目次へ

pythonスクリプトを作成する

対象とする読者

このテキストを読むために、特に準備すべき知識はありません。ただし、pythonについて最低限の知識をお持ちであることを想定します。これの前に書かれた、最初のテキスト 1～10章 に目を通されることをオススメします。

これからの説明では、以下の環境であることを想定します。

- Windows XP(Vista, 7)
- python 2.x (python3系はここでは扱わない)

実行すべき内容を、順に書く

最初のテキストでは、pythonの色々な命令を試すために、対話シェルというものを使いました。

原理上は、これだけでもやりたい仕事をほとんどこなすことができます。ただ、書き間違いなどでエラーが出た場合は最初から書き直しですし、そもそも対話シェルを終わらせたら今までに作

った変数や関数などがすべてなくなってしまいます。

なので、これらの命令をあらかじめ「シナリオ」のようにすべて並べて書いておいたものが、スクリプトファイルです。簡単なものです。

ためしに、下のような内容が書かれたスクリプトファイルというものがあったとして、これがどんな動作をするか考えてみましょう。

```
a = 100

b = 40

print a * b / 2
```

行同士にスキマを空けたのは、単に見やすくしようとしてみただけです。これらは

```
a = 100
b = 40
print a * b / 2
```

のように詰まっていても構いません。

上から順に実行されるのがスクリプトファイルの一般的な動作です。説明するまでもない気がしますが、これは 2000 という結果を表示することになるでしょう。三角形の面積の公式に、変数をつかって値を適用してみたという例です。

なんでもないことです。

ただ、慣れない方にとっては、いろいろと瑣末な問題が発生しがちなのも確かなのです。なので、基本テキストではこのスクリプトファイルを作る話題に触れませんでした。

スクリプトファイル作成の基本知識

まず、pythonのスクリプトファイルは"テキストファイル"である必要があります。それって何？

今まで、PCを使って文書を作るとき、"ワード"やら"一太郎"やらといったワープロソフトを使うのが普通だったという方もいるでしょう。「順番に命令を書けばいいんでしょ、よしやるよ」と言って、こういうワープロ系のソフトをさっそく起動してしまったら、まず間違いです。それらしい表示をする文書を作ったとしても、pythonで実行することはできません。

不正確であることを覚悟の上で言うと、「メモ帳」のようなアプリケーションで文書を作ってください。それがテキストファイルです。

「メモ帳」というアプリケーションをご存知でしょうか。意識的に使ったことは少ないかも知れませんが、実行してみれば「ああ、これね」と分かるでしょう。普通は、[スタート]→[プログラム]→[アクセサリ]の下あたりに見つかると思います。

なんで"ワード"みたいなのではダメで、「メモ帳」だと大丈夫なのか、ということを説明するのは、少しむつかしさを感じますが、いつかpythonでファイルの操作をするようになったら説明し

やすくなるかも知れませんので、ここではこの違いについてそれほどこだわらないことにします。一般に、書いている文字に色がつけられたり、下線が引けたり、文字サイズが色々と混在している文書が作れるときは、これはテキストファイルを作るソフトウェアじゃないんだなと考えることにするとよいでしょう。「メモ帳」って、文字に色もつかないし、文字サイズやフォントを混在させることもできないし、殺風景ですよね。あれがテキストファイルというものの姿です。

もうひとつ。

作るファイルの"拡張子"というものを気にすることができるようになりましょう。"拡張子"という言葉自体の意味を考える必要はありません。ファイルには、（当たり前ですが）名前をつけます。ファイルの名前にピリオド記号「.」が含まれているとき、それより先の部分を"拡張子"といいます。

例。「document.doc」というファイルがあったら、拡張子は「doc」です。

ああ、そういえば"ワード"を使って文章を作ると、「.doc」が後ろについているなあ、と思い出せたら上出来です。"エクセル"を使って計算シートを作るときは「.xls」とかいうのもついていますね。あれが拡張子というものです。

でも、"ワード"を使っていながらそれを見たことがない、という人もいるでしょう。ファイルにも「.doc」なんていうオマケはついていない、と言うでしょう。それも正しい姿です。というか、買ったままの姿でWindowsを使っている場合、大体こういう症状を見るでしょう。Windowsのお節介、というしかないのですが、使う人にわざわざ拡張子を見せないようにはじめは設定されているのです。見えないだけで、実はファイル名にはちゃんと拡張子が含まれているのです。

pythonなどのプログラミングを学んでみようという方は、この拡張子がいつも表示されるようにWindowsの設定を直してしまうことをオススメします。やりかたは、環境によってちょっとずつ違いますし、ここで詳しく説明しません。Yahoo! とか Google を使ってインターネットを検索して、「拡張子 表示」とでも調べてみましょう。丁寧に説明されている記事がたくさん見つかりますよ。

できましたか。表示されるようになりましたか。

その上でこんなことを言うのはなんですが、拡張子をこんなふうに表示できることはほんのちょっとだけリスクも含むので、それを注意するようにしてください。ファイル名を修正するときに、このままだと拡張子までもうっかり変えてしまうということが可能です。拡張子が変わると、たとえば「ワードのファイル」だったのが見た目が「エクセルのファイル」に変身してしまったり、「何だかよくわからん」見た目に変身してしまったりします。Windowsって、拡張子によって「何で使う予定のファイルなのか」を判断するようにできているのです。

うっかり拡張子を変えてしまうと、ダブルクリックなんかの動作ではまともに扱えなくなってしまいます。拡張子を元あったとおりに戻せば、またダブルクリックで開けるように直すことができますので、落ち着けば対処は簡単です。例えばワードで開く予定のファイルだったら、

「.doc」と、半角文字でちゃんと最後に書き足せば大丈夫です。全角文字だとダメなのでそこはよく注意しましょう。

「Windowsをそんな怖い状態にするのは嫌だなー」とおっしゃる方もいるでしょう。初心者にとっては確かにそれが親切だったのですが、プログラミングを学んでみようと思えるほどの方なら、そろそろこういった冒険をしてみるのもよいのではないでしょうか。慣れてみれば、冒険というほどのことでもありません。

拡張子の話で長くなりました。

pythonスクリプトの典型的な拡張子は、「.py」です。あとで実際にスクリプトファイルを作るときは、すべてこの拡張子でつくるようにしますので、覚えておいてください。

また、拡張子を「.py」にしておくと、このファイルがpythonスクリプトなのだと一目でわかるように、アイコンが変化します（Windowsで、インストーラーを使って導入した場合。）ためしに、どんな種類でもよいのですが、捨てても構わないファイルを何か作って（例えばワード用のものでも構いません）、拡張子をわざと「.py」に直してみましょう。アイコンの表示が変わりましたか。

まあ、実際にpythonで実行してみたとしてもエラーが起こるに決まっていますから、確認だけして、そのファイルはすぐにゴミ箱にでも捨ててしまってください。拡張子というものの存在はわかっていただけましたね。

最初のまともなスクリプト

この時点で、ファイルの拡張子を触ることにある程度慣れていると思います。

ここらで、最初のpythonスクリプトを作ってみましょう。最初はまず、デスクトップ上にでも作ることにします。（あとで、コマンドプロンプトとかを扱い始めると、デスクトップはいろいろ面倒になってきます。ここにファイルを作るのは今だけ。）

まずは、デスクトップ上の何も置いてない場所でマウスを右クリック。ここで出てくるメニューから[新規作成]を選びます。何を新規作成するのかをさらに聞いてくるはずですから、[テキストドキュメント]とか[テキストファイル]とかいう項目を見つけてマウスクリックしてください。新しいテキストファイル.txt とかいう名前のファイルができました。これの名前をまずは変えましょう。日本語を含むファイル名にはしておかないほうが、あとでいろいろと都合がよいです。first.txt とでもしておくことにしましょう。

pythonスクリプトとして実行させたいのですから、実はこのままでは便利ではありません。やっぱり、first.txt ではなく、 first.py とファイル名を変えてください。拡張子を直してみることは、さっきやってみたとおりです。アイコンの形が変化しましたか。

で、このファイルを今度は編集します。今のままでは、中身はまだ空っぽです。

いろいろな方法がありますが、今は、Windowsの「メモ帳」アプリケーションを使うことにしましょう。

まずは「メモ帳」を起動します。このテキストの最初のあたりに書いたとおり、Windowsのスタートメニューから探して実行してください。空っぽの編集画面が表示されますね。

ここに、さっき作ったpythonスクリプトを"ドラッグ・アンド・ドロップ"してください。よくある動作なので大丈夫だとは思いますが、今の「メモ帳」アプリをデスクトップの隅に寄せておいて、デスクトップにある first.py のアイコンをマウスクリックでつかんで「メモ帳」の編集枠のなかに移動し、マウスボタンを離すという動作です。

空っぽのファイルなのでまだ何も表示されませんが、メモ帳のタイトル欄が「first.py」となっていれば成功です。

テキストファイルを編集するためのアプリケーションは「メモ帳」以外にも巷に色々ありますし、編集を開始するための手順も、こんなドラッグ＆ドロップみたいな面倒なことは普通しないですむように作られているものです。でも、まあ、最初ですから。Windowsって「素」の状態ではたかだかこんなものなわけですし。

ともかく、今編集がはじまった first.py に、下の一行を書きましょう。一行書いたら、[Enter]キーで改行もしておくのが普通ですので、そうしましょう。

```
print "Hello World!"
```

どんな動作をするものかは、ここまで読まれた方にはもう説明しなくてもよいですね。

ファイルを保存します。[ファイル]メニューから[保存]とか[上書き保存]とかを選ぶのがよくある方法です。慣れている人は、保存のために[Ctrl]キーを押しながら[S]を押す、なんていう動作で済ませるかもしれませんね。

ここまでで、いったんこれの動作を確かめてみましょう。「メモ帳」を閉じるのはちょっと待って、いったん最小化しておくだけにしましょう。保存さえしていれば、メモ帳で開いたままでもスクリプトは実行できます。

さて、デスクトップ上の first.py をダブルクリック。

…

うまくいった人も、うまくいってない人も、たぶん不満でしょう。ほんの一瞬だけ何かが表示された気もしますが、目で確認する時間さえなく、すぐに消えてしまいました。これじゃあ役に立ちません。でも、ダブルクリックでスクリプトを実行するときは、こんなものです。

最小化していた「メモ帳」をもとに戻して、さっきの一行にもう一行追加します。下のようにしてみてください。

```
print "Hello World!"
raw_input()
```

このraw_input()が何か、まだあまり気にしなくてもよいです。（さっきから、色々な場面でまだ気にするな、まだ気にするな、と繰り返している気がします。あとでちゃんと全部説明しなお

して回収できるかな…）　カッコ記号をすぐに開いて閉じる書き方をしています。また、raw と input の間は、アンダースコア記号です。

ここまで書いたらまた保存して最小化（デスクトップの横とかにずらすだけでももちろんOK）、再びさっきのようにダブルクリックでスクリプトを実行しましょう。

書き間違いが含まれていると、相変わらず一瞬で全部終わってしまうのですが、うまく書けているときは、下のような表示がされて画面が残っているはずです。

```
Hello, World!
```

この画面表示を終了するには、[Enter]キーを押します。

日本語の表示ができるかも試しておきましょう。同じスクリプトを下のように書き換えてください。今度は3行になります。

```
# coding:cp932
print "こんにちは世界"
raw_input()
```

一行目のおまじないは何だろう。今は、「英数字以外の文字、たとえば日本語が入るときは必ずこういう宣言をしておく必要があるんだ」と覚えておいてください。シャープ記号「#」を必ず行頭に書いておくことに注意してください。

で、また保存してダブルクリックで実行してみましょう。下の表示が確認できれば成功です。

```
こんにちは世界
```

表示されずにあいかわらず一瞬で画面表示が終わってしまっている場合は、書き間違いがないかよくチェックしましょう。空白記号は全角文字になっていませんか。（これは、矢印キーでカーソルを移動したときに移動距離が薄いか厚いかで判断しましょう。）また、ダブルクォーテーションまで全角記号で書いていませんか。綴りにミスはありませんか。

ところで、対話シェルではうまく日本語が扱えないときがありましたが、これからは大丈夫です。対話シェルを使ったときは、この「# coding:cp932」という書き方ができなかったのです。スクリプトの形にしてこの宣言さえしておけばもう大丈夫。日本語使いたい放題です。（分かる人向け注意：これが通用するのは、テキストファイルがWindowsのシフトJISエンコードのときだけです。別のエンコードのときは、それに応じた宣言になります。「メモ帳」で作るファイルはほとんどの場合シフトJISだから、まだそんなに気にするところではないでしょう。）

ということで、最初のスクリプトファイル作成と実行のことは終了です。

エラーのときも表示が消えないように

さっきまでのように、スクリプトファイルを作って、実行結果を確かめて、という繰り返しで、自由にスクリプトを作っていくことができます。

でも、成功したときはよくても、書き間違いがあったときに表示が全部一瞬でなくなってしまうのはどうも困ります。

何度も繰り返して実行結果を確かめるためには、Windowsの「コマンドプロンプト」の上でスクリプトの実行をすることにすれば、こういった制限からは開放されます。これのやりかたを確かめておきましょう。

まず、あらかじめ、デスクトップでない場所に作業用のディレクトリを作るところからはじめましょう。理由はすぐに分かります。

普段使っているコンピュータのドライブ名は何でしょうか。Cドライブですか。Dドライブですか。それともももっと違うドライブ名でしょうか。どのドライブでもいいのですが、そこの直下に今から新しいフォルダを作ります。pythonのインストールができたくらいのユーザーなら、たぶんそこらにフォルダを作るための権限も最初から持っているはずです。

フォルダがCドライブにしか作れなさそうな人は、そこに。作業のためのドライブを別に分けているという人は、そこに。実際、どこでもよいです。USBメモリか何かをいつも挿していて、そのドライブで作業しているんだという人は、そこのドライブにフォルダを作ってもよいです。

で、作るフォルダ名は、何でもよいのですが、仮に workpy とでもしておきましょう。wで始まるフォルダは割と下のほうに表示されるので、あとで見つけやすいです。

フォルダを作りましたか。そしたら、さっきデスクトップに作った first.py をそのフォルダに移動しておきましょう。今後はいろいろスクリプトを書くときにももっぱらこの場所を使うことにします。（注意：ファイルを移動するときは、「メモ帳」などで編集中の画面を終了しておくようにしましょう）

さて、「コマンドプロンプト」を開始して、このフォルダ内に移動しましょう。…って、当たり前のように言いましたが、もちろん説明が必要ですね。

コマンドプロンプトは、マウスとかを使わない、もうひとつのWindows操作の姿です。…とか言ってみたくなりますが、まあ、コマンドプロンプトはコマンドプロンプトです。開始してみましょう。やりかたはいくつかありますが、手っ取り早い方法としてショートカットキーを使うテクニックをこの際覚えてしまうことにします。Windowsキーを押しながら[R]。Windowsキーは、スペースキーの横のあたりにある、[Win]とか、旗のたなびくマークとかが刻印してあるキーです。これを押しながら英文字の[R]を押してください。「ファイル名を指定して実行」という画面が出ますので、この入力欄に直接 cmd と小文字で入力して[Enter]キーを押してください。こいつがコマンドプロンプトです。なんか、pythonの対話プロンプトにそっくりではあります。でも表示されていることは少し違っていそうですね。

コマンドプロンプトでWindowsを操作しているときは、いつもどこかのフォルダが「現在位置」として設定されています。今から、この「現在位置」を、さっき作った作業フォルダに設定します。

現在どこのフォルダが「現在位置」かは、文字入力カーソルの左に表示されています。これは人の環境によって異なりますが、たぶんちょっと複雑な感じの場所が現在位置なのではないでしょ

うか。これを、C:¥workpy (人によっては D:¥workpy だったりE:¥workpy だったり）にしたいんですね。

「現在位置」の移動は、まずはドライブを、次にフォルダを、という順番で行います。ドライブの移動は、たとえば下のようなコマンドを打つことで行います。

```
C:￥Users￥yamamoto￥>d:    <-- 打ち込んだ部分は、 d:  だけ。決定は[Enter]
D:￥>
```

コロン記号をつけて、ドライブ名を打ち込むだけです。(Cドライブで作業する予定で、現在位置もC:¥で始まる場合は、この作業はいりません)

その後、改めて、フォルダレベルの移動を下のようなコマンドで行います。

```
D:￥>cd d:￥workpy     <-- 打ち込んだ部分は、 cd d:￥workpy
D:￥workpy>
```

cd ってのが、コマンドプロンプト的に、「現在位置をどこそこのフォルダにします」って意味の命令です。（change directory の略らしいよ。）円マーク「¥」は、「～の下の」とでもいった意味です。「Dドライブの真下に作った workpy というフォルダに移動してね」という意味になるわけです。

これで、うまくいったように見えます。ためしに、このフォルダの中に何があるかの一覧表を表示させて、さっきあらかじめ放り込んでおいた first.py があるかを確かめましょう。dir と入力して、[Enter]です。

```
d:￥workpy>dir
 ドライブ D のボリューム ラベルがありません。
 ボリューム シリアル番号は C813-4D76 です

 D:￥workpy のディレクトリ

2010/06/16  00:04    <DIR>          .
2010/06/16  00:04    <DIR>          ..
2010/06/16  00:04                52 first.py

               1 個のファイル                  52 バイト
               2 個のディレクトリ  470,771,773,440 バイトの空き領域
```

表示内容は環境によってところどころ異なりますが、とにかく first.py が入っているらしいことは分かります。ここまででやっと、スクリプトをこの中で実行してみる準備ができたことになります。（面倒ですね。でも一度やってしまえばしばらくこのままで他の作業を繰り返すのみになります。もっと便利な方法はもちろんありますが、今は「素」に近いWindowsでできること、という方針でやってますので...)

ここでおもむろに、スクリプトを実行するために first.py と入力して[Enter]を押してください。

```
d:￥workpy>first.py
こんにちは世界


d:￥work>
```

今度は、さっきと違って、Enterを押して実行を完了したあとも表示がなくなりません。

first.py をあらためてメモ帳で開いて（手順はもう大丈夫でしょうか？　今度はファイルがworkpyフォルダの中ににありますから、適当にフォルダを開いて探す必要がありますよ。「メモ帳」を開始してから、そのファイルをドラッグ＆ドロップします）、最終行のraw_input() を消して保存しなおしてください。コマンドプロンプトの上でもう一度first.py を実行してみて、やはり表示が消えてしまわないことを確かめてください。

次に、メモ帳の上で、わざとスクリプトに書き間違いを紛れ込ませてください。

```
# coding:cp932
priint ″こんにちは世界″      <--- こんな感じのミススペルとかね
```

で、これを保存したら、改めてコマンドプロンプトから実行。

```
d:￥workpy>first.py
  File ″D:￥workpy￥first.py″, line 2
    priint ″こんにちは世界″
           ^
SyntaxError: invalid syntax
```

ちゃんと間違ったところを指摘してくれて、しかも画面は消えません。これならスクリプトを書きながら実行結果を確かめて順次進めていく、ということがやりやすくなりますね。

ちなみに、コマンドプロンプトにおいても、さっきの入力を丸ごと再利用するために矢印キーの[↑]を押すというテクニックが使えます。慣れると快適になってきますよ。

補足みたいな形になりますが、なぜ普段の作業場所をデスクトップ上とかにしないのか。あそこはフォルダがずいぶん深い場所にあり、コマンドプロンプトで現在位置をそこに移動するのが大変だから、なのでした。

さあ、これで、仮にも、pythonのスクリプトを書いて実行結果を確かめるということができるようになったと考えてよいでしょうか。いよいよ、[練習問題編]をはじめる準備ができたことになる、わけですが。どうなんだろう。うん。

とりあえずこの問題ができるようなら、次にいく準備ができているということです。

練習問題00

【練習問題：00】ページの冒頭で示した三角形の面積を計算するスクリプトを実際に作成し、実行結果を確かめよ。 スクリプト名は triangle.py とすること。このスクリプトには日本語表示が含まれないので、coding:cp932 とかいった「オマジナイ」のための記述は省いてよい。

ところで、コマンドプロンプトをどうやって終了させるかということを説明していませんでした。単に、ウインドウ右上のXボタンをマウスクリックすればよいです。ただ、一度消してしまうと、もう一度これを開いて、現在位置を移動して…とかやるのは面倒なので、しばらく最小化でもしてとっておきながら、必要なときにもういちど復元して使う、という方針でやるのがよいと思います。（コマンドラインから exit と入力する終了方法もありますが、まあ、今は知ってても知らなくてもいいです。）