

2018年航空宇宙情報システム学第2
後半「プログラミングと数値計算」

第1回 プログラミング学習の準備

2018年5月8日

はじめに

- 航空宇宙システム学第2・後半では、プログラミング技術と数値計算アルゴリズムの基本を学ぶ
- プログラミング言語はPythonを使用する
 - 昨年からバージョン 3を利用．詳細は後述
- 場所は情報基盤センター「大演習室1」を使用
- 授業資料の配布、課題の提出、出欠調査、等は、ITC-LMSで行う

スケジュール(予定)

1. 5/8 : システムの使い方とプログラミング準備
2. 5/15 : Python 入門 1 (変数と関数)
3. 5/22 : Python 入門 2 (条件文・再帰呼び出し)
4. 5/29 : Python 入門 3 (繰り返し文・文字列)
5. 6/5 : Python 入門 4 (リスト・辞書・タプル)
6. 6/12 : Python 入門 5 (ファイル入出力)
7. 6/19 : 数値計算入門 1 (行列・ベクトル演算)
8. 6/26 : 数値計算入門 2 (常微分方程式)
9. 7/3 : 数値計算入門 3 (最小二乗法、最適化)
10. 7/10 : 数値計算入門 4 (考え中)
11. 7/17 : 予備日

* 5/29午前、7/17 は「補講日」に指定されています。他科目の補講がある場合は知らせてください。日程変更を検討します。

諸連絡

- 演習室のMac端末使用を前提とするが、自分のノートPCを持参して使用しても良い
 - 無線LAN (UTokyo WiFi) を使うためには、アカウント発行の必要あり
 - 詳しくは、<http://www.u-tokyo.ac.jp/ja/administration/dics/service/wifi.html>
 - 教育用計算機システム、ITC-LMS、上記UTokyo WiFiを利用するには、**UTokyo アカウント**が必要
- 成績は(1)出席(2)小課題(3)最終課題で判定
- TA(ティーチングアシスタント)
 - 新居君、苅野君(堀・矢入研修士課程)
 - 小課題の採点、質問への回答、を担当してくれます

教育用計算機システム (ECCS) と 授業支援システム (ITC-LMS) の利用

はじめに

- 正式名称: 東京大学情報基盤センター・教育用
計算機システム (略称: ECCS)
- ポータルサイト: <http://www.ecc.u-tokyo.ac.jp/>
- 各種サービスを提供
 - iMac 端末の利用
 - メールサービス (ECCSクラウドメール)
 - ITC-LMS (学習支援システム)
 - いずれも、UTokyo アカウントで利用可
 - 絶対にアカウントを他人に貸してはいけません

iMac端末の利用

- 学生共通アカウントでログイン
- 設置場所
 - 情報基盤センター・演習室、自習室
 - 工学部6号館2階、総合図書館メディアプラザ、等
 - 福武ホールB1F実習室
- Mac OS X 10とWindows 10が動作
 - 本授業ではMac OS の利用を前提とする
 - 自分のPCを使う場合は、OSはWindowsやLinuxでも構わない(後述)

ITC-LMSの使用(1)


- 授業資料の配布、課題の提出、質問等は、**学習支援システムITC-LMS**を利用する
 - <https://itc-lms.ecc.u-tokyo.ac.jp/portal/login>
- 共通アカウントでログインできる(はず)




ITC-LMSの使用(2)

ログイン後の画面(履修科目によって異なる)

ITC-LMS
ITC Learning Management System


LMS


コース検索


お知らせ

▶ 利用マニュアル

▶ お問い合わせ

▶ ヘルプ

▼ Language

▶ 個人設定

▶ ログアウト

ログイン: 矢入 健久 (1492296039) さん

! 更新情報(管理者)

2017/04/25 08:45

⊙ [金5 学術フロンティア講義 (航空宇宙工学 ー社会を豊かにする総合工学ー)]

教材(第2回教材(姫野))が更新されました

2017/04/25 08:43

⊙ [金5 学術フロンティア講義 (航空宇宙工学 ー社会を豊かにする総合工学ー)]

教材(第3回教材(小紫))が更新されました

2017/04/24 11:27

⊙ [金5 学術フロンティア講義 (航空宇宙工学 ー社会を豊かにする総合工学ー)]

教材(第4回教材(青木))が追加されました

2017/04/22 03:00

⊙ [金5 学術フロンティア講義 (航空宇宙工学 ー社会を豊かにする総合工学ー)]


履修者範囲設定が履修登録者に変更されました。

出講表

2017年度

S1

 履修者名簿

時限	月	火	水	木	金	土
1限		航空宇宙情報システム学第二 ⊙ 				
2限						
3限						
4限						
5限					学術フロンティア講義 (航空宇宙工学 ー社会を豊かにする総合工学ー)	

ITC-LMSの使用(3)

本授業のトップページ

The screenshot displays the ITC-LMS interface. At the top, there's a header with the ITC-LMS logo and navigation links like '利用マニュアル', 'Language', 'お問い合わせ', '個人設定', 'ヘルプ', and 'ログアウト'. Below the header, the course name '航空宇宙情報システム学第二' is shown. The main content area includes course details such as '工学部 03-341331 航空宇宙情報システム学第二', 'コース管理者' (堀浩一, 矢入 健久), and '開講学期' (S1S2/火曜/1限). A sidebar on the left contains links for '教材' (Materials), 'お知らせ' (Announcements), '更新通知' (Update Notifications), and '出席' (Attendance). The '出席' link is circled in red. At the bottom right, there's a 'Page Top' button and a direct link URL: <https://itc-lms.ecc.u-tokyo.ac.jp/>.

出欠はこの機能を使う

教材ダウンロードや課題プログラムの提出も
ここからできる

Python によるプログラミング 準備編

なぜ Python を使うのか？

Cをやめた理由

- 受講者の挫折率が高い(メモリ管理、ポインタなど)
- 標準の行列・ベクトル演算ライブラリが無い

Pythonを採用する理由

- 学習効率が**高い**(初心者優しい)
- 強力な**行列・ベクトル演算、科学計算ライブラリ**がある。
プロの研究者・技術者の利用も急速に増加
- 実行効率も改善されている
- 文字列処理、GUI作成等も比較的容易
「使える」アプリをすぐに作れる

他の候補

- Java, Ruby : 人気だが数値計算向きとは言えない
- Matlab : 商用、数値計算に特化し過ぎ
- Julia, R, Scilab : マニアック過ぎる

教科書・参考書

- オンラインチュートリアル

- “Think Python 2e”

- <http://greenteapress.com/wp/think-python-2e/>

- 教科書として利用。書籍としても販売

- NumPy Quickstart Tutorial

- <https://docs.scipy.org/doc/numpy-dev/user/quickstart.html>

- NumPyの使い方はこれで学ぶ

- その他参考情報源

- The Python Tutorial

- <https://docs.python.org/3.6/tutorial/>

- オフィシャルなチュートリアル。日本語訳もある



iMac端末でのPython環境

- ECCSでは既にPython のVersion 2 系と 3系がインストールされている
 - どちらも Anaconda というPythonディストリビューションでインストールされている
 - 使用バージョンの切り替えには Pyenv を使用
 - デフォルト(既定値) ではVersion 2系が起動するようになっている
- 本授業では Version 3 系を使用
 - 世間もVer.3 に移行しており、積極的にVer.2を使う利用は無い

iMac端末で Python 3 を使用する方法

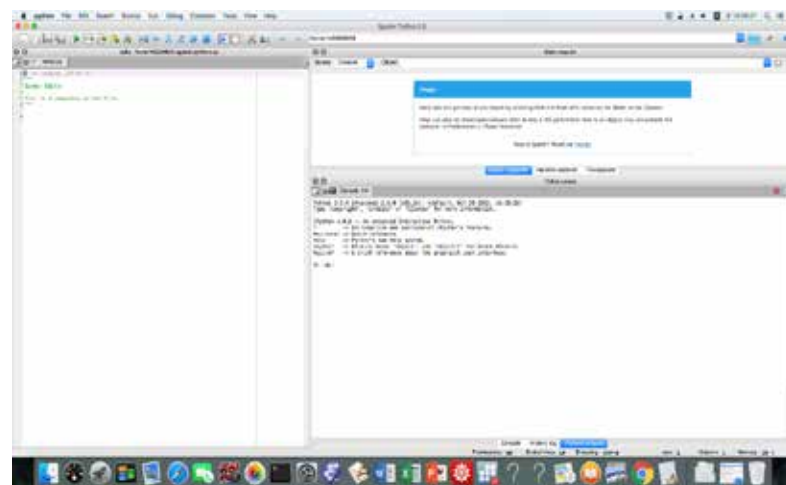
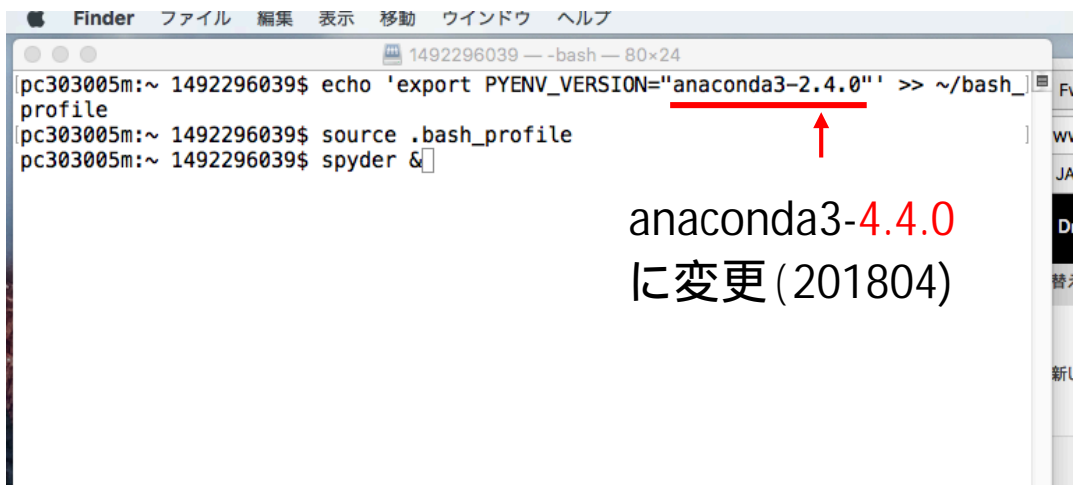
1. ターミナル を起動する (次スライド参照)
2. 以下のコマンドを順に打ち込む (次回以降は不要)

```
$ echo 'export PYENV_VERSION="anaconda3-4.4.0"' >> ~/.bash_profile
```

```
$ source .bash_profile
```

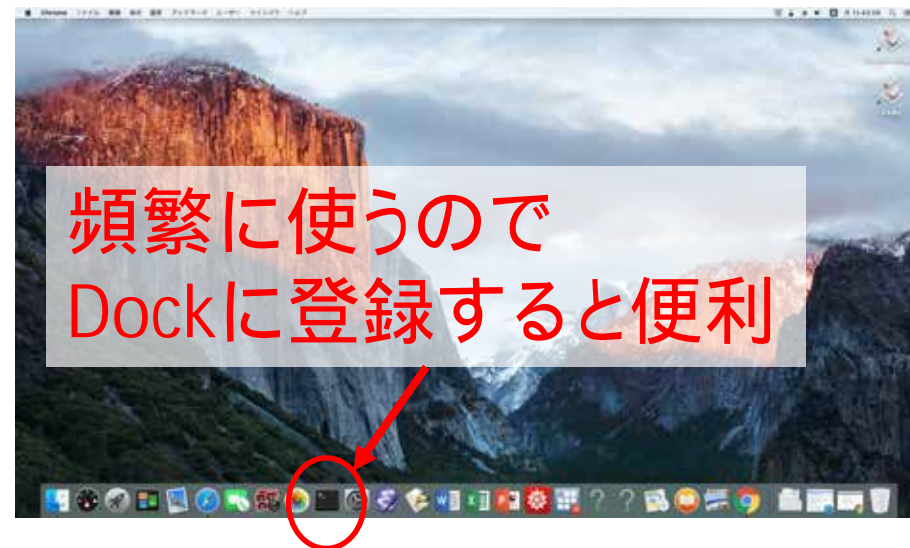
3. Python統合環境 Spyder を起動する

```
$ spyder &
```



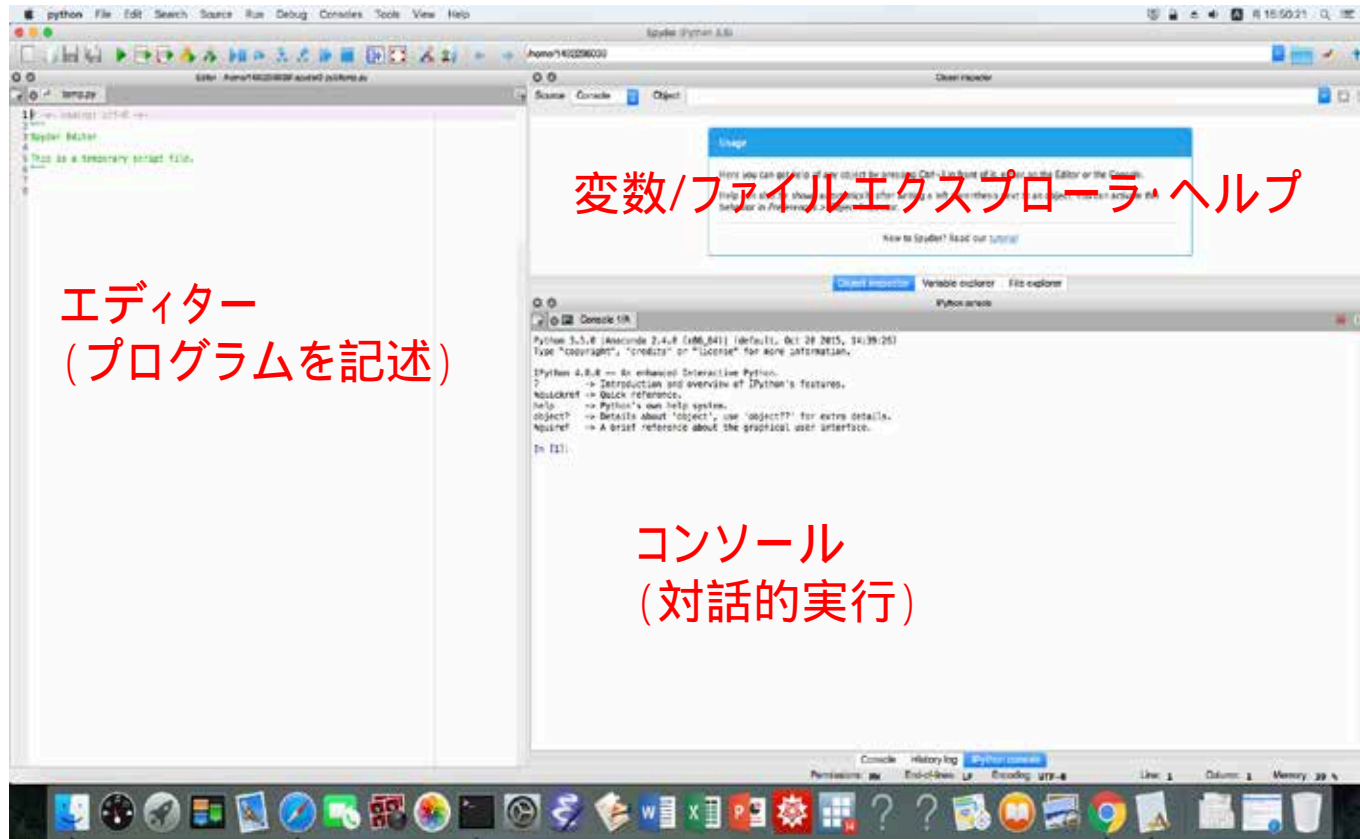
(備考) `pyenv local` コマンドを用いて、作業ディレクトリ毎にバージョン設定する方法もあるが、混乱しやすいので、初心者には勧めない。

MacOS で“ターミナル”を起動



統合開発環境 Spyder

- 統合開発環境 (IDE) : プログラム (スクリプト) の編集, 実行, デバッグ, インタプリタでの対話的操作が可能
- Spyder 以外の開発環境も存在 (IDLE, Eclipse, など)

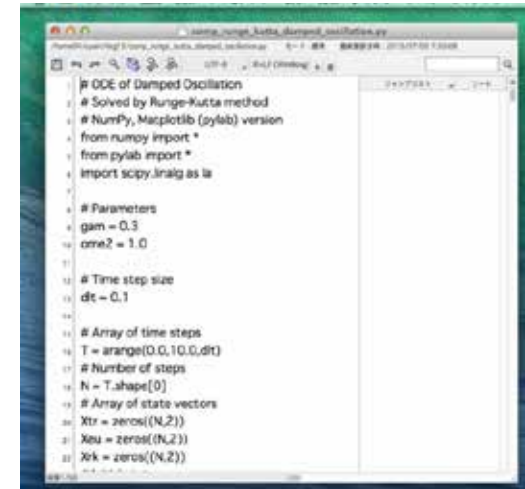


ターミナルとエディタを使用 (非推奨)

- 統合環境を使用しないミニマル主義な方法
- “ターミナル”を起動し、**python** (リターン) と打つと、pythonが「対話モード」で起動する
 - 終了時は、`>>> quit()` と入力
- 対話シェル IPython を使うと便利
- プログラム (スクリプト) の編集はテキストエディタ (TextEdit, mi, emacs, など) を使用する



ターミナルでpythonを対話モードで起動



テキストエディタ (mi) でpythonプログラムを編集

自分のPCにPythonを導入するには

- 一番楽な方法: オールインワンパッケージ(ディストリビューション)をインストール

Python 本体、NumPy, SciPy, Matplotlib など主要なパッケージが全て(余計なものも)導入される

- Anaconda (Windows, Mac, Linux): 全部入り

- 余計なものはいらない人は Miniconda

- 他に、Canopy, Python(x,y) など (Python 2系のみ)

- 自分で一つ一つインストールする場合 (非推奨)

1. オフィシャルサイトからpython本体をダウンロードしインストールする

2. パッケージマネージャー pip を使って必要なパッケージをインストール (例: `pip install numpy`)

Think Python:
“How to Think Like a Computer Scientist”
by [Allen Downey](#)

<http://greenteapress.com/wp/think-python-2e/>

Preface & Chap.1 The way of the program

まえがき(Preface)

- Java用の教材を1999年に執筆
 - 著者 Downy氏はFranklin W. Olin College of Engineering の教授
 - 履修者の落第率の改善が動機
 - GNU Free Documentation Licenseで公開！
- 2003年よりPython版を採用
 - 高校教師が最初に「翻訳」
 - Javaに比べて、学生の習得率、意欲がアップ
- 2nd Edition で Python 3 に対応

Chapter 1 The way of the program

- 本書のゴール:「計算機科学者のように考える」
 - 数学者のように形式言語 (formal language) を操る
 - 工学者のように設計し、システムを組み立てる
 - 科学者のように複雑系を観察し仮説を立て検証する
- 計算機科学者にとって最も重要なスキル:
「問題解決」 (Problem solving)
 - 問題の定式化、解法の創出、解法を明確に記述
 - プログラミングの学習は問題解決スキルを習得する絶好の機会

そもそも「プログラム」とは何？

- 定義：「計算 (computation) の仕方の指示書」
 - － 数学的な算術演算だけでなく、「文字を検索し置換する」というような操作も「計算」の一部
- 全てのプログラム言語に共通する基本指示
 - － 入力：キーボードやファイルなどからデータを取得
 - － 出力：画面に表示したりファイルに書き出す
 - － 数学：足し算や掛け算などの算術演算
 - － 条件実行：特定の条件に対してコードを実行
 - － 繰り返し：ある操作を(少し変えながら)繰り返し実行

Python を走らせる ("Running Python")

- 教科書では、Webで使える(クラウド版の)Pythonを紹介しているが、授業では Spyder のコンソールを使用
- `>>>` をプロンプトと呼ぶ。プロンプトの後にプログラムコードを入力して改行(Enter)を打つ
 - 注意: IPython の場合、プロンプトは、`In [1]:` のように表示される。

```
In [1]: 1+1
```

```
Out[1]: 2
```

```
In [2]:
```



自分で入力してEnter



答えが表示される



次のプロンプトが表示される

最初のプログラム

- “Hello World !”と表示するだけのプログラム
 - なぜかそういう慣習になっている
 - Wikipedia で様々な言語での例がまとめられている
- Python では1行で済む
 - インタラクティブモードで実行する場合

```
In [2]: print('Hello world !')  
Hello world !
```

print は関数

– バージョン2 では、次のように書いていた

```
>>> print 'Hello world !'
```

print は単純文

– シングル引用符(')の代わりに二重引用符(")でも可

算術演算子

- 四則演算: + - * /

```
>>> 40 + 2
```

```
42
```

```
>>> 43 - 1
```

```
42
```

```
>>> 6 * 7
```

```
42
```

```
>>> 84 / 2
```

```
42.0
```

← 浮動小数点とみなして計算が行われる。
(Ver.2 では整数として演算されていた)

- 冪演算: **

```
>>> 6**2 + 6
```

```
42
```

値と型

- 値 (value) の例: 1, 2, 'Hello world !'
 - 明らかに、1,2 と 'Hello world' とでは種類が違う
 - この値の種類のことを型 (type) と呼ぶ
- 型の例
 - 2 : 整数型 (integer)
 - 3.2 : 浮動小数点型 (float)
 - 'Hello' : 文字列型 (string)
- type 関数で型を調べられる

```
>>> type(42.0)
<class 'float'>
>>> type('Hello, World!')
<class 'str'>
```

形式言語と自然言語

- 自然言語: 人々によって話される言葉
 - 曖昧性、冗長性、比喩などを含む
- 形式言語: 特定の目的のために設計された言語
 - 数式: 数字と記号との関係を表すための形式言語
 - 化学式: 分子の化学構造を表現するための形式言語
 - プログラミング言語: 計算を表現するための形式言語
- 形式言語は厳密な文法規則を持つ
- 形式言語の構成要素
 - トークン: 単語、数字、化学的要素、などの基本要素
 - 構造: トークンの並び方、順番

デバッキング

- プログラミングにおけるエラーをバグと呼ぶ
- バグを探し取り除く過程をデバッグと呼ぶ
- プログラミング、特にデバッグは、時に強い感情（怒り、困惑、等）をもたらす
- 計算機のことを、「速さ」や「精度」等の長所と「共感を持たない」「全体像を見られない」「忖度しない」などの短所の両方を持つ「従業員」だと考えると良い
 - 「従業員」の長所を利用し、短所の影響を減らすようなマネージャーになろう

課題 (提出の必要なし)

- ITC-LMSの「計算機工学概論」のページから、
 - 「出席」を送信する
 - 今日の授業スライドPDFをダウンロードする
- Python 3 を利用できるように設定する
- Python の対話シェル(コンソール)で、
 - "Hello World !" を表示させる
 - 電卓として使えることも確認する
 - Tabキーによる自動補完機能の便利さを体感する
- 教科書 "Think Python" の1章を復習
- 自分のノートPCを持っている人は、
 - Python 3 をインストール

次回の課題からは、作成したプログラムをITC-LMSから提出してもらいます。