


1歩進んだサーバープログラミング

CGI.pmとJcode.pmを
活用して実用的な
CGIプログラミングをしよう

CPAN(Comprehensive Perl Archive Network)は、Perlプログラムとドキュメントの膨大なコレクションである。プログラミングに便利な機能を寄せ集めたもので、さしずめ知恵の宝庫といえよう。本連載では、CPANで4,000個以上提供されているPerlモジュールから便利なものを厳選し、その機能を自分のPerlプログラムから使い、効率よくプログラミングする方法を解説する。

CPAN

Powered by 

<http://www.cpan.org/>

第1回では、CPANとPerlモジュールという基本的なことを解説して、便利なモジュールの例としてCGI.pmを取り上げた。しかし、CGI.pmについてはどういうものかという概略の説明にとどまっていたので、今回はCGI.pmを使ってフォーム処理を例にとり実際のプログラミングについて説明していく。

CGI.pmを使ったサンプル:

「わかりやすい」フォームの生成

CGI.pmのようなモジュールを使わずにCGIプログラミングを行った場合、動的に生成するHTMLソースの中に変数の値を使う必要があるため、Perlプログラムの各所にHTMLタグを埋め込んだうえに、その中に変数の参照などが入り乱れるといった、とても読みにくいPerlコードになってしまう。そこでCGI.pmを使えば、Perlのプログラム中にHTMLタグを直接埋め込むことなく、CGI.pmのメソッドとしてプログラミング的にHTMLを生成できる。では実際にCGI.pmを使ってフォームを生成してみよう。もしまだCGI.pmをインストールしていないのならば、次のようにコマンドラインから入力すればインストールできる。

```
# perl -MCPAN -e "install 'CGI'"
```

詳しい説明は後にすると、まずindex.cgiのサンプルコードを見てほしい(コード1)。このコードにはどこにもHTMLタグは埋め込まれておらず、Perlプログラムとして読みやすいコードになっていると感ずるだろう。実際にこれはフォームを生成するもので、ウェブサーバーに設置して実行すると次のようなページが表示される。

CGI.pmを使ったフォーム生成のサンプルプログラム

以下のフォームに入力してください

お名前は何?

年齢は何?

CGI.pmを使ったことは? 知らない 知っている 使ったことがある

使ったことのあるOSは? (複数選択可) Unix系OS Windows系 MacOS

Perlのプログラムを初めて書く場合はここをチェック

ここに表示されているフォームはよくある簡単なフォーム画面だ。これがindex.cgiによって表示される。実際にCGI.pmのメソッドを使ったこのコードについて順を追って説明していこう。

プログラム自体は大きく3つのprint文になっている。これは意図的に「ヘッダーとフォームまでの部分」「フォーム部分」「フォーム

コード1 フォーム生成のサンプルコード(index.cgi)

```
#!/usr/bin/perl
use CGI qw(:standard);

print
  header(-charset=>'x-euc-jp'),
  start_html,
  h3('CGI.pmを使ったフォーム生成のサンプルプログラム'),
  p('以下のフォームに入力してください'),
  comment('フォームはここから');

print
  start_form(-action=>"form.cgi"),
  'お名前は?', textfield('Name'), p,
  '年齢は?',
  popup_menu(-name=>'Age',
    -values=>['0','1','2','3','4'],
    -default=>'2',
    -labels=>{'0'=>'10歳未満', '1'=>'10代',
      '2'=>'20代', '3'=>'30代',
      '4'=>'40歳以上'}), p,
  'CGI.pmを使ったことは?',
  radio_group(-name=>'useCGIpm',
    -values=>['no','known','yes'],
    -default=>'known',
    -labels=>{'no'=>'ない',
      'known'=>'知っている',
      'yes'=>'使ったことがある'}), p,
  '使ったことのあるOSは? (複数選択可)',
  checkbox_group(-name=>'OS',
    -value=>['unix','win','mac'],
    -labels=>{'unix'=>'Unix系OS',
      'win'=>'Windows系OS',
      'mac'=>'MacOS'}), p,
  checkbox(-name=>'Exp', -value=>'1',
    -label=>' Perlのプログラムを初めて書く場合は
    ここをチェック'), p,
  submit('送信'),
  reset('リセット'),
  button(-label=>'戻る',-onClick=>'history.back()'),
  end_form,
  comment('フォームはここまで');

print end_html;
```

以降の部分」と見やすくするために分けたもので、実際には1つのprintですべてを「,」で区切ってつないで問題ない。またこのサンプルプログラムによって生成された画面のソースを参照したときにわかりやすくするため、「フォームはここから」と「フォームはここまで」のコメントを入れてある。

それでは中身の説明に入ろう。まず、前回説明したとおりCGI.pmを呼び出し、headerとstart_htmlのメソッドを使ってヘッダーを表示している。また最後のprint文ではend_htmlメソッドによりHTMLソースを終了している。今回このサンプルプログラムはUnix系OS上で実行したので、headerメソッドではEUCコードを指定しているが、ウィンドウズ上でPerlプログラムを実行する場合にはSJISにすればよいだろう。次に、h3、p、commentメソッドで単純にテキストの表示を行っている。

そして真ん中のprint文が今回のテーマのフォーム生成部分である。

CGI.pmのフォーム関係のメソッド

各メソッドを順番に説明していこう。サンプルコードで使っているメソッド以外にも関係するものは説明してある。サンプルと照らし合わせてこの説明を読めばフォーム生成のメソッドについてはほとんど使えるようになるだろう。

start_formメソッド

<FORM> タグの生成を行う。end_formメソッドとセットで利用する。

【構文】

```
start_form([method],[action],[encoding])
```

【引数】

・method

リクエストメソッドの指定。POSTもしくはGETメソッドを指定できる。省略するとPOSTメソッドとなる。

・action

Submit時に実行されるCGIスクリプトのURL。省略すると現在のURL(自分自身)となる。

・encoding

エンコード形式。デフォルトでは「application/x-www-form-urlencoded」となる。start_formメソッドの代わりにstart_multipart_formメソッドを利用するとデフォルトのエンコード形式が「multipart/form-data」となる。start_formメソッドで明示的にencodingに「multipart/form-data」を指定しても動作は同じとなる。

【名前付きパラメーター】

上記の引数はそれぞれ「-method」「-action」「-encoding」でも指定できる。JavaScript関数用にフォームに名前を付ける「-name」、指定したフレーム内に表示する「-target」、フォーム送信時に実行させるJavaScriptを指定する「-onSubmit」がある。

text_fieldメソッド

テキストフィールドを生成する。

【構文】

```
text_field('name',['value'],size,maxlength)
```

【引数】

・name

入力要素の名前(必須)。

・value

テキストフィールドに設定する初期値。

・size

テキストフィールドの大きさ(文字数で指定)。

・maxlength

テキストフィールドに入力できる最大の文字数。

【名前付きパラメーター】

上記の引数はそれぞれ「-name」「-default」「-size」「-maxlength」でも指定できる。このほか「-override」「-onChange」「-Focus」「-onBlur」「-onSelect」も指定可能(それぞれ共通パラメーターの説明を参照)。

【参考】

テキスト入力ボックスを生成する「textarea」、パスワード入力フィールド(入力した文字が見えないテキストフィールド)「password_field」、CGIファイルアップロードのファイル名指定フィールドを生成する「filefield」の各メソッドは、書式とパラメーターは同様。ただし、textareaメソッドだけ「size」「maxlength」の代わりに「rows」「columns」で表示する行数と桁数を指定する。

popup_menuメソッド

ポップアップメニューを生成する。

【構文】

```
popup_menu('name',\@array['selected'],\%labelhash)
```

【引数】

・name

入力要素の名前(必須)。

・\@array

メニュー項目を列挙した配列。

・selected

初期状態で選択するメニュー項目。

・\%labelhash

各メニュー項目のラベルを格納したハッシュ。

【名前付きパラメーター】

上記の引数はそれぞれ「-name」「-values」「-default」「-labels」でも指定できる。このほか「-onChange」「-Focus」「-onBlur」も指定可能(それぞれ共通パラメーターの説明を参照)。

radio_groupメソッド、checkbox_groupメソッド

radio_groupは一組のラジオボタンを生成する。checkbox_groupは複数のチェックボックスを生成する。

【 構文 】

```
radio_group('name',\@array['selected',  
    'true',\%labelhash])  
checkbox_group('name',\@array['selected',  
    'true',\%labelhash])
```

【 引数 】

- ・name
入力要素の名前(必須)
- ・\@array
リスト項目を列挙した配列。
- ・selected
初期状態で選択するリスト項目。
- ・true
ラジオボタン(チェックボックス)間に改行を挿入。
- ・\%labelhash
各メニュー項目のラベルを格納したハッシュ。

【 名前付きパラメーター 】

上記の引数はそれぞれ「-name」「-values」「-default」「-linebreak」「-labels」でも指定できる。このほかに「-column」で使用する列の数、「-columnheader」で列の見出し、「-rows」で使用する行の数、「-rowheader」で行の見出しを指定できる。また、checkbox_groupでは、このほかに「-onClick」も指定可能(共通パラメーターの説明を参照)

checkbox メソッド

単独のチェックボックスを生成する。

【 構文 】

```
checkbox('name',['checked'],'value','label')
```

【 引数 】

- ・name
入力要素の名前(必須)
- ・checked
チェックされた状態にする。
- ・value
チェックされたときに返す値。
- ・label
チェックボックスのラベル。

【 名前付きパラメーター 】

上記の引数はそれぞれ「-name」「-checked」「-value」「-label」でも指定できる。このほかに「-onClick」も指定可能(共通パラメーターの説明を参照)

button メソッド

JavaScriptのボタンを生成する。

【 構文 】

```
button('name','function')
```

【 引数 】

- ・name
ボタンの名前。
- ・function
ボタンがクリックされたときに実行する関数。

【 名前付きパラメーター 】

上記の引数はそれぞれ「-name」「-onClick」でも指定できる。このほかに「-label」でボタンにラベルを付けられる。

submit メソッド

送信ボタンを生成する。

【 構文 】

```
submit(['label'],'value')
```

【 引数 】

- ・label
ボタンのラベル。
- ・value
フォームが送信されたときに返す値。

【 名前付きパラメーター 】

上記の引数はそれぞれ「-name」「-value」でも指定できる。このほかに「-onClick」も指定可能(共通パラメーターの説明を参照)

reset メソッド、default メソッド

フォームをデフォルト状態にリセットするためのボタンを生成する。reset、defaultともに動作は同じ。

【 構文 】

```
reset('label')  
default('label')
```

【 引数 】

- ・label
ボタンのラベル。

各メソッド共通

各メソッドに共通で指定できる名前付きパラメーターがある。

- ・-override=>1
前回このスクリプトが起動されたときの入力値を引き継がない。
- ・-onChange=>function
フィールドの内容に変更を加えたときfunctionを実行させる。
- ・-onFocus=>function
フォーカスがフィールド上に移動したときfunctionを実行させる。
- ・-onBlur=>function
フォーカスがフィールドから離れたときfunctionを実行させる。
- ・-onSelect=>function
フィールドの中の選択範囲を変えたときfunctionを実行させる。
- ・-onClick=>function
ユーザーがボックスをクリックしたときfunctionを実行させる。

Jcode.pm モジュール

JIS、EUC、S-JIS、Unicodeの日本語文字コード変換のためのモジュール

【カテゴリー】文字コード変換
【バージョン】0.82
【作者】Dan Kogai 氏
【URL】<http://search.cpan.org/author/DANKOGAI/Jcode-0.82/Jcode.pm>

CGI.pmとJcode.pmを使ったフォームの処理

先ほどのフォームの処理と出力を行ってみる。フォームデータの受け渡しは、第1回で説明したCGI.pmのparamメソッドを使ってフォーム内のnameで指定した変数名を直接参照するだけだ。サンプルプログラム(コード2、form.cgi)とその結果を示す。先ほどのフォームに対する出力だが、変数の引き渡しやHTMLタグなどの面倒な処理がほとんどなく、フォーム変数に対する純粋な処理だけが書かれていて非常に見やすい。

```
CGI.pmを使ったフォーム入力表示のサンプルプログラム
フォームに入力された値は次のとおりです

お名前は「情報技研」さん
年齢は20代
CGI.pmモジュールは使ったことがないです
使ったことのあるOSは Unix系OS MacOS です
実は Perl のプログラムは書いたことがありません
```

Jcode.pmとは

日本語の処理にJcode.pmモジュールを使っていることに注意してほしい。Jcode.pmは、文字コードを変換するモジュールである。フォームに入力される日本語は、ユーザーの環境によって文字コードがばらばらだ。サンプルプログラムはUnix系OS上でEUCコードで書かれ、出力のHTMLもEUCコードである。ここで別の文字コードで入力された日本語文字列をそのまま出力すると、文字化けの原因になる。そこで、文字列をフィールド入力させるフォーム変数の内容は文字コードを変換する必要がある。今回は出力をEUCコードに統一している。実際には、出力する文字コードを環境に合わせて統一するように気をつける必要がある。

Jcode.pmのインストールは次のように簡単だ。

```
# perl -MCPAN -e "install 'Jcode'"
```

Jcode.pmのココがスゴイ!

Jcode.pmは、以前から使われていたjcode.plの後継となるモジュール版だ。JIS、EUC、SJIS各コード間の変換だけでなくUnicodeにも対応し、半角、全角カナの変換やMIMEのデコード、エンコードなどできる。今回は「jcode(\$str)->euc」で変換しているが、eucの代わりにjis、sjisなど変換したいコードを指定すればその文字コードに変換してくれる。

perl 5.8からJcode.pmの全機能がEncode.pmとして標準で含まれている。最新のPerlではEncode.pmを使うのがいいだろう。

コード2 フォーム処理のサンプルコード (form.cgi)

```
#!/usr/bin/perl
use CGI qw(:standard);
use Jcode;

## Name 変数処理
$name = jcode(param("Name"))->euc;

## Age 変数処理
@agelist = ('10歳未満',
            '10代',
            '20代',
            '30代',
            '40歳以上');
$age = $agelist[param("Age")];

## useCGIpm 変数処理
%uselist = ('no', '使ったことがないです',
            'known', '使ったことはないが知っています',
            'yes', '使ったことがあります');
$useCGIpm = $uselist{param("useCGIpm")};

## OS 変数処理
%oslist = ('unix', 'Unix系OS',
           'win', 'Windows系OS',
           'mac', 'MacOS');
if (param("OS")){
    $str = '使ったことのあるOSは';
    foreach $i (param("OS")){$str .= " $oslist{$i}"}
    $OS = br("$str です");
}

## Exp 変数処理
if (param("Exp")){
    $EXP = br('実は Perl のプログラムは書いたことがありません');
}

## 出力処理
print
header(-charset=>'x-euc-jp'), start_html,
h3('CGI.pmを使ったフォーム入力表示のサンプルプログラム'),
p('フォームに入力された値は次のとおりです'),
br("お名前は「$name」さん"), br("年齢は$age"),
br("CGI.pmモジュールは$useCGIpm"), $OS, $EXP;

print end_html;
```

CPANの情報検索を極めよう

CPANサイトに検索ページがあり、役に立つモジュールを見つけるために使えることは前回説明したとおりだ。しかし、CPANには膨大な数のモジュールが登録されていて、それぞれのモジュールに詳しい解説があるので、モジュールをうまく検索したり、すべて英語で表示される結果を利用したりするには慣れが必要だ。CPANを使いこなすためには、まずCPAN検索ページの使い方をよく知るのが大切だ。

CPANサイトの検索ページ



検索のページを開くと、検索文字列を入力するテキストフィールドとその下にカテゴリーが表示される。検索の方法としては、カテゴリーのリストをたどっていく方法とキーワードを入力する方法がある。検索したいモジュールのキーワードに見当がつく場合には、Googleでウェブページを検索するときのようにキーワードを使った検索の方が楽だろうし、見当がつかない場合やCPANを眺めて楽しみたい場合にはYahoo!のディレクトリー検索を使うように、カテゴリーをたどっていくことになるだろう。

探したいモジュールの作者がわかっている場合には、ページの上部にあるAuthorsをクリックすれば、作者名別に一覧表示させることができる。気に入ったモジュールがあった場合に、その作者が作ったほかのモジュールを試してみるのもいいだろう。

CPANに慣れてくると、なにか便利なモジュールがないか頻繁に見に行くことになるだろう。そういうときにはページの上にあるRecentをクリックすればいい。リリースされた新しい順にモジュールの一覧が表示される。

キーワード検索で探す

キーワード検索を行う場合には、テキストフィールドに検索したいキーワードを入力して検索対象を選ぶ。検索対象は、Modules(モジュール名)、Distributions(配布アーカイブ名)、Authors(作者名)とAll(それらすべて)から選択できる。たとえばCGIモジュールではモジュール名は「CGI」、最新版の配布アーカイブ名はCGI.pm-2.90となる(実際にダウンロードされるファイルはCGI.pm-2.90.tar.gz)。デフォルトでは検索対象がAllになっているが、同じ

ような名前のものがたくさんあるときはModuleやDistributionsを対象に検索してみるといい。

たとえば、CGIという名前をつくモジュールを検索するならば、「CGI」というキーワードをテキストフィールドに入力し、検索対象のドロップダウンリストから「Modules」を選択してから、「CPAN Search」という検索ボタンをクリックする。検索結果として、モジュール名、簡単な説明、モジュールの配布アーカイブ名、最新のリリース日、作者の一覧が表示される。

キーワード検索結果

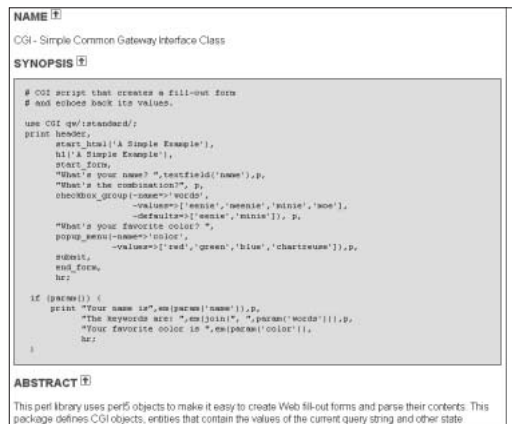


モジュールの詳細情報

検索結果のページでモジュール名をクリックするとモジュールの詳細な情報が表示される。

詳細表示はモジュールの名前から始まり、概要、サンプルからリファレンスまで記述されているので、そのモジュールについて詳しく知ることができる。

モジュールの詳細表示



Perlモジュールを多く使うようになると、使っているモジュール情報を調べるためにこのページを見ることも多くなるだろう。このページに表示される情報はモジュールの作者が作成するものなので、細かい内容にはある程度ばらつきがあり、完全に書き方が統一されているわけではない。とはいえ基本フォーマットはほぼ共通の、一般的に使われる表現になっているので、簡単に読み方を説明しておこう。基本的には、NAMEとSYNOPSISを見ればモジュールのだいたいの目的や使い方がわかるようになっている。

NAME	モジュール名と簡単な解説
SYNOPSIS	概要。多くの場合、典型的な使い方のコード例が示されるので、初めて使うモジュールではコード例をそのまま使ってみるのもいいだろう
DESCRIPTION	詳しい解説。モジュールに含まれるクラスやメソッド、細かい引数などの情報や、具体的なコードを使ったサンプルプログラムによる解説が含まれることが多い
AUTHOR	作者の情報
COPYRIGHT	著作権情報。利用する場合は必ず確認しておくべき情報だ
SEE ALSO	その他参照すべき情報。関連するモジュールや必須モジュールがある場合などはこの情報も確認しておくべきだろう

配布アーカイブ情報

検索結果のページでモジュールの配布アーカイブ名をクリックすると、そのモジュールが収録されている配布アーカイブ名が表示される。最新のCGI.pmはCGI.pm-2.90に収録されており、この配布アーカイブにはCGI.pmのほかにも複数のモジュールが収録されていることがわかる。モジュール一覧のモジュール名をクリックすると先ほどのモジュールの詳細情報が表示される。

このほかにも、検索結果のページで作者の名前をクリックすると、その作者の作成したモジュールの配布アーカイブが一覧で表示される。

配布アーカイブの表示

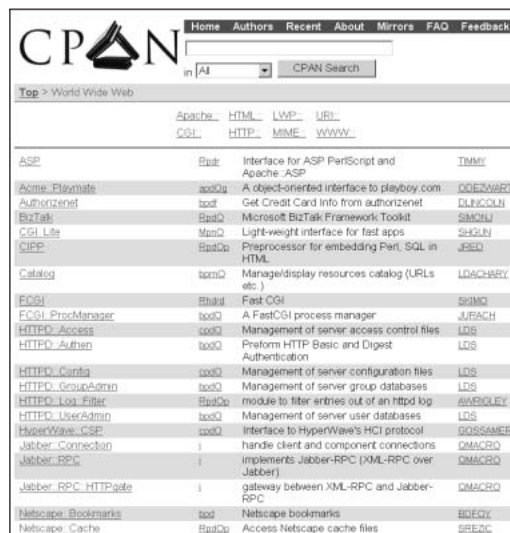


The screenshot shows the CPAN website interface. At the top, there's a search bar and navigation links like Home, Authors, Recent, About, Mirrors, FAQ, Feedback. Below that, the search results for 'CGI.pm-2.90' are displayed. It includes a 'This Release' section with download and view links, 'Other Releases' for previous versions, 'Special Files' like Changes, MANIFEST, and README, a 'Modules' list with version numbers, 'Documentation' links, and 'Other Files' like perlfaq.html and examples/index.html.

カテゴリーをたどって探す

次に、カテゴリーのリストをたどっていく場合には、自分が探しているモジュールのカテゴリーを見当をつけて探していく。たとえば「World Wide Web」のカテゴリーをクリックしてみよう。

World Wide Web カテゴリー



The screenshot shows the CPAN website interface with the search results for the 'World Wide Web' category. It lists various modules like ASP, Acme::Playmate, Authorizenet, BizTalk, CGI::Lite, CIPP, Catalog, CGI, CGI::Fast, CGI::ProcManager, HTTPD::Access, HTTPD::Authn, HTTPD::Config, HTTPD::Log::Filter, HTTPD::UserAdmin, HyperWave::CGI, Jabber::Connection, Jabber::RPC, Jabber::RPC::HTTPgate, and Netscape::Bookmarks. Each entry includes the module name, author, and a brief description.

カテゴリー表示では、そのカテゴリーのモジュールが一覧で表示される。画面上部に表示されているのはサブカテゴリーだ。

この表示結果では、各モジュールについてモジュール名、DLSIP、簡単な説明、作者が1行で表示される。モジュール名をクリックするとこれまで同様にモジュールの詳細情報が表示される。DLSIPとはモジュールの属性情報、Development Stage(開発のどの段階にあるか)、Language Used(モジュール作成に使った開発言語)、Support Level(質問や要望の受け付け方法)、Interface Style(クラス構造を含むか)、Public License(使用許諾)の5つの頭文字をとったものである。一覧では「apdOg」「RpdOP」など意味不明の文字が並ぶが、クリックするとその意味が表示されるので心配はない。

その他のCPAN検索サイト

CPANの情報を検索できるウェブサイトはほかにもいくつかあるので、たとえば検索に時間がかかるような場合にはほかの検索サイトを使うのもいいだろう。その中でも代表的なものが、Randy Kobes氏による検索ページ、通称「Kobesearch」[URL02](http://kobesearch.cpan.org/)だ。モジュールだけでなくPerl関係のドキュメントやFAQまで検索できる優れたサイトで、「最近よくダウンロードされたモジュール」[URL03](http://theoryx5.uwinnipeg.ca/CPAN/stats.html)などの情報もあり、便利だ。

[URL01](http://search.cpan.org/) <http://search.cpan.org/>

[URL02](http://kobesearch.cpan.org/) <http://kobesearch.cpan.org/>

[URL03](http://theoryx5.uwinnipeg.ca/CPAN/stats.html) <http://theoryx5.uwinnipeg.ca/CPAN/stats.html>



[インターネットマガジン バックナンバーアーカイブ] ご利用上の注意

このPDFファイルは、株式会社インプレスR&D(株式会社インプレスから分割)が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

<http://i.impressRD.jp/bn>

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容(技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど)は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者(執筆者、写真の撮影者、イラストの作成者、編集部など)が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元(株式会社インプレス R&D)、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D(当時は株式会社インプレス)と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接のおよび間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先

株式会社インプレスR&D

All-in-One INTERNET magazine 編集部

im-info@impress.co.jp