

sub get_user

目次

呼び出し方法

```
@rcdlines = &get_user("$Fm{'id'}");
```

FORM から入力された id を引数としてサブルーチン get_user を呼び出し、その戻り値をマイルコードとする。(後述)

ユーザーデータファイル名は、?????.dat なので、引数には「4桁の数字で構成される文字列」(またはそれを返す式)でなければならない。

ソースコードの解説

```
sub get_user {  
    return if $getuserflag;
```

1度このサブルーチンが実行されていれば、2度目は実行しない。

```
open(IN, "$usrdir$_[0]%.dat") || &error("ID $_[0] は存在しません");
```

引数は配列として特殊変数 @_ に格納されるので、ファイル名は \$_[0] になっている。

open 関数でユーザーデータファイル(例えば、0000.dat)を入力モードでオープンし、ファイルハンドル IN に結び付ける。

ファイルオープンに失敗すると sub error が実行され、エラーメッセージを HTML に出力する。

```
@_ = <IN>;  
close(IN);
```

ファイルハンドルに結び付けられたファイル(例えば、0000.dat)の内容を、1行ごとに配列 @_
の要素として代入する。

この時点で、以前の @_
の内容(ファイル名)は上書きされて失われる。

- \$_[0] = ファイル内容の 1 行目
- \$_[1] = ファイル内容の 2 行目
- :
- \$_[n] = ファイル内容の n+1 行目
- \$_[最後] = 未定義値

となる。

ファイルハンドルをクローズする。

```
if (!@_) { &error("読みこみエラー") }
```

配列 @_ が偽(この場合は空リスト)ならば、エラーメッセージを HTML に出力する。

```
($id,$nm,$ps,$jb,$ig,$sp,$bp,$bn,$ak,$dd,$hp,$xp,$ab,$db,
$av,$wn,$lz,$dt,$mn,$bg,$xi,$tm,$lf,$fe,$cm,$is,$ht,$fg,$rw,$kl) = split(/</>/,$_[0]);
```

配列 @_ の 0 番目の要素 (つまり、ファイルの 1 行目) を、<> を区切り文字として分割し、各スカラー変数に代入する。

各スカラー変数の内容は [リファレンスマニュアル](#) 参照。

```
&error(" パスワードが違います ") if $Fm{'ps'} ne $ps && $Fm{'ps'} ne $admpas && $Fm{'mode'} ne
'make_con';
```

次の全てに該当した場合、エラーメッセージを HTML に出力する。

- \$ps が FORM から入力されたパスワードと一致しない
- \$ps が \$admpas (管理者パスワード) と一致しない
- \$Fm{'mode'} が 'make_con' (Continue) ではない

```
$userline = shift(@_);
```

配列 @_ の先頭要素 (つまり、\$_[0] ファイル内容の 1 行目) を取り除いて、それを \$userline に代入する。

これによって、以降の配列 @_ の要素は、ファイル内容の 2 行目以降になる。

\$userline の内容は

```
• $userline = "id<> 名前 <> パスワード <>.....<>\n"
```

のように、<> で連結されている状態のもの。

```
$getuserflag = 1;
```

sub get_user を複数回実行しないようにフラグを立てる。

```
return @_;
}
```

@_ を戻り値として、サブルーチン終了。

呼び出しの項にあるとおり、この戻り値のリストが @rcdlines に代入される。

つまり、@rcdlines は、ファイル内容の 2 行目以降 = マイレコードということになる。

キーワード解説

open 関数

- open(FILEHANDLE,FILENAME) || オープン失敗時の処理 ;

FILEHANDLE ... ファイルハンドル名。大文字が原則。

FILENAME ... 開くファイル名。オプションについては割愛

ファイルを FILEHANDLE に結び付けてオープンする。

説明中に「入力モードでオープン」と書かれているが、これはスクリプト側に入力するという
ことで、「読み込み専用でファイルを開く」といったようなニュアンス。オプション指定で入力/出
力モードを指定。

ファイルをオープンした後は、スクリプト内でそのファイルにアクセスする場合はファイルハン
ドル名で指定することになっている。

|| オープン失敗時の処理 :

open 関数ではないが必須セットと言ってもいい。該当ファイルがない、ファイルハンドル名
が不適切などで、ファイルオープンに失敗した場合でも open 関数がエラーを出すわけではな
く、戻り値が未定義値になるだけで、その後の処理は何事もなかったかのように続く。SOS2
では、sub error でエラー処理を行っているので、open 関数の後に || &error("hoge") とし
てエラーを出すようになっている。解説本などには「open(略) or die 'メッセージ'」と書かれて
いることが多いが、or 演算子は Perl5 以降用なので、Perl4 以前でも使えるように || 演算子にな
っていると思われる。

<> 入力演算子

・ <FILEHANDLE>

FILEHANDLE ... ファイルハンドル (または リファレンス済みのスカラー変数)

FILEHANDLE に結び付けられたファイルの内容を 1 行ごとに読み込む。(デフォルトの状態)

改行コード(\n)までを 1 行として、左辺値がスカラーならば 1 行のみ、リストならば各行を要素
とする。

(ただし末尾の要素は未定義値)

特殊変数の値を変更することで行末文字(またはバイト数)を指定できるが、SOS2 でやる必要は
ない。

shift 関数

・ shift(@ARGV)

配列 @ARGV の先頭要素を取り除く。残りの要素は先頭方向に詰められる。

\$ARGV[0] がなくなり \$ARGV[1] から始まる配列に.....はならない。

元の \$ARGV[1] は \$ARGV[0] に、元の \$ARGV[2] は \$ARGV[1] に となる。このため、ループ
構文で shift 関数を使うときには注意が必要。

左辺値にスカラー変数を置けば、取り除かれた要素が代入される。

関連項目

・ コラム

このページのコメント