

メッセージ：自分で新規の Jupyter Notebook を作成し、Python コードを一緒にかけるノートだと思って、自分なりの「まとめ」を作ってみて、最初はそれを見ながら作業すると定着がいいかなと思います。あとでその Jupyter Notebook ファイルをレポートとしてメール提出してもらおうと思いますので、学びの記録と思って各自作成してみてください。！

「BMI 計算」や「FizzBuzz」など「やってみよう系」は後回しで先に進んで下さい。ただし、基本的に practice01~04 は python 入門のためにはこなすことを前提としているので、分かってきたらあとでやって見て下さいね！平均的にはこのペースで良さそうなので、遅れ気味になってきたなという人や休んだり遅刻で実際かなり遅れがあれば、ぜひ宿題としてやっておいてみてください。そうしておけば授業を楽しめるようになってくるかも！？

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

- フォローアップ Q & A
- Python のここまでのまとめ
- 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
- ファイルの読み込みと書き込み

4. 参考情報

- Python 公式チュートリアル
<https://docs.python.jp/3/tutorial/>
- Markdown
<https://guides.github.com/features/mastering-markdown/>

先週の質問への回答：

Q. デバッグを繰り返していると頻繁に shutdown を繰り返さなければならないのが面倒ですが仕方ないですか？

A. エラーが出て shutdown する必要はないし、まずは停止ボタン■(メニューだと Interrupt)で十分なのですが、どういう状況だろうか？また演習で聞いて下さい！(Interrupt までなら評価済みのセルの再評価は必要ありません)(interrupt については過去の Q and A をチェックしてみてください)

Q. プログラムの文をコピーした後に、ファイルをシャットダウンし、もう一度、同じファイルを開いて、貼り付けようとしたのですが、できなかつたです。普段よく使う、コピー&ペーストはどこに保存されているのでしょうか。

A. メモリ上に保存されています。セルを選択した状態でのコピー&ペーストと、セルの中の文を選択した状態でのコピー&ペーストは挙動が異なります。前者は Jupyter の機能、後者は Windows の機能で、もしかしたら前者でコピーしたとかかも？また演習で聞いて下さい！

Q. 趣味として個人的にホームページを作りたいと思っているのですが、簡易的なものでもどれくらいの難易度なのでしょうか。

A. どのようなホームページをイメージしているかによります。簡易的なものなら、HTML という Markdown のような規則にしたがってテキストファイルに書いただけなので、例えば、以下のファイルたった一つでも十分ホームページです。下記をメモ帳に貼り付けて保存したファイルをブラウザからファイルを開くで表示してみてください。

----ここから

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>こんにちは！</title>
</head>
<body>
<h1>瀧川のホームページ</h1>
<p>初めてのホームページを作ってみました！よろしくね！</p>
</body>
</html>
```

----ここまで

以上をメモ帳で作成して、「testpage.html」で保存しただけのものを下記におきました。

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

これに写真やスタイル(CSS というルールで指定します)もつければ瀧川のホームページ

<http://art.ist.hokudai.ac.jp/~takigawa/> くらいにはなります(ただしファイル群は Adobe DreamWeaver というソフトで管理しています)。ブログなど内容や更新が多い場合は「コンテンツ管理システム(CMS)」と呼ばれるソフトウェアを使うのが人気です。プロバイダのホームページ公開サービス(無料で付いてる場合も多い)ではサポートされている場合が多いと思いますが、代表的なのが WordPress です。例えば、私の研究室のページ <http://art.ist.hokudai.ac.jp> は WordPress ベースです。また趣味の個人ページなどだと、google site や github.io のような無料のサービスを利用して作成するか、ホームページビルダーのようなホームページ作成ソフトで作成する人もいます。より商用では動画や絵や Flash などデザイン面で凝った作りをしたり、google map などをはじめとする JavaScript を使ったインタラクティブな Web サービスで実装するなど、凝りだしたら無限にやれることがあります。まずは、上のような HTML と CSS で作る入門書や Web サイトを参考に作ってみると良いと思います。入り口のハードルはプログラミングよりだいぶ低いです。

Q. ()と[]の違いがあまり分かりません。

A. ()は「タプル型」、[]はリスト型、{}は辞書型になります。

Q. 「0の0乗」を計算すると1と出ますが、高校のとき、0の0乗は計算できないと言われました。どちらが正しいのでしょうか。

A. 数学的には高校の先生が正しく、0の0は「定義できません」。「10より大きい負の整数」みたいに「ナントカ乗(微分積分で習いますが指数関数というものです)」の概念上意味をなさない、という感じです。ただ、プログラミング言語の場合は $0^{**}0$ と簡単にできてしまうので、この場合「一番利便性の高い」1を返す、プログラミング言語がほとんどです。Wikipedia に「0の0乗」というそのままズバリのエントリがあるのでみてみて下さい。

Q.

```
set1 = {'3', '6', '9', '12', '15'}
set2 = {'5', '10', '15'}
print(set1 - set2)
```

とすると、{'6', '3', '9', '12'}と表示されるのですがset1は'3', '6', ...の順番なのに、表示されたものでは'6', '3', ...の順番になっているのはなぜでしょうか。

A. 「集合」としてみれば、{1, 2, 3}も{3,1,2}も数学的に全く同じです。なので「集合」型の場合は順番を保持する必要がないので、順番を保持するように作られていません。順番は適当にリセットされます。順番を保持したければ、リストにして小さい順に並べ替えが必要です。

Q. リスト型の復習で、delを使って複数の要素を取り除こうとしたらエラーが出たのはなぜですか。例えば

```
lis0 = ['a', 'b', 'c']
del lis0[1, 2]
```

A. del関係なく、lis0[1, 2]がそもそもエラーになります。['b', 'c']を消したいなら、lis0[1:3]ですね。リストの「スライス」と言いますが、下記の例、今日の演習、過去の演習の「リスト」のところをまたcheckしてみてください。

```
lis0 = ['A', 'B', 'C']
```

```
lis0[1,2]
```

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-2-72557c0aa994> in <module>()
----> 1 lis0[1,2]
```

TypeError: list indices must be integers or slices, not tuple

```
lis0[1:2]
```

```
['B']
```

```
lis0[1:]
```

```
['B', 'C']
```

```
lis0[1:len(lis0)]
```

```
['B', 'C']
```

```
lis0[1:3]
```

```
['B', 'C']
```

```
del lis0[1:3]
print(lis0)
```

```
['A']
```

Q. USBの接続についてなのですが、▲から切るのを忘れて、だいたいパソコンの電源を切ってから取ってるのですが、あまりよろしくないのでしょうか。

A. 本当に電源が切れてるなら問題ありません。ただ、演習室のパソコンは電源切る操作が再起動なので、物理的には電源が切れないと思います。その場合はあまりよろしくないです。(あまりないとはいえ、最悪データが失われます)

Q. -inf を用いた最大値の求め方がいまいちわからなかったです。

A. 「-inf」はどんな数字と比べても小さい特殊な数字なので、まあすごい小さい数字と思えばいいです。例えば、

```
lis = [3, -1, 7, 4, 100, 13, 46]
max = -1000000
for i in lis:
    if i > max:
        max = i
print(max)
```

100

ただこれだともしリストの値が以下の感じだったら、正しい答えが求まりません。

```
lis = [-3000000, -2000000, -5000000]
max = -1000000
for i in lis:
    if i > max:
        max = i
print(max)
```

-1000000

Q. 情エレの情報理工系で行われている研究の手法はプログラミングがメインですか？

A. プログラミングはある情報処理をコンピュータでやるのに必要なだけで、それ自体は研究の手法という感じではありません。紙と鉛筆で研究している先生もいますし、プログラミングを伴わない研究もあります。がまあ、ほとんどの場合は避けては通れないですね。みなさんが授業のレポート書くのに、日本語(か英語か)何か言語が必要なのと同じ感じです。

Q. 「my_module.py」がそんなものはないと言われて結局使えませんでした。保存場所も practice03 と同じにはしましたが …。

A. 演習で聞いて下さい。そんなものはないというエラーが出るならそんなものはないのだと思います。Jupyter で作成しましたか？ダウンロードしましたか？拡張子が違うか、ファイル名のミスタイプか、practice03 をおいてるフォルダを勘違いしているか、色々な理由が考えられます。

Q. まだ「practice02」の途中でおいてかれてる感がすごいで、もう少しヒントとか欲しいです。デバッグが苦痛です。

A. 「BMI 計算」や「FizzBuzz」など「やってみよう系」はあまりに分からなかったらとりあえず後回しで、practice03 と practice04 をしてみるといいと思います。だんだん分かってくるケースも多いので。デバッグはみんな苦痛ですが、これは何か明らかに間違えていることを確実に意味しているので、自分の思考のダメさ加減を認識する体験として、プログラミングにおいて避けては通れないです。そうやって簡単そうなことでも最初はミスをしながら、えっちらこっちら頑張っていれば、だんだんプログラミングに必要な思考パターンや法則が身につくし、そもそも慣れてきます。

Q. BMI の計算がよくわかりません。Practice01 が終わることなく 2 をやっているんですが大丈夫ですか？

A. BMI は回答が practice03 に載っているので、分からなかったら practice02 の「やってみよう」系は飛ばして進むのも大事です。ただ、最終的には全てこなすことを想定しているので一旦先に進んで慣れてきたら、分からなかったら、その都度、後戻りして確認する方がいいと思います。本当にわからなかったら Web を調べるとか、本を調べるとか、対処法を身につけていくことも大事です。全く何も仮定しないこの演習 16 コマだけではどっちみち Python の全ては全然カバーできないので、むしろ今後興味がある人が自分で学んでいけるようになるのが最も大切です。自転車の運転や、水泳や、英会話など、理屈でわかってても自分で練習しないと身につかない類のスキルだと思って下さい。

Q. 実際に瀧川先生が頼まれてプログラムを作ったとおっしゃっていましたが、実際に作ったもので、どんなものがありましたか。もしできるならみてみたいです。

A. python じゃないのでコードは出しませんが、例えば、基本英単語 12000 個 <https://www.alc.co.jp/vocgram/article/svl/> を覚えたくて、覚えたのを消して行って、覚えてないのから間違い率が高いのをランダムに出すみたいな、英単語覚えるプログラムを作るとか、天気などの情報をネットから取得して表示するプログラムとか、体重やランチを管理するとか、まあ、日常的な用途でも結構使えるプログラムは作れると思います。これを Web 出力で実装すればスマホアプリってことになりますね。これも python じゃないけど「テニスで英単語学習 ~TOEST~」という Android アプリは何年前に、3年

生のグループ実習の授業で私が面倒見たチームで授業中に作成したものです。で、Python でもこういうのは作れますが、面白い実例があんまり手元にないので、実用レベルのコードがどういうものかということでは、今日の practice04 の最後で触れたように、github とかでいろんなフリーのプログラムのソースコードを読むのが勉強(と刺激)になると思います。

Q. タプルはリストと比べて何が良いのですか？変えられないのはただ不便なのでは？

A. これは「非常に」良い質問ですね。初心者は一見、タプルなんて要らないのでは、と思うかもしれませんが、「変えられないこと」が必要な局面もあるのです。

その1：リストだとプログラムが複雑になると意図せず事故で内容を書き換えてしまうことが起こり得ます。で、こうしたバグは非常に特定がツライです。なので、ファイル名とか、パスワードとか、確実に処理の途中で変更されないものを保持する場合は、「変えられない」仕様になっている構造で保持しておくのがいいです。

その2：辞書のキーに使う場合、キーは途中で変わると引けなくなるので、変えられない型しか許されていません。したがって、リストでは辞書のキーにできません。以下の例を見て下さい。

```
t1 = ['takigawa', 'ichigaku']
t2 = ['hokudai', 'taro']
dic = {}
dic[t1] = 'sapporo'
dic[t2] = 'otaru'
```

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-11-e337d4992e6e> in <module>()
      2 t2 = ['hokudai', 'taro']
      3 dic = {}
----> 4 dic[t1] = 'sapporo'
      5 dic[t2] = 'otaru'
```

TypeError: unhashable type: 'list'

```
t1 = ('takigawa', 'ichigaku')
t2 = ('hokudai', 'taro')
dic = {}
dic[t1] = 'sapporo'
dic[t2] = 'otaru'

dic[('hokudai', 'taro')]
```

'otaru'