

## 第17章 Perlを使ってみよう(1/2) — 初めてのSSIとCGI

ここではPerl スクリプトによる簡単な CGI や SSI を動作させてみます。前の章で説明したように、SSI も CGI も Server Side で実行されるプログラムです。従って、Web サーバに実行の負担が掛かるため、全面的に禁止していたり、制限をかけている場合が多く見られます。本書で解説する SSI や CGI の例を実行させることが出来るかどうか、また実行する際にはどのような手続きが必要か、ということについては Web サーバの管理者に必ず確認して下さい。

### 17.1 初めてのSSI

SSI は Server Side Include の略称で、HTML ファイルに埋め込まれた命令を Web サーバ側で解釈し、実行するものです。Web サーバの設定によっては、拡張子が“.shtml”というファイルにのみ、SSI の解釈が行われることがあります。以降は“.html”ファイルも“.shtml”も HTML ファイルと呼ぶことにします。

例えば HTML ファイルの適当な部分に

```
<!--#config timefmt="%Y年%m月%d日 %H時%M分" -->
この文書の更新日時: <!--#echo var="LAST_MODIFIED" -->
```

と書けば、このタグ内が Web サーバ側で解釈され、Web ページ内に

この文書の更新日時: 2001年07月03日 19時24分

と表示されます。

また、アクセスしてきたユーザが使用しているマシンの IP アドレスやブラウザが知りたければ、HTML ファイルに

```
あなたが使っている PC の IP アドレスは<!--#echo var="REMOTE_ADDR" -->です。<br>
あなたが使っているブラウザは<!--#echo var="HTTP_USER_AGENT" -->です。
```

と書いておきます。Web サーバからこの SSI が埋め込まれた文書を読み出すと、前の例と同様に Web サーバ側でこのタグが解釈されて

あなたが使っている PC の IP アドレスは 133.88.120.87 です。

あなたが使っているブラウザは Mozilla/4.75 [ja] (Windows NT 5.0; U) です。

と表示されます。この例では、133.88.120.87 という IP アドレスを持つ PC 上で、Windows 2000 (Windows NT 5.0) から Netscape 4.75 (Mozilla 4.75) を使用してこの Web サーバにアクセスしていることが分かります。

SSI や CGI は、Web サーバ側で実行されるため、Web サーバでしか取得できない情報をユーザー側にフィードバックすることが可能になります。本章の最後に SSI と CGI を使用した簡単なアクセスカウンタの例を示しますが、これも Server Side で実行しなければ実現できないものです。

SSI は他にも様々な環境変数や書式・命令が使用可能です。詳細については付録を参照して下さい。

## 17.2 初めての CGI

SSI が HTML ファイルに埋め込まれて使用されるのに対し、CGI はもっと複雑な処理を行うプログラムを実行する仕組みを提供します。

まず最初に、以下の HTML を吐き出す CGI を書いてみましょう。

```
<html>
<title>First CGI</title>
<body>
<h1>初めての CGI</h1>
</body>
</html>
```

### 17.2.1 Perl スクリプトを書く

HTML を吐き出す CGI を Perl スクリプト<sup>1</sup>で記述すると次のようなものになります。

```
#!/usr/bin/perl

print "Content-type: text/html\n\n";
print "<html><title>First CGI</title>\n";
print "<body>\n";
print "<h1>初めての CGI</h1>\n";
print "</body>\n";
print "</html>\n";
```

C 言語等のプログラム言語を少し嚙ったことのある人なら大体想像が付くと思いますが、これは標準出力に

```
Content-type: text/html
```

```
<html>
```

<sup>1</sup>Perl というインタプリタが実行するプログラムの一種。

```
<title>First CGI</title>
<body>
<h1>初めての CGI</h1>
</body>
</html>
```

という文字列を出力しているだけのプログラムです。

### 17.2.2 CGI として Web サーバに転送する

次の手順を踏んで、CGI として動作させてみます。最初に Perl スクリプトを自分の PC で作成するわけですが、その時にはなるべく秀丸エディタのような高機能のテキストエディタを使い、改行コードを UNIX 形式である“LF”だけにして保存するようにして下さい。それができない時、例えばメモ帳でこのスクリプトを作成せざるを得ないような時には、FTP クライアントソフトウェアでこのスクリプトを転送する時には、必ず ASCII モードで転送するようにして下さい。そうすることで、Web サーバに転送する時に、改行コードを自動的に UNIX 形式に変換してくれます。もし改行コードが UNIX 形式になっていなければ、恐らく Web サーバ上でこのスクリプトはエラーを吐いてしまうでしょう。

1. Perl スクリプトを“hellow.cgi”という名前で作成します。
2. WS\_FTP で“public\_html/cgi-bin”ディレクトリを Web サーバ側に作成し、そこに“hellow.cgi”を転送します。
3. 転送したら、転送先の“hellow.cgi”を右クリックし、出てきたメニューから“chmod(UNIX)”を選択します。
4. “Owner”, “Group”, “Other”のそれぞれの“Execute”にチェックを入れ、“OK”ボタンを押して下さい。これで“hellow.cgi”は単独で動作するプログラムとして使用可能になりました。
5. ブラウザから“http://Webサーバ名(or IP アドレス)/自分のユーザ ID/cgi-bin/hellow.cgi”と URI 指定して、“hellow.cgi”を動作させてみて下さい。画面に「初めての CGI」と表示されていれば OK です。

### 17.2.3 エラーのときは …

“Internal Server Error…”等が表示され、うまく実行できないときは、まず画面に表示されている情報をきちんと読みとって下さい。大体次のケースが考えられます。

**”Permission Denied …”と表示される** CGI が実行できるディレクトリかどうかチェックして下さい<sup>2</sup>。  
CGI の実行を許可するディレクトリには特別に“.htaccess”というファイル<sup>3</sup>を作り

<sup>2</sup>Web サーバによっては CGI の実行が全面的に許可されていないこともある。Web サーバの管理者に確認すること。

<sup>3</sup>近年は Apache という Web サーバプログラムが高いシェアを誇っている。この Apache が動作している環境ではこのように書けば良いが、それ以外の Web サーバを使っていることもあり、管理者に確認すること。

### Options ExecCGI

などと書いておく必要がある場合もあります。

**”Not Found ...”と表示される** ちゃんと指定されたディレクトリに”hellow.cgi”を転送してありますか？

**”Internal Server Error ...”と表示される** Perl スクリプトが正しく動作できない状態にあります。その際には以下の項目を再度チェックして下さい。

- CGI が動作可能なディレクトリか？
- “hellow.cgi”のモードが正しく “Execute”になっているか？
- 第一行目の “#!/usr/bin/perl” の絶対パス指定が正しいか？
- 改行コードが UNIX 形式になっているか？
- そもそも Perl スクリプトを動作させるためのインタプリタが無かった, など他にも原因は考えられる。どうしてもわからないようであれば Web サーバ管理者に確認せよ

エラーの一度や二度出したからといってめげないで下さい。また, エラーが出たからといって, 単に「うまく動きません」と質問されても訳が分かりません。最低限「このようなエラーが出ていますが, どうすればいいのでしょうか?」と聞いて下さい。慣れてくれば自分でエラー内容が把握でき, そこから解決策を自分で導くことができるようになります。

## 17.3 SSI の応用 – 来訪者カウンタの仕組み

来訪者カウンタの仕組みはごく簡単です。ブラウザから Web ページに対してアクセスがあると

1. SSI が埋め込まれた Web ページが Web サーバによって呼び出される。
2. SSI が実行される。この際
  - (a) 呼び出す前に, ”counter.dat”をロック
  - (b) 現在までの呼び出し回数を”counter.dat”から入力する。
  - (c) ロック解除。
  - (d) 書き出す前に再度ロック。
  - (e) 呼出し後, 回数を 1 増やし”counter.dat”に出力。
  - (f) ロック解除。

という手順を行う。

3. SSI が埋め込まれた個所に数字が表示される。
4. 数字が埋め込まれた HTML ファイルとしてブラウザに表示される。

「ロック」とは、他のプロセスから同じファイルに読み書きできないようにすることです。特に Web サーバは複数の接続に対応できるよう、複数のプロセスで同時に走っている事があり、このアクセスカウンタが働く Web ページが並行して読み込まれることも考慮する必要があります。このようにロックをかけておくことで、一つのアクセスが終了するまでは `counter.dat` が他のプロセスによって書き換えられることが防止できます。

### 17.3.1 作ってみよう

この手順を Perl スクリプトにしたのが図 17.1 です。

```
#!/usr/bin/perl

#
# 来訪者カウンタ
#

# 来訪者数を記録するファイル
$counter = "/home/tkouya/public_html/cgi-bin/counter.dat";

# データファイルを開いて読み出す。
open (FILE, $counter) || die "データファイルが読めません\n";
flock (FILE, 2);
$visitor = <FILE>;
flock (FILE, 8);
close (FILE);

# 来訪者数を標準出力に書き出す。
print "Content-type: text/plain\n\n";
print ++$visitor;

#データファイルを開いて書き出す。
open (FILE, ">" . $counter) || die "データファイルに書き込めません\n";
flock (FILE, 2);
print FILE $visitor;
flock (FILE, 8);
close (FILE);
```

図 17.1: 来訪者カウンタの Perl スクリプト

SSI を埋め込む HTML ファイルは図 17.2 のように作成し、タグを図 17.3 のように埋め込んでお

きます。

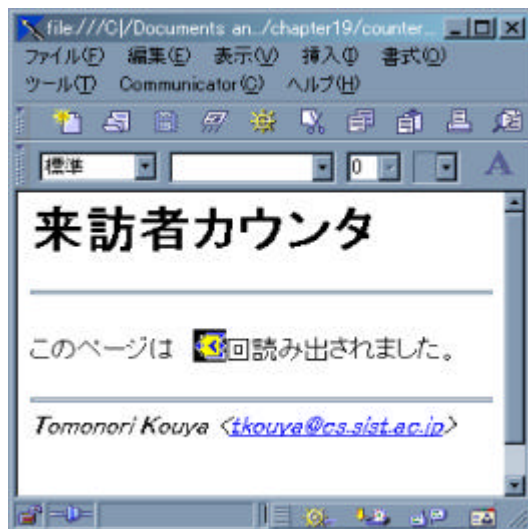


図 17.2: 来訪者カウンタ用 Web ページを作成

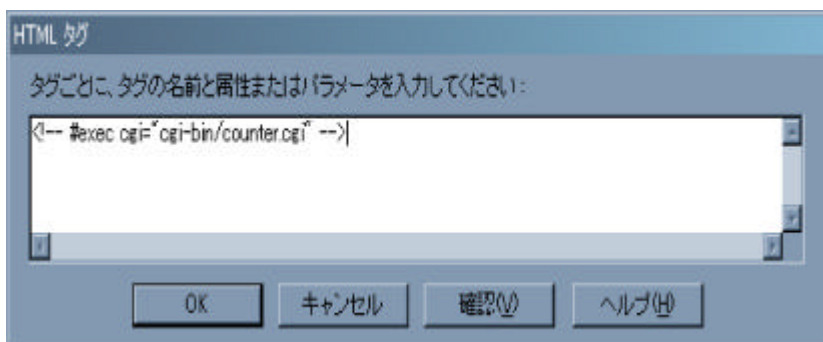


図 17.3: SSI のためのタグを挿入する

Web サーバの指定のディレクトリにこれらのファイルを転送し、ブラウザから動作確認を行ってください。うまく動作すると図 17.4 のように表示されます。

### 17.3.2 エラーのときは …

もしエラーになった時には、counter.cgi については先の「エラーの時は …」の項目をチェックし、これを呼び出している HTML ファイルについては次の項目をチェックしてみてください。

#### HTML ファイルについて

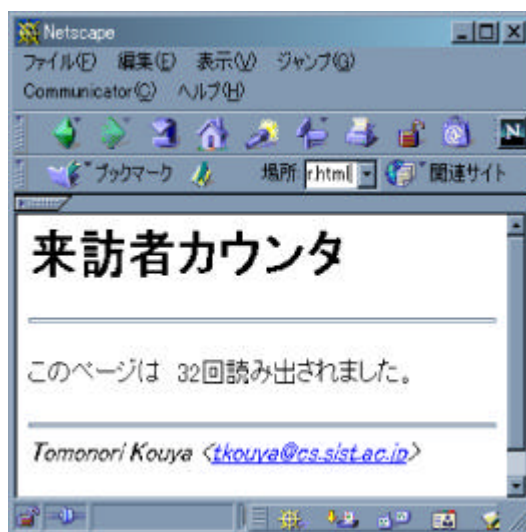


図 17.4: 来訪者カウンタ実行画面

**SSI の記述に間違いはないか？** ... 単に実行するべき “counter.cgi” を正しく指定していないのか、“# exe ...” のタグに問題がないのかチェックして下さい。

**HTML ファイルの拡張子は指定されたものを使っているか？** ... SSI を動作させるときには HTML ファイルの拡張子を “.html” ではなく、“.shtml” とする場合があります。Web サーバ管理者に確認しておいて下さい。また SSI を許可する時には CGI と同様、HTML ファイルと同じディレクトリに “.htaccess” というファイルを作り

#### Options Includes

と書いておく必要があるかもしれません。