【一般教育演習】プログラミングで問題を解く：集計から人工知能まで・第六回 2018 年 5 月 24 日

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/
   「毎回の作業」「Jupyter 使い方」を忘れた人はここで確認

2. 教室
   5/10（木）～8/2（木）E209 教室

3. 今日の内容
   - 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 for 文、if 文、while 文、型
   - Markdown で文書を飾ってみよう

4. 参考情報
   - Python 公式チュートリアル
     https://docs.python.jp/3/tutorial/
   - Progate (プロゲート)
     http://prog-8.com/languages/python
   - Paiza ラーニング
     https://paiza.jp/works/python3/primer
   - Markdown
     https://guides.github.com/features/mastering-markdown/

先週の質問への回答：

Q. 授業中に自分の PC を使って授業を受けるということはダメなのでしょうか。
A. 前回の授業中に何度か念押しして、前回のプリントにも書いてあると思うけど、むしろ自分の PC で受けられるならそうしてください。ただしネットの設定だけお忘れなく。詳しくはプリントや資料を見てください。

Q. ex を試していて思った質問をメールで送っても良いですか？ミニレポートにコードを手書きして質問するのは大変です。(や可能なら授業で扱ってないコード)
A. 授業の課題については takigawa@ist.hokudai.ac.jp まで質問してください！ 授業以外のコードについては初歩的なものは多分大丈夫ですがモノによります。でもまあ手間がかかりすぎるときは、コレは初心者に説明するには時間かかりすぎ！とか私はデバッグ係ではありません！とか言うだけなので一応質問してみてください。

Q. ex1.ipynb でいじっていたのですが、なんどやっても「b が定義されていない」といった内容のコードが出て終わってしまいました。例のまねをして b[〜]とするだけではダメだったのでしょうか？
A. myfirstcode.ipynb のことだとして回答します。で、すみません、説明し忘れていましたが、上からセルを全部実行しないとダメです。多分「b = […]」と書いてあるセルを実行し忘れているだけだと思います。インタプリタなので順番に実行していかないとダメです。なお、メニューから「Cell」→「Run All」とすると一旦全てのセルを実行してくれるので一回それで全部 Run したあとなら b にアクセスできると思います。

Q. プログラミングを独学でやるのは難しいのでしょうか。
A. (少なくとも基本レベルでは)全く難しくありません。紹介したオンラインのサイトをチェックしてみてください。というか、ある意味ではみんな独学で身につけているとも言えるかもしれません。私も授業でだれかに習ったので今プログラミングできているわけではなくて基本的には自分で身につけた感覚です。情報系だったのでもちろん授業とかではやりますが、それで完全にプログラミングが身につくってことはありません。中高の英語の授業「だけで」英語が喋れるようにならないのと似ています。スキルなので説明されて覚えていく感じのものではないような気がします。

Q. パソコンに内蔵されているメモリと USB や SD カードなどのフラッシュメモリ?はメモリとしては別物なのでしょうか？パソコンを買うとき 8GB か 16GB かで迷って、USB などで捕捉できるなら 8GB で良いのかなと思ったのですが…
A. 全く別物です。「パソコンに内蔵されているメモリ」は電源が入っているときのみ情報を保持できます(揮発性と言います)。それに対して、USB メモリや SD カードやハードディスクは電源がなくても情報が保持できます(不揮発性)。パソコンは情報処理をするとき色々なデータやプログラムを「パソコンに内蔵されているメモリ」に読み出して処理します。そのためこのメモリは非常に高速です。それに対して、USB メモリや SD カードやハードディスクは処理し終わった情報を保存しておくために使うので相対的に速度は遅いです。例えば、Web ブラウザでたくさんのページを開いたり、ゲームやメールも起動していたりするとき、それらのプログラムとデータは「パソコンに内蔵されているメモリ」に展開されています。なので、この領域が枯渇すると動かなくなるか、仕方なくハードディスクなどにちょっとづつはみ出る文を書き出したり読み出したりしながら処理するのですっごく遅くなります。例えるなら、台所で料理作る時のスペース(広いとやりやすいが狭いとやりくりが必要)というか、ものすごいたくさんの荷物が狭い部屋に置かれていて、頻繁にそれを部屋の外に出し入れしなければならない状態というか。一旦に荷物どけないと奥の荷物取れないので一回外に避けて戻して、で、また奥の取りたいとかなって、また外にちょっと避けて、だし終わったら戻して、みたいな非常に無駄な処理をやっている時、パソコンがすごく遅くなります。

Q. パソコンの起動とか動作が遅くなるのもパソコン内のアプリの起動準備が複雑なプログラムになっていたりするからですか。
A. パソコン(やスマホ)が遅くなるのは主に一つ前の Q の説明にある「パソコンに内蔵されているメモリ」がいっぱいで情報処理ができないであることがほとんどです。たくさんプログラムを起動していたりするとそれらは全部メモリ上に読み出されますので、他の処理をするスペースがなくなります。なので、一旦ハードディスクにちょっと避けて処理してまた戻すみたいな非常に無駄な処理が(メモリと比べて)非常に遅いハードディスクや USB や SD カードに対して起こるため遅くなります。一方、起動が遅くなるのはソフトウェアを使っていると起動時に準備してほしいこと(前回使った時の状態や設定など)が増えてきて起動時に処理しなければいけないことが増えるからです。

Q. 保存するときに形式を変更しなければいけないときとそのままでもいいときの違いが余りわからないです。
Q. 拡張子の意味がよく分かりません。教えてください。
Q. 拡張子を変えるとなぜ Jupyter Notebook が反応してくれたのでしょうか？
A. 拡張子というのはファイル名の最後に「.pdf」とか「.ipynb」とかつけるものです。これは単にファイル名の規則です

が、これによってそのファイル(パソコンにとってはただの 0 と 1 の並び)がどういう種類のファイルなのかが分かるので良く使われます。プログラム(例えば python.exe)もデータ(例えばデジカメの写真ファイルや動画ファイル)も、コンピュータの中ではただの 0 と 1 の並び(ずらっとならんだ素子に電圧かけるかけないのパターン)にすぎないので、何のファイルか分かりません。なので、例えば「.jpeg」で終わっていたら写真ファイルだよ、とかが分かるようにつけているだけです。これによって、例えば写真閲覧ソフト(例えば Photoshop)ではプログラムや python コードや pdf を間違えて開くことがないようにできます。ただし、これはただの名前なので、写真ファイル photo010101.jpeg の最後を photo010101.pdf にしてみたところで pdf になるわけではないことには注意してください。Jupyter Notebook のプログラム(jupyter-notebook.exe)では、最後が「.ipynb」で終わっているファイルだけを Notebook ファイルとみなすように多分プログラムされていて、逆に中身は notebook ファイルであっても名前を pdf とか jpeg とか htm とかにすると(プログラム側では中身が分かるわけではないので)関係ないファイルだとして処理してしまうので開けないのだと思います。

Q. スマホ(iPhone)でプログラミングの環境を揃えて実際にプログラミングすることはできますか？
A. これは「スマホ」だけでプログラミングをしたい、という意味か、「スマホ」で動くプログラム(アプリ)を作りたいという意味か、でだいぶ違いますが、パソコンを使わず iPhone だけでプログラミングできるか？という意味で回答します。で、それはもちろんできます。授業で説明した通り、lightbot も scrach も「プログラミング」だし iPhone アプリがあると思います。では、例えば、「Python」のプログラミングができるか？と言えば、実は「Pythonista」というアプリがありできます。(iPhone だけで iPhone アプリが開発できるかと言えば Web アプリは可能で、通常のアプリはできません。Web アプリとは何かについては前回までの Q & A を check してみてください)

Q. 数学の知識で線形代数以外に必要な知識は何ですか。
A. プログラミングを理解するのにってことかな？プログラミング自体に線形代数がいるのではなくて、線形代数をプログラミングすると例えば統計処理ができます、という感じです。なので、例えば、何かシュミレーションをしたいのなら、微分方程式や物理数学になるだろうし、機械学習をしたいのなら確率や統計になるだろうし、パズルを解くとかなら代数学や論理学になるだろうし、むしろ「プログラミング」で何をやりたいのか？ってことじゃないかな。

Q.「タッチタイピング」と「ブラインドタッチ」の違いとは何ですか？最近「ブラインドタッチ」とはあまり耳にしなくなった気がします。
A. 確かにと思って調べたら、最近は「タッチタイピング」と呼ぶそうです。ブラインドタッチは「キーボードが見えない状態(blind)で」タイピングすること、タッチタイピングは「タッチ(触った感覚)で」タイピングすること、というのが英語のニュアンスです。が、ブラインドとは盲目の人をさす言葉でもあるため、特にポリティカルコレクトに敏感な欧米でそういう呼び方は好ましくないのでタッチタイピングと呼びましょう、となっていったみたいですね。あえて日本語にすると、語感的には「ブラインドタッチ = 盲目タッチ」、「タッチタイピング=触覚タイピング」という感じなので、確かに後者の方がいいね。

Q. Python をそれなりに使えるまで学習した後に他の言語を学ぶとどの位違うものですか？おそらく 2 年生の時などに Java を学ぼうと思ってい流のですが、Python を学んでいると Java の学習効率が良くなるなどのメリットはありますか？
A. 一般的に、一つ言語を知っていると二つ目のハードルはめちゃめちゃ下がります。初心者から説明すべきではないかもですがプログラミング言語には流儀がいくつかあって「オブジェクト指向」「関数型」「手続き型」などがあります。で、それが共通していれば他の言語を学ぶのはすごく簡単です。まず、python の基本文法はかなりミニマルでどの言語も同様の構文を持つ場合が多いです。また python や Java は「オブジェクト指向」の言語なので、そこまで python で行ければ実は Java の言語を学ぶ上でのハードルはあまりないと思います。

Q. 時間があったので、瀧川先生のサイトにある参考情報からすごそうなプログラムをコピーして見てみたのですが、total running time 0.48 seconds と書いてあったのに、当然のように 10 秒くらいかかりました。Total running time と結果が表示される時間は別のものですか？
A. なんのプログラムか分からないけど「total running time」というのは単純にプログラムを実行して終わるまでの時間のことで、当然速いパソコンでやれば短いし、遅いパソコンでやれば長いです。このように実行時の環境によって違う情報を処理している(例えば実行した時の日付を表示、など)場合もあるし、どこで実行しても同じ(1+3 とか)の情報を処理していることもあります。

Q. anaconda と jupyter と python の関係がいまいちわかっていません。Python が言語なのはわかるんですが。
A.「anaconda」は「色々丸ごと入りパック」みたいなもんでその中に「jupyter」も「python」も他に使えそうな色々も丸ごと入っている全部入りみたいなものです。個別に個人個人で一つひとつインストールすると面倒なだけではなく結構ハマるのでこういうパックの需要があります。で、「jupyter」は「python」のプログラミングをブラウザ上でやるための編集管理ソフトみたいな感じかな？で、「python」がインタプリタ本体で「python のコード(テキストファイル)」を CPU で

処理できる 0 と 1 の命令の列に翻訳してくれるソフトウェアです。「jupyter」はブラウザに入力された文字列を受け取って、裏でこの「python」に送って、実行し、出力結果を受け取って、それを再度ウェブブラウザに表示する、というような処理をやっています。