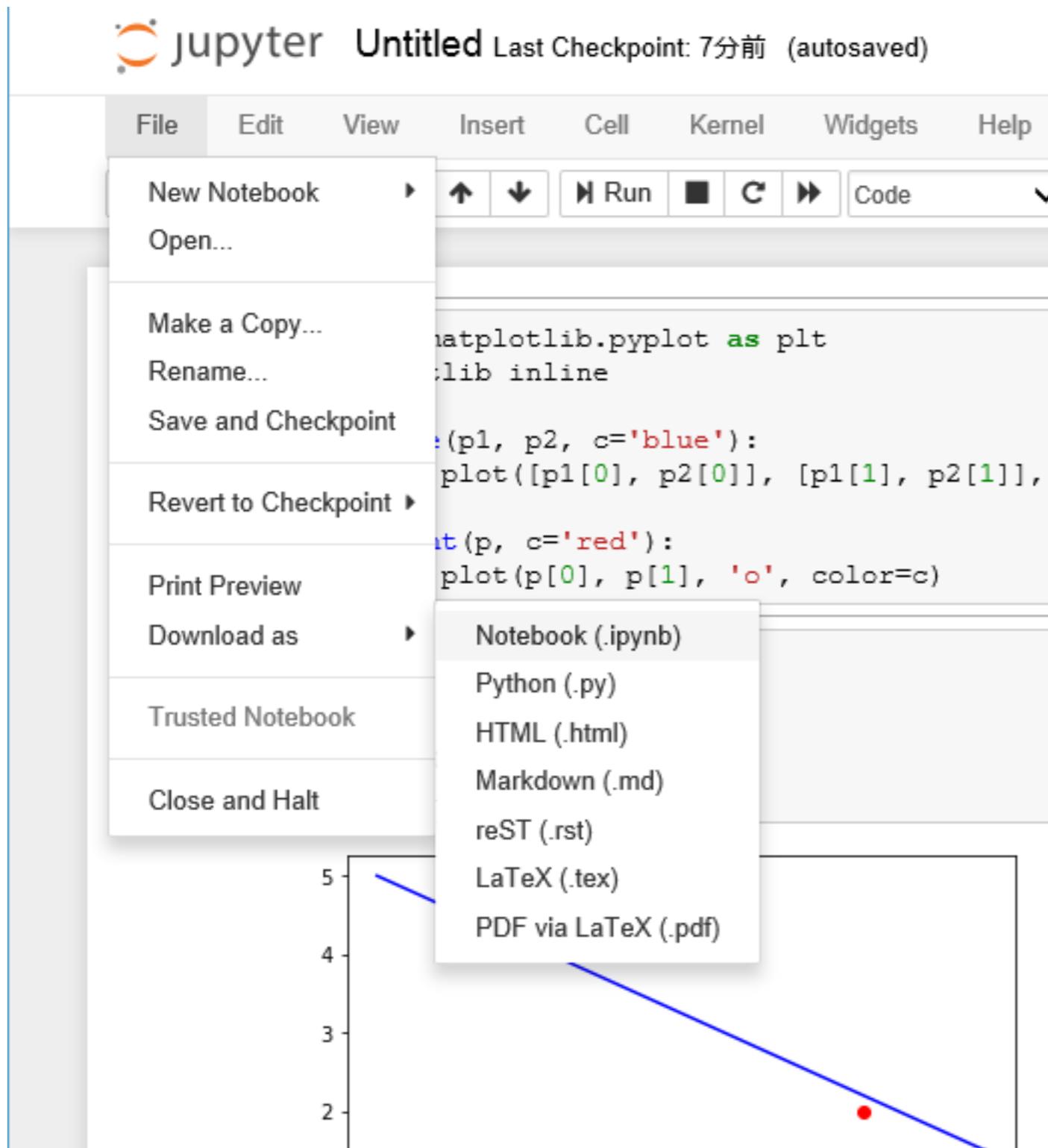
F:¥python¥subfolder ←これはF:ドライブのpythonフォルダの中のsubfolderフォルダ

Mac/Linuxの場合

コードセルで「!pwd」を実行して表示されるフォルダ

注意：「¥」記号は英語では「\」に対応するので同じことを指す。適宜読み替えてください

実行できたらそのJupyter Notebookを保存してみよう！



The screenshot shows the Jupyter Notebook interface. At the top, the title bar reads "jupyter Untitled Last Checkpoint: 7分前 (autosaved)". Below the title bar is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The "File" menu is open, displaying a list of actions: New Notebook, Open..., Make a Copy..., Rename..., Save and Checkpoint, Revert to Checkpoint, Print Preview, Download as, Trusted Notebook, and Close and Halt. The "Download as" sub-menu is also open, showing options: Notebook (.ipynb), Python (.py), HTML (.html), Markdown (.md), reST (.rst), LaTeX (.tex), and PDF via LaTeX (.pdf). In the background, a code cell contains Python code for plotting:

```
import matplotlib.pyplot as plt
plt.rcParams['figure.figstyle'] = 'inline'

def plot(p1, p2, c='blue'):
    plt.plot([p1[0], p2[0]], [p1[1], p2[1]],
             color=c)

def plot(p, c='red'):
    plt.plot(p[0], p[1], 'o', color=c)
```

Below the code cell, a plot is displayed. The y-axis is labeled from 2 to 5. A blue line connects the points (0, 5) and (5, 2). A red dot is plotted at the coordinates (4, 2).

Jupyter Notebookが起動し動いて入れば、この環境でかなり高度なこともできます。またネットに転がっているNotebookを実行して見ることもできます!

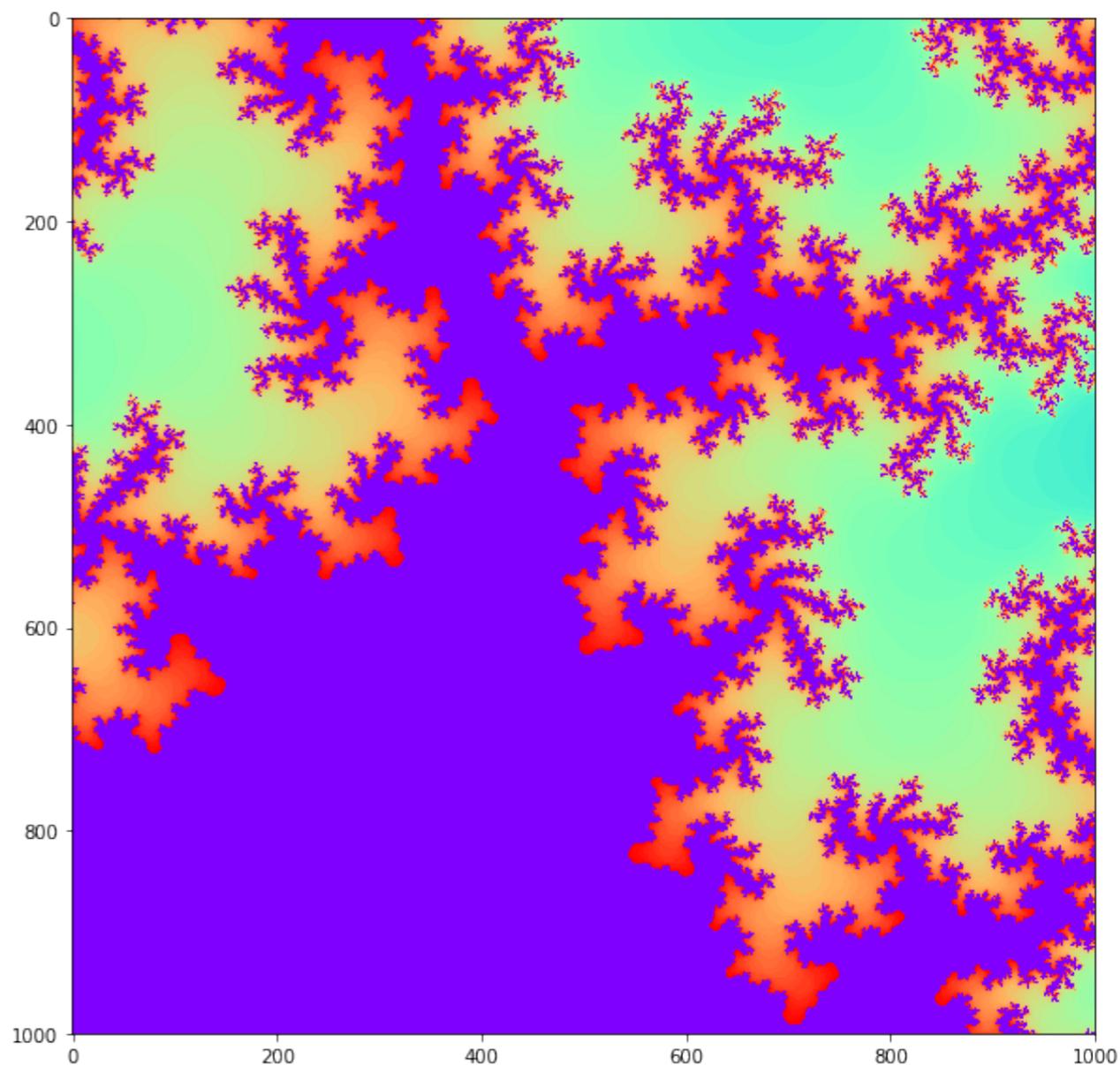
```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline

mpl.rcParams['figure.figsize'] = 10, 10
```

```
def mandelbrot(c, maxiter):
    z = c
    for n in range(maxiter):
        if abs(z) > 2:
            return n
        z = z*z + c
    return 0

def mandelbrot_set(xmin, xmax, ymin, ymax, w, h, maxiter):
    a = np.linspace(xmin, xmax, w)
    b = np.linspace(ymin, ymax, h)
    s = np.empty((w, h))
    for i in range(w):
        for j in range(h):
            s[i, j] = mandelbrot(a[i] + 1j*b[j], maxiter)
    return s
```

```
m = mandelbrot_set(-0.56, -0.55, -0.56, -0.55, 1000, 1000, 80)
plt.imshow(m, cmap='rainbow')
```



Pythonの学習について

- Python公式チュートリアル

<https://docs.python.jp/3/tutorial/>

- LearnPython.org interactive Python tutorial (英語)

<https://www.learnpython.org>

- CodeAcademy (英語)

<https://www.codecademy.com/learn/learn-python>

- Intro to Python for Data Science (英語)

<https://www.datacamp.com/courses/intro-to-python-for-data-science/>

- Python tutorials for beginners (英語)

<http://thepythonguru.com>

- Progate (プロゲート)

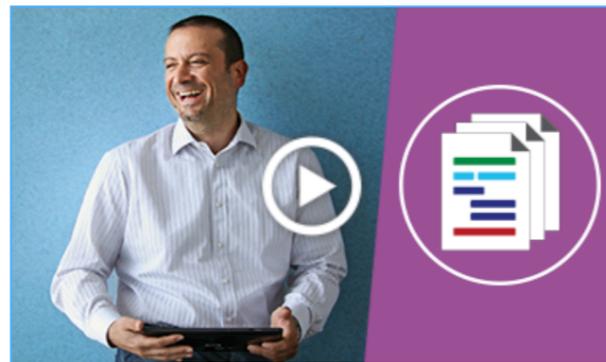
<http://prog-8.com/languages/python>

- ドットインストール

https://dotinstall.com/lessons/basic_python_v3

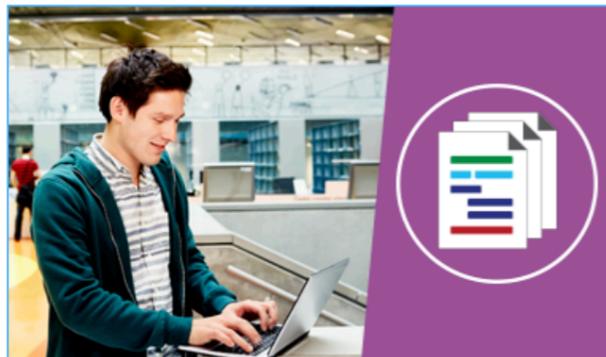
オンラインで無料の良い講義も多数！

<https://wiki.python.org/moin/BeginnersGuide/NonProgrammers>



Introduction to Python: Absolute Beginner

In this course that's perfect for true beginners, learn Python basics and start coding right away.



Introduction to Python: Fundamentals

Build on what you learned in the "Introduction to Python: Absolute Beginner" course, and dig into data structure basics.



FREE COURSE

Introduction to Python

Starting Out in Python 3

START FREE COURSE

- LearnPython.org interactive Python tutorial (英語)

Indentation

Python uses indentation for blocks, instead of curly braces. Both tabs and spaces are supported, but the standard indentation requires standard Python code to use four spaces. For example:

```
script.py  IPython Shell
1 x = 1
2 if x == 1:
3     # indented four spaces
4     print("x is 1.")
```

Run

Powered by DataCamp 

Exercise

Use the "print" command to print the line "Hello, World!".

```
script.py  IPython Shell
1 print("Goodbye, World!")
```

• CodeAcademy (英語)

The screenshot shows the Codecademy interface for a Python lesson. On the left, there is a sidebar with the Codecademy logo, a 'Learn' button, and a list of navigation options: 'Instructions', 'Community Forums', and 'Report a Bug'. The main content area is titled 'PYTHON SYNTAX' and 'Hello World!'. It contains a paragraph explaining that programming is teaching a computer to have a conversation with a user, and that this is done using the 'print' statement. Below the text is a code block with the following Python code:

```
print "Hello, world!"  
print "Water--there is not a drop  
of water there! Were Niagara but a  
cataract of sand, would you travel  
your thousand miles to see it?"
```

Below the code block, it says 'A print statement is the easiest way to get your Python program to communicate'. To the right of the text is a code editor window titled 'script.py' containing the code: `1 print "hoge"`. The output of the code is shown in a terminal window on the right, displaying the text 'hoge'. At the bottom of the code editor is a 'Run' button and a refresh icon. The bottom navigation bar includes a hamburger menu, '1. Hello World!', a 'Back' button, '1/14', a 'Next' button, and a 'Get Help' button.

- Intro to Python for Data Science (英語)

The screenshot shows a video player interface for a DataCamp course. At the top left is the DataCamp logo. In the center top, there is a 'Course Outline' button with a right-pointing arrow. On the top right, there are icons for notifications, a document, and a warning. Below the navigation bar, the video title 'Hello Python!' is displayed in large, bold, dark blue font. To the right of the title, a badge indicates '50XP'. The video content area features a blue header with a Python logo icon. Below this, a large white shield-shaped graphic contains the text 'INTRO TO PYTHON FOR DATA SCIENCE' and 'Hello Python!' in bold black font. A presenter, Filip Schouwenaars, is shown in a black polo shirt with the DataCamp logo on the chest. At the bottom left of the video frame, the name 'Filip Schouwenaars' and a timestamp '0:04' are visible. The video player controls at the bottom include a play button, a volume icon, a progress bar, a time display of '-3:32', a download icon, a '1x' speed control, a CC icon, and a full-screen icon. A yellow 'Got it!' button is located at the bottom right of the player.

● Progate (プロゲート)

<http://prog-8.com/languages/python>



文字列

文字列とは？

文字列

クォーテーション

先ほどの例で用いた「Hello Python」という文字は、プログラミングの世界では「文字列」と呼ばれます。

文字列はシングルクォーテーション「'」またはダブルクォーテーション「"」で囲む必要があります。どちらで囲んでも出力結果は同じとなります。どちらかで囲んでいない場合、コードは動かなくなります。

シングルクォーテーション(')
または、ダブルクォーテーション(")で囲む

```
script.py x  
print('Hello Python')  
print("Hello Python")
```

```
script.py x  
# エラー発生  
print(Hello Python)
```

出力結果

>_ コンソール

```
Hello Python  
Hello Python
```

出力結果は同じ

>_ コンソール

```
SyntaxError: invalid syntax  
~~~~~  
X エラー!!!: 必ず文字列はクォーテーションで囲む!
```





Pythonの基礎を学...

3/17

script.py



コンソール



数値

数値を出力してみましょう。

script.py

数値の7を出力してください。

スライドで確認

script.py

「+」を用いて、9に3を足した値を出力してください。

スライドで確認

```
1 # 数値の7を出力してください
2
3
4 # 9に3足した値を出力してください
5
6
7 # 「9 + 3」を文字列として出力してください
8
```

リセット

答えを見る

確認して次へ

見本



スライドを見る



11%

ご意見箱

script.py



```
1 # 数値の7を出力してください
2 print(7)
3
4 # 9に3足した値を出力してください
5 print(9 + 3)
6
7 # 「9 + 3」を文字列として出力してください
8 print('9 + 3')
```

>_ コンソール

```
7
12
9 + 3
□
```

>_ 見本

🏆 PYTHON Lv.3  (+ 100 EXP) ✕

Congratulations!

正解です！次のレッスンは
「計算してみよう」です！

解答を見る

次に進む

● paizaラーニング

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

paizaラーニング™

トップ

有料会員プラン

タグから検索

スキルチェック

paizaラーニングトップ > Python > Python3入門編レッスン一覧

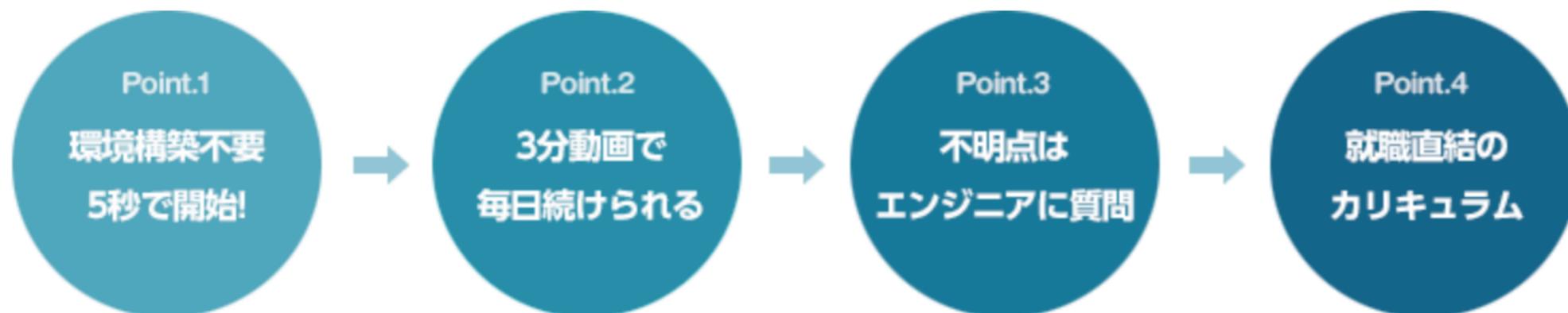


Python3入門編 (全6レッスン)

paizaラーニングの学び方

完全無料 Python

ツイート



paizaラーニングは、**プログラミング初心者～中級者向けのスキルアップ学習コンテンツ**です。

1チャプター3分動画なので、忙しくても毎日続けられます。

不明点は **エンジニアに質問が出来る**ので、つまづく事はありません。



本レッスンで身につけられるスキル

Pythonでプログラミングの初歩を学ぶレッスンです。(Python3を対象として学んでいきます)

標準出力や変数、配列、演算子、条件分岐、ループといったプログラミングの基本を手順を追って学びます。

D E

対象者



今日のお題：Pythonに慣れよう

- 前回の復習とふりかえり
- 演習室を使うための**毎回の準備作業**について
- Jupyter Notebookの動かし方
- Notebookをダウンロードして実行してみる
- Python言語の基本を学ぶ
 - print文、変数、数字の計算
- Markdownで文書を飾ってみよう
- ミニレポートタイム