

python自習テキスト WEB編 はじめに



python自習テキストをご愛顧いただきありがとうございます云々。

今までに書いたテキストでは、手元のコンピュータで何か意味のある仕事ができるようにという方向で内容を組み立てていました。だから、データの処理の仕方だの、その表示の整え方だのといった具合で、グラフィックやらサウンドといったキャッチーな要素の少ない、かなりストイックな作りだったと思います。全部読んでくださった方はよほどの我慢強いお人です。ありがとうございます。

ここからは、スクリプト言語を使ってできる仕事の中で、重要性がますます高くなりこそすれ低くなる見込みが一向にない、WEBアプリケーションを作るための基礎知識を紹介していきたいと思います。pythonはこの仕事にけっこう向いていますので、使う言語はやっぱりpython。

WEBアプリってのは、WEBブラウザの上で動くシステムです。Googleの検索サービスしかり、mixiやFacebookなんかのSNSサービスしかり、2ちゃんねるみたいな掲示板システムしかり。

まともなWEBアプリ（って略します）を作るために必要な知識は、膨大だといっていいでしょう。簡単に「ゼロから学べるWEBアプリ」とかそんなことを言われても信用しちゃあいけません。ゼロからWEBアプリを学ぶなんて無謀です。せめて簡単なスクリプト言語をひとつ、あとはHTMLの基礎くらい知っている人じゃないと、とても足場が定まった学習はできないです。

で、これまでの自習テキストでは、簡単なスクリプト言語(python)がある程度使えるようになってもらいました。HTMLの基本を理解してもらい、これを自動生成してみるという体験もしてもらいました。そ

れなら準備は万端と言えますね。なのでこんなふうによっとWEB編を書き始められるわけです。最初から書きたいと思ってたんですけど、よっとここに進めますね。

ところで、プロでない人が普段の仕事に使えるpythonの知識を授けたい、という思いはここでも変わりありません。そんな人たちが、WEBアプリを実際に仕事のために書くことって、そんなことあるのかな？ 実はけっこうありうることだと思うのです。下のようなことがしたくなることって、ありそうだと思いますか。

- 自社のWebを見に来てくれた人が感想を簡単にこちらに送るための手段を提供したい。メールフォームとか。
- 最新のニュースを提供するためのページを作って、いちいちHTMLを書き換えなくてもデータが整形されて見られるようにしたい。
- 特定のページが何回見られたのか知りたい。

大体、ほとんどの会社やら学校やらがWEBサイトを持っているんですから、WEBを管理する人は必ずどこにでも必要になりますね。するとそのうち、「見るだけ」ではないような、なんらかのアクションに反応するWEBページを運用したくなるものではないですか。そのとき、ごく簡単なものなら、サクッと作っちゃってもいいと思うわけです。

難しいWEBアプリまで自力でつくっちゃうのは、それはさすがに難しいです。もし個人がとつても努力してそれらしいものをこしらえたとしても、他人に理解できるくらい整理された作り方をしないと、結局保守もままならなくなってしまいますしね。WEBアプリが「仕事として」作れるってのは、作り直しや改造を含めて自由にできる程度にとどまっているときだけです。そこらへん、無理しないようにしましょうね。WEBアプリが作れるようになると、結構熱中して複雑なものをつくっちゃいがちです。

あとは何より、webアプリを作ってインターネットに公開するとしたら、あらゆる攻撃目的のアクセスに耐える必要がありますから、相当の気を遣う必要もあるのです。複雑なものを作れば作るほど、どこかに攻撃のスキを作ってしまう可能性が大きくなります。こういう意味でも、なるべく単純なものをしっかり作る場所に集中するほうがいいです。

自力でつくらないにしても、人に頼んだりすることが出てくるでしょう。そのときもやっぱり、これから書くような基本を知っているといいと思うんですよ。できることとできないこと、難しいことと易しいことを見分けることができるようになったら、変に予算を無駄に使ってしまったり、業者に無理をいって泣かせることも減ってくると思うわけでした。

まあそこらへんの理屈はいいや。とりあえず、知らない人には新鮮で楽しい知識だと思いますよ。それだけでも充分です。また、それこそが大事です。

WEBサーバーをさわってみる

知っている人には何をいまさら、といったことから始めます。

まず、WEBアプリとは、WEBサーバーとWEBブラウザが通信することで動作するものです。WEBブラウザとは、あなたの手もとで動いている、インターネットエクスプローラとか、Firefoxとか、そういうもの。

で、インターネットの中のどこかにあるWEBサーバーに「これが見たい」というリクエストを送ると、それに対応するHTMLとか画像が送られてきて、それがWEBブラウザの上ではきれいに整形されて見え

ます。WEBサーバーは、「これが見たい」と言ってきたWEBブラウザに情報を送ってあげるのが仕事です。WEBブラウザだけが相手ではないですけど、基本はそう。

WEBサーバーを見たことはありますか。ほとんどの場合、ないと思います。でも、WEBサーバーをつくるために専用のコンピュータが必ず必要なわけではありません。サーバーの役割をするプログラムを実行すれば、なんでもサーバーになります。ってことで、あなたのコンピュータをWEBサーバーに仕立てて、そこで色々な動作を試しながら、WEBアプリとは何ぞやという実感を持ってもらうことがいいかな、と考えました。つまり、一台のコンピュータの中に、WEBサーバーとWEBブラウザが同居して動いている状況をつくるというわけです。一人二役ですね。

巷には色々な種類のWEBサーバー用プログラムが存在します。今のところ一番使われているのはApache というソフトでしょう。こいつのWindows版もありますからそれを導入してみてもいいな、と思いましたが、なんだかオオゲサになってしまいます。IISというものもありますが、インストールがけっこう大仰ですよ、あれも。

とにかく簡単に動作が試せて安定感もあるソフトを考えると、AN HTTPD というアプリケーションが目的にぴったりしているように思えました。今回はこれを使わせてもらいましょう。作者さまには感謝します。

Windowsを使ってるひとが対象です。LinuxとかMacとかの人はすいません。Apacheを使うとなればほとんどの環境にあわせて説明できるんですが、まあこれはよほど多くの方が（幸運にも）要望してくださるときに考えます。

インストールと設定

ということで、早速 AN HTTPD を入手して自分の環境で動かしてみましょ。pythonがインストールできるくらいの権限をマシン上に持っている方なら、たぶん問題なく動作させられると思います。

<http://www.st.rim.or.jp/~nakata/> [<http://www.st.rim.or.jp/~nakata/>]

上のURLから、最新の配布ファイル（2011年2月9日現在では、[httpd142p.zip](#)）をダウンロードしてきて、好きな場所に展開します。ZIPファイルの展開のしかたは大丈夫ですか。どこかにダウンロードしておいて、それをダブルクリックで中身を見て、中にあるファイルとかフォルダを全部、どこか別のフォルダ（あらかじめ作っておく）の中にドラッグ&ドロップしてやればできますよ。



尺 圧縮率: (複数の値)
 西暦日時: 1995/08/20 10:00 2005/01/01 12:00

↑こんなふうにすべて選択して、どこかにドラッグ&ドロップ。

たとえば D:\work_py とかいうフォルダで今までの練習問題をやっていたりしたなら、さらにこの中に httpd とでも名前をつけた新しいフォルダをつかって、その中に入れるのとかでもいいでしょう。

そしたらその中の httpd.exe をダブルクリックなどで実行します。一瞬なにかが表示されるかもしれませんが、すぐ消えてしまうでしょう。それで正常です。デスクトップ右下の、小さいアイコンが横に並んでいるところ（タスクトレイっていいです）に、黄色い四角のアイコンが新しく追加されているようなら正常です。（Windows XPとかVistaとか7では、使用頻度が少ないアイコンは自動で隠れてしまうような設定もあります。付近のどこかを左クリックすると多分隠れている奴が現れますよ）

タスクトレイにその黄色い四角が見つかったら、そいつを一回だけ左クリックしてみましょう。下のようなウィンドウが表示されましたか。



ここで、ファイル(F)から終了(E)のメニューを選択するか、終了ボタンを押すと、Webサーバーの実行は終了です。最小化ボタンを押すと、一見ウィンドウが消えますが、またタスクトレイの中にアイコンが現れたままのはずです。これはまだ動作中という意味。

この時点でもうサーバーは動いています。下のようなURLをWebブラウザのアドレス欄に打ち込んでみて、アプリケーションの説明書が表示されたら成功です。

`http://localhost/readme.html` [`http://localhost/readme.html`]

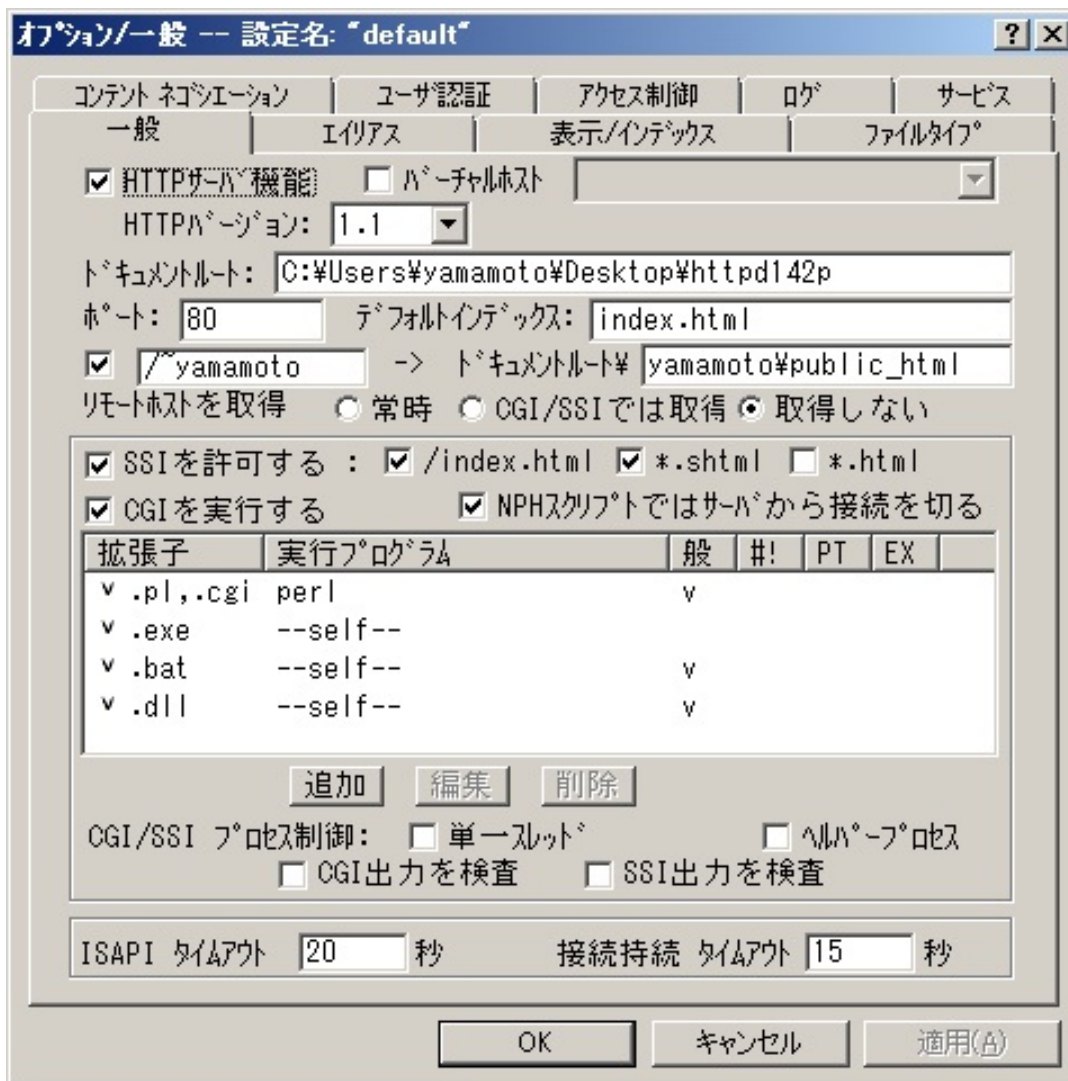
この時点で、あなたのPCの上にはウェブサーバーが稼動しています。

- `localhost` っていうのは、インターネット上のコンピュータにアクセスするのではなく、自分自身をサーバーとみなしてアクセスするという特殊な書き方です。`localhost` のかわりに、`127.0.0.1` と書くこともあります。

最初のWEBアプリ

で、pythonで最初のWEBアプリを書いて、このウェブサーバーの上で動かしてみるところまでをまずはやってみたいと思います。まずは、AN HTTP の設定を若干いじる必要があります。

まずは、さっきみたいにタスクトレイの上のアイコンを左クリックして、アプリケーションのウィンドウを出します。そしたらここのオプション(O)→一般(G) メニューを選択すると、下のような設定画面が出てきます。

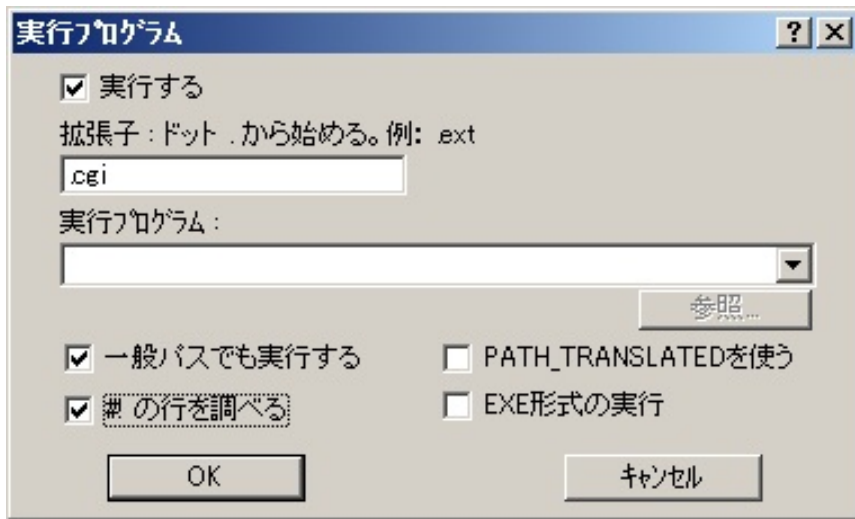


上のほうにはタブがいくつも並んでいて、とてもたくさん設定項目があります。今回いじるのはこれらの一部だけです。

さわるのは、「一般」タブの中身です。その中で、「CGIを実行する」というラベルの左にチェックがついていることを確認してください。ついてなければ、マウスクリックで「v」チェックをつけてください。

で、その下には「拡張子、実行プログラム、般...」といった感じのリストが並んでいて、4行くらい何かの行ができていますね。この4行はまずすべて消してしまいます。行を選んで、その下の「削除」ボタンを押す、という動作の繰り返しでできますよ。

全部消したら、新しい設定行をひとつ追加します。「追加」ボタンから。



「拡張子」という入力欄には、「.cgi」と入れます。最初のピリオドが重要なので忘れないで。

「実行プログラム」の欄はカラっぽで、で、「一般パスでも実行」はチェック。「#!の行を調べる」にもチェック。これでいいです。「OK」ボタンで決定。

もとの設定ウィンドウに戻って、一行設定が増えていることを確認します。

最後に、「CGI出力を検査」というチェック欄もマウスクリックで「v」チェックをつけておいてください。これで設定はおしまい。「オプションを変更してサーバーを再起動します」というメッセージが出てきたら「OK」でよいです。アプリケーションのウィンドウそのものも、最小化してしまいましょう。（「閉じる」だと、サーバーが終了してしまうので注意）

さて、このAN HTTPDを展開したフォルダの中には、「cgi-bin」という名前のフォルダがあると思います。この中にこれからスクリプトを書いて格納していきます。先客として、10数個のファイルがすでに入っていますが、これはすべて消してしまっても構いません。思い切って消しちゃいましょう。

で、最初のスクリプトを、下の内容をコピペでいいですから作りましょう。今のcgi-binフォルダの中に作ります。でファイル名は、「hello.cgi」としてください。「hello.py」じゃないですよ。今回は拡張子がcgiです。

hello.cgi

```
#!/C:/python26/python.exe
print "Content-Type: text/plain"
print
print "hello WEB APP world"
```

- 最初の行は、使っているpythonのバージョンや、pythonをインストールした場所によって変わってきます。インストールするときにパスを明示的にいじった人はそれに合わせて記述を調整してください。（パスを変えられるくらいだから、たぶんそこらへんの調査も大丈夫でしょう。）また、python2.5をインストールしている人は python25、python2.7をインストールしている人は python27 といった具合に書き換えてください。

このスクリプトをちゃんと書き終わったら、WEBブラウザから今度は下のURLを打ち込んでアクセスしてみてください。

<http://localhost/cgi-bin/hello.cgi> [<http://localhost/cgi-bin/hello.cgi>]

どうですか、hello WEB APP world という文字がブラウザ上に表示されていますか。それが確認できれば成功ですよ。最初のWEBアプリですよ。

これがどんな原理で動作したのかを、次回から説明していきます。