

Jupyter notebook 入門

このノートブックは、Jupyterのいくつかのコア機能の概要を説明しています。

ノートブックでPythonコードを実行する方法に慣れていない場合は、最初にこのノートブックを読む必要があります。

ノートブックの使用に既に慣れている場合は、このコンテンツをスキップするか、参照として保持できます。

このノートでは：

- コードセルとマークダウンセル
- 実行中のコード
- インターフェースの概要
- いくつかの便利なショートカット
- マジックコマンド

コードセルとマークダウンセル

ノートブックは一連のセルで構成されています。細胞にはさまざまな種類があります。

コードセルには実行するコードが含まれ、マークダウンセルにはマークダウン構文を使用して記述された説明（ドキュメント）が含まれます。

以下はコードセルです。

- セルの中央をクリックしてアクティブにします
- ツールバーの「実行(Run)」アイコンをクリックしてコードを実行します

In []:

```
# コードセル

# 「#」で始まる行はコメントです。
# 実行されない行です。コードの説明

a = 500
b = 202

print(a + b)
```

代わりに、これは**マークダウン(Markdown)** セルです。

テキストの中央をダブルクリックすると、実際のマークダウンコードが表示されます。

マークダウンコードを見栄えの良いドキュメントにレンダリングするには、セルを実行する必要があります。

マークダウンを使用すると、次のようなものを含むドキュメントをフォーマットできます

大きなヘッダー

小さいヘッダー

はるかに小さいヘッダー

箇条書き：

- One
- Two
- Three

番号付きリスト：

1. One
2. Two
3. Three

italic、**bold**、 monospace のテキスト。

セルをコードまたはマークダウンとして定義する

ツールバーのドロップダウンメニューを使用してタイプを変更できます。

アクティブセルに応じて、現在のセルタイプがわかります（たとえば、この特定のセルのマークダウン、下のセルのコード）

In []:

```
# コードセル  
print("テキスト Jupyter Notebook です。")
```

コードを実行する

コードセルを実行するには、いくつかのオプションがあります。

- ツールバーの「実行」アイコンをクリックします
- [セル]メニュー、[セルを実行]（または他の[実行...]オプションのいずれか）をクリックします。
- Shift + Enterを使用します
- alt + enterを使用します（これによりアクティブセルが実行され、すぐ下に空のセルが追加されます）

通知

実行の順序は重要です。ノートブックは上から下に実行する必要があります。

ノートブック環境はステートフル(全てのセル状態を記憶)です。セルを実行した後、その状態は次のセルで使用できます（下の例を参照）

In []:

```
var = "Python Jupyter Notebook"  
  
print("この {} ツールを勉強している.".format(var))
```

In []:

```
# 上記のセルを実行した場合、変数varは引き続き使用できます  
print(var)
```

コードセルを実行すると、最後の式の出力がデフォルトで出力され、`print()` 明示的にする必要はありません。 :

In []:

```
var
```

In []:

```
a = 500  
b = 400
```

```
a + b
```

インターフェースの概要

各コードセルの横に、実行順序のプレースホルダーがあります。これにより、実行されたセルと順序がわかります。

例

次のセルを実行し、クリックして再度実行します。

増加する数を観察します。

In []:

```
print("これを複数実行する")
```

主な機能

[編集]メニューを使用すると、コピー/貼り付けやマージなどのセルの操作を実行できます。

「表示」メニューでは、ヘッダー、ツールバー、および行番号を視覚化または非表示にできます。

「挿入」メニューでは、新しい空のセルを挿入できます。

「セル」メニューでは、さまざまな方法でセルを実行し、セルタイプを変更して、出力をクリアできます。

「カーネル」メニューでは、カーネル（ノートブックの背後にあるPythonインタプリタ）と対話できます。

ヒント

コードセルが動かなくなった場合（無限ループなど）、カーネルを**中断**または**再起動**する必要がある場合があります。

いくつかの便利なショートカット

キーボードショートカットを使用すると、生産性や快適性が向上する場合は、十分な数があります。

ショートカットを使用できるようにするには、[Esc]キーを押して「コマンドモード」に切り替える必要があります。

Esc を押すと、アクティブなコードセルは（緑の境界線ではなく）青い境界線を表示し、キーボードコマンドを受け入れることができます。

「編集モード」に戻るには、「Enter」を押すか、セルの中央をクリックします（セルの境界線が緑色に変わります）。

- Esc+M : セルタイプをMarkdownに変更する
- Esc+Y : セルタイプをコードに変更する
- Esc + A : 上に新しい空のセルを追加
- Esc + B : 新しい空のセルを下に追加
- Esc + DD (2つのD) : 現在のセルを削除
- Esc + X : 選択したセルを切り取ります
- Esc + C : 選択したセルをコピーします
- Esc + V : 選択したセルを貼り付け

さらに多くのコマンドとショートカットが利用可能で、以下を使用してリストを視覚化できます。

- Esc + P : コマンドパレットを表示

Jupyter Magics

マジック（またはマジックコマンド）は、Jupyterノートブックを強化する便利なヘルパー機能です。

コードセルは魔法のコマンドを実行できます。

マジックには2つのタイプがあります。ラインマジック（単一の % で始まる）とセルマジック（二重の %% で始まる）です。

利用可能なものの完全なリストを参照してください：

In []:

```
%lsmagic
```

実際に役立つマジック

test.pyを作成してから、

```
#test.py  
print("Hello World!")
```

以下のセルを実行して、外部ファイルからコードをロードします

In []:

```
%load ./test.py
```

以下のセルを実行して、test.pyファイルからコードを実行します

In []:

```
%run ./test.py
```

以下の2つのセルを実行して、コードの時間を計ります。

注：これはセルマジックの例です。

In []:

```
%%time  
  
slow_sum = 0  
  
for i in range(1000000):  
    slow_sum = slow_sum + i  
  
slow_sum
```

In []:

```
%%time  
  
faster_sum = sum(range(1000000))  
  
faster_sum
```

In []: