

1. Web サイト

スライド、資料、事務連絡などはすべてココでシェアする予定↓

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

2. 教室

5/10 (木) ~8/2 (木) E209 教室

3. 今日の内容

- 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 の動かしかた
- 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

4. 参考情報

- paiza.IO
<https://paiza.io/ja>
- 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>
- Paiza ラーニング
<https://paiza.jp/works/python3/primer>
- ドットインストール
https://dotinstall.com/lessons/basic_python_v3
- Markdown
<https://guides.github.com/features/mastering-markdown/>

技術的な問題について：

- 自分の PC を使いたい人は補足資料に沿ってネットワークの設定をしてください。
- それ以外は USB メモリも要らないし変なインストールも要らないしコマンドプロンプトも要らないため、最も楽です。インストールもインストーラが提案してくる設定に OK 押し続けければ良いです。インストール先の変更とかありません。というか本来こういう風に楽に使えるものです。今回の USB メモリとかコマンドプロンプトでコマンド入力とかは全てここの演習室の特殊設定に対処するためだけのものです。
- Windows 10 の人は「コマンドプロンプト」は入っていますが演習室の説明の形では使えないので注意 (Powershell というのが導入されたことによる)。そもそも自分の PC 使う人はコマンドプロンプト使わないでこの演習できるので無視してください。
- Mac の人は家の PC でインストールした Anaconda が Mac 用だった場合、そもそも演習室では動きません。なので、演習室を使うならば USB メモリの準備は演習室でやってください。ただし、演習でも自分の Mac を使うなら、普通に Anaconda をインストールすれば普通に使えます。Jupyter Notebook のアイコンも Application フォルダにできると思います。
- とにかく Jupyter Notebook さえ起動できれば、OS は Windows でも Mac でも (Linux/UNIX でも) 同じように Python を実行できます。少なくともこの演習の範囲では (OS 固有の機能を用いたプログラミングや操作をしなければ、という意味)。

先週の質問への回答：

Q. 自分の Mac で授業を受ける場合、授業であった色々な手順をふむ必要はありますか？自分はやりたくないです。
A. 授業でも言っていますが、全くその必要はありません。そもそも Mac ではあの操作できません (Windows 用の操作)。上記情報や補足情報を参照のこと。

Q. ELMS の時間割かつ出席確認をすると全て「欠」となっていますが、この授業ではどのように出欠を取っているのでしょうか。

A. カードのスキャンで取ると事務はいつてましたが、システム確認したら初回ガイダンス以外、「全員が全欠」(マジか!) になってたので、システム上の問題な気がします。事務のシステム担当に聞いておきます。教室変わったりしてるのが反映されてないんだろうかねえ。この授業の履修上は毎回この「ミニレポート」出してれば問題ありません。

Q. プログラマーの一般的なタイピングの速度はどれくらいでしょうか？日本語はそれなりに打ってきたのですが英語はうつ機会があまりなく、タイピング速度が遅いです。

A. 人によりますが、ブラインドタッチはほぼ必須なので、もしあまり得意でない人は最初に頑張って練習してください。演習室の PC にも練習用のソフトが入っていますし、瀧川も大学入るまでパソコン使ったことがなかったので 1 年生の 1 学期に頑張って練習しました。記号や数字のブラインドタッチは練習しないと多分できませんがプログラマはできた方が良いでしょう。なお、これはキーボードの言語に依存するので注意してください。実はプログラミングで使う記号は「英語キーボード」の方が良い位置に配列されているため、英語キーボードを好む人もいます。私も英語キーボードを普段使っています。なお、マニア(私)は「無刻印」(キートップに何も印字されていない) キーボードを使ったりします。

なお、その筋では有名なのですが下記の TypeRacer が好きで私もたまに気晴らしでやります。登録しなくてもできるし基本英語なので英語タイピングの訓練の成果確認にどうぞ。練習モードもあるし、ネット対戦もできます。

<http://play.typeracer.com>

瀧川で平均 75wpm、ベスト 85wpm くらいです。Wpm は word per minute で一分間に入力できる語数でタイピングの速度を表すのに良く使われます。TypeRacer の目安では「Beginner 0 - 24, Intermediate 25 - 30, Average 31 - 41, Pro 42 - 54, Typemaster 55 - 79, Megaracer 80+」だそうです。なお、やってみるとわかりますが、英語タイピングが速くなるためには

実は英語力(スペルが正確にパッと把握できる程度の)も必要かも。日本語入力は確かにもっと速い気がしますね。なお日本語ではタイピストレベルの速度を出すには「ローマ字入力」ではなく「かな入力」らしいです。入力数が少ないので。

Q. Python でスマホのアプリって作れたりしますか?自分専用の使い勝手の良いアプリ作って見たいです!!

A. 基本は Android なら Java、iPhone なら Swift という言語で作る必要がありますが、Web アプリ(Web アプリケーション)という形で Python でもスマホやタブレットのアプリ開発は可能です。Youtube、Instagram、DropBox、Pinterest、などは Python で開発された有名なものですね。Python で Web アプリなら Django というフレームワークを使うものが多い(Instagram など)です。ただし Web を経由(アプリ使うのに通信が発生)する必要があり Web の仕組みについての知識が入ります。フォーカスが強すぎて、この演習では扱えないので、Web アプリケーションや Django で調べてみてください。また、一般のアプリ開発に興味あったら、授業で紹介したドットインストールの下記を見てみてください。無料です。iPhone アプリ開発入門(全 13 回)(ドットインストール) https://dotinstall.com/lessons/basic_iphoneapp_v3
Android アプリ開発入門(全 11 回) https://dotinstall.com/lessons/basic_android_v3

Q. コードの入力で変数を用いることができるのですが、連立方程式を用いた計算もできるのでしょうか。少し試してみましたが、やり方が分からず、エラーに終わってしまいました。

A. プログラミング言語の「変数」は値に名前をつけているだけで、数学の変数とは全く違います。「=」は等号ではなく、「右の値を左の変数に代入する」という規則を意味します。Python で式を書いてもそれは数学的な意味を全く持ちません。したがって何か方程式と捉えて解を計算してくれるわけではありません。コンピュータの CPU やメモリは各素子の電圧を上げたり(1)下げたり(0)してるだけなので。授業で解説した通り、Python インタプリタはそれをコンピュータの中核(CPU)にわかる予め定められたコマンド(四則演算やコピーなど程度)に焼き直しているだけです。数学的なことがやりたいならば、それをどういう手順でやるか(例えば、移項して、変数消去して、...みたいな手順)を全てプログラミングする必要があります。これはもちろん単純なことではなくれっきとした研究分野があります。では、Python では連立方程式は解けないのかということ実は解けます。がそれは Python が解ける訳ではなく、連立方程式をとくアルゴリズムを誰か先人が作りこんだものが付属して(ライブラリやモジュールと言います)、それを呼び出すことで解くだけです。この授業では数学には寄せないつもりですが、ライブラリなどは追いついて授業で説明します! 方程式を解くには線形代数(とくに LU 分解という行列の因数分解)の知識が必要でそれを Python でプログラミングすれば自力でも解けます。線形代数の授業か、あるいは、工学部に行けばどっか授業でやることになると思います。

Q. プログラミングの本を買くと、参考のコードが本の公式サイトで配布されていたりしますが、わざわざ自分で打ち直さなくてもコードを何となく読んで実行すれば十分でしょうか?

A. 本にもよりますが多分十分です。私の授業でも基本的にはコードは配ります。少しちゃんと理解したければ、そのコードをちょこっと書き換えてどうなるか、見てみるというのが最も効果的だと思います。

Q. USB のコマンドプロンプトで打ったコードの意味は何ですか?また「E:」や「F:」とは何なのでしょう?

A. コードの意味:

Jupyter Notebook で Python3 を実行する場合は Python に便利機能をつけた iPython というバージョンを背後でこっそり起動してブラウザに入力された情報をそこへ送って実行しています。そのコアプログラムを Jupyter ではカーネルと読んでいますが、このカーネルをユーザ領域(USB ではなくて PC の方のハードディスク)に書きこんでそれを使うようにプログラムされているようです。ということで、USB にあるこのカーネルに相当する ipython のカーネルをユーザ領域にコピーしないと Jupyter Notebook が動きません。ということで、「(USB メモリに入っている)ipython の kernel のプログラムを user 領域に install(コピー)せよ」という意味です。Python インタプリタである Python.exe を実行しますが、この時に「ipykernel」という ipython の kernel に関するモジュール(-m)の中にある install という機能を「-user」(ユーザ領域へ)と指定して実行しています。

A. 「E:」や「F:」の意味:

これは Windows の機能で授業とは関係ないですが、「ドライブレター」と呼ばれるもので、ハードディスクや DVD ドライブや USB メモリなど周辺機器を A~Z のアルファベットの名前をつけて判別しているだけです。例えば USB メモリを二つさしたら異なるドライブレターがつかますしそれ以外にスマホも充電がてらさしていたらそれにもドライブレターがつかます。(そういうので「E:」の人と「F:」の人がいただけなのかも?)もうちょっとした説明は Windows の入門書か、例えば下記の初心者向け説明を読んでみてください。

<http://wa3.i-3-i.info/word11920.html>