

The Kyoto College of
Graduate Studies
for Informatics

kcg.edu

コンピュータプログラミング概論 (a)

Fundamentals of Computer Programming

2021年秋第1回 eラーニング資料

安 平勲

h_an@kcg.ac.jp

プログラム, プログラミング言語, Pythonとは？

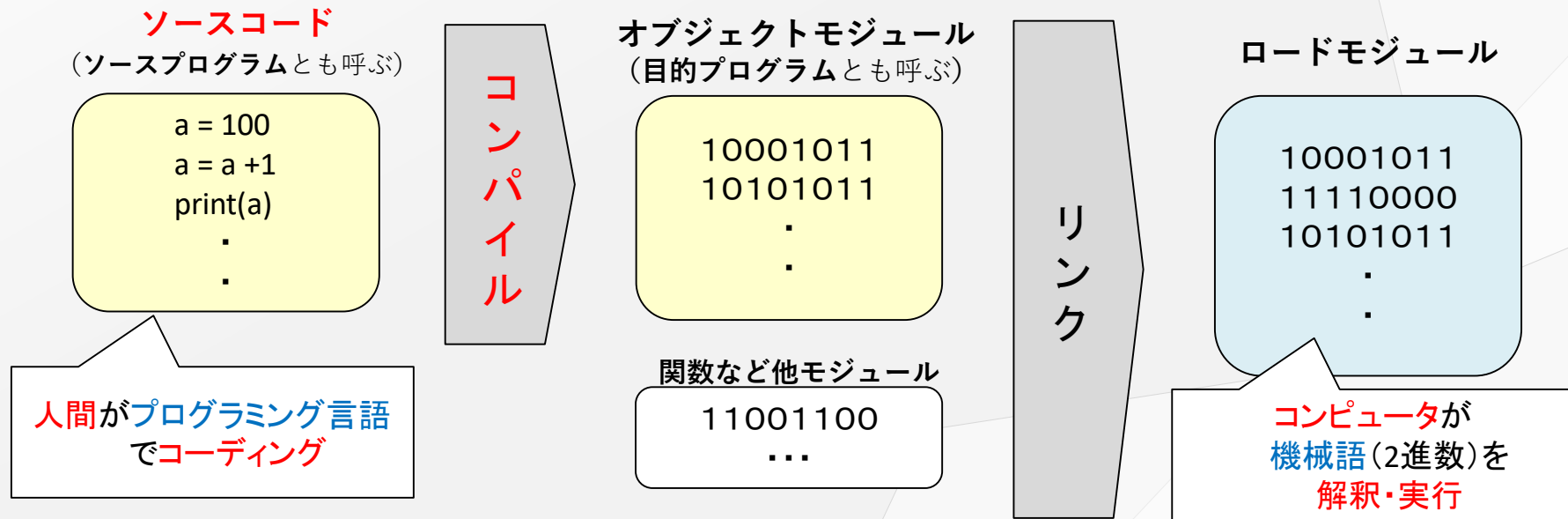
プログラム，プログラミングとは？

■ コンピュータプログラムの定義例（ASCII.jpデジタル用語辞典より）

1. **コンピュータが解釈・動作できるデータ**。コンピュータに対する命令をプログラムとして記述すると，コンピュータはプログラムに指示された手順で計算，入出力などの処理を実行していく
2. 通常は**プログラミング言語**でソースコードを記述し，これをCPUが理解できる機械語に変換（コンパイル）してから実行させる
3. プログラミング言語や中間言語で記述されたプログラムを機械語に変換しながら実行する，**インタプリタ方式**もある

プログラム、プログラミングとは？

- コンピュータは2進数の機械語で動く。2進数の機械語は人間には難しいので、**プログラミング言語**で**ソースコード**を書き、それを2進数の機械語（**ロードモジュール**）に変換（**コンパイル・リンク**）する



プログラム，プログラミングとは？

- **コンパイル方式**とは別に，プログラミング言語や中間言語で記述されたプログラムを1行ずつ**機械語に変換しながら（翻訳）**実行する，**インタプリタ方式**もある

ソースコード

```
a = 100  
a = a + 1  
print(a)  
.  
.
```

人間がプログラミング言語
でコーディング

翻訳 : a=100

翻訳 : a=a+1

翻訳 : Print(a)

.

.

(Python等) インタプリタ

```
10001011  
11110000  
10101011  
.  
.
```

インタプリタが
ソースコードを1行ずつ
翻訳(変換・実行)

コンパイル式 対 インタプリタ式

- インタプリタ式言語では動作（実行）時、インタプリタ*が必須

コンパイル式言語

ロードモジュール

オペレーティングシステム (Win, Mac...)

ハードウェア (Intel, AMD ...)

インタプリタ式言語

pythonソースコード

pythonインタプリタ*

オペレーティングシステム (Win , Mac...)

ハードウェア (Intel, AMD ...)

* : JavaならJVM (Java Virtual Machine) がインタプリタに相当
JavaScriptならブラウザ (Google ChromeやMS Edge) がインタプリタに相当

代表的なプログラミング言語（1 / 2）

■ C 言語

- 豊富な演算子やデータ型，制御構造をもち，構造化プログラミングに適する
- C++, C# はC 言語にオブジェクト指向性を追加した言語

■ Java

- オブジェクト指向性を備え，構造化プログラミングに適する。また，プラットフォーム・フリー（無依存）で人気

■ COBOL

- 汎用機のプログラム言語として，企業の業務システムに広く使われた

■ FORTRAN

- 汎用機での科学技術計算を目的としたプログラミング言語。今は Matlab, Rが使われる

代表的なプログラミング言語（2 / 2）

■ Perl, Ruby, PHP

- 主にWebアプリケーションのサーバーサイド開発に利用される，インタプリタ式言語

■ JavaScript


- 主にWebアプリケーションのクライアントサイド開発に利用される，オブジェクト指向のプログラム言語

■ Visual Basic

- 「フォーム」と呼ばれるウィンドウに部品を張り付け，その設定や部品間の関係を指定することでWindowsのGUIアプリを開発する言語。当初はインタプリタ式だったが，後にコンパイル式もサポート

プログラミング言語いろいろ

コンパイル式言語	インタプリタ式言語
C, C++, C# COBOL Swift . .	Python Javascript, PHP, Perl, Ruby . .

 Javaはコンパイル式とインタプリタ式の両方の性質を持つ

Pythonはインタプリタ式言語

■ インタプリタ式言語のメリット

- a. コンパイルとリンク作業が不要 ⇒ すぐ動かせる
- b. 会話型処理ができる ⇒ 1行ずつ動かせる
- c. オペレーティングシステム（Win, Mac, Linuxなど。以降OSと呼ぶ）に依存せず動かせる

※コンパイル式の場合，OSが異なる（例えば
Unix⇒Linux移植）と再コンパイル等が必要

■ インタプリタ式言語のデメリット

- a. コンパイル式より処理速度が遅い

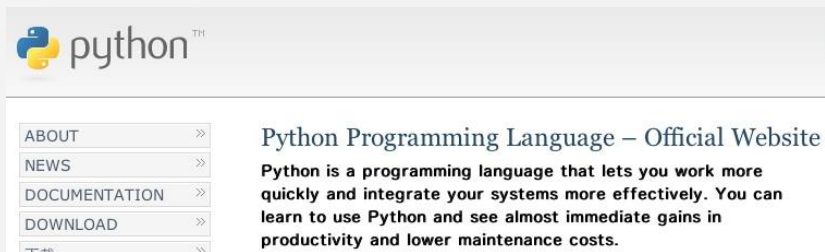
Pythonはどんなプログラミング言語か

- Python : 蛇 (へび) の一種



- Guido van Rossum (現マイクロソフト社) を中心とする技術者コミュニティがオープンソース方式で開発している
 - 1991年に公開 ⇒ 現在 ver.3.9

公式 ⇒
サイト




<http://www.python.org/>

Guido ⇒
van Rossum




Pythonの特徴

- a. オブジェクト指向のスクリプト言語
- b. Windows, Mac, Linux, iOS, Androidなど様々な環境で動く (platform-free, multi-platform)
- c. 学習コストが低い。ソースコードが読みやすい
 - ・文法が簡単。但し、インデント（字下げ）が必要
- d. 色々なライブラリが提供されている
 - ・プログラミング初等教育から商用・学術用アプリケーション（特に人工知能）まで、用途が幅広い

 基本情報技術者試験は2020年春から、COBOLがPythonに…

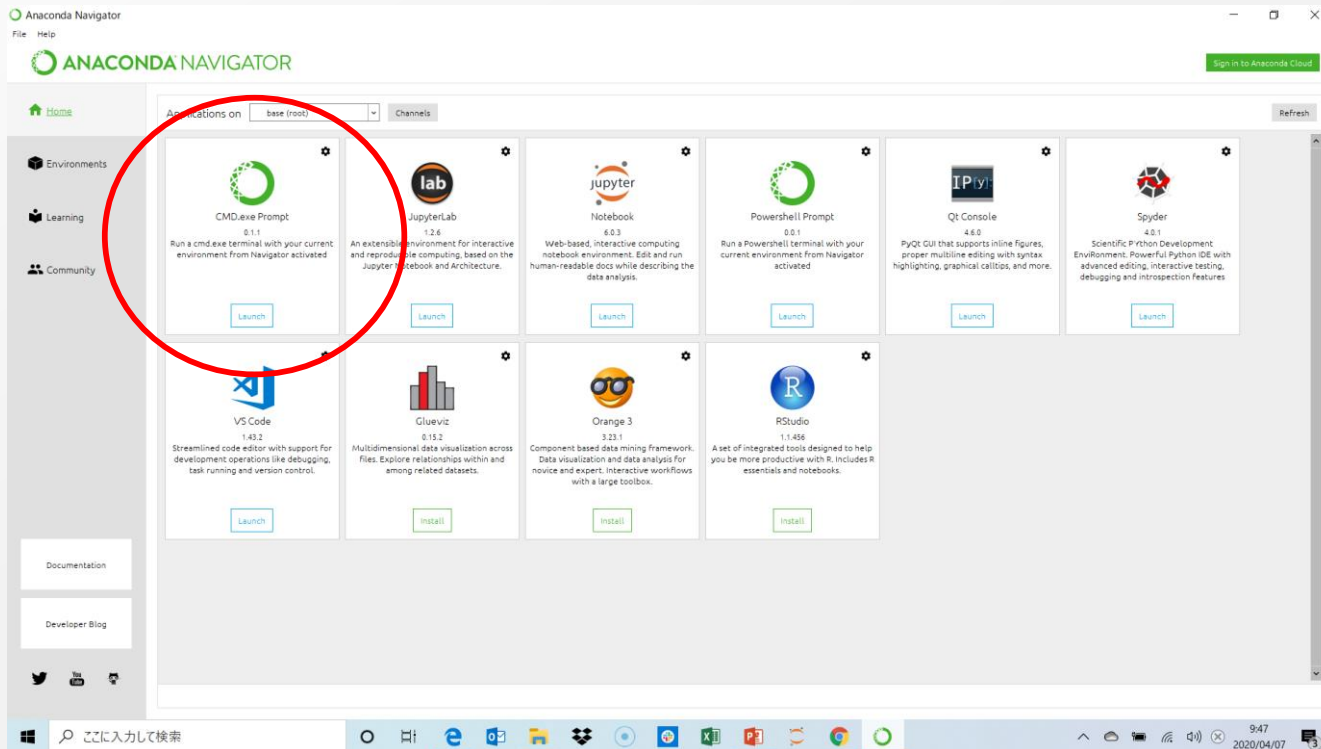
Pythonを会話型（インタラクティブ）で動作

- Windows ; コマンドプロンプトを起動
- Mac ; ターミナルを起動
(アプリケーション⇒ユーティリティ⇒ターミナル)

 WindowsではAnaconda NavigatorのCMD EXE Promptが便利

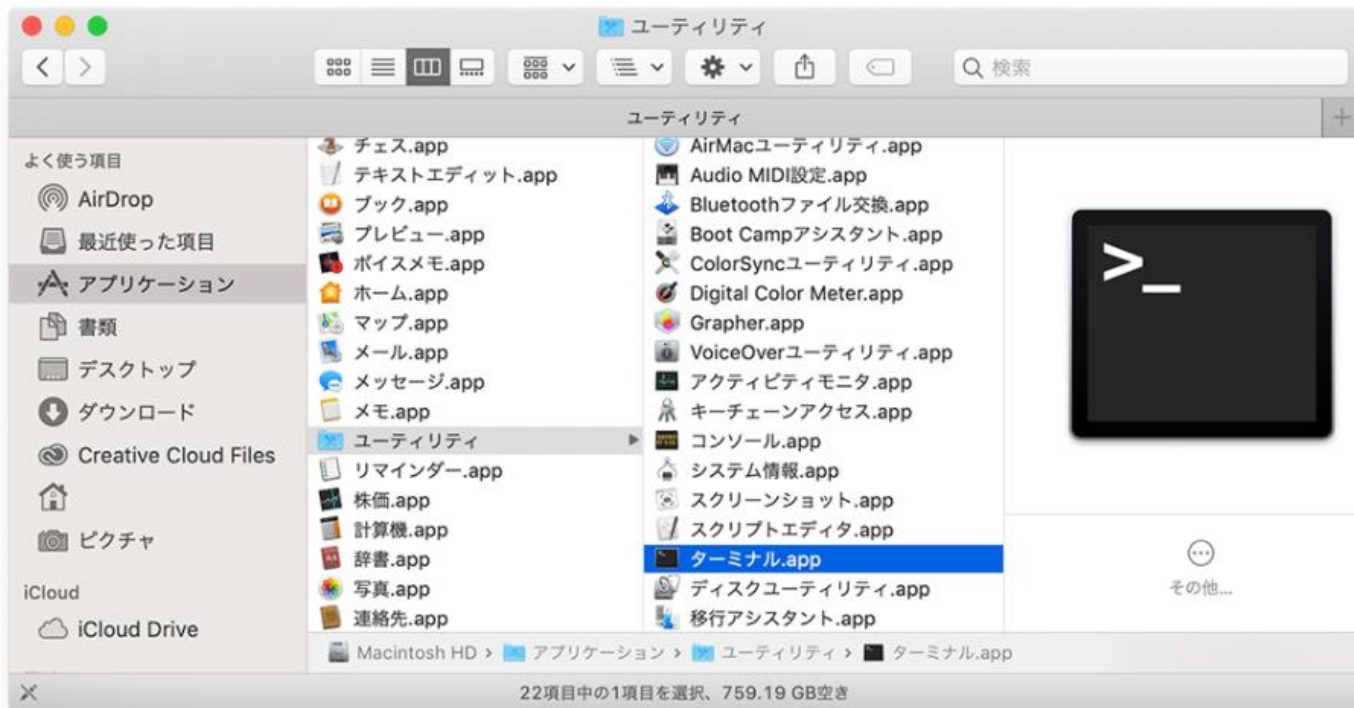
Pythonを会話型で動作（以下はWindows）

CMD EXE PromptをLaunch



Pythonを会話型で動作（以下はMac）

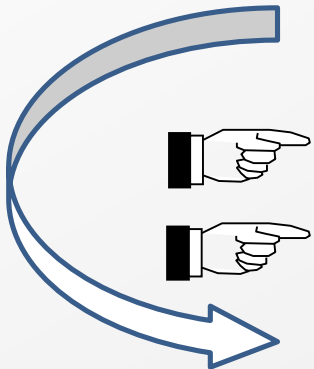
Mac のターミナルは「アプリケーション」→「ユーティリティ」にあります。



Pythonの起動と文字列の表示

- コマンドプロンプト（ターミナル）に **python** と入力

“python”で起動



“exit()”で終了

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18363.720]
(c) 2019 Microsoft Corporation. All rights reserved.

(base) C:\Users\KCG>python
Python 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64-bit]
Type "help", "copyright", "credits" or "license()" for more
>>> 'hello world'
'hello world'
>>> 'こんにちは！私は安です'
'こんにちは！私は安です'
>>> exit()
(base) C:\Users\KCG>
```



文字列は半角の “ ” か “ ” でくくる。文字列は全角文字でもOK

会話型(インタラクティブ)で四則演算

- 加算 (+)、減算 (-)、乗算 (*)、除算 (/) ができる
 - ・ 数字および演算子 (+, -, *, /) は半角

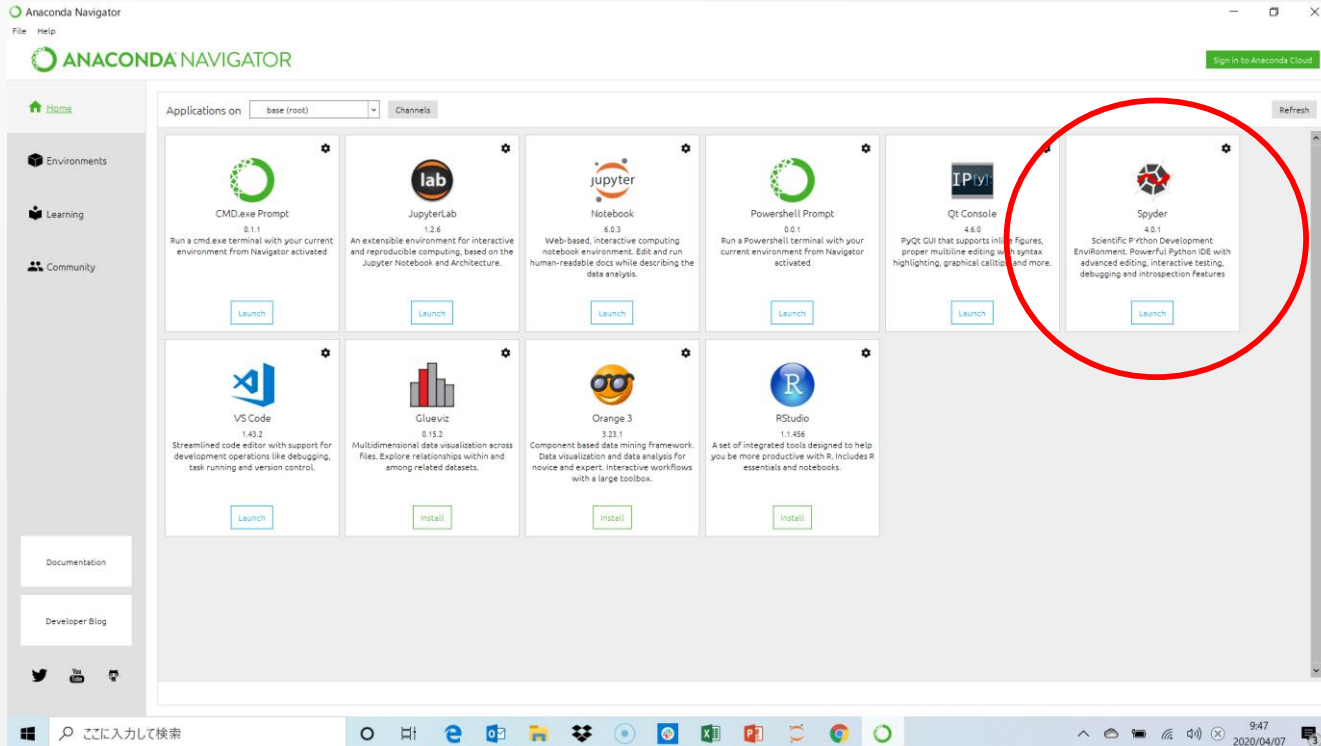
```
C:\Windows\system32\cmd.exe - python
>>> 10 + 3
13
>>> 10 - 3
7
>>> 10 * 3
30
>>> 10 / 3
3.3333333333333335
>>> 10 // 3
3
>>> 10 * 3 + 100
130
>>> 10 * (3 + 100)
1030
>>> _
```

- ✓ //は余りを切り捨ての除算
- ✓ 演算順序は乗除算 (*, /) が加減算 (+, -) に優先。()内がさらに優先

PythonをSpyder（プログラム・ファイル）で動作

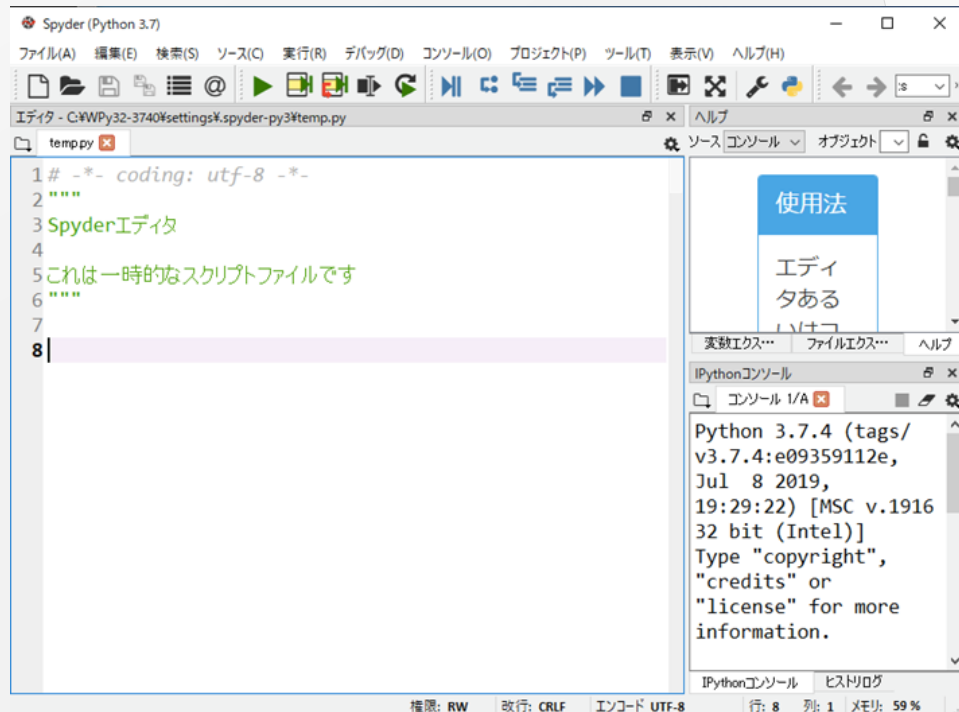
PythonをSpyderで動作

■ SpyderをLaunch (以下はWindows)

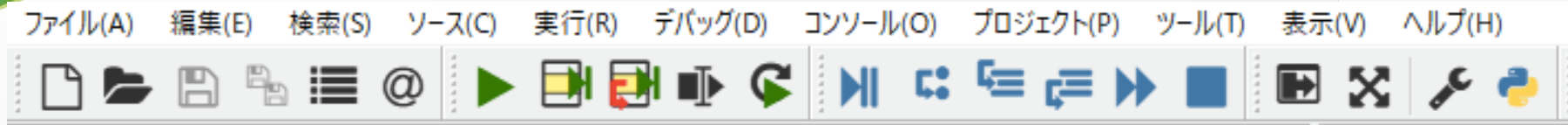


Spyderを試す

- Spyderは、**Pythonプログラムの編集→保存→実行**をまとめて行うための**統合開発環境**



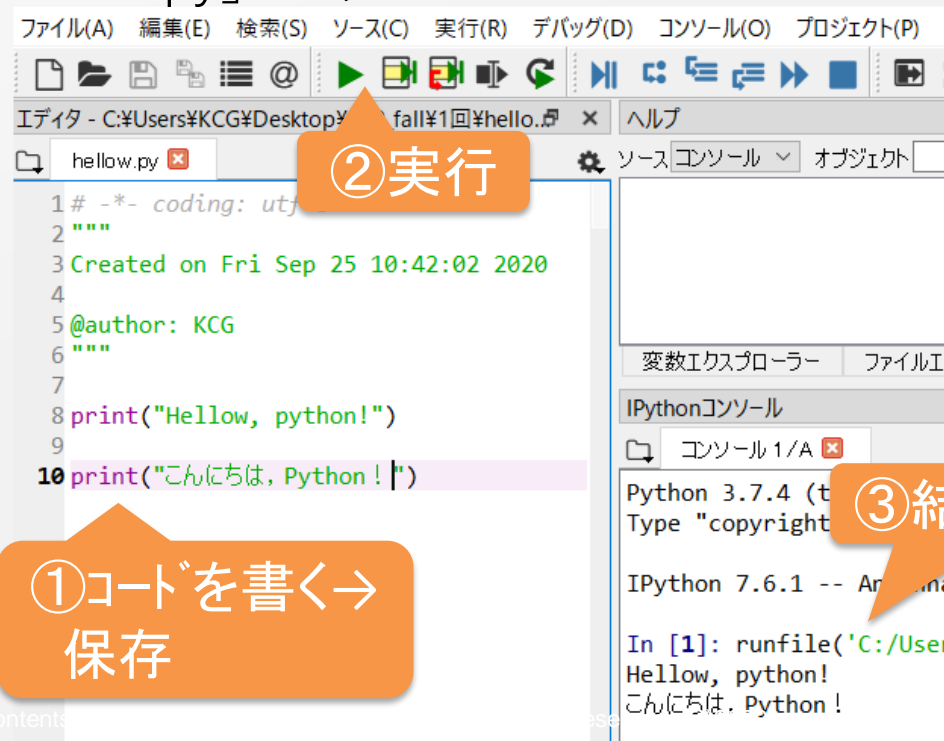
Spyderを試す



- ファイル→新規ファイル... で新規**プログラム**作成
 - Ctrl + N でも同じ
 - **ファイルの拡張子は「*.py」**
- ファイル→開く... で既存**プログラム**を開く
- ファイル→保存... で**プログラム**を保存
 - Ctrl + S でも同じ
- 実行→実行 で編集集中の**プログラム**を実行する
 - 実行結果は右下の「**コンソール**」に**表示**される

まずは Hellow, Python!

- 文字列を出力する**プログラム**を書いて実行しよう
- ファイル名は「hellow.py」にする



コンソール表示の関数

- **print()** ; コンソール (Macではターミナル) に文字列を表示する組み込み**関数**function

print("出力する文字列")

関数の名前

引数;関数の実行時に使うデータ

- 関数の引数argumentの括弧()は省略不可
- 文字列は引用符 (" " または ' ') でくくる

練習1-1

- 以下の文字列をコンソールに出力する**プログラム**をSpyderで書いて、実行してください
 - ファイル名は「print_test.py」にする

Python programing

2021.10.5

今日は 10月 5日です

練習1-2

- 以下の四則演算の結果をコンソールに出力する**プログラム**をSpyderで書いて、実行してください
 - ファイル名は「cal_test.py」にする

10 + 3

10 - 3

10 * 3

10 / 3

注) 文字列ではない。print関数の引数の引用符は不要

Pythonのデータ型

- データには型がある

Pythonのデータ型 (data type)

文字列(string)		“kcgi”, '京都', “123” など (“ または ' で囲む)
数値	整数 (integer)	123, -3, 0 など
	浮動小数点(float)	123.0, 3.14, 0.0など
真偽値(boolean)		True または False

※ 数値の **123** と文字列の **"123"**は別物