Pythonプログラミング入門

2018年3月10日（土）第１回～第５回
講師：渡邉 貴之

概要

- Pythonについて・・・
  - コードが簡潔に記述できて読みやすいなどの特徴があり、様々な分野で人気を得ています。
  - 最近は機械学習やディープラーニングといった人工知能やデータサイエンスの分野においても、豊富な追加モジュールが揃っていることから、急速に人気が高まっています。

- 本講座では・・・
  - Pythonに初めて触るプログラミング初学者の方を対象として、プログラミングの初歩から講義と演習を行います。
    - 変数やリスト、タプル、辞書といったデータ型
    - 関数の概念
    - 制御構文
    - クラスとオブジェクト
    - 簡単なデータの集計やグラフ作成など

- 参考図書
  - 柴田, "みんなのPython第4版", SBクリエイティブ株式会社

本日の内容

- タイムテーブル

第1回	**13:00 ～ 13:45**
	Pythonの特徴とプログラミング環境の確認
第2回	**13:45 ～ 14:30**
	変数とデータ型
第3回	**14:30 ～ 15:15**
	制御構文
第4回	**15:15 ～ 16:00**
	関数
第5回	**16:00 ～ 16:45**
	内包表記

※途中、休憩時間を適宜取ります

実習室利用ガイダンス

- おねがい
  - 教育・研究のための利用に限ります。
  - 室内での飲食は禁止されています。

- コンピュータの起動とログオン
  - 別紙「講座用ユーザIDとパスワード」をお手元にご用意下さい。
  - コンピュータの電源を入れ、ログオン画面にてCtrl+Alt+Deleteを押し、「ユーザーの切り替え」「他のユーザー」をクリックします。
  - ログオン用IDとパスワードを入力し、矢印ボタンをクリックします。
  - ログイン後、中央モニタに講師のプロジェクタと同じ画面が映ります

その他

- 自販機
  - 1Fの階段横にあります。
- 喫煙場所
  - 棟内は全面禁煙です。
  - 棟外の灰皿のある場所のみ喫煙できます。
    - 経営情報学部棟を出て看護学部の入り口から少し下った場所にある階段を降りた先です。
- 食堂
  - 春休み期間中につき休業しています。
  - お弁当をお持ちの方は3F地域経営研究センターまたは、1F吹き抜けのホールをご利用ください。

第1回
Pythonの特徴と
プログラミング環境の確認

目次

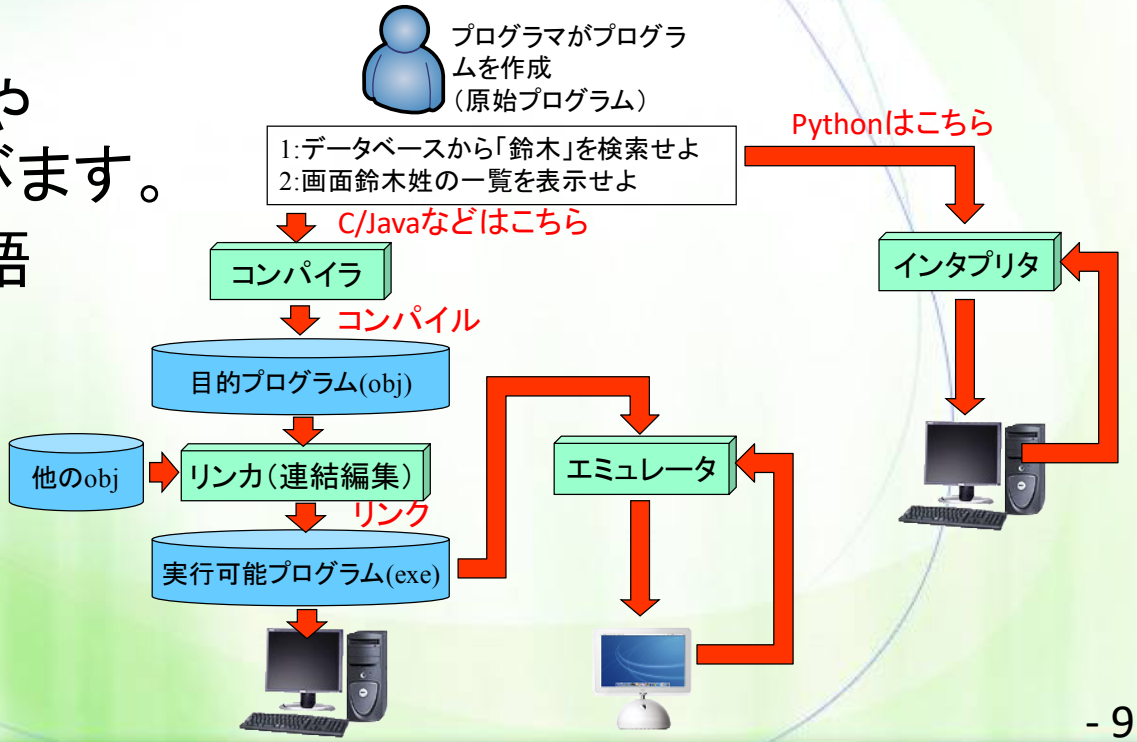
Pythonの歴史

- グイド・ヴァンロッサム氏
  - オランダ国立情報工学・数学研究所に勤めていた1989年末のクリスマス休暇中に暇つぶしに作り始めた言語
  - 1991年に公開したところ、設計のよさから開発者の間で話題に
  - 1994年にバージョン1.0を公開
  - 2000年にバージョン2.0を公開
  - 2005年、Googleに移籍
    - 勤務時間の50%をPythonの開発に費やす契約を結ぶ
  - 2008年にバージョン3.0を公開
  - 2013年、Dropboxに移籍
    - 現在もPythonソフトウェア財団の一員としてPythonの開発を継続

Guido van Rossum OSCON 2006 cropped.png

Pythonの特徴（１）

- すぐに実行できる
  - 人間が記述したプログラムのファイルを、コンピュータが実行できる「機械語」のファイルに変換する処理をコンパイルやリンクと言いますが、Pythonはそれらの前処理が不要です。
  - Pythonのプログラムを、Pythonの実行環境（インタプリタ）に読み込ませると、すぐに1行1行を解釈して実行してくれます。
  - このような言語を「スクリプト言語」や「軽量言語」と呼びます。
  - 他のスクリプト言語
    - Perl、PHP、Ruby、JavaScriptなど



Pythonの特徴（2）

- 実行環境を選ばない
  - Pythonのプログラムは、Windows、macOS、Unix/Linuxや携帯機器でも動かすことができます。
    - パソコンからサーバまで実行環境を選びません。
  - 私たちが普段使用している様々なWebサービスもPythonで記述されています。
    - YouTube、Instagram、Pinterest、Dropboxなど

  - ビッグデータ分析、人工知能（機械学習、ディープラーニング）など先端分野でも活用されています。
    - 先端分野の機能を手軽に実行できる追加機能（モジュール）が豊富に提供されていることから人気が高まっています。
    - ソフトバンクのロボット「Pepper」のソフト開発にも採用されています。

Pythonの特徴（3）

- 読み書きしやすい
  - プログラムで使う命令のうち、基本となるものを予約語といいます（日本語に例えると「てにをは」）。Pythonは予約語の数が少ないので、覚えやすいと言えます。
  - 書き方に曖昧さを許さない設計
    - 誰が書いても同じような書き方になる
      - 他の人が書いたプログラムが読みやすくなる（可読性が高くなる）
      - 自分が書いたプログラムも読みやすくなるので、間違いを見つけやすくなる

言語	予約語数
Python3	33
ruby	41
C	44
Java	52
swift	61
JavaScript	61
C++	86
Perl	220

- その他名前の由来
  - Python（ニシキヘビ）は、グイド氏が好きだったイギリスのテレビ局 BBC が製作したコメディ番組「空飛ぶモンティ・パイソン」から
  - ロゴはニシキヘビがモチーフ

Pythonの環境を用意する

- Pythonの入手方法
  - 日本Pythonユーザ会のサイトから入手できます。
    - https://www.python.jpを開く
    - 「環境構築ガイド」を開く
    - インストールしたい環境を選択して表示された手順に従ってインストールを進める

Anacondaとは？

- データサイエンス向けに作成されたPythonパッケージ
  - データサイエンス用の多くのモジュールが標準でインストールされるのでとても便利です。

環境構築ガイド 〉 Anaconda

Anaconda

Anaconda ¦ Windows 版のインストール ¦ MacOS 版のインストール ¦ Unix版 のインストール ¦ Conda コマンド

Anaconda はデータサイエンス向けに作成された Pythonパッケージで、科学技術計算などを中心とした数多くのモジュールやツールが独自の形式で同梱されています。

macOSやUnix環境では、ほとんどのモジュールがコンパイル済みパッケージを提供しているため、Anaconda を使わなくとも、通常の pip コマンドでも簡単に環境を構築できます。

しかし、Windows環境のように、簡単にインストール可能なモジュールが提供されていない環境で、機械学習などのためにPython を使用するなら、多くのモジュールがデフォルトでインストールされる Anaconda はとても便利です。

また、Anaconda でインストールされるモジュールは、Intel Math Kernel Library を使ってビルドされるなどの、特定用途向けパッケージならではの工夫が加えられている場合もあります。

さらにここから好みの環境を選ぶとインストールの手順が表示される

Pythonプログラミングの基本

- プログラムとは？
  - コンピュータに何かさせたいことを、文字や記号を組み合わせた命令として書いたものです。
- プログラムを書くルール
  - 書き方には文法や構文と呼ばれるルールが決まっています。
- ソースコードとは？
  - プログラム言語の持つ文法に従って書かれた命令の集合体をソースコードやコード、ソースファイルなどと呼びます。

```
for i in range(1, 10):
    for j in range(1, 10):
        print(str(i * j) + "¥t", end="")
    print("¥n")
```

九九の表を出力する例

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

Pythonコードを実行する3つの手段

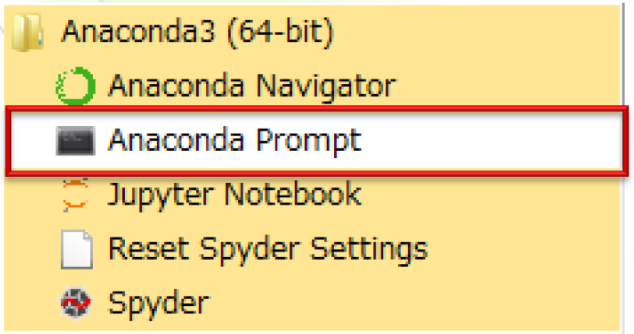
方法	概要	利点	欠点
インタラクティブシェルで実行	Pythonのプログラムをちょっと試したい時に便利な手段。Pythonと対話をしながらプログラムを実行できる。	ソースファイルを保存する必要がない。	大きなプログラムや繰り返し実行したいプログラムの実行には不向き。
ファイルに保存してインタプリタで実行	あらかじめ実行したい命令群をソースファイルに記述して保存しておき、pythonコマンド等で一気に実行する方法。	大きなプログラムや繰り返し実行したいプログラムは、ファイルに書いて保存して置いた方が便利。	ちょっとした処理を試したい場合はファイルを保存するのが面倒。実行するまでエラーがどこにあるかわからない。
Jupyter Notebook（http://jupyter.org/try）	Webブラウザを使ってソースを入力し、Webブラウザで結果を表示できるしくみ。計算はネットワーク上のサーバで行う。	インタラクティブシェルのようにファイルを保存する必要がなく、なおかつ自動的に保存されるので、繰り返して実行が可能。他者とのコードの共有も可能。	不正アクセス（ネットワークセキュリティ）に配慮する必要がある。

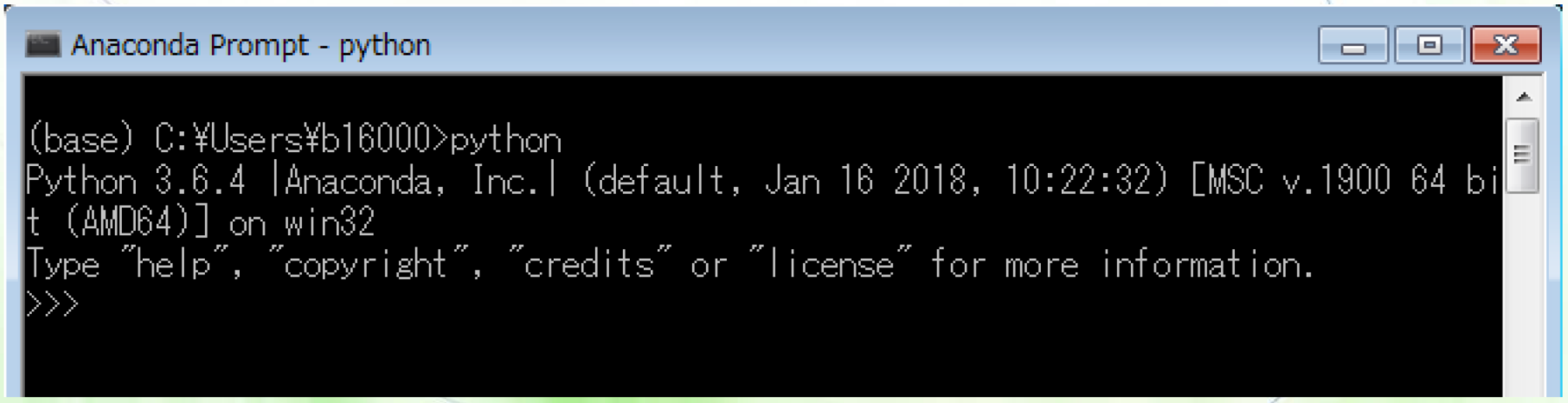
インタラクティブシェルを試す

- インタラクティブシェルの起動
  - [スタート]-[すべてのプログラム]-[Anaconda3]を開く
  - Anaconda Promptを起動する
  - 起動したPromptのウインドウに、

    Python [Enterキー]

    と打ち込むとインタラクティブシェルが起動する





プロンプトとコード入力

- プロンプトとは？
  - 不等号が3つ並んだ「>>>」という文字をプロンプトと呼びます。プロンプトは入力を待っている状態になります。
  - 従って、サンプルコードの先頭に>>>と書いてある場合は、自分で入力する必要はありません。

- 簡単なコード入力
  - print()という命令を使って文字列を表示してみます。
    - キーボードからprint("Hello Python!")と入力して改行します。
    - すると、""で囲まれた文字列がすぐ下に表示されます。プロンプト無しの行は、Pythonからの応答です。そして再びプロンプトが表示されて、入力待ちの状態になります。

```
>>> print("Hello Python!")
Hello Python!
>>>
```

簡単な計算を試す

- Pythonで四則演算を試す
  - Pythonに計算をさせたい場合は、演算子を使って計算式を記述します。演算子の例を下記に示します。
    - 加算は＋　　減算は－　　乗算は＊　　除算は／
  - 計算式をprint()命令のカッコの中に記述することで、結果がすぐ下に表示されます。
    - 計算式を""で囲ってしまうと、文字列として扱われてしまうので、計算が行われず、計算式がそのまま出力されてしまいます。
    - インタラクティブシェルでは、print()命令を省略しても結果が表示されます（ファイルにプログラムを保存する方法では省略できないので注意）。

```
>>> print("6 * 8 - 2")
6 * 8 - 2
>>> print(6 * 8 - 2)
46
>>> 6 * 8 - 2
46
>>>
```

タートルグラフィクス

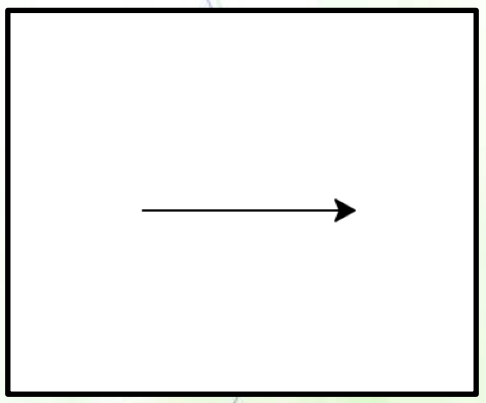
- 一筆書きで図形を描く「タートルグラフィクス」を使ってPythonに慣れましょう。
  - 以下、英数字は全て半角小文字で、日本語は使わないで入力してください。
  - エラーが表示された場合は、もう一度よく確認しながら打ち直してください。

  - turtleモジュールを使用できるように読み込む
    - タートルグラフィクスの命令は標準では使用できないので、事前にimportする（読み込む）必要があります。

```
>>> from turtle import *
```

  - 右に100px線を引く

```
>>> forward(100)
```



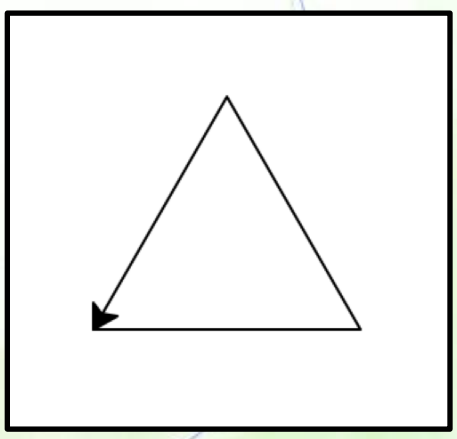
タートルグラフィクスを表示するウインドウが開く

関数と引数

- foward()のような命令のことを関数と呼びます。
- 関数のカッコの中にあるデータを引数と呼びます。
  - 関数に引き渡す数という意味です。
  - 関数名（引数）のように命令を記述します。
  - foward関数では、引数を変えると線の長さが変わります。

- 引き続き、正三角形を描いてみる
  - left関数は、線を描く方向を指定の角度まで反時計回りに回転させる関数です。

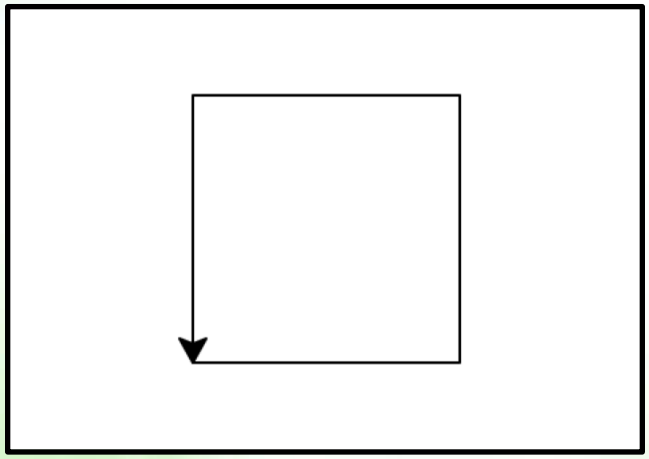
```
>>> left(120)
>>> forward(100)
>>> left(120)
>>> forward(100)
```



課題1-1

- 正方形を描いてみる
  - 一旦、これまで描いた図形を消去して、forward関数とleft関数を使って正方形を描いてみてください。
  - 一辺の長さは自由です。
  - 図形の消去には、reset関数を使います。reset関数では、カッコの中には何も指定しません（引数なし）。

```
>>> reset()
```

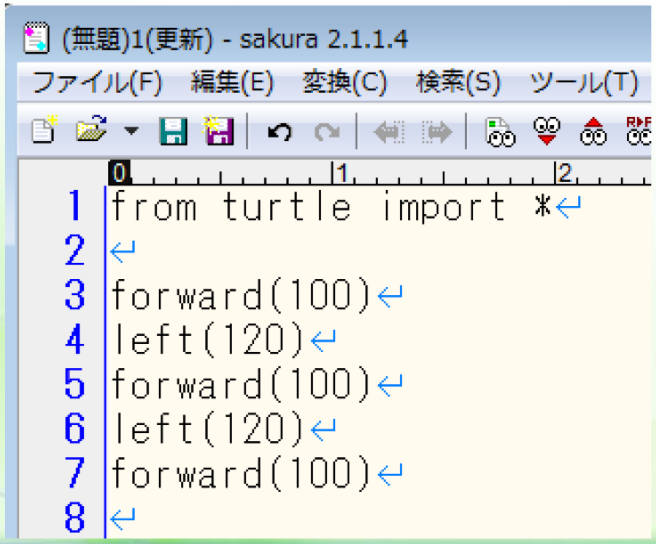


- インタラクティブシェルを終了する
  - インタラクティブシェルのウインドウ内で、Ctrl+z（Ctrlキーを押しながらzキーを押す）を2回入力して、Enterキーを押すと終了できます。

ファイルに保存して実行を試す

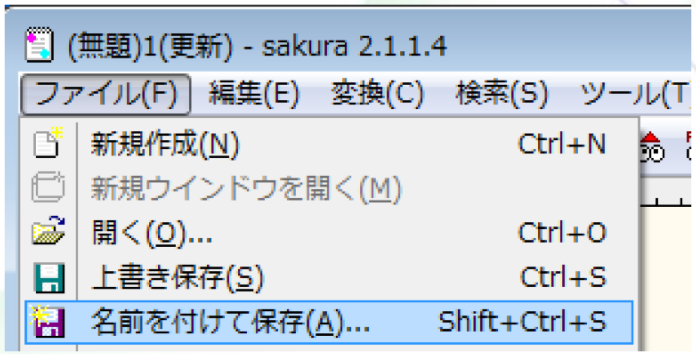
- 次に、ソースコードをファイルに保存してインタプリタで実行する方法を試しましょう。
  - ソースコードをファイルとして保存するには、テキストエディタと呼ばれるアプリが必要です。
    - Windowsにはメモ帳というエディタが標準で用意されていますが、より高機能なサクラエディタを使います。サクラエディタは無料で配布されています。
- 手順
  - デスクトップのサクラエディタのアイコンを起動します。

  - ソースコードを打ち込みます。この方法では、プロンプトは表示されないので、下記のように全ての命令を打ち込みます。

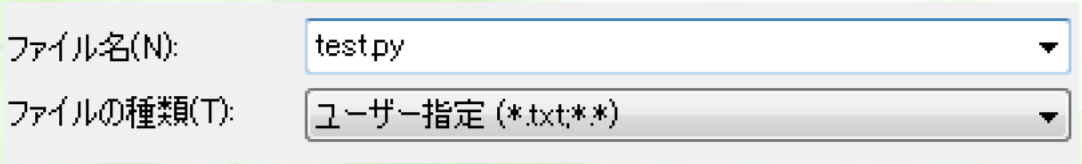


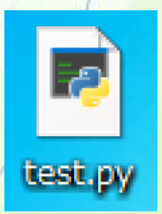
インタプリタで実行する(1)

- ファイルの保存
  - サクラエディタのメニューから[ファイル]-[名前を付けて保存]を選択します。



  - ファイルを保存する場所は「デスクトップ」、ファイル名として半角で「test.py」と付けます。お尻の「.py」は、このファイルがPythonのソースファイルであることを表しています。「test」の部分は、自由に命名できますが、今回は統一で「test」としました。
  - 問題なければ「保存」ボタンをクリックします。
  - デスクトップに保存されたファイルのアイコンが表示されるので確認します。



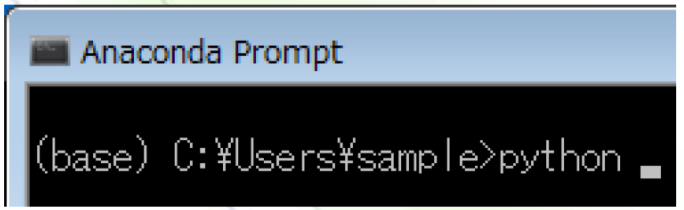


インタプリタで実行する(2)

- デスクトップのtest.pyのアイコンをダブルクリックして実行します。
- 実行できない場合は下記を試します。
  - Anaconda Promptのウインドウにあらかじめ

    python [半角スペース]

    と打ち込んでおきます(Enterキーはまだ押さないでください)。



  - デスクトップに保存したtest.pyのアイコンをAnaconda Promptのウインドウにドラッグ&ドロップします。
  - Enterキーを押すと、実行できます。

- 実行後に画面が消えてすぐ終了してしまうことがわかると思います。
- 右の8行目のように、最後にinput関数を記述しておくと、ユーザが何か文字を入力するまで画面は消えません。
- サクラエディタを修正して上書き保存して試してみてください。

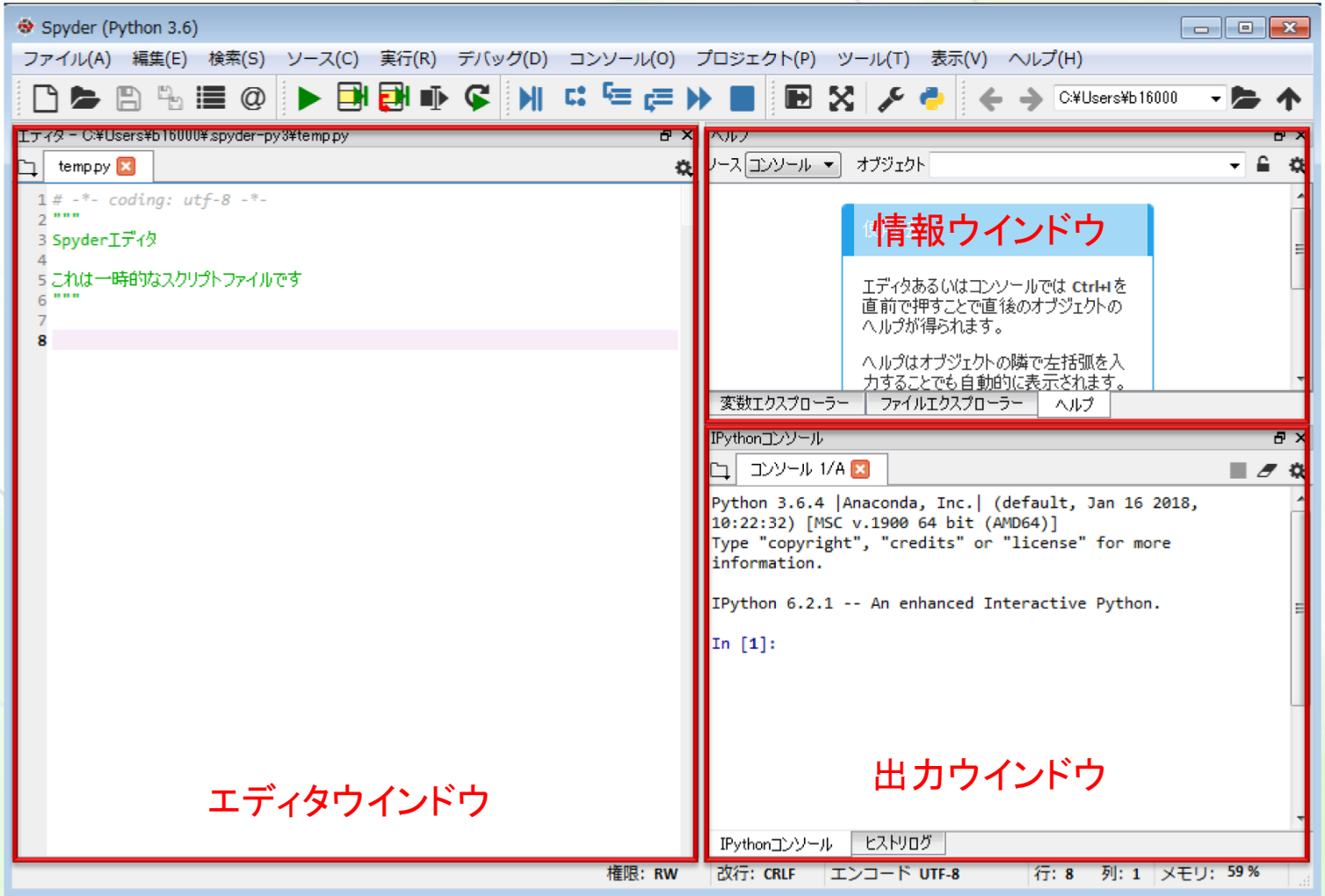
```
1 from turtle import *
2
3 forward(100)
4 left(120)
5 forward(100)
6 left(120)
7 forward(100)
8 input()
```

統合開発環境を使用する

- 統合開発環境とは？
  - IDE（Integrated Development Environment）とも呼ばれる、プログラム開発のための専用ソフトのことです。
  - メモ帳やサクラエディタなどの単なるテキストエディタを用いた開発では、エディタとインタプリタが別れているので保存から実行までが面倒でした。IDEを使えば、エディタとインタプリタが統合されているので、保存したらボタン1つで実行できます。
  - IDEでは、プログラムが楽になる様々な仕掛け（支援機能）が用意されています。
  - Pythonの統合開発環境
    - PyScripter
    - Eclipse + Pythonプラグイン
    - Spyder
    - ・・・

Spyderを起動する

- [スタート]-[すべてのプログラム]-[Anaconda3]を開く
- Spyderを起動する



簡単なコードを記述して実行する

- 下記のような2行のprint関数を記述し保存、実行する。
  - 右下の出力ウインドウ（コンソール）の結果を確認する。