1 はじめに

このファイルではpython3をとりあえず動かして, 動作のための基礎を学んでいきます.

2 目次

3 pythonのファイル

プログラミング言語を書くには, テキストエディタでファイルの中にプログラムを書く必要があります.

pythonのファイルであることを示す拡張子は `.py` が利用されています.
そのため, pythonのファイルは `ファイル名.py` というようにしましょう.

4 pythonの学習フォルダで作業する

自分の好きな場所に `python学習用のフォルダ` を作成しましょう.
ホームディレクトリから遠くない場所に, 半角英数字から始まる(できれば全部半角英数字の方が良いと思う)フォルダ名を作成しましょう.

以下に例を示します.
(ちなみに `~/` というのはホームディレクトリを示します.)

```
~/
 ├hoge/
 ├fuga/
 ├python_ws/
```

```
|      └ learn/  # <-- ここ!
|
└piyo/
```

学習用のフォルダができたら, コマンドプロンプトを開き(`windowsキー` → `cmd` と入力→ `Enter` キー), python学習用のフォルダへ移動, VSCodeを開いていください.

もし先ほどのようなフォルダ構成であれば, 以下のようにやっていきます.

```
$ cd ~/python3_ws/learn
$ code ./
```

そうすると, エディタが現在のディレクトリをルートとして起動します.

ちなみに `code` はVScodeを起動する時のコマンドで, 後ろに開きたいフォルダやファイルを指定できます. `./` はカレントディレクトリ(現在のいるディレクトリ)を示します.

階層構造を表す記号たち

フォルダやファイルの位置を示す時, 2種類の方法があります.
それは `絶対パス` と `相対パス` です. `パス` というのは対象までの経路/ルートをのことを指します.
`相対パス` は対象のディレクトリやファイルの位置を, 現在自分がいる階層から相対的に示す方法です.
それに対して `絶対パス` は対象へのパスをルートディレクトリから示していきます.

以下のような階層構造において例を示します.

```
/
 ├bin/
 ├etc/
 ├usr/
 |   ├ AAA/
 |   |   ├hoge.py
 |   |   └fuga.py
 |   |
 |   └ BBB/
 |       └ bbb.txt
 └ lib/
     └ log.txt
```

一番上の `\` はルート(root)ディレクトリと言い, 一番大元(根っこの)のディレクトリと成っています.
現在, `AAA` フォルダの中にいるとして, `fuga.py` ファイルを参照しようとします.
相対パスなら `./fuga.py` もしくは `fuga.py` と表すことができます.
絶対パスなら `/usr/AAA/fuga.txt` のように表せます.

このように相対パスは絶対パスに比べて, (参照ファイルは近隣にあることが多いため)簡潔にかける場合が多いです.
逆に絶対パスではより深い階層であればあるほど文字列が長くなりがちですが, どこで参照してもパスが変わらないという利点があります.

そのため状況に応じて使い分けましょう.

またパスを表すための重要な記号としては `./` , `../` , `~/` を覚えておけばとりあえず問題はないです.
`./` はカレントディレクトリ(現在いるところ), `../` は一個上のディレクトリを示し, `~/` はホームディレクトリです.

そのため, `AAA` フォルダにいるときに `bbb.txt` ファイルを参照するときは, 相対パスでは `../BBB/bbb.txt` のように書くことができます.

また, 二階層上がるには `../../` のように二回連ねて書けばよく, `AAA` フォルダから `lib/log.txt` を参照するには, `../../lib/log.txt` のようにすればOK.

5 Hello, world!

ではとりあえず `Hello, world!` を標準出力に出すプログラムを作成しましょう.

ソースファイル

余談ですが, プログラムを作成する時, プログラミング言語で書かれているファイルなどを総称して `ソースファイル(source file)` と呼ぶことがあります.

テキストエディタも開けたので, 新規ファイルを作成し, プログラムを書いていきましょう.
VSCode左上の `ファイル` を選択し, `新規ファイル` を選択してください. ファイル名は `hello.py` とでもしておいてください.

ファイルを作成し, 開くことができたら, 以下の文章をタイピングして入力してみてください.

```
print("hello, world!")
```

```
hello, world!
```

実行

terminalかコマンドプロンプトを開き, そのファイルがあるディレクトリまで `cd` していって, `python hello.py` と入力し, `Enter` してください.

(python2系や他のバージョンのpythonがある場合, `python3 hello.py` や `python3.7 hello.py` である場合もあります.)

そうすると

```
hello, world!
```

のように出力されるはずです.

一部説明

今回利用した `print()` は `関数` というもので, 後々説明します.
とりあえず何かを表示したいときは `print()` へぶち込んでおくということだけ今は覚えておいてください.

6 対話モード(インタラクティブモード)を利用する

terminalかコマンドプロンプトで `$ python3` もしくは `$ python` と入力してください.
(`$` は入力しなくていいです)

そうするとコードを一行ずつ入力できるモードへ移ることができます.
簡単なコードはここで試すのも手です.

```
C:\>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05)
[MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> str = "hello, world!"
>>> print(str)
hello, world!
>>> exit()
C:\>
```