

1 VIII. Python 入門

2

3 目的

4 プログラミングについて学習する

5 プログラミング言語 Python の初歩を学ぶ

6 Jupyter Notebook の使用法を学ぶ

7

8 1. Python 概要

9 (1) 高水準 (高級) プログラミング言語

10 (2) 開発者はガイド・ヴァンロッサム

11 (3) 文法を極力単純化してコードの可読性が高く書きやすい

12 (4) 標準ライブラリやサードパーティ製のライブラリ、関数など、さまざまな領域に特
13 化した豊富で大規模なツール群

14 (5) オブジェクト指向、命令型、手続き型、関数型などの形式で書ける

15 (6) Web アプリケーションやデスクトップアプリケーションなどの開発はもとより、シ
16 ステム用の記述 (script) や、各種の自動処理、理工学や統計・解析などに応用

17 (7) Python のリファレンス実装である CPython は、フリーかつオープンソースのソフ
18 トウェア

19 (8) 現在, Python 2.7 と 3.7 の 2つの系統

20

21 2. Python の特徴

22 (1) インタープリタで動作することを前提

23 (2) 動的型付け (変数などの型を実行時に決定)

24 (3) ガーベージコレクション (不要になったメモリを自動的に解放)

25 (4) モジュールやクラスなどプログラムの要素にプログラム内部からアクセス可能で,
26 リフレクション (プログラムがプログラムを書き換えること) ができる

27 以上, Wikipedia “Python” より

28

29 3. Anaconda

30 (1) Python のディストリビューションの 1つ

31 (2) NumPy や matplotlib など外部ライブラリをあらかじめ用意

32 (3) プログラム編集ツール “Jupyter Notebook” を持つ

33

34 …と色々書いてありますが, 知らないことが沢山あります。未知の世界に向かって旅立
35 ちましょう

36

37 4. Anaconda のインストールとテスト

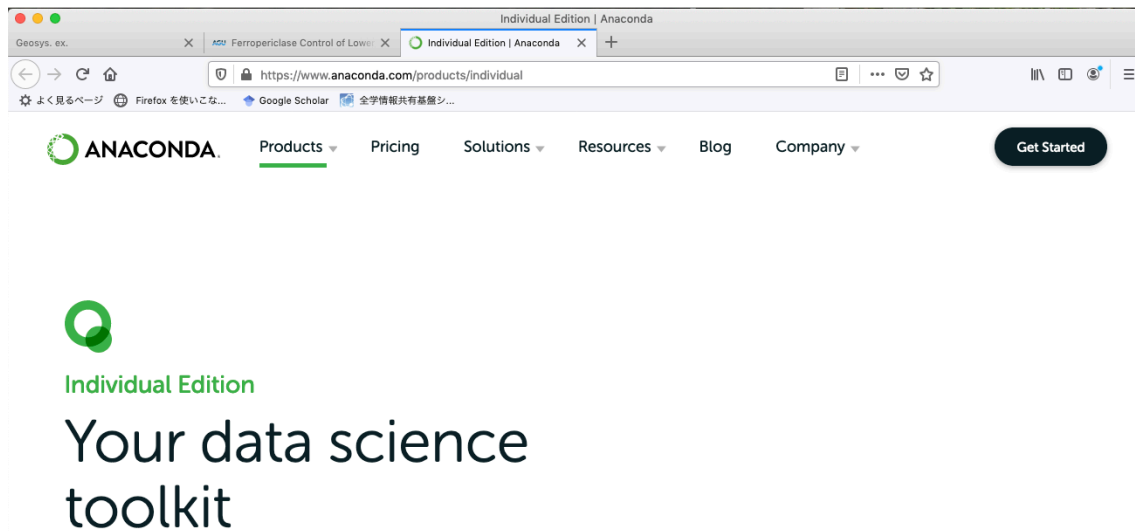
38

39 4.1 インストール

40 (1) 次の URL にアクセスする。

41 <https://www.anaconda.com/products/individual>

42



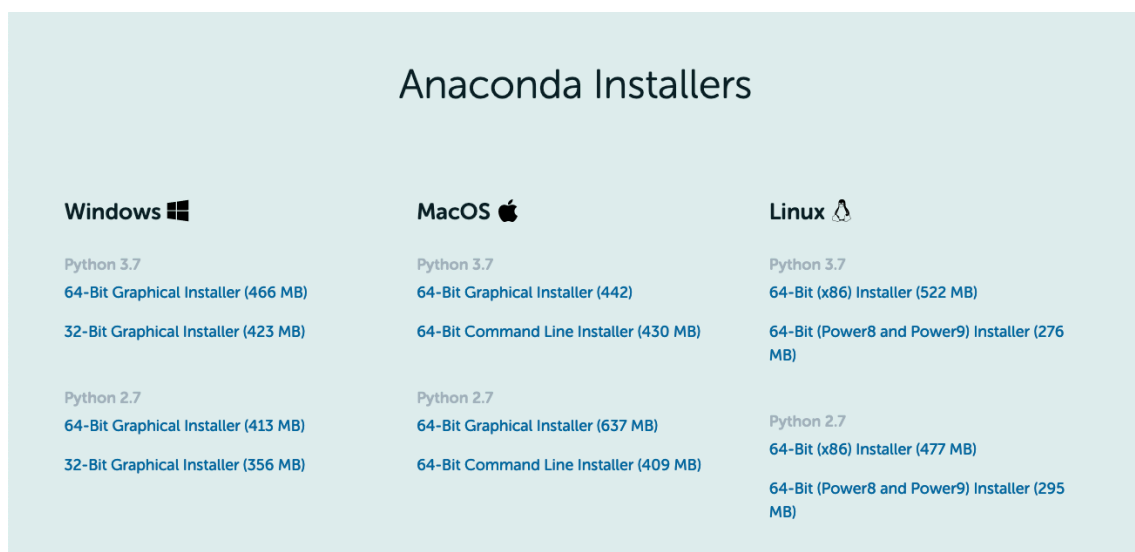
43

44 図 4.1 Anaconda 公式ダウンロードサイト

45

46 (2) 下の方へスクロールして OS にあったインストーラを選ぶ。

47



48

49 図 4.2 様々な OS に対応したインストーラ

50

51

52 (3) ダウンロードしたインストーラを起動する

53



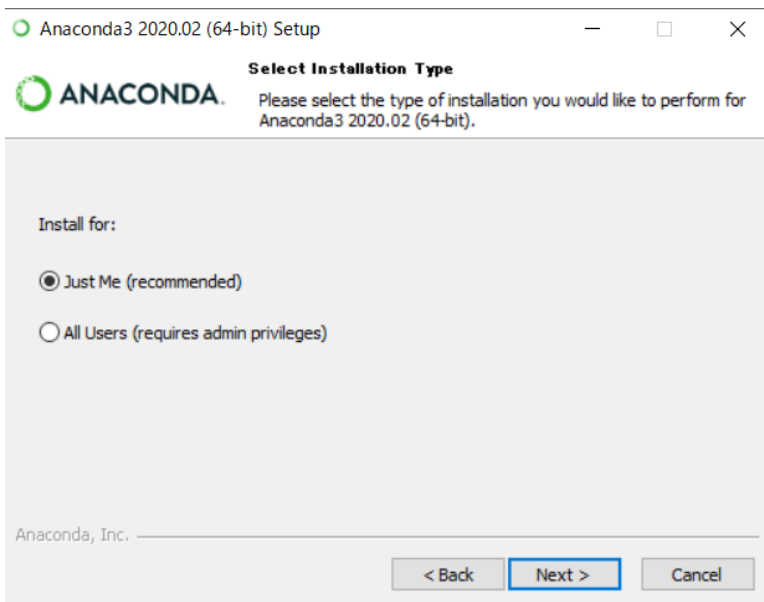
54

55 図 4.3 Anaconda インストーラが起動

56

57 (4) デフォルトインストールで良いので[Next]を押し続ける

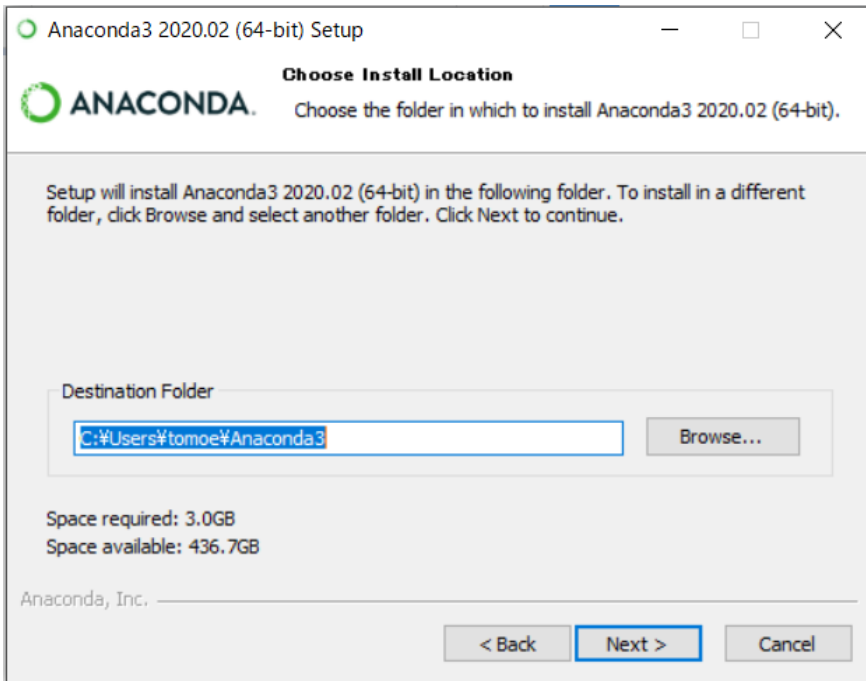
58



59

60 図 4.4 Anaconda インストーラが起動

61



62

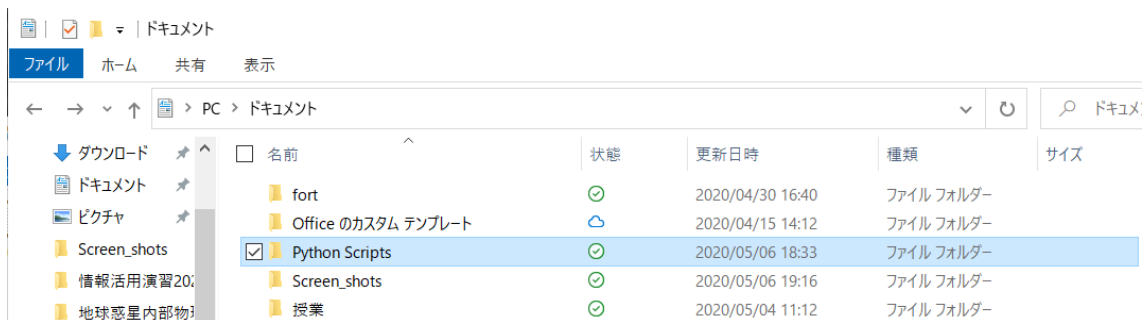
63 図 4.5 Anaconda のインストール先選択・デフォルトが良い

64

65 (4) インストーラを終了する。

66 (5) ドキュメントフォルダの中の“Python Scrips”を確認。

67



68

69 図 4.6 Anaconda のインストール先選択

70

71 (6) スタートメニューに Anaconda フォルダとその中のアプリケーションはで
72 きていることを確認。

73

74

75 4.2 インストール後のテスト

76 (1) [Anaconda3 (64bit)]→[Anaconda Powershell Prompt (Anaconda3)]をクリック

77 Powershell Prompt では Unix コマンドが使える

78 (2) Python インタプリタが使えるコマンドプロンプトが起動する

79 (3) “python --version”と入力して python のバージョンを確認する

80



81

82 図 4.7 Anaconda Prompt の起動と Python のバージョンを確認

83

84

85 5. Python インタプリタ

86

87 5.1 Python インタプリタの起動とプログラムの実行

88 (1) “python”と入力すると python インタプリタが起動する

89 (2) プロンプトが“>>>”に変わる

90 (3) 次のように入力する。1行毎にエンターキーを押す

91 >>> a = 3

92 >>> b = 1

93 >>> c = a + b

94 >>> print (c)

95

```

Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\tomoe> python --version
Python 3.7.6
(base) PS C:\Users\tomoe> python
Python 3.7.6 (default, Jan  8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = 3
>>> b = 1
>>> c = a + b
>>> print ( c )
4
>>> -

```

96

97 図 5.1 Python インタプリタとプログラムの逐次入力・実行

98

99 (4) 上のように入力結果が表示される

100 (5) [Ctrl]+[D]を押すと python インタプリタが終了する

101

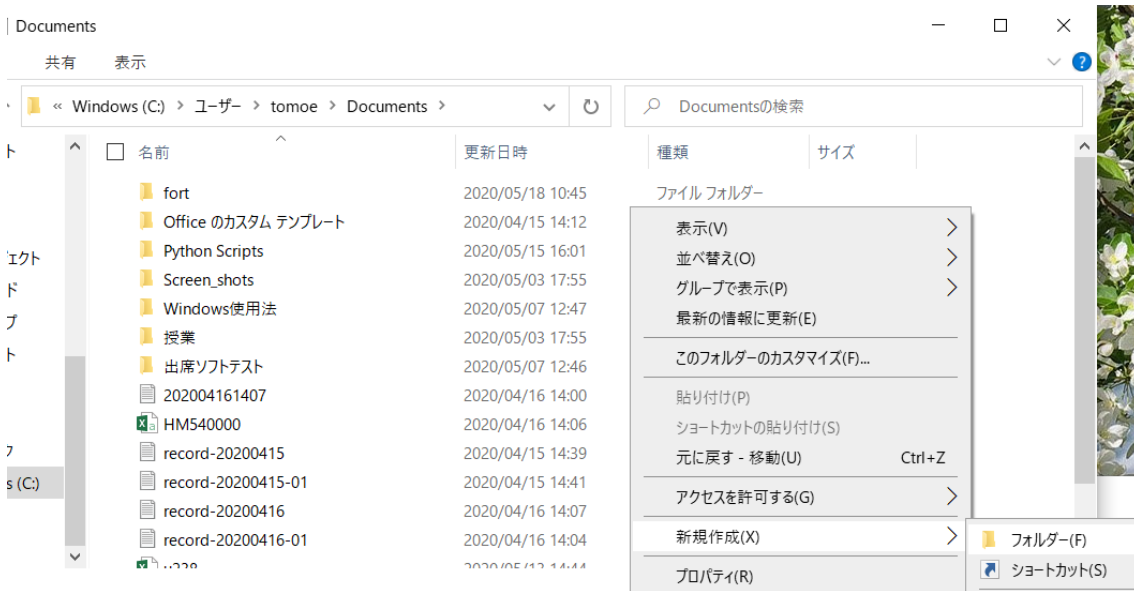
102 5.1 プログラムの保存と保存したプログラムの実行

103 (1) Explorer (ファイルブラウザ) で Python プログラムを保存する場所を作る

104 [右クリック]→[新規作成]→[フォルダー]

105 例えば, C:\Users\yourID\Documents\python_prog など

106



107

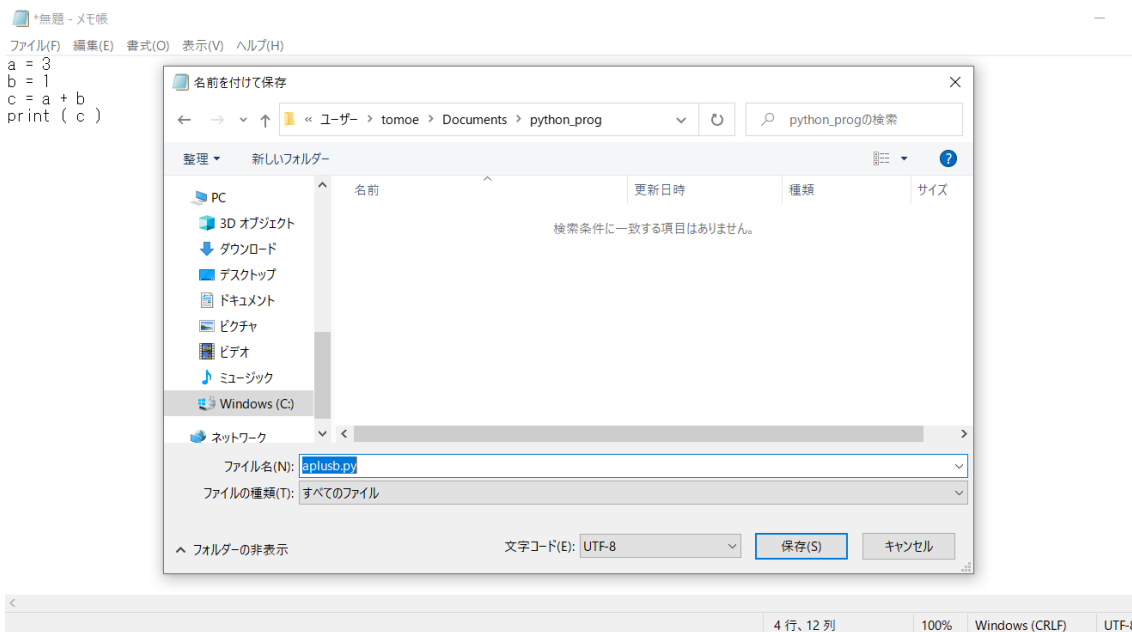
108 図 5.2 Python プログラム保存先の作成

109

110 (2) 「メモ帳」や「TerraPad」などのテキストエディタを起動する

111 「メモ帳」は[スタートメニュー]→[Windows アクセサリ]の中にある

- 112 (2) 「メモ帳」などが起動したら、5.1 のプログラムを入力する。
 113 (3) [名前をつけて保存]を選ぶ
 114 (4) プログラム保存フォルダ C:\Users\yourID\Documents\python_prog に移動する
 115 (4) ファイルの種類を選ぶプルダウンメニューで、[すべてのファイル]を選ぶ
 116 (5) ファイルの名前を入力、[保存]ボタンを押す
 117



- 118
 119 図 5.3 Python プログラム保存先の作成
 120
 121 (6) 先ほど python インタプリタを動かした Anaconda PowerShell Prompt を前面に
 122 持ってくる
 123 (7) プロンプトに現在のフォルダがでているので、
 124 `¥Users¥yourID`
 125 になっているか確認する
 126 (8) 次のコマンドでフォルダを移動
 127 `> cd Documents¥python_prog`
 128 (9) `ls` または `dir` と入力して先ほど入力したプログラムがこの場所に保存されているか
 129 確認
 130 (10) 次のように python 実行コマンドを入力して、プログラムを実行する
 131 `> python aplusb.py`
 132 (11) 結果が表示されているか確認する
 133

```
Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\tomoe> cd Documents\python_prog
(base) PS C:\Users\tomoe\Documents\python_prog> ls

ディレクトリ: C:\Users\tomoe\Documents\python_prog

Mode                LastWriteTime         Length Name
----                -
-a----            2020/05/19  22:45             36 aplusb.py

(base) PS C:\Users\tomoe\Documents\python_prog> python aplusb.py
4
(base) PS C:\Users\tomoe\Documents\python_prog>
```

134

135 図 5.4 プログラム確認と実行

136

137

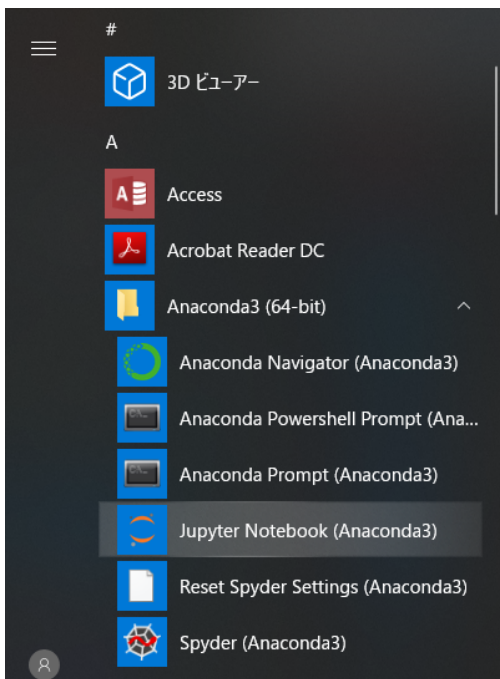
138 5. Jupyter Notebook 使用法

139

140 5.1 Jupyter Notebook の基本

141 (1) [スタートメニュー]→[Anaconda3]]→[Jupyter Notebook]を選ぶ

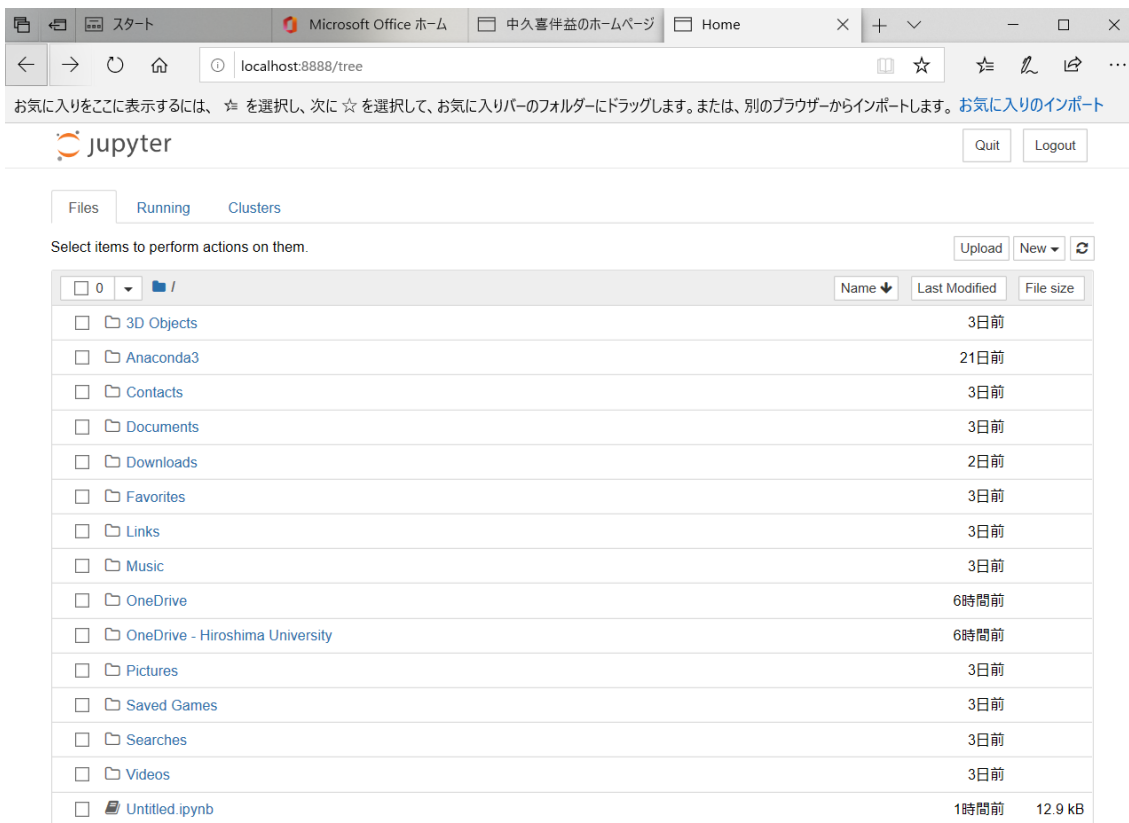
142



143

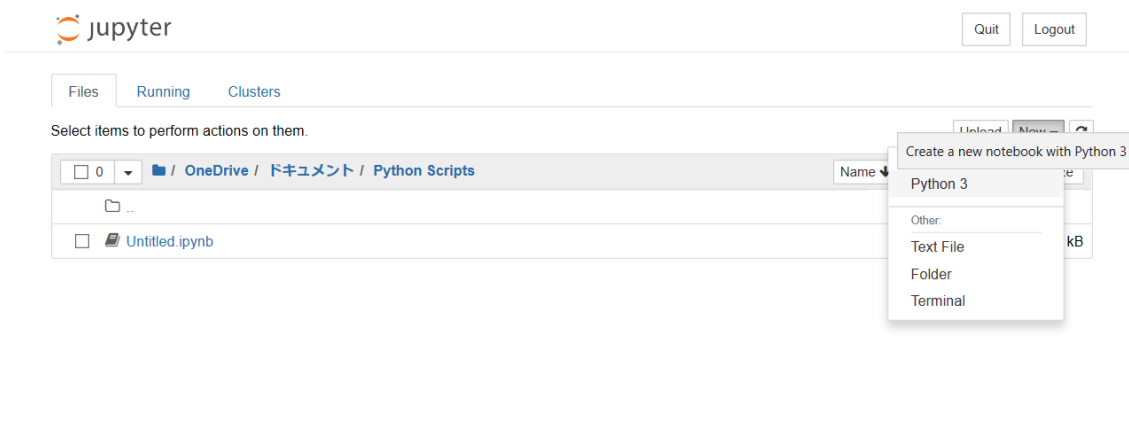
144 図 5.1 Anaconda インストール後のスタートメニュー

- 145 (2) Jupyter Notebook がブラウザ上で起動する。起動すると、Jupyter Notebook のフ
 146 ァイルブラウザが表示される。
 147



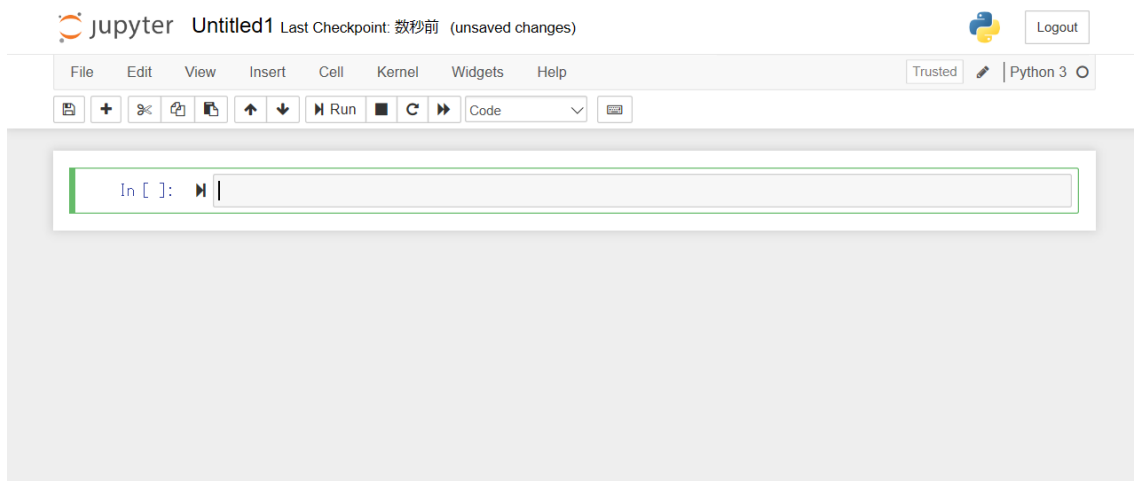
- 148
 149 図 5.2 Jupyter Notebook が起動したブラウザ (Microsoft Edge)
 150

- 151 (3) 右の[New]から[Python 3]を選ぶ
 152



- 153
 154 図 5.2 新しいプログラムの選択

155 (4) ブラウザ画面上にある横長の四角い枠の中に、1行ずつプログラムを入力する。
156



157

158 図 5.3 Jupyter Notebook が起動したブラウザ (Microsoft Edge)

159

160 (5) [Enter]キーを押すと、行が自動的に増えていくので、次の行が入力できる。

161

```
In [5]: ▶ %matplotlib inline
import matplotlib.pyplot as plt

# データ
x = list( range( 0, 5 ) )
y = [ ]
for i in range( 5 ):
    y.append( 2 * x[ i ] ** 2 ) # y = 3x - 24

# グラフ
plt.plot( x, y )
plt.grid( color = '0.8' )
plt.show
```

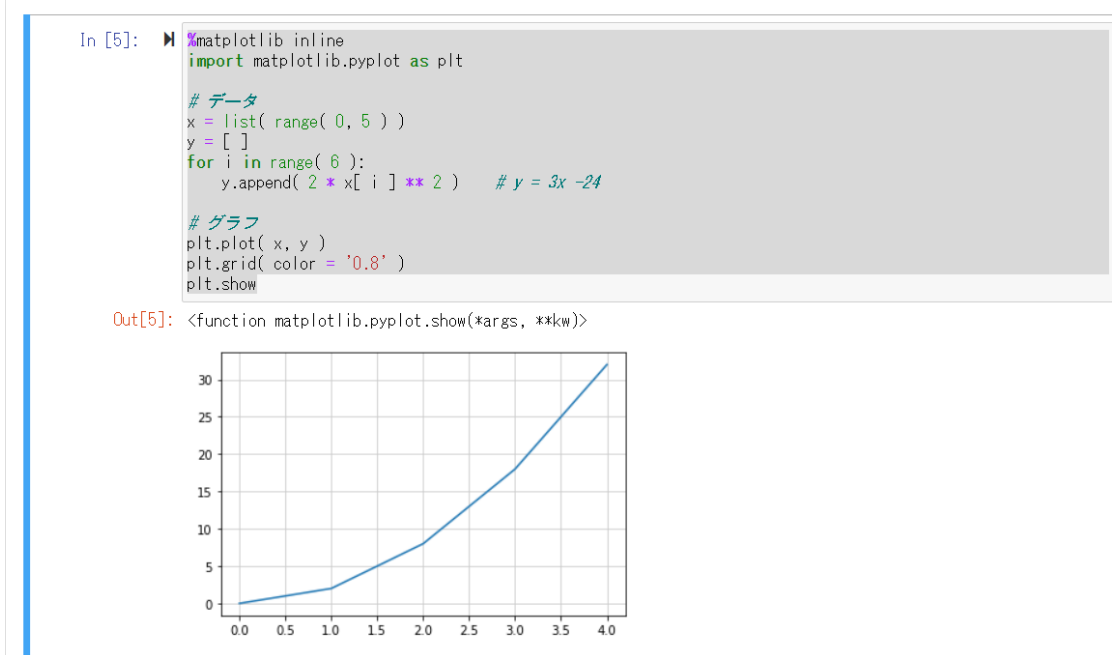
162

163 図 5.4 プログラム入力中の表示

164

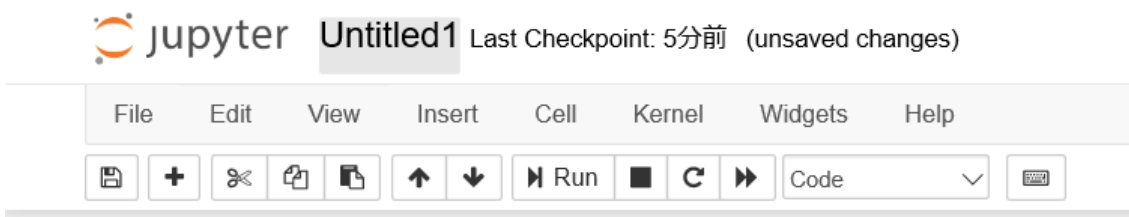
165

166 (6) 入力が終わったら, [Run]ボタンを押す。正しく入力されているときにはプログラム
167 が実行される。このプログラムではグラフが表示される。
168



169
170 図 5.5 実行後の表示

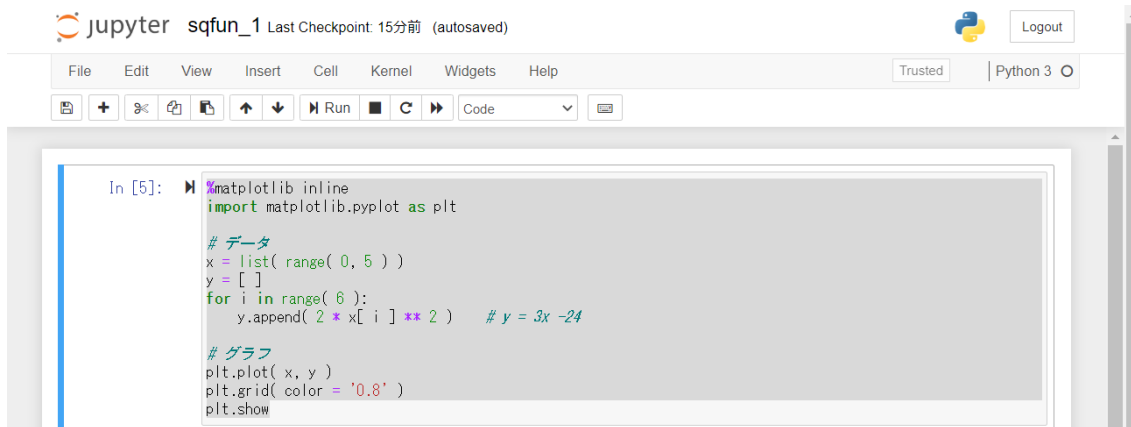
171
172 (5) ファイルとして保存するときは“Untitled”をクリックするか, [File]メニューから
173 [Rename]を選ぶ



174
175 図 5.6 ファイルの保存

176
177

178 (6) ここでは、“sqfun_1”という名で保存した。ここでは、python のファイル(テキスト
179 トファイル)でなく、Jupyter Notebook のファイルとして保存する (そのために、
180 ファイル名に拡張子.py つけてはいけない。自動的に.ipynb になる)。
181



182
183 図 5.7 ファイル保存後の表示

184
185 (7) 再度ファイルを開くときには、起動画面のファイル表示のところからファイルを選
186 ぶ (クリックする)。
187



188
189 図 5.8 Jupyter Notebook ファイルブラウザの表示
190

191 **6. プログラミングの基本**

192

193 **6.1 プログラムの基本構造**

194 プログラムを使うのは、数値シミュレーションの結果を得たり、実験の結果取得
195 したデータを処理したり(手動でするより高速に)するためである。したがって、
196 プログラムは必ず次のような構造を持つことになる。

197 ・データを入力する：物理パラメータや実験データ

198 ・データを処理する

199 ・処理した結果を出力する：可視化につなげる

200

201 **6.2 データの入力元や出力先**

202 (1) データが少ない

203 ・入力：プログラム中に値を代入，キーボードから入植

204 ・出力：画面

205 (2) データが多い

206 ・入出力：ファイル (数値や画像)

207 (3) データの入出力を行うため，`print()`などの入出力関数を用いる

208

209 **6.3 プログラムの流れに関する処理**

210 プログラムは最初の行から順番に2進数のCPU命令(機械語)に翻訳されて実行される。

211 しかし、次のような場合には処理の流れを変更する必要がある。

212 (1) 同じ処理を何度も行う

213 ・繰り返し処理

214 (2) 条件によって、異なる処理を行う

215 ・条件分岐

216 (3) 繰り返し処理や条件分岐処理のため，`for`文や`if`文が用意されている

217