

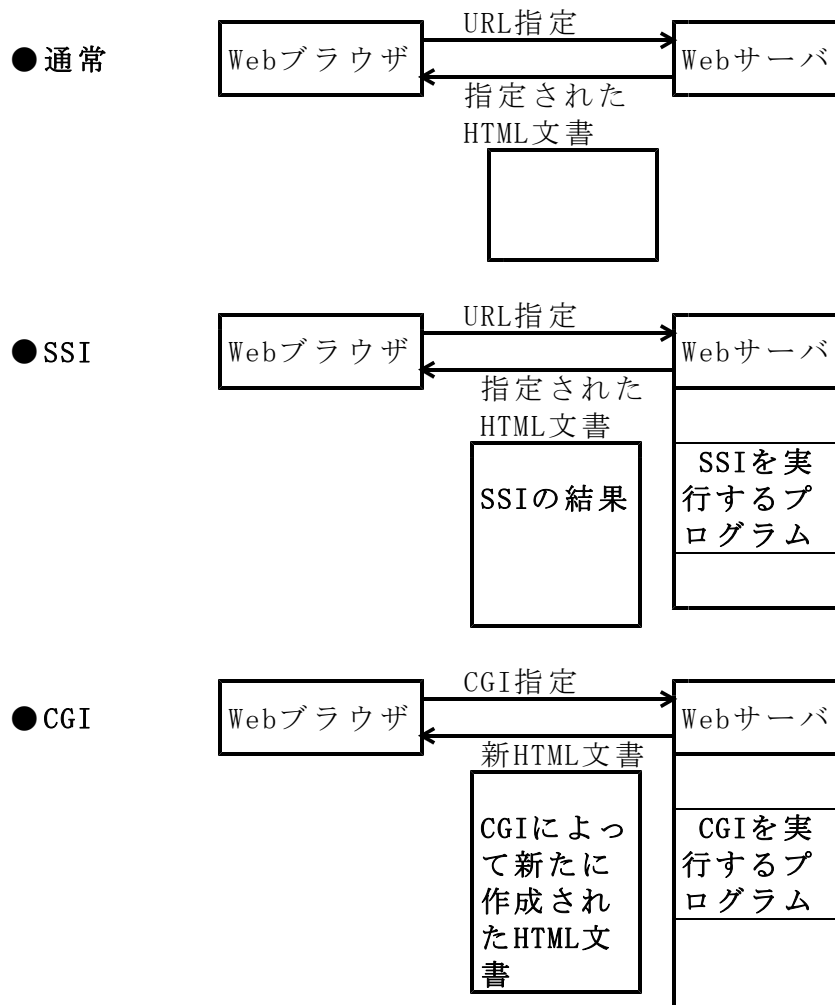
SSI/CGIの基礎と応用

0. 目次

1. はじめに
2. SSI
 2. 1 設定と機能
 2. 2 環境変数の表示
 2. 3 シェルコマンドの実行
 2. 4 Perlプログラムの実行
3. SSIの応用
 3. 1 あいさつ表示
 3. 2 アクセス総数表示
 3. 3 アクセス総数と本日分アクセス回数表示
4. CGI
 4. 1 標準入力から受信したデータの変換 (GET使用)
 4. 2 標準入力から受信したデータの変換 (POST使用)
 4. 3 標準入力から受信したデータ (漢字を含む) の変換
5. CGIの応用
 5. 1 アクセスログ
 5. 2 ダウンロード

1. はじめに

通常、WebブラウザからURLを指定してWebサーバにアクセスすると、Webサーバ内に存在するHTMLで書かれたファイル(HTML文書)が送られてくる。Webブラウザは送られてきたHTML文書を解釈して画面に表示する。ところが、HTML文書中で**SSI (Server Side Include)**が指定されているとSSIを実行するプログラムが実行された後、HTML文書中のSSIを指定しているところが実行結果に置き換えられ、Webブラウザに送られる。また、Webブラウザから**CGI (Common Gateway Interface)**を指定してWebサーバにアクセスすると、Webサーバ内に存在するCGIプログラムが新たに作成したHTML文書がWebブラウザに送られてくる。



SSIまたはCGIを実行可能にするには、Webサーバに設定(サーバ管理者が行う)が必要である。ユーザ側は、SSIを実行するHTMLファイルの拡張子、CGIプログラムを格納するディレクトリ、およびCGIプログラムの拡張子を知る必要がある。ここでは、

SSIを実行するHTMLファイルの拡張子：`.shtml`

CGIプログラムを格納するディレクトリ：`public_html/cgi-bin`

CGIプログラムの拡張子：`.cgi`

とする。

2. SSI

2.1 設定と機能

SSIを実行するには、HTMLファイルの拡張子を.shtmlとし、HTML文書中に次の形式で書く。

```
<!--#コマンド 引数="値"-->
```

HTMLファイル中、この行が実行され結果がこの行と置き換わる。環境変数の表示(現在時刻の表示、ファイル最終更新時刻)やシェルコマンドの実行ができる。

2.2 環境変数の表示

コマンド	機能
config	実行結果の表示形式を指定する。 引数として、timefmt, sizefmtなどがある。
echo	環境変数を表示する。
flastmod	指定したファイルの最終更新時刻を表示する。 fileは、このコマンドが書かれている.shtmlファイルが存在するディレクトリからの相対パス名を指定する。 virtualは、public_htmlからのパス名。
fsize	指定したファイルのサイズを表示する。 bytesはバイト単位表示を指定する。abbrevはキロバイト単位またはメガバイト単位表示を指定する。

```

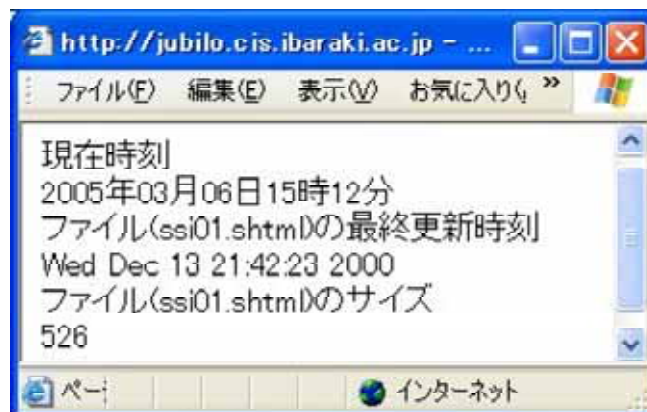
1 <!-- << ssi01.shtml >> -->
2 <html>
3 <head>
4   <title>SSI01</title>
5 </head>
6 <body bgcolor="white" text="black">
7 <!-- 現在時刻の表示 -->
8 現在時刻<br>
9 <!--#config timefmt="%Y年%m月%d日%H時%M分"-->
10 <!--#echo var="DATE_LOCAL"--><br>
11 <!-- ファイルの最終更新時刻 -->
12 ファイル(ssi01.shtml)の最終更新時刻<br>
13 <!--#config timefmt="%c"-->
14 <!--#flastmod file="ssi1.shtml"--><br>
15 <!-- ファイルサイズの表示 -->
16 ファイル(ssi01.shtml)のサイズ<br>
17 <!--#config sizefmt="bytes"-->
18 <!--#fsize virtual="/~isemba/HP/SSICGI/ssi01.shtml"-->
19 </body>
20 </html>

```

許可情報の設定

```
% chmod 644 public_html/HP/SSICGI/ssi01.shtml
```

実行結果



2.3 シェルコマンドの実行

コマンド	機能
exec	シェルコマンドやプログラムを実行する。

dateコマンドを実行する。

```

1 <!-- << ssi02.shtml >> -->
2 <html>
3 <head>
4   <title>SSI02</title>
5 </head>
6 <body bgcolor="white" text="black">
7 <!-- S S I によるシェルコマンドの実行 -->
8 S S I によるシェルコマンドの実行<br>
9 <!--#exec cmd="date"-->
10 </body>
11 </html>

```

許可情報の設定

```
% chmod 644 public_html/HP/SSICGI/ssi02.shtml
```

実行結果



2.4 Perlプログラムの実行

この機能を使って環境変数を表示する。

```

1 <!-- << ssi03.shtml >> -->
2 <html>
3 <head>
4   <title>SSI03</title>
5 </head>
6 <body bgcolor="white" text="black">
7 SSIによるPerlプログラムの実行<br>
8 <!--#exec cmd="./ssi03.pl"-->
9 </body>
10 </html>

```

```

1 #!/usr/local/bin/perl
2 # << ssi03.pl >>
3 # 環境変数とその値を表示。
4 print"DATE_LOCAL = $ENV{¥"DATE_LOCAL¥"}<br>";
5 print"DOCUMENT_NAME = $ENV{¥"DOCUMENT_NAME¥"}<br>";
6 print"SERVER_NAME = $ENV{¥"SERVER_NAME¥"}<br>";
7 exit(0);

```

許可情報の設定

```

% chmod 644 public_html/HP/SSICGI/ssi03.shtml
% chmod 755 public_html/HP/SSICGI/ssi03.pl

```

実行結果



3. SSIの応用

3.1 あいさつ表示

```

1 <!-- << ssill.shtml >> -->
2 <html>
3 <head>
4   <title>SSI11</title>
5 </head>
6 <body bgcolor="white" text="black">
7   あいさつ表示<br>
8   <!--#exec cmd="./ssill.pl"-->
9 </body>
10 </html>

```

```

1 #!/usr/local/bin/perl
2 # << ssill.pl >>
3 # 機能：朝昼夜のあいさつを表示。
4 @week = ("日","月","火","水","木","金","土");
5 ($sec,$min,$hour,$mday,$mon,$year,$yday) = localtime(time);
6 $today = sprintf("%d年%d月%d日 (%s) ",1900+$year,$mon+1,$mday,
7   $week[$yday]);
8 # timeには1970年1月1日0時から現在までの経過時間が秒で返される。
9 # localtime関数はtimeから、秒(0-59),分(0-59),時(0-23),日(1-31),
10 # 月(0-11),年(西暦-1900),曜日(0-6)に変換する。
11 printf("今日は %s です<br>¥n",$today);
12 if( $hour <= 6 ) { print"こんばんは<br>¥n"; goto end; }
13 if( $hour <= 12 ) { print"おはようございます<br>¥n"; goto end;}
14 if( $hour <= 18 ) { print"こんにちは<br>¥n"; goto end;}
15 print"こんばんは<br>¥n";
16 end:
17 exit(0);

```

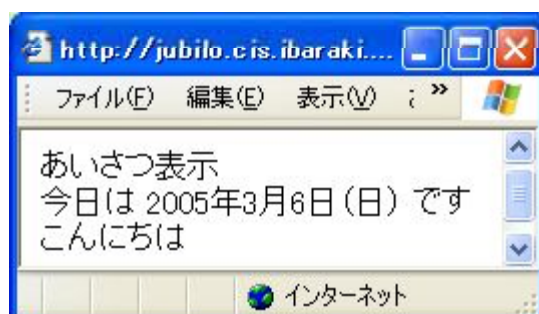
許可情報の設定

```

% chmod 644 public_html/HP/SSICGI/ssill.shtml
% chmod 755 public_html/HP/SSICGI/ssill.pl

```

実行結果



3. 2 アクセス総数表示

```

1 <!-- << ssi12.shtml >> -->
2 <html>
3 <head>
4   <title>SSI12</title>
5 </head>
6 <body bgcolor="white" text="black">
7   アクセス総数表示<br>
8   <!--#exec cmd="./ssi12.pl"-->
9 </body>
10 </html>

```

```

1 #!/usr/local/bin/perl
2 # << ssi12.pl >>
3 # 機能：アクセス総数を表示。
4 $log = "./ssi12.log";           # アクセス回数を保存するファイル。
5 open(FILE, "+<$log");         # 読み書きできるファイルとして開く。
6 flock(FILE, 2);              # ファイルをロックする。
7                               # すでに使用されている場合は待つ。
8 $count = <FILE>;             # アクセス回数を読み込む。
9 chop($count);                # 末尾の改行コードを削除。
10 $count++;                     # 1増やす。
11 seek(FILE, 0, 0);            # ファイルの先頭から書き込む
12                               # ようにする。
13 print FILE "$count¥n";       # アクセス回数を書き込む。
14 flock(FILE, 8);              # ファイルのロックを解除。
15 close(FILE);                 # ファイルを閉じる。
16 printf("訪問者数：%d", $count); # アクセス回数を標準出力に出力。
17 exit(0);                     # 終了。

```

ログファイル (ssi12.log)

0

許可情報の設定

```

% chmod 644 public_html/HP/SSICGI/ssi12.shtml
% chmod 755 public_html/HP/SSICGI/ssi12.pl
% chmod 666 public_html/HP/SSICGI/ssi12.log

```

実行結果



3. 3 アクセス総数と本日分アクセス回数表示

```

1 <!-- << ssi13.shtml >> -->
2 <html>
3 <head>
4   <title>SSI13</title>
5 </head>
6 <body bgcolor="white" text="black">
7   アクセス総数と本日分アクセス回数表示<br>
8   <!--#exec cmd="./ssi13.pl"-->
9 </body>
10 </html>

```

```

1 #!/usr/local/bin/perl
2 # << ssi13.pl >>
3 # 機能：アクセス総数と本日分アクセス回数を表示。
4 $log = "./ssi13.log";
5 @week = ("日", "月", "火", "水", "木", "金", "土");
6 ($sec, $min, $hour, $mday, $mon, $year, $wday) = localtime(time);
7 $today = sprintf("%d年%d月%d日 (%s) ", 1900+$year, $mon+1, $mday,
8   $week[$wday]);
9 open(FILE, "+<$log");
10 flock(FILE, 2);
11 chop(@log = <FILE>);           # $log[0]は今日までのアクセス総数。
12                               # $log[1]は今日のアクセス回数。
13                               # $log[2]は今日の日付。
14 if( $today ne $log[2] ) { $log[1] = 0; $log[2] = $today; }
15 $log[0]++; $log[1]++;
16 seek(FILE, 0, 0);           # ファイルポインタを先頭に移動。
17 print FILE "$log[0]¥n";
18 print FILE "$log[1]¥n";
19 print FILE "$log[2]¥n";
20 truncate(FILE, tell);      # tellで現在のファイルポインタの位置を取得。
21                               # truncate関数で現在のファイルポインタ
22                               # の位置以降を削除する。
23 flock(FILE, 8);
24 close(FILE);
25 printf("訪問者数： %d人 (本日%d人) ", $log[0], $log[1]);
26 exit(0);

```

ログファイル (ssi13.log)

```

0
0

```

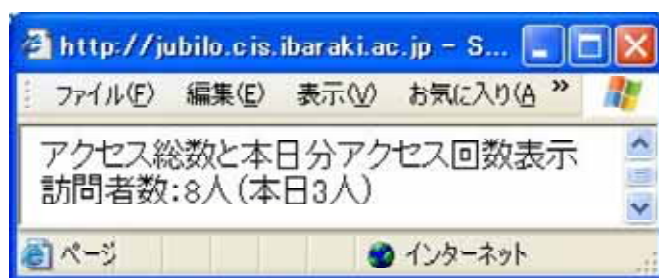
許可情報の設定

```

% chmod 644 public_html/HP/SSICGI/ssi13.shtml
% chmod 755 public_html/HP/SSICGI/ssi13.pl
% chmod 666 public_html/HP/SSICGI/ssi13.log

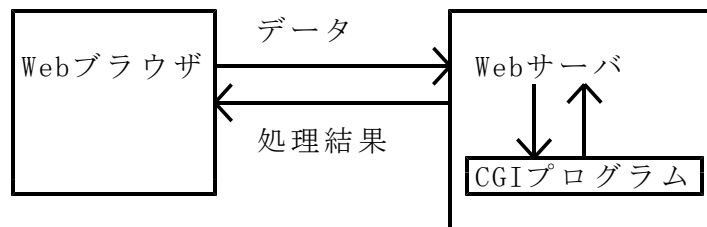
```

実行結果



4. CGI

Webブラウザからデータを受け取り、処理結果をHTML文書にしてWebブラウザに返す会話型の処理を行うプログラムをCGIプログラム（CGIスクリプト）という。データを標準入力から受け取り、実行結果を標準出力から出力するように書く。CGIプログラムは、Webサーバ上で動作するプログラム言語（シェルやPerlなど）ならなんでもよい。



CGIプログラムを実行可能にするには、Webサーバに設定が必要である。ここでは、ディレクトリ（public_html/cgi-bin）を作りCGIプログラム（拡張子は.cgi）置くことにする。Webサーバ上のCGIプログラムは、URLで指定して起動する。たとえば、`…`

●CGIへの引数

WebブラウザからCGIプログラムへのデータは、空白は+、特殊文字は%と16進数2桁の文字コードに変換されて送られる。method属性をGETやPOSTに指定すると、送信データ形式は、

名前=値&名前=値&名前=値&…

となっているのでCGIプログラム側で元に戻す処理をする必要がある。たとえば、`data1=abc+123&data2=ABC%3F%0D%0A`と受け取ったデータは、名前data1の値がabc 123、名前data2の値がABC?、%0D%0Aは¥nとして処理しなければならない。

漢字を扱う場合、漢字コード変換パッケージjcode.plをpublic_html/cgi-binの下にjcode.cgi（許可情報755）として保存しておく必要がある。

GET指定の場合、環境変数QUERY_STRINGに送信データがセットされCGIプログラムに渡される。POST指定の場合、環境変数CONTENT_LENGTHに送信データの長さがセットされCGIプログラムに渡される。データは、read文で読み込む。送信データが長い場合、POST指定がよく使われる。GET指定またはPOST指定のいずれかは、環境変数REQUEST_METHODにセットされる。

●CGIからの出力

CGIプログラムからWebブラウザへのデータには、データの種別を指定する必要がある。HTML形式のデータならtext/html、テキストデータならtext/plainとなる。

```

Content-type: text/html
空行
データ本体
  
```

4. 1 標準入力から受信したデータの変換 (GET使用)

```

1 <!-- << cgi01.html >> -->
2 <html>
3 <head>
4   <title>CGI01</title>
5 </head>
6 <body bgcolor="white" text="black">
7 <form action="../../cgi-bin/cgi01.cgi" method="GET">
8   送信データ : <input type="text" name="data1" size="20"><br>
9   送信データ : <input type="text" name="data2" size="20"><br>
10  <input type="submit" value="送信">
11  <input type="reset" value="取消">
12 </form>
13 </body>
14 </html>

```

```

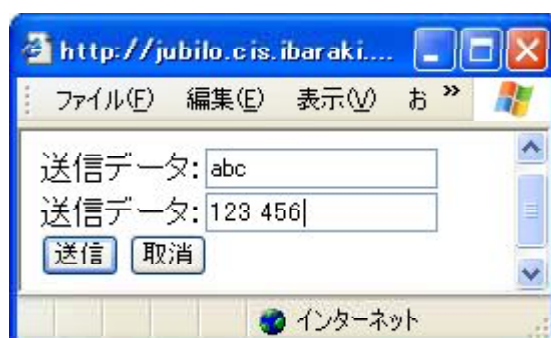
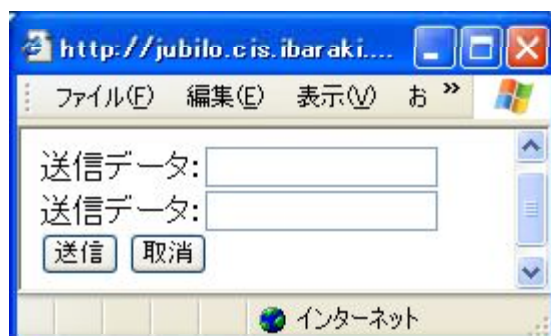
1 #!/usr/local/bin/perl
2 # << cgi01.cgi >>
3 # 機能 : 受信したデータを変換する。
4 $data = $ENV{"QUERY_STRING"}; # 受信データ。
5 @w = split(/&/, $data); # 受信データを&で分割。
6 # 受信データの変換処理。
7 foreach $x (@w) {
8   ($key, $val) = split(/=/, $x);
9   # +を空白に変換。
10  $val =~ tr/¥+ / /;
11  # 特殊文字を変換。
12  $val =~ s/%([0-9A-F][0-9A-F])/pack("c", hex($1))/eg;
13  $val =~ s/¥r¥n/¥n/g;
14  $val =~ s/¥r/¥n/g;
15  $in{$key} = $val;
16 }
17 # 処理結果をHTML文書として出力。
18 printf("Content-type: text/html");
19 printf("¥n¥n");
20 printf("<html>");
21 printf("<head><title>変換後 (GET使用) </title></head>");
22 printf("<body bgcolor=¥white¥ text=¥black¥>");
23 printf("受信データ : %s<br>", $data);
24 printf("変換データ : %s<br>", $in{"data1"});
25 printf("変換データ : %s<br>", $in{"data2"});
26 printf("</body>");
27 printf("</html>");
28 exit(0);

```

許可情報の設定

```
% chmod 644 public_html/HP/SSICGI/cgi01.html  
% chmod 755 public_html/cgi-bin/cgi01.cgi
```

実行結果



4. 2 標準入力から受信したデータの変換 (POST使用)

```

1 <!-- << cgi02.html >> -->
2 <html>
3 <head>
4   <title>CGI02</title>
5 </head>
6 <body bgcolor="white" text="black">
7 <form action="../../cgi-bin/cgi02.cgi" method="POST">
8   送信データ : <input type="text" name="data1" size="20"><br>
9   送信データ : <input type="text" name="data2" size="20"><br>
10  <input type="submit" value="送信">
11  <input type="reset" value="取消">
12 </form>
13 </body>
14 </html>

```

```

1 #!/usr/local/bin/perl
2 # << cgi02.cgi >>
3 # 機能 : 受信したデータを変換する。
4 $n = $ENV{"CONTENT_LENGTH"}; # 受信データの長さ。
5 read(STDIN, $data, $n); # 受信データを読み込む。
6 @w = split(/&/, $data); # 受信データを&で分割。
7 # 受信データの変換処理。
8 foreach $x (@w) {
9   ($key, $val) = split(/=/, $x);
10  # +を空白に変換。
11  $val =~ tr/+/ /;
12  # 特殊文字を変換。
13  $val =~ s/%([0-9A-F][0-9A-F])/pack("c", hex($1))/eg;
14  $val =~ s/¥r¥n/¥n/g;
15  $val =~ s/¥r/¥n/g;
16  $in{$key} = $val;
17 }
18 # 処理結果をHTML文書として出力。
19 printf("Content-type: text/html");
20 printf("¥n¥n");
21 printf("<html>");
22 printf("<head><title>変換後 (POST使用) </title></head>");
23 printf("<body bgcolor=¥white¥ text=¥black¥>");
24 printf("受信データ : %s<br>", $data);
25 printf("変換データ : %s<br>", $in{"data1"});
26 printf("変換データ : %s<br>", $in{"data2"});
27 printf("</body>");
28 printf("</html>");
29 exit(0);

```

許可情報の設定

```
% chmod 644 public_html/HP/SSICGI/cgi02.html  
% chmod 755 public_html/cgi-bin/cgi02.cgi
```

実行結果



4. 3 標準入力から受信したデータ（漢字を含む）の変換

```

1 <!-- << cgi03.html >> -->
2 <html>
3 <head><title>CGI03</title></head>
4 <body bgcolor="white" text="black">
5 <form action="../../cgi-bin/cgi03.cgi" method="POST">
6   送信データ : <input type="text" name="data1" size="20"><br>
7   <input type="submit" value="送信">
8   <input type="reset" value="取消">
9 </form>
10 </body>
11 </html>

```

```

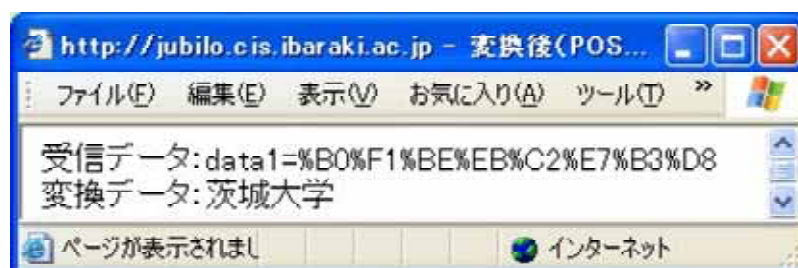
1 #!/usr/local/bin/perl
2 # << cgi03.cgi >>
3 # 機能：受信したデータ（漢字を含む）を変換する。
4 require 'jcode.cgi'; # パッケージjcodeを利用できるようにする。
5 $n = $ENV{"CONTENT_LENGTH"}; # 受信データの長さ。
6 read(STDIN, $data, $n); # 受信データを読み込む。
7 @w = split(/&/, $data); # 受信データを&で分割。
8 # 受信データの変換処理。
9 foreach $x (@w) {
10   ($key, $val) = split(/=/, $x);
11   $val =~ tr/¥+/ /;
12   $val =~ s/%([0-9A-F][0-9A-F])/pack("c", hex($1))/eg;
13   $val =~ s/¥r¥n/¥n/g;
14   $val =~ s/¥r/¥n/g;
15   # convertはパッケージjcode内のサブルーチンで変数$val内の
16   # 漢字コードをeucコードに変換する。
17   &jcode'convert(*val, 'euc');
18   $in{$key} = $val;
19 }
20 # 処理結果をHTML文書として出力。
21 printf("Content-type: text/html");
22 printf("¥n¥n");
23 printf("<html>");
24 printf("<head><title>変換後 (POST使用) </title></head>");
25 printf("<body bgcolor=¥\"white¥\" text=¥\"black¥\">");
26 printf("受信データ : %s<br>", $data);
27 printf("変換データ : %s<br>", $in{"data1"});
28 printf("</body>");
29 printf("</html>");
30 exit(0);

```


許可情報の設定

```
% chmod 644 public_html/HP/SSICGI/cgi03.html  
% chmod 755 public_html/cgi-bin/cgi03.cgi  
% chmod 755 public_html/cgi-bin/jcode.cgi
```

実行結果



5. CGI の応用

5.1 アクセスログ

HTML文書にアクセスがあると、アクセス記録(時刻、訪問者のホスト名)がファイル(access.log)に保存される。アクセスログをクリックすると最大10名分のアクセス記録が表示される。

```

1 <!-- << cgi11.shtml >> -->
2 <html>
3 <head>
4   <title>アクセスログ</title>
5 </head>
6 <body bgcolor="white" text="black">
7 <!--#exec cmd="./cgill.pl"-->
8 <a href=" ../cgi-bin/cgill.cgi">アクセスログ</a><br>
9 </body>
10 </html>

```

```

1 #!/usr/local/bin/perl
2 # << cgill.pl >>
3 # 機能：訪問者のホスト名と時刻を最新のものから一定数記録する。
4 $log = "./cgill.log";
5 $MAXLOG = 10; # 最大記録人数。
6 $ip = $ENV{"REMOTE_ADDR"}; # 訪問者のIPアドレス
7 # IPアドレスからドメイン名へ変換。
8 if( $ip eq "" ) {
9   $rhostname = "-";
10 } else {
11   $rhostname = gethostbyaddr(pack("C4", split(/\. /, $ip)), 2);
12 }
13 @week = ("日", "月", "火", "水", "木", "金", "土");
14 ($sec, $min, $hour, $mday, $mon, $year, $wday) = localtime(time);
15 $today = sprintf("%d年%d月%d日 (%s) %d時%d分%d秒", 1900+$year,
16               $mon+1, $mday, $week[$wday], $hour, $min, $sec);
17 $data = "$today:$rhostname¥n";
18 #アクセス記録を保存する。
19 open(FILE, "+<$log");
20 flock(FILE, 2);
21 @log = <FILE>;
22 unshift(@log, $data); # 配列@logの先頭に$dataを挿入。
23 while( $#log>$MAXLOG ) { pop(@log); } # $#logは配列@logの要素数。
24                                     # pop関数は@logの最後の要素を削除。
25 seek(FILE, 0, 0);
26 print FILE @log;
27 truncate(FILE, tell);
28 flock(FILE, 8);
29 close(FILE);
30 exit(0);

```

```

1  #!/usr/local/bin/perl
2  # << cgit1.cgi >>
3  # 機能：アクセスログの表示。
4  $log = "../HP/SSICGI/cgit1.log";
5  open(FILE, "+<$log");
6  flock(FILE, 2);
7  chop(@log = <FILE>);
8  printf("Content-type: text/html");
9  printf("\n\n");
10 printf("<html>");
11 printf("<head><title>アクセスログ表示</title></head>");
12 printf("<body bgcolor=%\"white\" text=%\"black\">");
13 printf("アクセスログ<br>");
14 foreach $data (@log) {
15     ($date, $rhostname) = split(/:/, $data);
16     print "$date $rhostname<br>";
17 }
18 printf("</body>");
19 printf("</html>");
20 flock(FILE, 8);
21 close(FILE);
22 exit(0);

```

許可情報の設定

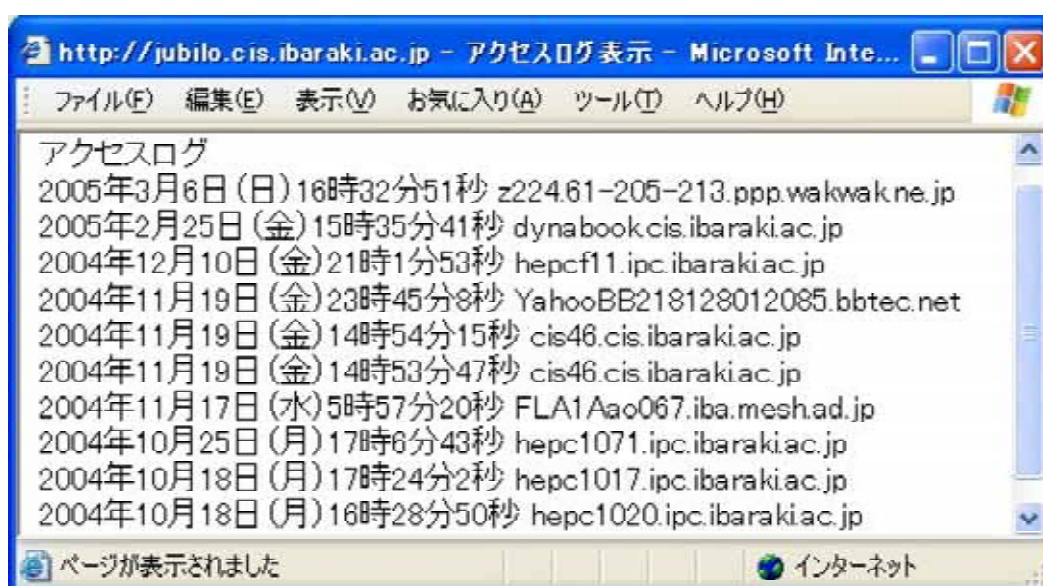
```

% chmod 644 public_html/HP/SSICGI/cgit1.shtml
% chmod 755 public_html/HP/SSICGI/cgit1.pl
% chmod 755 public_html/cgi-bin/cgit1.cgi
% chmod 666 public_html/HP/SSICGI/cgit1.log

```

実行結果





5. 2 ダウンロード

ダウンロードを行う方法を紹介する。

```

1 <!-- << cgi12.html >> -->
2 <html>
3 <head>
4   <title>ダウンロード</title>
5 </head>
6 <body bgcolor="white" text="black">
7 <a href="../../cgi-bin/cgi12.cgi">ダウンロード</a>
8 </body>
9 </html>

```

```

1 #!/usr/local/bin/perl
2 # << cgi12.cgi >>
3 # 機能：ダウンロードの実行。
4 # ダウンロードを許可するファイル名。
5 $file = "../HP/SSICGI/DOWNLOAD/test.dat";
6 print "Location: $file\n\n";
7 exit(0);

```

ファイル(test.dat)

```

123456789
abcdefg
茨城大学

```

許可情報の設定

```

% chmod 644 public_html/HP/SSICGI/cgi12.html
% chmod 755 public_html/cgi-bin/cgi12.cgi
% chmod 644 public_html/HP/SSICGI/DOWNLOAD/test.dat

```

実行結果



- ファイルとして保存するには、**ファイル**メニューをクリックし、**名前を付けて保存**を選択する。
- 文字化けしているときは、**表示**メニューから**エンコード**項目を選び、適切な日本語コードを選択する。