

---

# Jupyter Notebook環境構築～HelloWorld

---

- 写経元
  - <https://qiita.com/kshigeru/items/2cd504e927869163b4c8>

## 環境構築編

---

- docker pull

- ```
$ docker pull jupyter/datascience-notebook
```

- 8888 ポートでjupyter を起動

- ```
$ docker run -d --name notebook -p 8888:8888 jupyter/datascience-notebook
```

- 動いた



■

### Token authentication is enabled

If no password has been configured, you need to open the notebook server with its login token in the URL, or paste it above. This requirement will be lifted if you [enable a password](#).

The command:

```
jupyter notebook list
```

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

```
Currently running servers:
http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks
```

or you can paste just the token value into the password field on this page.

See [the documentation on how to enable a password](#) in place of token authentication, if you would like to avoid dealing with random tokens.

Cookies are required for authenticated access to notebooks.

## ■ コンテナに入って

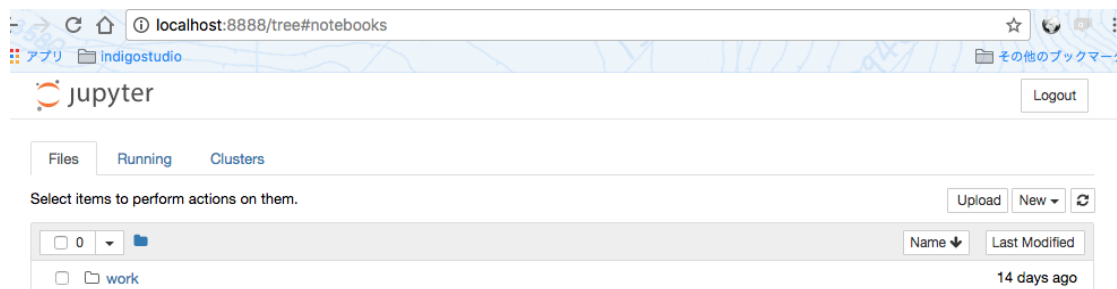
```
$ docker exec -it notebook bash
```

## ■ tokenを確認する

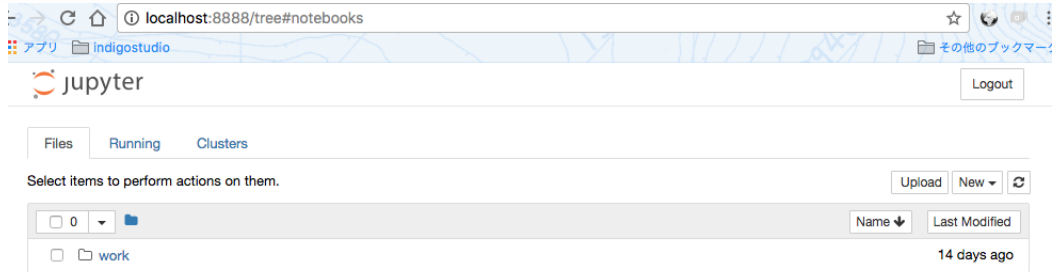
```
$ jupyter notebook list
Currently running servers:
http://localhost:8888/?
token=cfc996979890b0cf3cd80b4a34ff3f20a92754849065e8a3 ::
/home/jovyan
```

# Hello World編

## ● tokenを入力して Login する



## ○ New で Python3 を選択する



- sin/cosカーブを書く

- Jupyter NotebookのGUIと操作方法

- In []ごとにコードブロックと実行結果がある

- 

- コードブロックにコードを書いて Shift + Enter を押すと実行される

- ```
import math
```

```
import bokeh.plotting as bplt
import numpy as np
```

- ```
bplt.output_notebook()
```

- ```
cycle = math.pi * 2
x = np.linspace(-1 * cycle, cycle, 100)
y1 = np.sin(x)
y2 = np.cos(x)
```

- `numpy.linspace`

- <https://deepage.net/features/numpy-linspace.html>

- NumPyの `np.linspace` は、線形に等間隔な数列を生成する関数です。同様の数列を `np.arange` で生成することもできますが、`np.linspace` を使用したほうがコード量を減らすことができ、読みやすくスマートになります

- $-\pi$  から  $+\pi$  までを 100等間隔 にした場合の数列を生成

- ```
p = bplt.figure(title='sin/cos curve', plot_width=640,
plot_height=320)

p.circle(x, y1, color='red')
p.triangle(x, y2, color='blue')
bplt.show(p)
```

■