PHP Hypertext Preprocessor

Part-3

PHP関数リファレンス

第３部

PHP

第３部では、PHPをより高度に使いこなすための多くの組み込み関数および設定用オプションについて解説します。

関数の引数および返り値に関しては以下のように記述します。

・指定を省略できる引数は [] でくくって表します。
・型はint（整数）、string（文字列）、array（配列）、object（オブジェクト）、bool（論理型）、mixed（複数の型を使用可能）と記述します。
・関数がエラーを発生した場合にFALSEを返す場合は F、-1を返す場合は M、処理が成功した際にTRUEを返す場合は T を関数名の横に記しています。
・標準ではない拡張モジュールの機能については、「PHP構築オプション」として、PHPの構築時にconfigureスクリプトに指定するオプションを記述しています。
・PHPの設定は設定ファイル（php.ini）で行ないます。各モジュールの設定オプションを「php.ini設定オプション」に示しています。
・参照渡しの引数には'&'を付記しています。参照渡しの引数には関数の戻り値が代入されます。

Hypertext Preprocessor

PHP関数

Chapter-1 データベース関連

PHPはデータベースとの連携機能がすぐれており、以下のような多くの主要なリレーショナルデータベース（RDBMS）をサポートしています。

オープンソース系RDBMS	：PostgreSQL、MySQL、Firebirdなど
商用RDBMS	：Oracle、IBM DB2、Microsoft SQL Server、Sybase、InterBaseなど

そのほか、Microsoft Windows環境の標準的データベースインタフェースであるODBCやフラットファイル形式の簡易データベースdbmおよびdBASEがサポートされています。

ここでは、PHPで一般に使用される以下のデータベース関数について解説します。

PostgreSQL	MySQL	Oracle
InterBase/Firebird	IBM DB2/ODBC	Microsoft SQL Server

また、ディレクトリサービスの標準プロトコルであるLDAPについても広義のデータベースとして本章で取り上げます。

Column 持続的接続（Persistent Connection）について

データベースとの接続には通常の接続と持続的接続があります。通常の接続はPHPの実行終了時に自動的にクローズされます（通常の接続の場合であれば、明示的に接続を閉じる必要はありません）。

一方、持続的接続の場合、スクリプトの実行終了後も接続はオープンされたままとなり、同じ引数で接続をオープンする場合に再利用されます。

持続的接続を使用することにより、持続処理の時間が短縮できるため、応答速度の向上が期待できます。特にOracleのような接続処理に負荷がかかるデータベースの場合は、大きな性能向上が期待できます。反面、接続をオープンすることにより、リソースを消費するため、使用するシステムのリソースに合わせて最大接続数を調整することが必要です。持続的な接続はWebサーバにモジュールとして組み込んだPHPでのみ使用可能です。

1.1 PostgreSQL関数

PostgreSQLは、オブジェクト指向RDBMSの概念を取り入れた高機能なRDBMSです。PostgreSQLは無料で利用でき、かつ有志により早くから日本語対応が行なわれたため、LinuxやFreeBSDといったフリーのUNIXにおいて現在最も人気があるRDBMSのひとつになっています。

▶PHP構築オプション

--with-pgsql[=DIR]　　PostgreSQLサポートを有効にします。

▶php.ini設定オプション

パラメータ	デフォルト値	説明
pgsql.allow_persistent	1	持続的接続を許可する場合に1
pgsql.max_persistent	-1	持続的接続の最大数（-1の場合は制限なし）
pgsql.max_links	-1	続数の最大数（-1の場合は制限無なし）
pgsql.auto_reset_persistent	0	1の場合、接続的接続が断となった際にリセット

▶ 参考URL

PostgreSQLの開発･配布元　：**http://www.postgresql.org/**
日本PostgreSQLユーザー会：**http://www.postgresql.jp/**

▶ 注意

1. PostgreSQL関数で接続ID（cid）を省略した場合は、デフォルトの接続IDが使われます。
2. pg_connect()関数によるデータベース接続時にユーザ認証を行なわない場合、Webサーバの実行ユーザID（Vine Linux 2.5の場合はapache）が接続ユーザ名として使用されます。createuserコマンドにより該当するデータベースユーザを作成し、GRANT句により適切なアクセス権限を設定する必要があります。
3. 本節で説明するサンプルテーブルの定義を付属CD-ROMに収録してあります。テーブル作成は以下のように行ないます。

```
psql -f database_pgsql.sql
```

4. いくつかのPostgreSQLサポート関数の関数名がPHP 4.2.0以降で変更されています。旧バージョンの対応を**表3-1**に示します。

表3-1 PostgreSQLサポート関数の名前の対応

PHP 4.2.0より前	PHP 4.2.0以降	PHP 4.2.0より前	PHP 4.2.0以降
pg_cmdtuples()	pg_affected_rows()	pg_loexport()	pg_lo_export()
pg_exec()	pg_query()	pg_loimport()	pg_lo_import()
pg_fieldisnull()	pg_field_is_null()	pg_loopen()	pg_lo_open()
pg_fieldname()	pg_field_name()	pg_loread()	pg_lo_read()
pg_fieldnum()	pg_field_num()	pg_loreadall()	pg_lo_read_all()
pg_fieldprtlen()	pg_field_prtlen()	pg_lounlink()	pg_lo_unlink()
pg_fieldsize()	pg_field_size()	pg_lowrite()	pg_lo_write()
pg_freeresult()	pg_free_result()	pg_numfields()	pg_num_fields()
pg_getlastoid()	pg_last_oid()	pg_numrows()	pg_num_rows()
pg_loclose()	pg_lo_close()	pg_result()	pg_fetch_result()
pg_locreate()	pg_lo_create()		

■int pg_affected_rows(resource result)

F

INSERT、UPDATE、DELETE文により変更されたレコード数を返します。

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");
 $result = pg_query($con, "INSERT INTO otenki VALUES('2000-8-20','曇',26,20)");
 print pg_affected_rows($result); // 1を出力
?>
```

■bool pg_cancel_query(resource cid)

F T

非同期クエリをキャンセルします。

■string pg_client_encoding([resource cid])

クライアント側の文字エンコーディング（SQL_ASCII、EUC_JP、UNICODE、MULE_INTERNAL、LATIN1、SJISなど）を返します。

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");
 print pg_client_encoding($con);
?>
```

■bool pg_close([resource cid])

T

データベースへの接続を閉じます。

■resource pg_connect([string connection_string])

F

データベースに接続し、接続IDを返します。同時に複数の接続をオープンすることができます。

ユーザ認証を行なわない場合、ユーザ名としてWebサーバの実行ユーザID（Vine Linux 2.5の場合はapache）が使用されます。ユーザ認証を行なう場合、ユーザ名とパスワードも同時に指定します。

```
<?php // ユーザ認証を行なう場合の例
 $con = pg_connect("host=localhost port=5432 dbname=foo user=apache password=secret");
 if($con){
   print "接続しました。";
 }
?>
```

■bool pg_connection_busy(resource cid)

非同期クエリにおいて、ビジー状態の場合にTRUEを返します。

■bool pg_connection_reset(resource cid)

F T

接続をリセットし、再接続します。

■int pg_connection_status(resource cid)

接続のステータス（PGSQL_CONNECTION_BADまたはPGSQL_CONNECTION_OK）を返します。

■bool pg_copy_from(resource cid, string table, array rows [, string delim [,string null_as]])

T F

配列からテーブルにコピーします。デリミタdelim（デフォルトはタブ '¥t'）およびヌルを置換する文字（デフォルトは '¥¥N'）を指定可能です。

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");
 $a[0] = "2000-08-21¥t雨¥t29¥t100"; // 追加するレコードのデータ
 print pg_copy_from($con, "otenki", $a) ? "成功" : "失敗";
?>
```

■array pg_copy_to(resource cid, string table [, string delim [,string null_as]])

F

テーブルを配列にコピーします。デリミタdelim（デフォルトはタブ '¥t'）およびヌルを置換する文字（デフォルトは '¥¥N'）を指定可能です。

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");
 $a = pg_copy_to($con, "otenki"); // テーブルotenkiのデータを配列に取得
 print_r($a); // テーブルotenkiのデータを表示
?>
```

■string pg_dbname([resource cid])

接続しているデータベースの名前を返します。

```
<?php
 $con = pg_connect("host=localhost dbname=foo") or die("postgres connection failed.");
 print "データベース ".pg_dbname($con)."¥n";
?>
```

■bool pg_end_copy([resource cid])

F T

バックエンドに同期してコピーコマンドを完了します。

■string pg_escape_string(string data)

文字列をエスケープして返します。

```
<?php
 $tenki="it's rainy.";
 print pg_escape_string($tenki); // 「it''s rainy.」を出力
?>
```

■string pg_escape_bytea(string data)

bytea型のバイナリデータをエスケープして返します。

■array pg_fetch_array(resource result [, int i [, int type]])

i番目のレコード*1を取得し、配列として返します。残りのレコードがない場合にはfalseを返します。カラム名をキーとした連想配列としても結果にアクセスできます。*2

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");
 $result = pg_query($con, "SELECT * FROM otenki"); // クエリ実行
 while($a = pg_fetch_array($result)){ // クエリ結果を配列に取得
   print "{$a['day']} {$a['tenki']}¥n"; // クエリ結果を表示
 }
?>
```

■object pg_fetch_object(resource result [, int i [, int type]])

クエリ結果からi番目のレコード*1を取得し、オブジェクトとして返します。残りのレコードがない場合にはfalseを返します。カラム名をプロパティとして指定し、結果にアクセスします。*2

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");
 $result = pg_query($con, "SELECT * FROM otenki"); // クエリ実行
 while($a = pg_fetch_object($result)){ // クエリ結果をオブジェクトに取得
   print $a->day ." ". $a->tenki . "¥n"; // クエリ結果を表示
 }
?>
```

■mixed pg_fetch_result(resource result [, int i], mixed col)

クエリ結果からi番目のレコード*1のカラムcol*3のデータを取得します。

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");
 $result = pg_query($con, "SELECT * FROM otenki"); // クエリ実行

 for ($i=0; $i< pg_num_rows($result); $i++){
   $day = pg_fetch_result($result, $i, "day");
   $tenki = pg_fetch_result($result, $i, "tenki");
   print "$day $tenki¥n"; // クエリ結果を表示
 }
?>
```

■array pg_fetch_row(resource result [, int i])

クエリ結果からi番目のレコード[*1]を取得し、配列として返します。残りのレコードがない場合にはFALSEを返します。

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");
 $result = pg_query($con, "SELECT * FROM otenki"); // クエリ実行
 while($a = pg_fetch_row($result)){ // クエリ結果を配列に取得
   print implode("¥t",$a)."¥n"; // クエリ結果を表示
 }
?>
```

■bool pg_field_is_null(resource result [, int i], mixed col)

i番目[*1]のレコードのカラムcol[*3]がNULLの場合に1、そうでない場合には0を返します。

■string pg_field_name(resource result, int i)

i番目（先頭カラム：0）のカラムの名前を返します。カラムに関する情報を取得する関数を表3-2に示します。

表3-2　カラム情報を取得する関数[*4]

関数名	返り値
string pg_field_name(resource result, int i)	カラム名
int pg_field_size(resource result, int i)	カラムのサイズ[*5]
string pg_field_type(resource result, int i)	カラムの型

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");
 $result = pg_query($con, "SELECT * FROM otenki"); // クエリ実行

 for ($i=0; $i<pg_num_fields($result); $i++){
   print pg_field_name($result, $i)."¥t"; // カラム名を表示
   print pg_field_type($result, $i)."¥t"; // カラムの型を表示
   print pg_field_size($result, $i)."¥n"; // カラムのサイズを表示
 }
?>
```

int pg_field_num(resource result, string col)

カラムcolのカラム番号を返します。

int pg_field_prtlen(resource result [, int i], mixed col)

i番目[*1]のレコードのカラムcol[*3]のデータの表示長を返します。

bool pg_free_result(resource result)

F T

結果保持用のメモリを解放します。メモリはスクリプト実行終了時に自動的に解放されるため、通常は不要です。

resource pg_get_result([resource cid])

F

非同期クエリの結果を取得し、結果リソースを返します。

string pg_host([resource cid])

接続しているホストの名前を返します。

```
<?php
 $con = pg_connect("host=localhost dbname=foo user=apache") or die("postgres
connection failed.");
 print "ホスト名 ".pg_host($con)."¥n";
 print "ポート番号 ".pg_port($con)."¥n";
?>
```

string pg_last_error([resource cid])

F

直近のエラーメッセージを取得します。

*1
先頭レコードは0、省略時は次のレコードとなります。

*2
引数typeに定数PGSQL_ASSOC、PGSQL_NUM、PGSQL_BOTHを指定することが可能です。この場合、返り値の配列はそれぞれ連想配列、数値添字配列、両方でアクセス可能となります。

*3
カラム名またはカラム番号（先頭カラム：0）で指定します。

*4
引数iは、カラム番号（先頭カラム：0）です。

*5
データ長が-1の場合は可変長データです。

■string pg_last_notice(resource cid)

F

直近の通知メッセージを取得します。

■string pg_last_oid(resource result)

F

直前のINSERTクエリにより挿入されたレコードのオブジェクトID（oid）を返します。

■bool pg_lo_close(resource lobj)

ラージオブジェクトを閉じます。*6

■int pg_lo_create([resource cid])

F

ラージオブジェクトを生成し、オブジェクトID（oid）を返します。*6

■bool pg_lo_export([resource cid,] int oid, string filename)

F T

ラージオブジェクトをファイルシステムにエクスポートします。*6

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");

 $result = pg_query($con, "SELECT * from image_db");
 $im = pg_fetch_array($result); $oid = $im['id']; // オブジェクトID取得

 pg_query($con, "BEGIN");
 pg_lo_export($con, $oid, "dog.jpg"); // ラージオブジェクトをファイルに書き出す
 pg_query($con, "COMMIT");
?>
```

■int pg_lo_import([resource cid,] string file)

F

ラージオブジェクトをファイルからインポートし、オブジェクトIDを返します。*6

■resource pg_lo_open([resource cid,] int oid, string mode)

F

ラージオブジェクトをオープンし、ラージオブジェクトのリソースを返します。ラージオブジェクトのリソースをクローズする前に接続を閉じてはいけません。modeには読み書きモード（r、w、r+、w+ のいずれか）を指定します。*1

string pg_lo_read(resource lobj [, int len])

ラージオブジェクトから最大lenバイト（省略時は8Kバイト）読み込み、文字列として返します。*6

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");

 $result = pg_query($con, "SELECT * from image_db");
 $im = pg_fetch_array($result); $oid = $im['id']; // オブジェクトID取得

 pg_query($con, "BEGIN");
 $lobj = pg_lo_open($con, $oid, "r") or die("large object open failed");
 header('Content-type: image/jpeg');
 print pg_lo_read($lobj,40960); // ラージオブジェクト読み込み
 pg_lo_close($lobj);
 pg_query($con, "COMMIT");
?>
```

int pg_lo_read_all(resource lobj)

ラージオブジェクトを読み込んで出力します。読み込んだバイト数を返します。主にイメージなどのバイナリデータを送信するために使用します。*6

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");

 $result = pg_query($con, "SELECT * from image_db");
 $im = pg_fetch_array($result); $oid = $im['id']; // オブジェクトID取得

 pg_query($con, "BEGIN");
 $lobj = pg_lo_open($con, $oid, "r") or die("large object open failed");
 header('Content-type: image/jpeg');
 pg_lo_read_all($lobj); // ラージオブジェクトを読み込んで出力
 pg_lo_close($lobj);
 pg_query($con, "COMMIT");
?>
```

*6
ラージオブジェクト（lo）インタフェースを使用する場合、クエリをトランザクションブロック「BEGIN～END（またはCOMMIT）」の中に入れる必要があります。

■bool pg_lo_seek(resource lobj, int offset [, int whence])

ラージオブジェクトにおいてポインタの位置をoffsetに移動します。*6

■int pg_lo_tell(resource lobj)

ラージオブジェクトにおいてポインタの位置を返します。*6

■bool pg_lo_unlink([resource cid,] string oid)

F T

ラージオブジェクトを削除します。*6

■int pg_lo_write(resource lobj, string buf [, int len])

文字列bufをラージオブジェクトに最大lenバイト書き込み、実際に書き込んだバイト数を返します。*6

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");
 pg_query($con, "BEGIN");
 $oid = pg_lo_create($con);
 $obj = pg_lo_open($con, $oid, "w");

 $filename = "mydog.jpg";
 $fp = fopen($filename,"r");
 $image = fread($fp, filesize($filename));
 fclose($fp);

 $size = pg_lo_write($obj, $image);
 pg_lo_close($obj);
 pg_query($con, "COMMIT");

 pg_query($con, "INSERT INTO image_db VALUES($oid,'".$filename."')");
                                                                  // oidを保存
 print "$size バイト書き込みました。";
?>
```

■int pg_num_fields(resource result)

クエリ結果のカラム数を返します。

■int pg_num_rows(resource result)

クエリ結果のレコード数を返します。

■string pg_options([resource cid])

接続に関するオプションを文字列として返します。

■resource pg_pconnect([string connection_string])

F

持続的な接続をオープンし、接続IDを返します。引数の指定方法はpg_connectと同じです。

■int pg_port([resource cid])

接続しているポートの番号を返します。

■bool pg_put_line([resource cid,] string query)

F T

バックエンドのサーバに文字列を送信します。

■resource pg_query([resource cid,] string query)

F

クエリqueryを実行し、結果IDを返します。

■string pg_result_error(resource result)

クエリ結果に関するエラーメッセージを取得します。

■int pg_result_status(resource result)

クエリ結果のステータスとして、PGSQL_EMPTY_QUERY、PGSQL_COMMAND_OK、PGSQL_TUPLES_OK、PGSQL_COPY_OUT、PGSQL_COPY_IN、PGSQL_BAD_RESPONSE、PGSQL_NONFATAL_ERROR、PGSQL_FATAL_ERROR のいずれかを返します。

```
<?php
 $status = array(PGSQL_EMPTY_QUERY=>"EMPTY_QUERY",
                 PGSQL_COMMAND_OK=>"COMMAND_OK",
                 PGSQL_TUPLES_OK=>"TUPLES_OK");
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");
 $result = pg_query($con, "SELECT * FROM otenki"); // クエリを実行
 print $status[pg_result_status($result)]; // クエリステータスを出力
?>
```

■bool pg_send_query(resource cid, string query)

F T

非同期クエリを送信します。

■int pg_set_client_encoding([resource cid,] string encoding)

クライアントのエンコーディングを設定します。SQL_ASCII、EUC_JP、UNICODE、MULE_INTERNAL、SJISなどを指定します。

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");
 pg_set_client_encoding($con, "SJIS"); // 出力用文字コードをシフトJISに設定
 $result = pg_query($con, "SELECT * FROM otenki"); // クエリ実行
 while($a = pg_fetch_array($result)){ // クエリ結果を配列に取得
   print "{$a['day']} {$a['tenki']}¥n"; // クエリ結果を表示（シフトJISで出力）
 }
?>
```

■bool pg_trace(string file [, string mode [, resource cid]])

F T

デバッグ用にフロントエンド/バックエンド間の通信をファイルに保存します。fileおよびmodeの指定方法はfopen()と同じです。

```
<?php
 $con = pg_connect("dbname=foo") or die("postgres connection failed.");
 pg_trace("/tmp/pgsql.log", "w", $con); // 通信ログをとる
 $result = pg_query($con, "SELECT * FROM otenki"); // クエリの実行
 pg_untrace($con); // 通信ログの終了
?>
```

■string pg_tty([resource cid])

サーバ側のデバッグ出力が送られるtty名を返します。

■bool pg_untrace([resource cid])

バックエンドとの通信のトレースを停止します。

1.2 MySQL関数

MySQLは、高速・高機能なオープンソースのRDBMSで、特にWebアプリケーション用として人気があります。バージョン3.23.20以降GPL2（GNU Public License 2）に基づき配布されています。

▶PHP構築オプション

--with-mysql[=DIR]	MySQLサポートを有効にします（デフォルトで有効）。
--without-mysql	MySQLサポートを有効にします。

▶php.ini設定オプション

パラメータ	デフォルト値	説明
mysql.allow_persistent	1	持続的接続を許可する場合に1
mysql.max_persistent	-1	持続的接続の最大数（-1の場合は制限なし）
mysql.max_links	-1	接続数の最大数（-1の場合は制限なし）
mysql.default_port	NULL	接続ポートのデフォルト値
mysql.default_socket	NULL	接続ソケットのデフォルト値
mysql.default_user	NULL	接続ユーザのデフォルト値
mysql.default_password	NULL	接続パスワードのデフォルト値

▶参考URL

MySQLの開発・配布元　：**http://www.mysql.com/**
日本MySQLユーザー会：**http://www.mysql.gr.jp/**

▶注意

1. MySQL関数で接続リソース（cid）を省略した場合、直近にオープンされた接続リソースを使用します。一度も接続がオープンされていない場合は、新規に接続をオープンします。

2. 本節で説明するサンプルテーブルの定義を付属CD-ROMに収録してあります。テーブル作成は以下のように行ないます。

```
mysql test < database_mysql.sql
```

■int mysql_affected_rows([int cid])

直近のINSERT、UPDATE、DELETE文により変更されたレコードの数を返します。

```
<?php
 mysql_connect("localhost") or die("connection failed.");
 mysql_select_db("test");
 $result = mysql_query("INSERT INTO otenki VALUES('2000-8-20','曇',26,20)");
 print mysql_affected_rows(); // 1を出力
?>
```

■bool mysql_close([int cid])

T

接続を閉じます。通常は自動的に閉じられるため、不要です。

■resource mysql_connect([string hostname[:port][:/path/to/socket] [,string username[, string password[, bool new]]]])

F

接続をオープンし、接続リソースを返します。引数はすべてオプションで、省略時はデフォルト値（localhost、サーバプロセスの所有者のユーザ名、空のパスワード、FALSE）が使用されます。同じ引数で接続がすでにオープンされている場合、そのリソースが返されます。ただし、引数newにTRUEを指定した場合には、常に新規に接続がオープンされます。接続後は、**表3-3**に示す関数により接続に関する情報を取得できます。

表3-3　接続に関する情報を取得する関数

関数名	返り値
string mysql_get_client_info()	MySQLクライアントライブラリのバージョン
string mysql_get_host_info([resource cid])	MySQLサーバのホスト名とソケット
int mysql_get_proto_info([resource cid])	プロトコルのバージョン
string mysql_get_server_info([resource cid])	MySQLサーバのバージョン

```
<?php
 mysql_connect("localhost","apache","secret") or die("connection failed.");
 print mysql_get_client_info()."¥n"; // MySQLクライアントライブラリのバージョン
 print mysql_get_host_info()."¥n"; // MySQLサーバのホスト名とソケット
 print mysql_get_proto_info()."¥n"; // プロトコルのバージョン
 print mysql_get_server_info()."¥n"; // MySQLサーバのバージョン
?>
```

■bool mysql_data_seek(resource result, int i)

F T

クエリ結果の内部ポインタをi番目（先頭レコード：0）のレコードに移動します。

```
<?php
 mysql_connect("localhost") or die("connection failed.");
 mysql_select_db("test");
 $result = mysql_query("SELECT * FROM otenki");
 mysql_data_seek($result,mysql_num_rows($result)-1); // 最後のレコードに移動
 $a = mysql_fetch_array($result); // クエリ結果を取得
 print_r($a);
?>
```

■int mysql_errno([resource cid])

直近に発生したエラーのエラー番号を返します。

■string mysql_error([resource cid])

直近に発生したエラーのエラー文字列を返します。

■string mysql_escape_string(string query)

MySQLクエリにおける特殊文字をエスケープ処理して返します。

```
<?php
 $tenki="it's rainy.";
 print mysql_escape_string($tenki); // 「it¥'s rainy.」を出力
?>
```

■array mysql_fetch_array(resource result [, int type])

F

クエリ結果からレコードを取得し、配列として返します。typeに返り値の型：MYSQL_ASSOC（連想配列）、MYSQL_NUM（数値配列）、MYSQL_BOTH（両方、デフォルト）を指定可能です。*7

```
<?php
 mysql_connect("localhost") or die("connection failed.");
 mysql_select_db("test");
 $result = mysql_query("SELECT * FROM otenki");
 while($a = mysql_fetch_array($result)){ // クエリ結果を取得
   print_r($a);
 }
?>
```

■object mysql_fetch_field(resource result [, int k])

F

k番目（先頭カラム：0）のカラム（省略時はカレントのカラム）に関する情報を取得し、**表3-4**に示すプロパティを有するオブジェクトとして返します。

カラムに関する情報は、**表3-5**に示す関数で取得可能です。

表3-4　mysql_fetch_field()で取得されるプロパティ

プロパティ	カラム（カラム）情報
name	カラム名
table	テーブル名
max_length	最大長
not_null	NOT NULLの場合に1
primary_key	PRIMARY KEYの場合に1
multiple_key	MULTIPLE KEYの場合に1
unique_key	UNIQUE KEYの場合に1
numeric	numericの場合に1
blob	blobの場合に1
type	カラム型（int、real、date、string、blob）
unsigned	符号なし（unsigned）の場合に1
zerofill	ZEROFILLの場合に1

表3-5　カラム情報を取得する関数

関数名	返り値
string mysql_field_name(resource result, int k)	カラム名
string mysql_field_table(resource result, int k)	テーブル名
string mysql_field_type(resource result, int k)	型（int、real、string、blob）
string mysql_field_flags(resource result, int k)	設定されているフラグ *8
int mysql_field_len(resource result, int k)	カラムの長さ

*7
続けてコールした場合、次のレコードが返されます。レコードが存在しない場合、FALSEが返されます。

*8
次のフラグを返します。

not_null
primary_key
unique_key
multiple_key
blob
unsigned
zerofill
binary
enum
set
auto_increment
timestamp

```
<?php // mysql_fetch_field()のサンプル
mysql_connect("localhost") or die("connection failed.");
mysql_select_db("test");
$result = mysql_query("SELECT * FROM otenki");

for ($i=0; $i<mysql_num_fields($result);$i++){
 $obj = mysql_fetch_field($result); // カラム情報を取得
 print $obj->name . "¥t";
 print $obj->table . "¥t";
 print $obj->type . "¥t";
 print $obj->not_null ? "not_null¥t" : "¥t";
 print $obj->primary_key ? "primary_key¥t" : "¥t";
 print $obj->max_length . "¥n";
}
?>
```

```
<?php // mysql_field_*()のサンプル
mysql_connect("localhost") or die("connection failed.");
mysql_select_db("test");
$result = mysql_query("SELECT * FROM otenki");

for ($i=0; $i<mysql_num_fields($result);$i++){
 print mysql_field_name($result, $i)."¥t";
 print mysql_field_table($result, $i)."¥t";
 print mysql_field_type($result, $i)."¥t";
 print mysql_field_flags($result, $i)."¥t";
 print mysql_field_len($result, $i)."¥n";
}
?>
```

■array mysql_fetch_assoc(resource result)

F

クエリ結果からレコードを取得し、連想配列として返します。レコードを取得できない場合はFALSEを返します。*7

■array mysql_fetch_lengths(resource result)

F

直近に取得されたレコードの各カラムのデータ長を0から始まる配列に代入して返します。NULLを含むバイナリデータの場合も正しい長さを返します。

```
<?php
 mysql_connect("localhost") or die("connection failed.");
 mysql_select_db("test");
 $result = mysql_query("SELECT * FROM otenki");
 while($a = mysql_fetch_array($result)){ // レコードを取得
   $len = mysql_fetch_lengths($result); // 各カラムのデータ長を取得
   print_r($len);
 }
?>
```

■object mysql_fetch_object(resource result)

F

クエリ結果からレコードを取得し、オブジェクトとして返します。*7

```
<?php
 mysql_connect("localhost") or die("connection failed.");
 mysql_select_db("test");
 $result = mysql_query("SELECT * FROM otenki");
 while($a = mysql_fetch_object($result)){ // クエリ結果を取得
   print_r($a);
 }
?>
```

■array mysql_fetch_row(resource result)

F

クエリ結果からレコードを取得し、0から始まる配列として返します。*7

■bool mysql_field_seek(resource result, int k)

F T

カラムオフセットをk（先頭カラム：0）に設定します。

■bool mysql_free_result(resource result)

F T

クエリ結果を保持するメモリを解放します。通常、スクリプト終了時に自動的に解放されるため不要です。

■int mysql_insert_id([resource cid])

直近のINSERT文においてAUTO_INCREMENTEDカラムで自動的に生成されたIDを返します。

resource mysql_list_dbs([resource cid])

F

データベース名のリストを取得し、結果リソースを返します。

```
<?php
 mysql_connect("localhost") or die("connection failed.");
 $result = mysql_list_dbs();
 while($a = mysql_fetch_array($result)){ // データベースのリストを表示
   print_r($a);
 }
?>
```

resource mysql_list_fields(string database, string table [, resource cid])

F

指定したデータベーステーブルのカラム情報を取得し、結果リソースを返します。

```
<?php
 mysql_connect("localhost") or die("connection failed.");
 $fields = mysql_list_fields("test","otenki");
 for ($i=0; $i < mysql_num_fields($fields); $i++){
   print mysql_field_name($fields, $i)."¥n"; // カラム名の一覧を表示します。
  }
?>
```

resource mysql_list_tables(string database [, resource cid])

F

データベースにあるテーブルのリストを取得し、結果リソースを返します。

```
<?php
 mysql_connect("localhost") or die("connection failed.");
 $result = mysql_list_tables("test");
 while($a = mysql_fetch_array($result)){ // テーブル名のリストを表示
   print_r($a);
 }
?>
```

int mysql_num_fields(resource result)

クエリ結果のカラム数を返します。*9

■int mysql_num_rows(resource result)

クエリ結果のレコード数を返します。

■resource mysql_pconnect([string hostname[:port][:/path/to/socket] [, string user [, string password]]])

持続的な接続をオープンし、接続リソースを返します。引数の指定方法は、mysql_connect()と同じです。

■mixed mysql_query(string query [, resource cid [, int mode]])

F

アクティブなデータベースにクエリを送信し、SELECT、SHOW、EXPLAIN、DESCRIBE文の場合は結果リソース、それ以外は結果を表す値（TRUEまたはFALSE）を返します。クエリ結果のバッファリング動作をmodeに指定可能（MYSQL_STORE_RESULT：有効（デフォルト）、MYSQL_USE_RESULT：無効）です。

■mixed mysql_result(resource result, int i [, mixed field])

F

i番目のレコードのカラムfield（デフォルト：先頭カラム）のデータを返します。カラムは、**オフセット**、**カラム名**、**テーブル名.カラム名**で指定できます。*10

```
<?php
 mysql_connect("localhost") or die("connection failed.");
 mysql_select_db("test");
 $result = mysql_query("SELECT * FROM otenki");
 for($i=0; $i<mysql_num_rows($result); $i++){ // クエリ結果を取得して表示
   for ($j=0; $j<mysql_num_fields($result)-1; $j++){
     print mysql_result($result, $i, $j)."¥t";
   }
   print "¥n";
 }
?>
```

■bool mysql_select_db(string database [, resource cid])

F T

アクティブなデータベースを設定します。

■**resource mysql_unbuffered_query(string query [, resource cid [, int mode]])**

アクティブなデータベースにクエリを送信し、結果を返します。引数および動作はmysql_query()と同じですが、結果のバッファリングは行ないません。

1.3 InterBase/FireBird関数

InterBaseはBorland製の商用RDBMSです。InterBase 6には有償の商用版以外に無償配布されているオープンソース版が存在します。PHPは、InterBaseおよびオープンソースの派生版であるFireBirdもサポートしています。

▶PHP構築オプション

--with-interbase[=DIR]　InterBase/FireBirdのサポートを有効にします。

▶php.ini設定オプション

パラメータ	デフォルト値	説明
ibase.allow_persistent	1	持続的接続を許可する場合に1
ibase.max_persistent	-1	持続的接続の最大数(-1の場合は制限なし)
ibase.max_links	-1	接続数の最大数(-1の場合は制限なし)
ibase.default_user	NULL	接続ユーザ名のデフォルト値
ibase.default_password	NULL	接続パスワードのデフォルト値
ibase.timestampformat	%m/%d/%Y %H:%M:%S	TIMESTAMP型のカラムの出力型式
ibase.dateformat	%m/%d/%Y	DATE型のカラムの出力型式
ibase.timeformat	%H:%M:%S	TIME型のカラムの出力型式

▶参考URL

InterBaseの開発元(Borland社)：**http://www.borland.co.jp/interbase/**
FireBirdの開発・配布元：**http://firebird.sourceforge.net/**

▶注意

1. 接続ID(conn)を省略した場合は、デフォルトの接続が使われます。
2. 本節で説明するサンプルテーブルの定義を付属CD-ROMに収録してあります(database_interbase.sql)。テーブル作成はisqlにより行ないます。

*9
mysql_list_fields()のサンプルを参照してください。

*10
すべてのカラムを取得する場合は、mysql_fetch_array()のほうが効率的です。

■bool ibase_close([resource conn])

F T

データベースとの接続を閉じます。

■bool ibase_commit([resource conn,] resource tr)

T F

トランザクションをコミットします。trはトランザクションのリソースです。

■resource ibase_connect(string database [, string user [, string password [, string charset [, int buffers [, int dialect [, string role]]]]]])

F

サーバへの接続をオープンします。リモートのサーバに接続する場合、使用する接続プロトコルに応じてデータベース名の前にhostname:（TCP/IP）、//hostname/（NetBEUI）、hostname@（IPX/SPX）を指定します。charsetはデフォルトの文字セット、buffersはデータベースバッファの数、dialectはSQL言語仕様のデフォルト値です。引数を省略した場合、デフォルト値が使用されます。

■string ibase_errmsg()

F

直近のエラーメッセージを返します。

```
<?php
 $conn = @ibase_connect("/tmp/foo.gdb", "SYSDBA", "masterkey");
 if (!$conn){ // 接続できなかった場合
   die(ibase_errmsg()); // エラーメッセージを出力して終了
 }
 print "データベースに接続しました。";
?>
```

■resource ibase_execute(resource query [, mixed bind_args , ...])

F

ibase_prepare()によりパースされたクエリを実行します。SELECT文のような結果を返すクエリーの場合は、結果セットのリソースを返し、自動的にカーソルがオープンされます。それ以外の場合は、TRUEを返します。

```
<?php
ibase_connect("/tmp/foo.gdb", "SYSDBA", "masterkey") or die("connection failed");
$stmt = ibase_prepare("INSERT INTO otenki VALUES(?, ?, ?, ?)");

$data[] = array("2000-8-20","晴れ",29,20);
$data[] = array("2000-8-21","雨",25,100);

foreach ($data as $value) { // 挿入するデータを設定し、クエリを実行
  list($day,$tenki,$ondo,$uryou) = $value;
  ibase_execute($stmt, $day, $tenki, $ondo, $uryou);
}
?>
```

■object ibase_fetch_object(resource result [, int blob_flag])

F

1件分のレコードをオブジェクトとして取得します。

```
<?php
 $conn = ibase_connect("/tmp/foo.gdb", "SYSDBA", "masterkey") or die("connection
failed");
 $result = ibase_query($conn, "SELECT * FROM otenki"); // クエリを実行
 ibase_timefmt("%F",IBASE_DATE); // 日付の出力型式を"2000-08-03"のようにする
 while ($row = ibase_fetch_object($result)) { // レコードを取得、表示
   print $row->HIZUKE . "¥t" . $row->TENKI ."¥n";
 }
?>
```

■array ibase_fetch_row(resource result [, int blob_flag])]

F

結果セットから1件分のレコードを配列として取得します。

```
<?php
 $conn = ibase_connect("/tmp/foo.gdb", "SYSDBA", "masterkey") or die("connection
failed");
 $result = ibase_query($conn, "SELECT * FROM otenki"); // クエリを実行
 ibase_timefmt("%F",IBASE_DATE); // 日付の出力型式を"2000-08-03"のようにする
 while ($row = ibase_fetch_row($result)) { // レコードを取得、表示
   print_r($row);
 }
?>
```

■array ibase_field_info(resource result, int i)

F

i番目のカラムに関する情報を連想配列として返します。連想配列のキーは、"name"、"alias"、"relation"、"length"、"type"となります。

```
<?php
$conn = ibase_connect("/tmp/foo.gdb", "SYSDBA", "masterkey") or die("connection
failed");
$result = ibase_query($conn, "SELECT * FROM otenki"); // クエリを実行
for ($i=0; $i < ibase_num_fields($result); $i++){
  $info = ibase_field_info($result, $i); // カラム情報を取得
  print_r($info);
}
?>
```

■bool ibase_free_query(resource query)

T

ibase_prepare()によりパースされたクエリのリソースを解放します。

■bool ibase_free_result(resource result)

T

クエリ結果を保持するメモリを解放します。

■int ibase_num_fields(resource result)

F

クエリ結果におけるカラム数を取得します。

■resource ibase_pconnect(string database [, string username [, string password [, string charset [, string role]]]])

F

スクリプト終了時に接続を閉じないことと、同じ引数ですでに接続がオープンされている場合にはその接続を使用すること以外は、ibase_connect()と同じです。

■resource ibase_prepare([resource conn,] string query)

F

クエリをパースし、クエリのリソースを返します。

■resource ibase_query([resource conn,] string query [, int bind_args]])

F

クエリを実行し、SELECT文のような結果を返すクエリーは結果セットのリソース、それ以外はTRUEを返します。

■bool ibase_rollback([resource conn,] resource tr)

T F

トランザクションをロールバックします。trはトランザクションのリソースです。

■bool ibase_timefmt(string format [, int type])

F T

timestamp、date、time型カラムの出力フォーマットをC言語のstrftime()関数形式で設定します。設定するカラムをtypeにIBASE_TIMESTAMP（デフォルト）、IBASE_DATE、IBASE_TIMEのように指定可能です。*11

■resource ibase_trans([int trans [, resource conn]])

F

トランザクションを開始し、トランザクションのリソースを返します。transには動作モード（IBASE_READ、IBASE_COMMITTED、IBASE_CONSISTENCY、IBASE_NOWAIT）を指定します。

*11
ibase_fetch_object()の例を参照してください。

1.4 Oracle関数

Oracleは実績のあるRDBMSであり、多くの実用システムで使用されています。Oracle 8/9はLinux版もリリースされています。

▶PHP構築オプション

--with-oci8[=DIR]　OCI8（Oracle 8以降の接続API）のサポートを有効にします。Net8（SQL*Net）がインストールされているディレクトリを指定します。

--enable-sigchild　PHP独自のSIGCHLDハンドラを有効にします。

Oracle 8.1.7の場合の例：./configure --with-oci8=/u01/app/oracle/product/8.1.7 --enable-sigchild [そのほかのオプション]

Apache起動オプション（Vine Linuxの場合：/etc/sysconfig/apache）に以下の環境変数の設定を追加します。なお、ディレクトリ構成およびデータベースSIDはシステムの設定により異なります。

```
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/8.1.7
export ORACLE_SID=orcl
export NLS_LANG=Japanese_Japan.JA16EUC
export ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

▶参考URL

Oracle Technology Network Japan：**http://technet.oracle.co.jp/**

▶注意

本節で説明するサンプルテーブルの定義を付属CD-ROMに収録してあります。サンプルを実行する場合は、sqlplusなどによりdatabase_oracle.sqlを実行してください。

*12
抽象データ型（LOB、ROWID、BFILE）をバインドする場合、ocinewdescriptor()によりリソースを確保しておく必要があります。

*13
typeには下記のディスクリプタの種類を指定します。

- OCI_B_FILE（バイナリファイル）
- OCI_B_CFILE（テキストファイル）
- OCI_B_CLOB
- OCI_B_BLOB
- OCI_B_ROWID

■int ocibindbyname(int stmt, string name, mixed &var, int len [, int type])

PHP変数varをOracle変数name *12 にバインドします。lenはバインド時の最大長（-1の場合は現在の長さ）を設定します。*13

```
<?php
 $conn = ocilogon("scott","tiger", "orcl") or die("connection faild.");
 $stmt = ociparse($conn,"INSERT INTO otenki VALUES(:day,:tenki,:ondo,:uryou)");
                                                                    // クエリをパース
 // Oracle変数をPHP変数にバインド
 ocibindbyname($stmt,":day",$day, 12);
 ocibindbyname($stmt,":tenki",$tenki, 6);
 ocibindbyname($stmt,":ondo",$ondo, 4);
 ocibindbyname($stmt,":ryou",$uryou, 6);

 $data[] = array("2000-8-20","晴れ",29,20);
 $data[] = array("2000-8-21","雨",25,100);

 foreach ($data as $value) { // 挿入するデータを設定し、クエリを実行
   list($day,$tenki,$ondo,$uryou) = $value;
   ociexecute($stmt);
 }
?>
```

■bool ocicancel(resource stmt)

T F

カーソルからの読み込みを中止します。

■bool ocicommit(resource conn)

F T

カレントのトランザクションをコミットします。

■int ocidefinedbyname(int stmt, string col, mixed &var [, int type])

SELECT結果のカラム（大文字で指定）[*12]をPHP変数（参照で指定）で参照可能とします。[*13]

```
<?php
 $conn = ocilogon("scott","tiger", "orcl") or die("connection faild.");
 $stmt = ociparse($conn,"SELECT * FROM otenki"); // クエリをパース

 ocidefinebyname($stmt, "DAY", $day); // カラムDAYを$dayにバインド
 ocidefinebyname($stmt, "TENKI", $weather); // カラムTENKIを$weatherにバインド
 ociexecute($stmt); // クエリを実行

 while (ocifetch($stmt)) { // レコードを取得、表示
   print "$day $weather¥n";
 }
?>
```

■array ocierror([resource stmt|conn|global])

ステートメント（stmt）、接続（conn）、グローバルリソース（global）に関する直近のエラーを返します。エラーが発生していない場合はFALSEを返します。返される配列は、'code'（エラーコード）、'message'（エラーメッセージ）の2要素となります。*14

■bool ociexecute(resource stmt [, int mode])

F T

パース済みのクエリを実行します。modeに実行モード（デフォルトはOCI_COMMIT_ON_SUCCESS）を指定可能です。カラムに関する情報を**表3-6**に示す関数により取得できます。

```
<?php
 $conn = ocilogon("scott","tiger", "orcl") or die("connection faild.");
 $stmt = ociparse($conn,"SELECT * FROM otenki");
 ociexecute($stmt);

 for ($i=1; $i<=ocinumcols($stmt);$i++){ // カラム情報を出力
   print ocicolumnname($stmt, $i)."¥t"; // カラム名
   print ocicolumntype($stmt, $i)."¥t"; // カラム型
   print ocicolumntyperaw($stmt, $i)."¥t"; // カラム型（数値）
   print ocicolumnsize($stmt, $i)."¥t"; // カラム長
   print ocicolumnscale($stmt, $i)."¥t"; // スケール
   print ocicolumnisnull($stmt, $i) ? "is_null¥t" : "¥t"; // IS_NULL
   print ocicolumnprecision($stmt, $i)."¥n"; // 有効桁数
 }
?>
```

表3-6　カラム情報を取得する関数

関数名と引数 *15	説明
bool ocicolumnisnull(resource stmt, mixed column)	カラムがNULLの場合にTRUEを返す
string ocicolumnname(resource stmt, mixed column)	カラムの名前を返す
int ocicolumnprecision(resource stmt, mixed column)	カラムの有効桁数を返す
int ocicolumnscale(resource stmt, mixed column)	カラムのスケールを返す
int ocicolumnsize(resource stmt, mixed column)	カラムの長さを返す
string ocicolumntype(resource stmt, mixed column)	カラムの型 *16 を返す
int ocicolumntyperaw(resource stmt, mixed column)	カラムの型を表す整数を返す

*16
DATE
NUMBER
LONG
RAW
LONG RAW
VARCHAR
REFCURSOR
CHAR
BLOB
CLOB
BFILE
ROWID

*14
ocilogon()のサンプルを参照してください。

*15
カラム名columnにはカラム番号（先頭カラム:1）またはカラム名を指定します。

bool ocifetch(resource stmt)

F T

クエリ結果から次のレコードを取得します。

```
<?php
 $conn = ocilogon("scott","tiger", "orcl") or die("connection faild.");
 $stmt = ociparse($conn,"SELECT * FROM otenki"); // クエリをパース
 ociexecute($stmt); // クエリを実行

 while (ocifetch($stmt)){ // レコードを取得
   for ($i=1; $i<=ocinumcols($stmt); $i++){
     print ociresult($stmt, $i)."¥t"; // 各カラムのデータを表示
   }
   print "¥n";
 }
?>
```

int ocifetchinto(resource stmt, array &result [,int mode])

F

クエリ結果から次のレコードを配列に取得し、カラム数を返します。modeに**表3-7**に示すオプションを指定可能です。

表3-7 レコード取得時のモード

modeの値	返り値
OCI_ASSOC	連想配列を返す
OCI_NUM	通常配列（先頭カラム：0）を返す（デフォルト）
OCI_BOTH	連想配列と通常配列を返す
OCI_RETURN_NULLS	NULLカラムを返す
OCI_RETURN_LOBS	記述子の代わりにLOBのIDを返す

```
<?php
 $conn = ocilogon("scott","tiger", "orcl") or die("connection faild.");
 $stmt = ociparse($conn,"SELECT * FROM otenki"); // クエリをパース
 ociexecute($stmt); // クエリを実行

 while (ocifetchinto($stmt, $row, OCI_ASSOC)) { // レコードを取得、表示
   print_r($row);
 }
?>
```

■int ocifetchstatement(resource stmt, array output [,int skip [,int maxrows [,int flags]]])

クエリ結果からすべてのレコードを二次元配列に取得し、取得したレコード数を返します。skipに読み飛ばすレコード数、maxrowsに最大レコード数（-1（デフォルト）の場合はすべて）を指定することが可能です。flagsにOCI_FETCHSTATEMENT_BY_COLUMN（デフォルト）を指定するとカラム単位の配列、OCI_FETCHSTATEMENT_BY_ROWを指定するとレコード単位の配列となります。

```
<?php
 $conn = ocilogon("scott","tiger", "orcl") or die("connection faild.");
 $stmt = ociparse($conn,"SELECT * FROM otenki"); // クエリをパース
 ociexecute($stmt); // クエリを実行

 $nrows = ocifetchstatement($stmt, $record); // クエリ結果を二次元配列$recordに取得
 for ($i=0; $i<$nrows; $i++){ // レコードを表示
   print $record['DAY'][$i]."¥t".$record['TENKI'][$i]."¥n";
 }
?>
```

■bool ocifreestatement(resource stmt)

T

ステートメントに関連するリソースを解放します。

■resource ocilogon(string user, string password [, string database])

F

データベースへの接続をオープンし、接続リソースを返します。databaseには、ローカルなOracleインスタンス名またはtnsnames.oraに記述した接続エントリ名を指定します。省略時は、環境変数ORACLE_SIDまたはTWO_TASKの値を使用します。

```
<?php
 $conn = @ocilogon("scott","tiger", "orcl"); // Oracleデータベースに接続
 if (!$conn){ // 接続エラーの場合
   $err = ocierror(); // エラー内容を取得
   print "エラーコード: {$err['code']}¥n";
   print "エラーメッセージ: {$err['message']}¥n";
 } else {
   print "Oracleに接続しました。";
 }
?>
```

■resource ocinewcursor(resource conn)

新規にカーソルをオープンして返します。

■string ocinewdescriptor(resource conn [, int type])

新規に空のLOB（デフォルト）またはFILEの記述子を初期化します。typeには、OCI_D_FILE、OCI_D_LOB、OCI_D_ROWIDを指定可能です。LOBの場合はload、save、savefileメソッド、FILEの場合はloadメソッドを使用可能です。

■resource ocinlogon(string user, string password [, string database])

F

新規にデータベースへの接続をオープンします。引数の指定方法はocilogonと同じです。

■int ocinumcols(resource stmt)

クエリ結果のカラム数を返します。

■resource ociparse(resource conn, string query)

F

クエリをパースし、ステートメントのリソースを返します。

■resource ociplogon(string user, string password [, string database])

F

データベースへの持続的接続をオープンします。引数の指定方法はocilogonと同じです。

■mixed ociresult(resource stmt, mixed column)

F

クエリ結果の指定カラムを文字列（ROWID、LOB、FILEのような抽象型を除く）として返します。*17

■bool ocirollback(resource conn)

F T

カレントのトランザクションをロールバックします。

*17 ocifetch()のサンプルを参照してください。

■int ocirowcount(int stmt)

更新をともなうクエリで影響を受けたレコードの数を返します。SELECTクエリによる取得レコードは含まれません。

■bool ocisavelob(object lob)

F T

ラージオブジェクトを保存します。

■string ociserverversion(resource conn)

サーバのバージョン情報を取得します。

```
<?php
 $conn = ocilogon("scott","tiger", "orcl") or die("connection faild.");
 print ociserverversion($conn); // Oracleのバージョンを出力
?>
```

■bool ocisetprefetch(resource stmt, int prefetch_rows)

T

ステートメントにおいてprefetch_rowsを実行する際にあらかじめ取得しておくレコードの数を設定します。

■string ocistatementtype(resource stmt)

F

OCI命令の型（下記）を返します。

SELECT、UPDATE、DELETE、INSERT、CREATE、DROP、ALTER、BEGIN、DECLARE、UNKNOWN

```
<?php
 $conn = ocilogon("scott", "tiger", "orcl") or die("connection faild.");
 $stmt = ociparse($conn, "SELECT * FROM otenki"); // クエリをパース
 print ocistatementtype($stmt); // "SELECT"を出力
?>
```

1.5 Microsoft SQL Server関数

Microsoft SQL Serverは、MS Windows上で動作する高性能な商用RDBMSです。PHPからMS Windows上のSQL Serverにアクセスするには以下の二通りの方法があります。

1. MS Windows上のPHPからローカル(またはリモート)アクセスする。
   ① PHPの設定ファイルphp.iniに以下の行を追加します。

```
extension = php_mssql.dll
```

2. Linux（またはほかのUNIX互換のOS）上のPHPからリモートアクセスする。
   ① FreeTDSをインストールします。

```
$ tar xzvf freetds-0.53.tgz; cd freetds-0.53
$ ./configure --with-tdsver=7.0
$ make; make install
```

   ②/usr/local/freetds/etc/freetds.confの最後の部分を以下のように書き変えます。

```
[gateway]
host = [Windowsサーバのアドレス]
port = 1433
tds version = 7.0
client charset = sjis
language = japanase
```

   ③ Sybaseサポートを有効にして、PHPを構築します。

```
$ cd php-4.2.1
$ ./configure --with-sybase=/usr/local/freetds [そのほかのオプション]
$ make; make install
```

▶ 参考URL

FreeTDS：**http://www.freetds.org/**

▶ 注意

・MS SQL関数において接続リソースを省略した場合、最後にオープンされた接続が仮定されます。
・MS SQL関数は、Sybase SQL Serverとの接続にも使用可能です（**mssql_***を**sybase_***として使用します）。

・FreeTDSを使用する場合、ライブラリのパス（例：/usr/local/freetds/lib）を/etc/ld.so.confまたは環境変数LD_LIBRARY_PATHに追加する必要があります。
・FreeTDSを使用する場合、一部の関数（mssql_field_*()など）はサポートされません。

■bool mssql_bind(resource stmt, string param, mixed var, int type [, int is_output[, int is_null[, int maxlen]]])

F T

ストアドプロシージャにおいて、PHP変数varを型typeのSQLパラメータparamにバインドします。*18 リソースstmtはmssql_init()の返り値です。プロシージャでの設定値を取得する場合はis_outputにTRUEを指定します。

```
<?php // システムストアドプロシージャsp_columnsによりカラム情報を取得
 $conn = mssql_connect("gateway", "sa", "") or die("connection failed.");
 mssql_select_db("foo",$conn); // データベース選択
 $stmt = mssql_init("sp_columns", $conn); // システムストアドプロシージャを初期化
 $table = "otenki"; // テーブル名を指定
 mssql_bind($stmt, "@table_name", $table, SQLVARCHAR); // 引数を指定
 $result = mssql_execute($stmt); // ストアドプロシージャ実行
 $row = mssql_fetch_assoc($result); // 結果セットを取得
 print_r($row); // カラム情報を表示
?>
```

■bool mssql_close([resource conn])

T

データベースへの接続を閉じます。

■resource mssql_connect([string server [, string user [, string password]]])

F

ユーザuserおよびパスワードpasswordを用いて接続をオープンし、接続リソースを返します。同じ引数ですでに接続がオープンされている場合には、既存の接続リソースが返されます。

■bool mssql_data_seek(resource result, int i)

F T

レコードポインタをi番目のレコードに移動します。

*18 FreeTDSを使用する場合、サポートされません。

*19 mssql_bind()のサンプルを参照してください。

*20 もうレコードが存在しない場合にはfalseを返します。

■bool mssql_execute(resource stmt)

F

mssql_init()から返されたステートメントのリソースを指定してストアドプロシージャを実行し、結果セットのリソースを返します。*18 *19

■array mssql_fetch_array(resource result [, int type])

F

レコードを1件取得し、配列として返します。*20 フィールド名をキーとした連想配列としても結果にアクセス可能です。typeに配列のアクセス方法(デフォルト：MSSQL_BOTH)を指定できます。*21

```
<?php
 $conn = mssql_connect("gateway", "sa", "") or die("connection failed.");
 mssql_select_db("foo",$conn); // データベース選択
 $result = mssql_query("SELECT * FROM otenki"); // クエリ実行

 while($row = mssql_fetch_array($result)){ // レコードを取得、表示
   printf("%s¥t%s¥t%s¥t%s¥n",
     $row['day'],$row['tenki'],$row['ondo'],$row['uryou']);
 }
?>
```

■array mssql_fetch_assoc(resource result [, int type])

F

レコードを1件取得し、連想配列として返します。*20 typeに配列のアクセス方法(デフォルト：MSSQL_ASSOC)を指定できます。*21

■object mssql_fetch_field(resource result [, int k])

F

k番目のフィールド(省略時はカレントフィールド)に関する情報を取得し、**表3-8**のプロパティを有するオブジェクトとして返します。

表3-8 mssql_fetch_field()が返すオブジェクトのプロパティ

プロパティ	説明
name	カラム名
max_length	カラムの最大長
column_source	カラムがあるテーブル
numeric	カラムが数値である場合に1
type	型名

*21
MSSQL_NUM：数値配列、MSSQL_ASSOC：連想配列、MSSQL_BOTH：数値配列及び連想配列。

*22
メモリはスクリプト実行終了時に自動的に解放されるため、通常は必要ありません。

■object mssql_fetch_object(resource result [, int type])

F

レコードを1件取得し、カラム名をプロパティとして有するオブジェクトとして返します。*20 typeに配列のアクセス方法（デフォルト：MSSQL_ASSOC）を指定できます。*21

```
<?php
 $conn = mssql_connect("gateway", "sa", "") or die("connection failed.");
 mssql_select_db("foo",$conn); // データベース選択
 $result = mssql_query("SELECT * FROM otenki"); // クエリ実行

 while($row = mssql_fetch_object($result)){ // レコードを取得、表示
   printf("%s¥t%s¥t%s¥t%s¥n",
     $row->day,$row->tenki,$row->ondo,$row->uryou);
 }
?>
```

■array mssql_fetch_row(resource result [, int type])

F

レコードを1件取得し、配列として返します。*20 typeに配列のアクセス方法（デフォルト：MSSQL_NUM）を指定できます。*21

```
<?php
 $conn = mssql_connect("gateway", "sa", "") or die("connection failed.");
 mssql_select_db("foo",$conn); // データベース選択
 $result = mssql_query("SELECT * FROM otenki"); // クエリ実行

 while($row = mssql_fetch_row($result)){ // レコードを取得、表示
   print implode("¥t",$row)."¥n";
 }
?>
```

■string mssql_field_name(resource result [, int offset])

F

指定したオフセット(省略時はカレントのカラム)のカラムの名前を取得します。*18
カラムに関する情報は、**表3-9**に示す関数で取得可能です。

表3-9　カラム情報を取得する関数

関数名	返り値
int mssql_field_length(resource result [, int offset])	カラム長
string mssql_field_name(resource result [, int offset])	テーブル名
string mssql_field_type(resource result [, int offset])	テーブル型

```
<?php // MS Windows上でのみ使用可能
 $conn = mssql_connect("gateway", "sa", "") or die("connection failed.");
 mssql_select_db("foo",$conn); // データベースを選択
 $result = mssql_query("SELECT * FROM otenki"); // クエリを実行

 for ($i=0; $i<mssql_num_fields($result);$i++) { // カラム情報を出力
   print mssql_field_name($result, $i)."¥t";
   print mssql_field_type($result, $i)."¥t";
   print mssql_field_length($result, $i)."¥n";
 }
?>
```

■bool mssql_field_seek(resource result, int offset)

F T

指定したオフセットにカレントのカラムを移動します。

■bool mssql_free_result(resource result)

F T

クエリ結果保持用のメモリを解放します。*22

■string mssql_get_last_message()

データベースサーバからの最新のメッセージを取得します。

■resource mssql_init(string sp_name [, resource conn])

F

ストアドプロシージャsp_nameを初期化し、そのリソースを返します。*18 *19

■bool mssql_next_result(resource result)

F T

ストアドプロシージャなどで複数の結果セットが得られた場合に、次の結果セットに移動します。*18

■int mssql_num_fields(resource result)

クエリ結果のカラム数を返します。

■int mssql_num_rows(resource result)

クエリ結果のレコード数を返します。

■resource mssql_pconnect([string server [, string user [, string password]]])

F

持続的な接続をオープンし、接続リソースを返します。引数の指定方法はmssql_connect()と同じです。

■resource mssql_query(string query [, resource conn])

F

クエリを送信し、結果セットのリソースを返します。

■string mssql_result(resource result, int i, mixed col)

F

i番目のレコードの指定カラムの内容を返します。colには、カラムの**オフセット**、**カラム名**、**カラム名.テーブル名**のいずれかを指定します。

■int mssql_rows_affected(resource conn)

直近のクエリで影響を受けたレコードの数を返します。*18 *23

■bool mssql_select_db(string database [, resource conn])

F T

アクティブなデータベースを設定します。*19

*23
FreeTDS経由の場合はmssql_affected_rows()とします。

1.6 IBM DB2/ODBC関数

ODBC（Open DataBase Connectivity）は、データベース接続用の標準インタフェースであり、主にMicrosoft Windows環境で使用されています。ODBCサポート関数により、ODBCインタフェースをサポートする多くのデータベースと接続することが可能です。

ODBC関数により信頼性の高い商用データベースであるIBM DB2と接続することが可能です。

▶PHP構築オプション

PHPは、複数のODBCドライバ/マネージャをサポートします。使用する環境に合わせた構築オプションを選択します。Microsoft Windows環境では、デフォルトでODBCサポートが有効となっています。

1. データベース固有のODBCドライバ（特定のデータベースをサポート）

--with-ibm-db2[=DIR] IBM DB2サポート関数を有効にします。DIRにSQLLIBがインストールされているパス（例：/home/db2inst1/sqllib）を指定します。

［IBM DB2用設定］

Apache起動オプション（Vine Linuxの場合：/etc/sysconfig/apache）に以下の環境変数の設定を追加します。*24 *25

```
./home/db2inst1/sqllib/db2profile
export LANG=ja_JP.ujis
```

--with-sapdb[=DIR] SAP DBのサポートを有効にします。

--with-empresss[=DIR] Empressのサポートを有効にします。

2. 汎用ODBCドライバ（複数のデータベースをサポート）

--with-unixODBC[=DIR] unixODBCサポートを有効にします。

--with-iodbc[=DIR] iODBCサポートを有効にします。

*24
DB2ライブラリのパス（例：/usr/IBMdb2/V7.1/lib）を/etc/ld.so.confまたは環境変数LD_LIBRARY_PATHに追加する必要があります。

*25
ディレクトリ構成などはシステムの設定により異なります。

▶php.ini設定オプション

パラメータ	デフォルト値	説明
odbc.allow_persistent	On	持続的接続を許可する場合にOn
odbc.check_persistent	On	既存の接続の有効性を利用前に確認する場合にOn
odbc.defaultbinmode	1	バイナリカラムデータ処理モード： 0（バイナリデータとして通過） 1（そのまま返す）、2（charに変換）
odbc.default_db	NULL	デフォルトのデータベース
odbc.default_pw	NULL	デフォルトの接続パスワード
odbc.default_user	NULL	デフォルトの接続ユーザ
odbc.defaultlrl	4096	LONGカラムの最大サイズ（バイト数）
odbc.max_links	-1	接続の最大数（-1：制限なし）
odbc.max_persistent	-1	持続的接続の最大数（-1：制限なし）

▶参考URL

IBM DB2の開発・配布元　：**http://www.ibm.co.jp/**
unixODBCの開発・配布元　：**http://www.unixodbc.org/**
Openlink Software（iODBCの開発・配布元）：**http://www.openlinksw.com/**

▶注意

1. ODBC関数で接続ID（cid）を省略した場合は、カレントの接続IDが使われます。
2. 本節で説明するサンプルテーブルの定義（IBM DB2用）を付属CD-ROMに収録（database_odbc.sql）してあります。

■int odbc_autocommit(resource conn [, int OnOff])

F T

オートコミットを設定（1：On（デフォルト）、0：Off）します。2番目の引数を省略した場合は現在の設定値を返します。

■bool odbc_binmode(resource result, int mode)

T

バイナリカラムデータの処理モードを以下のように設定します。

ODBC_BINMODE_PASSTHRU：バイナリデータとして通過
ODBC_BINMODE_RETURN　：そのまま返す（デフォルト）
ODBC_BINMODE_CONVERT　：charに変換して返す

■void odbc_close(resource conn)

ODBC接続を閉じます。*26

■void odbc_close_all(void)

サーバへのすべての接続を閉じます。*26

■bool odbc_commit(resource conn)

F T

すべてのトランザクションをコミットします。

■resource odbc_connect(string dsn, string user, string password [, int cursor])

ユーザ名、パスワードを指定してデータソースdsnに接続し、接続リソースを返します。複数の接続を同時にオープンすることができます。

cursorによりカーソルの型（SQL_CUR_USE_IF_NEEDED、SQL_CUR_USE_ODBC、SQL_CUR_USE_DRIVER、SQL_CUR_DEFAULT）を指定可能です。

```
<?php
 $conn = @odbc_connect("sample", "db2inst1", "db2inst1");
 if (!$conn){ // 接続できなかった場合はエラーメッセージを出力
   print "エラーコード: ". odbc_error()."¥n";
   print "エラーメッセージ: ". odbc_errormsg()."¥n";
   die('connection failed');
 }
 print "データベースに接続しました。";
?>
```

■resource odbc_columns(resource conn [, string catalog [, string schema [, string table [, string column]]]])

F

指定テーブルにおいてカラム名のリストを取得します。

*26
未完了のトランザクションがある場合にはエラーとなり、接続はオープンされたままとなります。

■resource odbc_columnprivileges(resource conn , string catalog , string schema , string table , string column)

F

カラムとその権限のリストからなる結果セットを取得します。

■string odbc_cursor(resource conn)

F

カーソル名を文字列として返します。

■string odbc_error([resource conn])

直近のエラーのエラーコードを返します。

■string odbc_errormsg([resource conn])

直近のエラーのエラーメッセージを返します。

■resource odbc_exec(resource conn, string query [, int flags])

F

クエリをパース、実行し、結果リソースを返します。

■bool odbc_execute(resource result [, array param])

F T

odbc_prepare()でパースしたクエリを実行します。オプション配列によりクエリのパラメータを指定できます。

```
<?php
 $conn = odbc_connect("sample", "db2inst1", "db2inst1") or die("connection
failed");
 $stmt = odbc_prepare($conn,"INSERT INTO otenki VALUES(?, ?, ?, ?)");
                                                          // クエリをパース
 $data[] = array("2000-8-20","晴れ",29,20);
 $data[] = array("2000-8-21","雨",25,100);

 foreach ($data as $value) { // 挿入するデータを設定し、クエリを実行
   odbc_execute($stmt, $value);
 }
?>
```

■int odbc_fetch_into(resource result, array arr [, int i])

F

i番目のレコード（省略時は次のレコード）を配列に取得し、カラム数を返します。

```
<?php
 $conn = odbc_connect("sample", "db2inst1", "db2inst1") or die("connection
failed");
 $result = odbc_exec($conn, "SELECT * FROM otenki"); // SQLクエリ実行
 while ($ncols = odbc_fetch_into($result, $record)) {
                                  // レコードを配列$recordに取得
   for ($i=0; $i<$ncols; $i++){ // レコードを表示
     print $record[$i]."¥t";
   }
   print "¥n";
 }
?>
```

■bool odbc_fetch_row(resource result [, int i])

F T

i番目のレコード（省略時は次のレコード）を取得します。もうレコードが存在しない場合にFALSEが返されます。

```
<?php
 $conn = odbc_connect("sample", "db2inst1", "db2inst1") or die("connection
failed");
 $result = odbc_exec($conn, "SELECT * FROM otenki"); // SQLクエリ実行
 while (odbc_fetch_row($result)) { // レコードを取得/表示
   for ($i=1; $i<=odbc_num_fields($result); $i++) {
     print odbc_result($result, $i)."¥t";
   }
   print "¥n";
 }
?>
```

■string odbc_field_name(resource result, int i)]

F

i番目（先頭カラム：1）のフィールドの名前を返します。カラムに関する情報を取得する関数を**表3-10**に示します。

*27
引数iはカラム番号（先頭カラム：1）です。

表3-10　カラム情報を取得する関数*27

関数名	返り値
int odbc_field_len(resource result, int i)	カラムの桁数
string odbc_field_name(resource result, int i)	カラム名
int odbc_field_scale(resource result, int i)	カラムのスケール
string odbc_field_type(resource result, int i)	カラムの型

```
<?php
 $conn = odbc_connect("sample", "db2inst1", "db2inst1") or die("connection failed");
 $result = odbc_exec($conn, "SELECT * FROM otenki"); // SQLクエリ実行
 for ($i=1; $i<=odbc_num_fields($result); $i++){
   print odbc_field_name($result, $i) . "¥t"; // カラム名を出力
   print odbc_field_type($result, $i) . "¥t"; // カラム型を出力
   print odbc_field_scale($result, $i) . "¥t"; // カラムのスケールを出力
   print odbc_field_len($result, $i) . "¥n"; // カラム長を出力
 }
?>
```

■int odbc_field_num(resource result, string col)

F

指定カラムのカラム番号を返します。

■int odbc_free_result(resource result)

T

クエリ結果を保持するメモリを解放し、TRUEを返します。通常は自動的に解放されるため、不要です。

■resource odbc_foreignkeys(resource conn, string pk_cat, string pk_schema, string pk_table, string fk_cat, string fk_schema, string fk_table)

F

指定したテーブルの外部キーのリスト、または、指定したテーブルの主キーを参照するほかのテーブルの外部キーのリストを結果セットとして返します。

■resource odbc_gettypeinfo(resource conn, int type)

F

データソースによりサポートされているデータ型に関する情報を含む結果セットを返します。typeに取得するデータ型（デフォルトはすべて取得）を指定可能です。

```
<?php
 $conn = odbc_connect("sample", "db2inst1", "db2inst1") or die("connection
failed");
 $types = odbc_gettypeinfo($conn); // サポートされるデータ型のリストを取得
 odbc_result_all($types);
?>
```

■int odbc_longreadlen(resource result, int len)

T

LONG/LONGVARBINARY型カラムで読み込むバイト数をlenバイトに設定します。

■bool odbc_next_result(resource result)

F T

更に結果セットがある場合にこの結果セットを取得し、TRUEを返します。

■int odbc_num_fields(resource result)

クエリ結果のカラム数を返します。

■int odbc_num_rows(resource result)

クエリ結果のレコードの数を返します。INSERT、UPDATE、DELETE文の場合は作用したレコード数、SELECT文の場合は取得可能なレコードの数（いくつかのODBCドライバを除く）を返します。

■resource odbc_pconnect(string dsn, string user, string password [, int cursor])

F

持続的な接続をオープンし、接続リソースを返します。引数の指定方法はodbc_connect()と同じです。

■resource odbc_prepare(resource conn, string query)

F

クエリをパースし、ステートメントのリソースを返します。*28

*28
odbc_execute()のサンプルを参照してください。

■**resource odbc_primarykeys(resource conn,
string catalog, string schema, string table)**

F

テーブルの主キーを構成するカラム名からなる結果セットを取得します。

■**resource odbc_procedures(resource conn
[, string catalog, string schema, string proc])**

F

データソースのプロシージャ名のリストからなる結果セットを取得します。

■**resource odbc_procedurecolumns(resource conn
[, string catalog, string schema, string proc, string column])**

F

プロシージャに関する入出力パラメータのリストを取得します。

■**string odbc_result(resource result, mixed col)**

F

指定したカラムの値を返します。colはカラム番号（先頭カラム：1）またはカラム名とします。

■**int odbc_result_all(resource result [, string format])**

F

クエリ結果からすべてのレコードをHTMLテーブル形式で出力し、出力したレコード数を返します。formatにより、テーブルのフォーマットをHTMLタグで指定可能です。

```
<?php
 $conn = odbc_connect("sample", "db2inst1", "db2inst1") or die("connection
failed");
 $result = odbc_exec($conn, "SELECT * FROM otenki"); // クエリ実行
 odbc_result_all($result); // クエリ結果をHTMLテーブル形式で出力
?>
```

■**bool odbc_rollback(resource result)**

F T

すべての未解決の命令をロールバックします。

■int odbc_setoption(resource conn|result, int func, int option, int param)

ODBC接続のオプションoptionをparamに設定します。funcは適用するODBC関数の種類で、SQLSetConnectOption()の場合に1、SQLSetStmtOption()の場合に2とします。

■resource odbc_specialcolumns(resource conn , int type, string catalog, string schema, string table, int scope, int nullable)

F

データソースの特別なカラム（主キーおよびインデックス）からなる結果セットを取得します。

■resource odbc_statistics(resource conn , string catalog, string schema, string table, int unique, int reserved)

F

テーブルのインデックス関連情報からなる結果セットを取得します。

■resource odbc_tables(resource conn [, string catalog [, string schema [, string table [, string type]]]])

F

接続中のデータソースのシステムカタログに保持されているテーブル名と関連情報のリストを結果セットとして返します。オプションで対象とするカタログ名、スキーマ名、テーブル名、テーブル型の検索パターン（すべての文字にマッチする'%'を使用可能）を指定可能です。

```
<?php
 $conn = odbc_connect("sample", "db2inst1", "db2inst1") or die("connection
failed");
 $tables = odbc_tables($conn); // テーブル名のリストを取得
 odbc_result_all($tables);
?>
```

■resource odbc_tableprivileges(resource conn, string catalog, string schema, string table)

F

各テーブルに関連する権限のリストからなる結果セットを取得します。

1.7 dba関数

dbaは、UNIX標準の簡易データベースシステムであるBerkeley DB型式のデータベースをサポートします。通常、パスワードの管理などの簡単な処理を行なう際に使用されます。dbaでは、複数の実装（dbm、ndbm、GNU gdbm、db2、db3、cdb）がサポートされています。

▶PHP構築オプション

--with-enable-dba	DBA関数を有効にします。
--with-db2[=DIR]	db2（Sleepycat Software製）のサポートを有効にします。
--with-db3[=DIR]	db3（Sleepycat Software製）のサポートを有効にします。
--with-dbm[=DIR]	dbmのサポートを有効にします。
--with-cdb[=DIR]	cdbのサポートを有効にします。
--with-gdbm[=DIR]	gdbm（GNU database manager）のサポートを有効にします。
--with-ndbm[=DIR]	ndbmのサポートを有効にします。

▶参考URL

Sleepycat Software（db2/db3の開発・配布元）：**http://www.sleepycat.com/**
cdbの開発・配布元　：**http://cr.yp.to/cdb.html**

■void dba_close(int handle)

データベースを閉じます。

■bool dba_delete(string key, int handle)

F T

指定したキーのレコードを削除します。

■bool dba_exists(string key, int handle)

キーが存在する場合にTRUE、見つからない場合にはFALSEを返します。

```
<?php
 $id = dba_open("/tmp/test.db", "r", "gdbm") or die("dba open failed.");
 print dba_exists("2000", $id) ? "存在する" : "存在しない";
 dba_close($id);
?>
```

string dba_fetch(string key, int handle)

F

キーkeyが指す値を文字列として取得します。

```
<?php
 $id = dba_open("/tmp/test.db", "r", "gdbm") or die("dba open failed.");
 for ($key = dba_firstkey($id); $key !== FALSE; $key = dba_nextkey($id)){
  print dba_fetch($key, $id)."¥n"; // 値を表示
 }
 dba_close($id);
?>
```

string dba_firstkey(int handle)

F

内部ポインタをリセットし、先頭要素のキーを取得します。

```
<?php
 $id = dba_open("/tmp/test.db", "r", "gdbm") or die("dba open failed.");
 for ($key = dba_firstkey($id); $key !== FALSE; $key = dba_nextkey($id)){
   print $key."¥n"; // キーを表示します。
 }
 dba_close($id);
?>
```

bool dba_insert(string key, string value, int handle)

F T

キーをkeyとして値valueを挿入します。指定したキーがすでに存在する場合はfalseを返します。

```
<?php
 $id = dba_open("/tmp/test.db", "n", "gdbm") or die("dba open failed.");
 $data = array("2000"=>"400","2001"=>"600","2002"=>"900");
 foreach ($data as $key => $value) {
   if(dba_insert($key, $value, $id) === FALSE){
     die("insert failed.");
   }
 }
 dba_close($id);
?>
```

■string dba_nextkey(int handle)

F

次の要素のキーを返し、内部ポインタを前に進めます。

■int dba_popen(string path, string mode, string handler)

F

持続的接続をオープンします。引数の指定方法はdba_open()と同じです。

■int dba_open(string path, string mode, string handler)

F

指定したパス（path）にあるデータベースへの接続をオープンし、ハンドラIDを返します。modeには c（新規作成）、r（読み込み）、w（書き込み）、n（読み込み/書き込み）を、handlerには使用するデータベース（gdbm、dbm、ndbm、cdb、db2、db3のうち使用可能なもの）を指定します。

■bool dba_optimize(int handle)

F T

vacuumなどによりデータベースを最適化します。

■bool dba_replace(string key, string value, int handle)

F T

キーをkeyとして値valueを挿入します。指定したキーがすでにある場合は上書きします。

■bool dba_sync(int handle)

F T

データベースの同期をとります。

1.8 LDAP関数

LDAP（Lightweight Directory Access Protocol）はディレクトリサービスを提供するしくみであり、階層構造を有するデータベースへのAPIを提供するものです。

ディレクトリサービスは、ファイルやリソースの管理などに使用することが可能で、データの更新よりも検索のほうがはるかに多い用途ではリレーショナル型データベースよりも管理および性能の面で優れています。

日本語を使用した検索を行なう場合には、LDAPv3をサポートするライブラリを使用し、文字コードはマルチバイト関数などによりUTF-8形式とする必要があります。

PHPでは、以下のLDAPライブラリをサポートしています。

- **OpenLDAP：http://www.openldap.org/**
  バージョン2以降でLDAPv3をサポートします。
- **Netscape社による LDAP SDK：http://developer.netscape.com/tech/directory/**

個々のデータは、以下のようなDN (distinguished name) によりアクセスします。

```
cn=鈴木太郎,o=phpusers,c=JP
```

DNは右から左に解釈され、順にcountry、organization、commonNameを表します。

▶PHP構築オプション

--with-ldap[=DIR] LDAPのライブラリがインストールされているディレクトリを指定します。

▶php.ini設定オプション

パラメータ	デフォルト値	説明
ldap.max_links	-1	最大接続数（-1の場合は制限なし）

▶注意

1. データおよび検索フィルタ指定に日本語を使用する場合、文字コードをUTF-8とする必要があります。
2. 本節のサンプルで用いるLDIFファイルを付属CD-ROMに収録してあります。テーブル作成は以下のように行ないます。

```
ldapadd -D "cn=Manager,o=phpusers,c=JP" -w secret -x < sample.ldif
```

なお、slapd.confに以下の値が設定されていることを仮定します。

```
suffix  "o=phpusers,c=JP"
rootdn  "cn=Manager,o=phpusers,c=JP"
rootpw  secret
```

*29 実用アプリケーションで使用する場合、パスワードの暗号化が必要です。

■bool ldap_add(resource link, string dn, array entry)

F T

LDAPディレクトリにエントリを追加します。属性は以下のように設定します。

```
entry["属性1"] = 値
entry["属性2"][0] = 値1    // 同一属性に複数の値を定義する場合
entry["属性2"][1] = 値2
```

```
<?php
 $ds = ldap_connect("localhost") or die("connection failed.");
 $r = ldap_bind($ds,"cn=Manager,o=phpusers,c=JP","secret"); // 認証つきバインド

 // エントリデータを定義
 $cn = mb_convert_encoding("清水シロー","UTF-8","EUC-JP");
 $sn = mb_convert_encoding("清水","UTF-8","EUC-JP");
 $entry = array("cn"=>$cn,"sn"=>$sn,"telephoneNumber"=>"1234-5578",
          "objectClass"=>"person");

 if(ldap_add($ds, "cn=$cn, o=phpusers, c=JP", $entry)){
   print "エントリを追加しました。";
 }
?>
```

■bool ldap_bind(resource link [, string dn, string pwd])

F T

リソースのDNおよびパスワードを指定してLDAPディレクトリにバインドします。dnおよびpwdを指定しない場合は匿名バインドとなります。

```
<?php
 $ds = ldap_connect("localhost") or die("connection failed.");
 $r = @ldap_bind($ds,"cn=Manager,o=phpusers,c=JP","secret"); // 認証つきバインド
 if (!$r){ // バインドできなかった場合
   $err = ldap_err2str(ldap_errno($ds)); // エラーメッセージを取得
   die($err);
 }
 print "LDAPディレクトリにバインドしました。";
?>
```

■bool ldap_compare(resource link, string dn , string attr, string value)

M

属性およびその値を指定して指定したDNの同じ属性の値と比較します。値が一致する場合はTRUE、一致しない場合はFALSE、エラーの場合は-1を返します。

```
<?php
 $ds = ldap_connect("localhost");
 ldap_bind($ds); // 匿名バインド
 $cn = mb_convert_encoding("鈴木太郎","UTF-8","EUC-JP");
 // 比較を実行
  if ((($r = ldap_compare($ds,"cn=$cn,o=phpusers,c=JP","objectclass","person"
))===-1){
   die(ldap_error($ds));
 }
 print $r === TRUE ? "一致" : "不一致";
?>
```

■resource ldap_connect([string host [, int port]])

F

LDAPディレクトリサーバに接続し、リンクリソースを返します。hostを指定しない場合、既存の接続リソースが返されます。portにより接続ポート（デフォルトは389）を指定可能です。

■int ldap_count_entries(resource link, resource result)

検索結果のエントリ数を返します。

■bool ldap_delete(resource link, string dn)

F T

ディレクトリからエントリを削除します。

```
<?php
 $ds = ldap_connect("localhost") or die("connection failed.");
 $r = ldap_bind($ds,"cn=Manager,o=phpusers,c=JP","secret"); // 認証つきバインド
 $cn = mb_convert_encoding("清水シロー","UTF-8","EUC-JP");
 if(ldap_delete($ds, "cn=$cn, o=phpusers, c=JP")){ // エントリデータを削除
   print "エントリを削除しました。";
 }
?>
```

■string ldap_dn2ufn(string dn)

F

DNをUFN（User Friendly Naming）形式に変換します。

```
<?php
 $ds = ldap_connect("localhost") or die("connection failed.");
 $r = ldap_bind($ds); // 匿名バインド
 $dn = "cn=Taro Suzuki, o=phpusers, c=JP";
 print ldap_dn2ufn($dn); // 「Taro Suzuki, phpusers, JP」を出力
?>
```

■string ldap_err2str(int errno)

エラー番号からエラー文字列を取得します。

■int ldap_errno(resource link)

カレントのエラー番号（ない場合には0）を返します。

■string ldap_error(resource link)

カレントのエラー文字列を返します。

■array ldap_explode_dn(string dn, int with_attrib)

DNを要素別に分割して配列として返します。キーcountにて要素数を取得可能です。

with_attrib に1を指定すると値のみ、0を指定するとRDN形式（属性＝値）で取得されます。

```
<?php
 $ds = ldap_connect("localhost") or die("connection failed.");
 $r = ldap_bind($ds); // 匿名バインド
 $dn = "cn=Taro Suzuki, o=phpusers, c=JP";
 $a = ldap_explode_dn($dn, 1);
 print_r($a); // array("count"=>3, "0"=>"Taro Suzuki", "1"=>"phpusers", "2"=>JP)
?>
```

■bool ldap_free_result(resource result)

T

結果セット保持用のメモリを解放します。

■int ldap_first_entry(resource link, resource result)

F

先頭のエントリのIDを返します。

```
<?php
 $ds = ldap_connect("localhost") or die("connection failed.");
 ldap_bind($ds); // 匿名バインド
 $result = ldap_search($ds,"o=phpusers,c=JP","sn=*"); // 検索を実行
 $entry = ldap_first_entry($ds, $result);
 while ($entry) {
   $dn = ldap_get_dn($ds, $entry);
   print mb_convert_encoding($dn, "EUC-JP", "UTF-8")."¥n";
   $entry = ldap_next_entry($ds, $entry);
 }
?>
```

■string ldap_first_attribute(resource link, resource result, resource &berid)

F

エントリにおける最初の属性を返します。beridには最初の属性を指す位置ポインタのリソースが代入されます。

■array ldap_get_attribute(resource link, resource result)

結果エントリから属性をすべて取得し、以下のような多次元の配列として返します。

```
$value["count"]   = そのエントリの属性の数
                    $value[0]             = 最初の属性
                    $value[i]             = (i+1)番目の属性
                    $value["a"]["count"]  = 属性"a"に関する値の数
                    $value["a"][0]        = 属性"a"に関する最初の値
                    $value["a"][i]        = 属性"a"に関する(i+1)番目の値
```

```
<?php
 $ds = ldap_connect("localhost") or die("connection failed.");
 ldap_bind($ds); // LDAPに匿名バインド
 $sr = ldap_search($ds,"o=phpgr,c=JP","sn=S*"); // 検索を実行
 $entry = ldap_first_entry($ds, $sr);
 $attrs = ldap_get_attributes($ds, $entry);
 print  "取得した属性の数: " .$attrs["count"];

 for ($i=0; $i<$attrs["count"]; $i++){
     print $attrs[$i]."¥n";
 }
?>
```

■string ldap_get_dn(resource link, resource result)

F

結果エントリのDNを返します。

■array ldap_get_entries(resource link, resource result)

結果エントリをすべて取得し、以下のようなツリー構造を有する配列として返します。

$value["count"]	= エントリの数
$value[i]["dn"]	= i番目のエントリ DN
$value[i]["count"]	= i番目のエントリの属性の数
$value[i][j]	= i番目のエントリのj番目の属性
$value[i]["属性"]["count"]	= i番目のエントリの属性に関する値の数
$value[i]["属性"][j]	= i番目のエントリの属性におけるj番目の値

■bool ldap_get_option(resource link, int option, mixed &retval)

F T

指定したオプションの値を変数retvalに取得します。optionには以下の定数を指定します。

LDAP_OPT_DEREF、LDAP_OPT_SIZELIMIT、LDAP_OPT_TIMELIMIT、LDAP_OPT_PROTOCOL_VERSION、LDAP_OPT_ERROR_NUMBER、LDAP_OPT_REFERRALS、LDAP_OPT_RESTART、LDAP_OPT_HOST_NAME、LDAP_OPT_ERROR_STRING、LDAP_OPT_MATCHED_DN

```
<?php
 $ds = ldap_connect("localhost") or die("connection failed.");
 if (ldap_get_option($ds, LDAP_OPT_PROTOCOL_VERSION, $ver)){
   print "プロトコルバージョン: $ver";
 }
?>
```

■array ldap_get_values(resource link, resource result, string attrib)

F

検索結果のエントリから値をすべて取得し、配列として返します。キーcountにてエントリ数を取得可能です。

```
<?php
 $ds = ldap_connect("localhost") or die("connection failed.");
 $r = ldap_bind($ds); // 匿名バインド
 $sr = ldap_search($ds,"o=phpusers,c=JP","sn=*"); // 検索を実行
 $entry = ldap_first_entry($ds, $sr);
 while ($entry) {
   $info = ldap_get_values($ds, $entry, "sn"); // snエントリを取得
   print mb_convert_encoding($info[0], "EUC-JP", "UTF-8")."¥n";
   $entry = ldap_next_entry($ds, $entry);
 }
?>
```

■array ldap_get_values_len(resource link, resource result, string attrib)

F

検索結果のエントリから値のバイナリ値として取得し、配列として返します。キーcountによりエントリ数を取得可能です。

■resource ldap_list(resource link, string base_dn, string filter [, array attrs [, int attrsonly [, int sizelimit [, int timelimit [, int deref]]]]])

F

スコープをLDAP_SCOPE_ONELEVELとし、指定したDNの直下のレベルにおいて検索を行ないます。引数の指定方法はldap_search()と同じです。

■bool ldap_mod_add(resource link, string dn, array entry)

F T

カレント属性に属性の値を追加[*30]します。追加は属性レベルで行なわれます。

■bool ldap_mod_del(resource link, string dn, array entry)

F T

カレントの属性から属性を削除します。*30 削除は属性レベルで行なわれます。

■bool ldap_mod_replace(resource link, string dn, array entry)

F T

属性の値を指定した値に更新します。*30 更新は属性レベルで行なわれます。

■string ldap_next_attribute(resource link, resource result, resource &berid)

F T

次の属性を返します。位置ポインタのリソースberidはは次の属性に移動されます。ldap_first_attribute()と同じberidを指定し、連続的に属性を取得することが可能です。

■int ldap_next_entry(resource link, resource result)

F

次のエントリのIDを返します。ldap_first_entry()をコールしたあと、連続的にエントリを取得することが可能です。

■resource ldap_read(resource link, string base_dn, string filter [, array attrs [, int attrsonly [, int sizelimit [, int timelimit [, int deref]]]]])

F

スコープをLDAP_SCOPE_BASEとし、指定したDNでLDAPツリーを検索し、エントリを取得します。引数の指定方法は、ldap_search()と同じです。

```
<?php
 $ds = ldap_connect("localhost") or die("connection failed.");
 $r = ldap_bind($ds); // 匿名バインド
 $result = ldap_read($ds,"o=phpusers,c=JP","objectClass=*"); // 検索を実行
 $entry = ldap_first_entry($ds, $result);
 print  ldap_get_dn($ds, $entry)."¥n";
?>
```

■bool ldap_rename(resource link, string dn, string newrdn, string newparent, bool delete)

F T

指定したエントリ（dn）を新しいエントリ（newrdn）に移動します。deleteにTRUEを設定すると移動前のエントリは削除されます。

*30 エントリの指定方法はldap_add()と同じです

■resource ldap_search(resource link, string base_dn, string filter [, array attrs [, int attrsonly [, int sizelimit [, int timelimit [, int deref]]]]])

F

スコープをLDAP_SCOPE_SUBTREEとし、指定したベースDN以下でLDAPツリーを検索し、結果セットのリソースを返します。filterには検索用のフィルタを指定します。attrsに必要とする属性を配列で指定することが可能です。

```
<?php
 $ds = ldap_connect("localhost") or die("connection failed.");
 $r = ldap_bind($ds); // 匿名バインド
 $sr = ldap_search($ds,"o=phpusers,c=JP","sn=*"); // 検索を実行
 $info = ldap_get_entries($ds, $sr); // エントリを取得
 for ($i=0; $i<$info["count"]; $i++){ // エントリを表示
   printf("%s¥t%s¥n",
          mb_convert_encoding($info[$i]["cn"][0],"EUC-JP","UTF-8"),
          mb_convert_encoding($info[$i]["sn"][0],"EUC-JP","UTF-8"));
 }
?>
```

■bool ldap_set_option(resource link, int option, mixed newval)

F T

指定したオプションの値を設定します。optionには以下の定数を指定します。オプションごとに指定する値（newval）の型が異なります。

整数型：LDAP_OPT_DEREF、LDAP_OPT_SIZELIMIT、LDAP_OPT_TIMELIMIT、LDAP_OPT_PROTOCOL_VERSION、LDAP_OPT_ERROR_NUMBER
文字列：LDAP_OPT_HOST_NAME、LDAP_OPT_ERROR_STRING、LDAP_OPT_MATCHED_DN
論理値：LDAP_OPT_REFERRALS、LDAP_OPT_RESTART
配列　：LDAP_OPT_SERVER_CONTROLS、LDAP_OPT_CLIENT_CONTROLS

```
<?php
 $ds = ldap_connect("localhost") or die("connection failed.");
 $r = ldap_bind($ds); // 匿名バインド
 ldap_set_option($ds, LDAP_OPT_SIZELIMIT, 2); // エントリ最大数を2に設定
 $sr = @ldap_search($ds,"o=phpusers,c=JP","sn=*"); // 検索を実行
 $info = ldap_get_entries($ds, $sr); // エントリを取得
 print $info["count"]; // 2以下の値を返す
?>
```

■bool ldap_start_tls(resource link)

F T

SSL/TLSセッションを開始します。

■bool ldap_unbind(resource link)

T

LDAPディレクトリへのバインドを解除します。

Chapter-2 変数関連

PHPでは、配列などのさまざまな変数を使用することができます。また、文字列を処理する際に便利な正規表現を処理する関数もサポートされています。

2.1 配列関数

PHPでは、通常の配列以外に文字列をキーとする連想配列が使用できます。また、多次元の配列も使用可能です。

▶ 組み込み方法

配列関数は標準で使用可能です。

■array array([mixed ...])

引数を値とする配列を返します。*1

*1 array()は実際には関数ではなくPHPスクリプト言語の一部です。

```
<?php
 $a = array(3,5,2); // $a[0] = 3,$a[1] = 5, $a[2] = 2となる
 print_r($a);
 // Perlのように'=>'演算子によりキーを指定することも可能
 $a = array ("a"=>"lemon", "b"=>"banana", "c"=>"apple");
 print $a['b']."¥n"; // bananaが出力される

 // 配列添字を不連続に指定したあとに指定を省略した場合、
 // 最後の添字の次の数字が使用される
 $a = array(0=>"first", 2=>"second", "third");
 // $a[0]="first".$a[1]="",$a[2]="second",$a[3]="third" となる
 print_r($a);
?>
```

■array array_change_key_case(array arr [, int case])

連想配列のキーを小文字（または大文字）にして返します。caseにCASE_LOWER（デフォルト）を指定した場合は小文字、CASE_UPPERを指定した場合は大文字となります。

■array array_chunk(array arr, int size [, bool preserve_keys])

配列を大きさsizeの複数の部分に分割します。preserve_kersにTRUEを設定すると配列の元の配列のキーが保持されます。

```
<?php
 $a = array(1, 2, 3, 4, 5);
 $b = array_chunk($a, 2, TRUE);
 print_r($b); // $b=array(array(1,2),array(3,4),array(5));となる
?>
```

■array array_count_values(array arr)

配列の値をキーとし、各値の出現回数を値とした配列を返します。

■array array_diff(array arr1 , array arr2 [, array ...])

F

最初の配列の要素のうち、二つ目以降の配列に含まれない要素からなる配列を返します。

■array array_fill(int start, int num, mixed val)

F

添字startから始まる要素数num、値valの配列を返します。

■array array_filter(array arr [, mixed callback])

配列の各要素にコールバック関数を適用し、コールバック関数がTRUEを返した要素のみからなる配列を返します。

```
<?php // ex2-3.php
 function odd($var) {
     return ($var % 2 == 1);
 }

 $a = array("a"=>1, "b"=>2, "c"=>3, "d"=>4);
 print_r(array_filter($a, "odd")); // "a"=>1, "c"=>3 を出力
?>
```

■array array_flip(array arr)

配列を逆順にして返します。

■array array_intersect(array arr1, array arr2 [, array ...])

複数の配列に共通する要素を配列として返します。

```
<?php
 $a = array("name"=>"taro", "age"=>21);
 $b = array("name"=>"taro", "address"=>"tokyo");
 $c = array_intersect($a,$b);
 print_r($c); // array("name"=>"taro") を出力
?>
```

■bool array_key_exists(mixed key, array arr)

指定したキーまたは添字が配列に存在する場合にTRUEを返します。

```
<?php
$a = array("name"=>"taro","age"=>21);
print array_key_exists("name",$a) ?  "ある" : "ない"; // "ある"を出力
?>
```

■array array_keys(array arr [, mixed value])

配列からすべてのキー（数値および文字列）を配列として返します。2番目の引数を指定した場合、その値を含むキーのみが返されます。

■array array_map(mixed callback, array arr1 [, array arr2 ...])

配列の各要素にコールバック関数を適用します。

```
<?php
 function sqr($n) { // 値を2乗する
     return $n*$n;
 }

 $a = array(1, 2, 3);
 $b = array_map("sqr", $a);
 print_r($b); // $b = array(1, 4, 9)となる
?>
```

■array array_merge(array array1, array array2 [, array ...])

指定した配列を順番に結合し、ひとつの配列として返します。同じキーが指定されていた場合にはあとから指定された値で上書きされます。

■array array_merge_recursive(array array1, array array2 [, array...])

array_merge()と同じですが、多次元の配列をひとつの配列としてマージします。

■bool array_multisort(array ar1 [, mixed arg [, mixed ... [, array ...]]])

T

複数の配列を一度にソートします。以下のオプションを指定可能です。

ソートの順番：SORT_ASC（昇順）またはSORT_DESC（降順）
ソートの種類：SORT_REGULAR(通常のソート)、SORT_NUMERIC(数値としてソート)、
SORT_STRING（文字列としてソート）

■array array_pad(array arr, int len, mixed val)

長さがlenとなるように配列を拡張します。拡張した部分には値valが代入されます。lenが負の場合は左側、正の場合は右側が拡張されます。

```
<?php
 $a = array(3, 4, 10);
 print_r($a);
 $b = array_pad($a, 5, 0);
 print_r($b); // 配列$bはarray(3, 4, 10, 0, 0)となります。
 $c = array_pad($a, -6, 'a');
 print_r($c); // 配列$cはarray('a', 'a', 'a', 3, 4, 10)となる
?>
```

■mixed array_pop(array arr)

配列から最後の値を取り出して返します。取り出された要素は配列から削除されます。

■int array_push(array arr, mixed var [, mixed ...])

指定した要素を配列の最後に追加します。複数の値を指定可能です。

```
<?php
 $a = array(3, 4, 10);
 print array_pop($a)."¥n"; // 10が出力され、$a=array(4,10)となる
 array_push($a, 'a', 'b');
 print_r($a); // 配列$aはarray(4, 10, 'a', 'b')となる
?>
```

■mixed array_rand(array arr [, int num_req])

配列から要素/キーの組みをランダムに取得します。num_reqに取得する要素の数（デフォルト：1）を指定できます。使用前にsrand()関数による乱数の初期化が必要です。

■mixed array_reduce(array arr [, mixed callback [, int initial])

配列にユーザ定義のコールバック関数を再帰的に適用し、スカラー値として返します。再帰処理の初期値（initial）を指定可能です。

```
<?php
 function sum($v, $w) {
     $v += $w;
     return $v;
 }

 $a = array(2, 3, 4, 5, 6);
 print array_reduce($a, "sum", 1); // 21を出力
?>
```

■array array_reverse(array arr [, bool preserve_keys])

配列を逆順にして返します。preserve_keysをTRUEにした場合、キーと値の関係は維持されます。

■mixed array_search(mixed srch, array arr [, bool strict])

配列arr内でsrchを検索し、見つかった要素のキーまたは添字、それ以外の場合にFALSEを返します。strictにTRUEが指定された場合、srchの型が一致することも確認します。

■mixed array_shift(array arr)

配列の先頭の値を取り出して返します。配列の要素数はひとつ減ります。

```
<?php
 $a = array("taro", "hanako");
 print array_shift($a)."¥n"; // "taro"が出力される
 print_r($a); // $aは"hanako"となる
?>
```

■array array_slice(array arr, int pos [, int len])

配列の位置posから長さlen分の要素を取得します。posが正の場合は先頭から、負の場合は最後からの位置を指定します。lenに負の値を指定した場合は配列の終端からの-lenの位置まで返され、lenを指定しない場合は最後の要素まで返されます。

```
<?php
 $a = array("a", "b", "c", "d", "e");
 $b = array_slice($a, 2);
 print_r($b); // $bはarray("c", "d", "e")となる
 $c = array_slice($a, 2, -1);
 print_r($c); // $cはarray("c", "d")となる
 $d = array_slice($a, -2, 1);
 print_r($d); // $dはarray("d")となる
 $e = array_slice($a, 0, 3);
 print_r($e); // $eはarray("a", "b", "c")となる
?>
```

■array array_splice(array arr, int pos [, int len [, array rep]])

配列の位置posから長さlen分の要素を削除し、削除した値を配列として返します。配列repを指定した場合は、その要素で置換します。位置および長さの指定方法はarray_slice()と同じです。

■mixed array_sum(array arr)

配列の要素の合計を返します。

■array array_unique(array arr)

F

配列の重複する値を削除して返します。

■int array_unshift(array arr, mixed var [, mixed ...])

F

配列の先頭に指定した値を追加します。追加後の要素の数を返します。

■array array_values(array arr)

配列arrからすべての値を配列として返します。

■int array_walk(array arr, string func [, mixed data])

配列の各要素の値を最初の引数、キーを2番目の引数としてユーザ定義の関数funcを実行します。funcの追加の引数としてdataを指定可能です。arrの値を変更したい場合には参照渡しとします。

```
<?php
 $names = array('a'=>"taro", 'b'=>"hanako", 'c'=>"yamada");

 function func1($val, $key){ print "$key => $val<BR>¥n"; }

 function func2(&$val, $key, $data){ $val = $data; }

 array_walk($names, 'func1'); // taro,hanako,yamadaが出力される
 array_walk($names, 'func2', "anonymous"); // $namesの各要素にanonymousが代入される
 print_r($names);
?>
```

■bool arsort(array arr [, int sort_flags])

F T

キーと要素との関係を維持しつつ配列を降順にソートします。*2

```
<?php
 $name = array('c'=>"jiro", 'a'=>"taro", 'b'=>'hanako');
 arsort($name); // 降順にソート
 for(reset($name); $key = key($name); next($name)) {
    print $name[$key] . "<BR>¥n"; // taro、jiro、hanakoの順に出力される
 }
?>
```

■bool asort(array arr [, int sort_flags])

F T

キーと要素との関係を維持しつつ配列を昇順にソートします。*2

■array compact(mixed varnames [, mixed...])

変数名の文字列または変数名の配列を引数とし、その変数名をキー、その変数の値を要素の値とする配列を返します。引数の数は可変です。存在しない変数名は無視されます。

```
<?php
 $n1 = "taro"; $n2 = "hanako"; $n3 = "jiro";
 $a = array("n1","n2");
 $b = compact("n3","n4",$a); // $n4は存在しないため無視される
 print_r($b); // $bはarray('n3'=>'jiro', 'n1'=>'taro', 'n2'=>'hanako')となる
?>
```

■int count(mixed var [, int mode])

変数var（通常は配列）に含まれる要素の数を返します。

COUNT_NORMAL　　：配列に関して再帰処理を行ないません（デフォルト）。
COUNT_RECURSIVE：多次元の配列を再帰的に数えます。

```
<?php
 $a = array(1, 2, 3, array(4, 5, 6)); // 4番目の要素は配列
 print_r($a);
 print count($a)."¥n"; // 4を出力
 print count($a, COUNT_RECURSIVE)."¥n"; // 7を出力
?>
```

■mixed current(array arr)

配列のカレント要素の値を返します。配列ポインタが最終要素の位置を超えている場合や、値が空文字列（""）または0の場合にFALSEを返します。

■array each(array arr)

配列から次の要素のキー/値を配列array('key'=>キー, 'value'=>値)または array(キー, 値)として返し、配列ポインタを進めます。要素がもうない場合はFALSEを返します。

```
<?php
 $name = array('c'=>"jiro", 'a'=>"taro", 'b'=>'hanako');
 reset($name); // 配列ポインタを先頭要素にセットする
 while (list($key, $val) = each($name)) {
   print "$key => $val<BR>¥n"; // jiro, taro, hanakoが出力される
 }
?>
```

■mixed end(array arr)

配列の内部ポインタを最後の要素まで進め、最後の要素の値を返します。

*2
オプションsort_flagsの使用法についてはsort()を参照してください。

■int extract(array arr [, int type [, string prefix]])

シンボルテーブルに変数をインポートし、インポートした変数の数を返します。配列arrのキーを変数名、値を変数の値とします。typeにより既存の変数名と衝突した場合の処理（**表3-11**）を指定します。

表3-11　typeの設定値と動作

typeの値（定数）	動作
EXTR_OVERWRITE	衝突時に既存の変数は上書きされる（デフォルト）
EXTR_SKIP	衝突時に既存の変数は上書きされない
EXTR_PREFIX_SAME	衝突時にprefixを前につけた変数とする
EXTR_PREFIX_ALL	すべての変数の前にprefixをつける
EXTR_PREFIX_INVALID	無効な変数名の場合にのみprefixを前につける
EXTR_IF_EXISTS	既存の変数の場合にのみ、その変数を上書きし、そのほかの場合には何もしない
EXTR_PREFIX_IF_EXISTS	同名の変数がある場合にのみ前にprefixを付けた変数名を作成する

```
<?php
 $name = "taro";
 $a = array("adrs"=>"tokyo", "name"=>"hanako");
 extract($a, EXTR_PREFIX_SAME, "n"); // $nameは衝突するので"n_"が付加される
 print "$adrs, $name, $n_name¥n";  // "tokyo, taro, hanako"が出力される
?>
```

■bool in_array(mixed srch, array arr [, bool strict])

配列arr内でsrchを検索し、見つかった場合にTRUE、それ以外の場合にFALSEを返します。strictにTRUEが指定された場合、srchの型が一致することも確認します。

```
<?php
 $a = array("1", 2);
 // それぞれ"一致"、"不一致"となる
 print in_array(1, $a) === TRUE ? "一致¥n" : "不一致¥n";
 print in_array(1, $a, TRUE) === TRUE ? "一致¥n" : "不一致¥n";
?>
```

■mixed key(array array)

配列の内部ポインタが指す要素のキーを返します。*5

int krsort(array array)

配列をキーで逆順にソートします。

bool ksort(array array [, int sort_flags])

F T

配列をキーでソートします。*2

```
<?php
 $a = array("c"=>"taro", "a"=>"hanako", "b"=>"jiro");
 print_r($a);
 ksort($a); // キーによりソートする
 for(reset($a); $key = key($a); next($a)) { // キーの順番（a、b、c）に表示する
     print "$key => {$a[$key]}<BR>¥n";
 }
?>
```

void list(mixed...)

同時に複数の変数に値を代入するために使用します。*3 *4

mixed max(mixed arg1 [, mixed arg2 [, mixed ...]])

F

配列または一連の引数の中で最大の値を返します。

mixed min(mixed arg1 [, mixed arg2 [, mixed ...]])

F

配列または一連の引数の中で最小の値を返します。

bool natsort(array arr)

T

自然ソートにより配列をソートします。

*3
list()は実際には関数ではなくPHPスクリプト言語の一部です。

*4
each()のサンプルを参照してください。

*5
ksort()のサンプルを参照してください。

```
<?php
 $a = array("12.png", "10.png", "2.png", "1.png");
 sort($a); // 通常のソート
 print_r($a); // 1.png、10.png、12.png、2.pngの順となる
 natsort($a); // 自然ソート
 print_r($a); // 1.png、2.png、10.png、12.pngの順となる
?>
```

■bool natcasesort(array arr)

T

自然ソートにより配列をソートします。大文字小文字を区別しません。

■mixed next(array arr)

配列の内部ポインタを次に進め、移動後の要素(終端を超える場合はFALSE)を返します。*5

■mixed prev(array arr)

配列の内部ポインタをひとつ戻し、移動後の要素（始端を超える場合はFALSE）を返します。

■array range(mixed low, mixed high)

F

[low,high]の範囲の整数または文字を値とする配列を生成します。

```
<?php
 $a = range(1,4);
 print_r($a);// $a[0]=1,$a[1]=2,$a[3]=3,$a[3]=4 となる
 $a = range('c','f');
 print_r($a);// $a[0]=c,$a[1]=d,$a[2]=e,$a[3]=f となる
?>
```

■mixed reset(array arr)

F

配列の内部ポインタを先頭に戻し、先頭の要素の値を返します。*5

■bool rsort(array arr [, int sort_flags])

F T

配列を降順にソートします。*2

```
<?php
 $a = array("taro","hanako","jiro");
 rsort($a);  // 配列$aを降順にソートします。
 reset($a);
 while(list($key, $value) = each($a)) {
     print "$key => $value<BR>¥n"; // taro、jiro、hanakoの順番に表示される
 }
?>
```

■bool shuffle(array arr)

F T

配列の順番をランダムに並べ替えます。使用前にsrand()関数による乱数の初期化が必要です。

■bool sort(array arr [, int sort_flags])

F T

配列を昇順にソートします。sort_flagsでソート方法を指定できます。

SORT_REGULAR：通常の比較（デフォルト）
SORT_NUMERIC ：数値として比較
SORT_STRING　：文字列として比較

```
<?php
 $a = array("3","abc",2);
 sort($a);
 print_r($a); // "3","abc",2の順番となる
 sort($a,SORT_NUMERIC);
 print_r($a); // "abc",2,"3"の順番となる
 sort($a,SORT_STRING);
 print_r($a); // 2,"3","abc"の順番となる
?>
```

■bool uasort(array arr, function func)

F T

ユーザ定義の比較関数funcにより配列を値でソートします。配列の添字と値の関係はソート後も不変です。

```
<?php
 function comp($x, $y) { // 逆順ソート用比較関数（$x、$yは配列の値）
     if ($x == $y) return 0;
     return ($x > $y) ? -1 : 1;
 }

 $a = array(5 => "five", 3 => "three", 20 => "twenty");
 uasort($a, 'comp'); // ユーザ関数により値でソート（添字と値の関係は不変）
 print_r($a); // 20=>twenty, 3=>three, 5=>five となる
 usort($a, 'comp'); // ユーザ関数により値でソート
 print_r($a); // 0=>twenty, 1=>three, 2=>five となる
?>
```

■bool uksort(array arr, function func)

F T

ユーザ定義の比較関数funcにより配列をキーでソートします。

```
<?php
 function kcomp($x, $y) { // 逆順ソート用比較関数（$x、$yは配列キー）
     if ($x == $y) return 0;
     return ($x > $y) ? -1 : 1;
 }

 $a = array(5 => "five", 3 => "three", 20 => "twenty");
 uksort($a, 'kcomp'); // ユーザ関数によりソート
 print_r($a); // 20=>twenty, 5=>four, 3=>three の順となる
?>
```

■bool usort(array arr, function func)

F T

ユーザ定義の比較関数funcにより配列を値でソートします。*6

*6
uasort()のサンプルを参照してください。

2.2 正規表現関数

PHPでは、POSIX互換の正規表現、Perl互換の正規表現、マルチバイト対応の正規表現を使用可能です。

2.2.1 POSIX互換正規表現

PHPには、POSIX 1003.2に基づく拡張正規表現ライブラリが付属しています。

▶ 組み込み方法

POSIX正規表現は、標準で組み込まれます。configureのオプション**--with-regex=TYPE**により組み込まれる関数の種類を指定可能です。TYPEには、php（PHP付属の関数、デフォルト）、apache（Apache付属の関数）、system（システムの標準関数）を指定します。

▶ 主なPOSIX正規表現

正規表現パターン

パターン	マッチする対象
.	任意の一文字
^	文字列の先頭
$	文字列の末尾
x*	直前の部分正規表現（例ではx）の0回以上の反復
x+	直前の部分正規表現（例ではx）の1回以上の反復
x?	直前の部分正規表現（例ではx）の0または1回の反復
[abc]	括弧の中の文字集合（例ではabc）のどれか
[^abc]	括弧の中の文字集合（例ではabc）に含まれないもの
[a-z]	先頭文字から終端文字までの範囲の文字（例ではa,b,c, ... ,z）
abc|xyz	パターンの和（例ではabcまたはxyz）
x{n}	xのn回反復にマッチ
x{n,}	xのn回以上の反復にマッチ
x{n,m}	xのn回以上m回までの反復にマッチ

文字クラス

パターン	マッチする対象
[[:alnum:]]	英数字
[[:digit:]]	数値
[[:alpha:]]	英字（アルファベット）

■bool ereg(string regex, string str [, array &regs])

正規表現により大文字･小文字を区別して検索し、マッチした場合にTRUE、それ以外の場合にFALSEを返します。regsを指定した場合、regexの()でくくられた部分にマッチした部分文字列が順次代入され、$regs[0]はstr、$regs[i]はi番目の括弧にマッチした文字列となります。

```
<?php
 $str = "2002-1-1 Tokyo/Japan";
 if (ereg("([0-9]{4})-([0-9]{1,2})-([0-9]{1,2})", $str, $regs)) {
   print_r($regs); // array("2002-1-1","2002","1","1") を出力
 }
 if (ereg("[[:space:]]([[:print:]]+)$", $str, $regs)){
   print_r($regs); // array("Tokyo/Japan","Tokyo/Japan") を出力
 }
?>
```

■string ereg_replace(string regex, string rstr , string str)

正規表現により文字列を検索し、マッチした文字列を文字列rstrで置換します。regex内に()でくくられた文字列は、rstrの中で¥¥iにより参照できます。

```
<?php
 $url = "http://www.phpserv.jp/index.php";
 print ereg_replace("[[:alpha:]]+://[^<>[:space:]]+[[:alnum:]/]",
                    "<a href=¥"¥¥0¥">¥¥0</a>", $url);
 // <a href="http://www.phpserv.jp/index.php">http://www.phpserv.jp/index.php</a>
 // を出力
?>
```

■bool eregi(string regex, string str [, array regs])

ereg()と動作は同じですが、大文字･小文字の区別を行ないません。

■string eregi_replace(string regex, string rstr, string str)

ereg_replace()と動作は同じですが、大文字･小文字の区別を行ないません。

■array split(string regex, string str [, int lim])

正規表現により文字列を最大limの要素に分割し、配列として返します。

```
<?php
 $passwd = "taro:secret:100:100:Suzuki Taro:/home/taro:bin/bash";
 list($user,$pass,$extra)= split(":", $passwd, 3);
 print "$user,$pass¥n"; // taro,secretを出力
?>
```

■array spliti(string regex, string str [, int lim])

spliti()と動作は同じですが、大文字・小文字の区別を行ないません。

■string sql_regcase(string str)

大文字小文字を区別せずに文字列strにマッチする正規表現を返します。

```
<?php
 print sql_regcase("taro"); // [Tt][Aa][Rr][Oo]を出力します。
?>
```

2.2.2 Perl互換の正規表現

PHPにはPerl互換の正規表現ライブラリ（PCRE）が付属しており、標準でPerl互換の正規表現を使用可能です。

▶Perl正規表現のオプション

以下の正規表現マッチング用オプションを指定できます（例：/abc/i）。

オプション	説明
i	大文字小文字を区別せずにマッチする
x	パターンの空白文字を無視する
e	preg_replace()において使用され、置換後の文字列をPHPのコードとしてパース/実行する
u	マッチ対象文字列の文字コードをUTF-8として評価する（マルチバイト文字使用可能）

▶参考URL

PCREの開発・配布元：**http://www.pcre.org/**

■array preg_grep(string regex, array arr)

正規表現にマッチする要素を配列として返します。

```
<?php
 $a = array('may'=>.2,'june'=>3,'july'=>2.4);
 $r = preg_grep("/^(¥d+)?¥.¥d+$/", $a); // 浮動小数点を含む要素を返す
 print_r($r); // Array([may] => 0.2 [july] => 2.4)を出力
?>
```

■int preg_match(string regex, string str [,array &regs])

F

正規表現にマッチする場合に1、マッチしない場合に0、エラーの場合にFALSEを返します。配列regsを指定した場合、最初の要素にマッチした文字列全体、以降の要素には括弧でくくられたサブパターンにマッチする文字列が代入されます。

```
<?php
 $url="http://www.php.net/";
 if(preg_match("/^(http:¥/¥/)?([^¥/]+)/i",$url,$regs)){
   print_r($regs);
     // array("http://www.php.net","http://","www.php.net")を出力
 }
?>
```

■int preg_match_all(string regex, string str ,array &regs [,int order])

正規表現にマッチした部分を配列に取得し、マッチした回数を返します。orderには以下の二種類のオプションを指定できます。

PREG_PATTERN_ORDER（デフォルト）：$regs[0]がパターン全体にマッチする配列、$regs[i]がi番目の括弧でくくられたサブパターンにマッチする文字列の配列となります。

PREG_SET_ORDER：$regs[i-1]がi番目にマッチした文字列の配列となります。

```
<?php
 $html="<b>this</b> is a pen.";
 if(preg_match_all("/<[^>]+>(.*)<¥/[^>]+>/i",$html,$regs)){
   print_r($regs); // $regs[0][0]="<b>this</b>",$regs[1][0]="this"となります。
 }
 if(preg_match_all("/<[^>]+>(.*)<¥/[^>]+>/i",$html,$regs,PREG_SET_ORDER)){
   print_r($regs); // $regs[0]はarray("<b>this</b>","this")となります。
 }
?>
```

■string preg_quote(string str [, string delim])

正規表現の特殊文字を'¥'でエスケープして返します。delimで指定した文字もエスケープされます。

```
<?php
 $str = "d=(a+b)/c";
 print preg_quote($str, "/"); // d¥=¥(a¥+b¥)¥/c を出力
?>
```

■string preg_replace(mixed regex, mixed rstr ,mixed str [, int max])

文字列strにおいて正規表現regexに一致する部分をrstrに置換します。最大max回（デフォルト：無制限）のマッチが行なわれます。正規表現にオプションeを指定した場合、置換文字列でPHPスクリプトのコードを記述することができます。

```
<?php
 $patterns = array("|(¥d{4})/(¥d{1,2})/(¥d{1,2})|","/tokyo/i");
 $replace = array("¥¥1-¥¥2-¥¥3","Osaka");
 $subject = "2002/1/1 Tokyo,Japan";
 print preg_replace($patterns,$replace,$subject)."¥n";
                                                  // 2002-1-1 Osaka,Japanを出力
 print preg_replace("/(¥d{4})/e",'¥¥1-2000',$subject)."¥n";
                                                  // 2/1/1 Tokyo,Japanを出力
?>
```

■strin preg_replace_callback(mixed regex, mixed callback ,mixed str [, int max])

文字列strにおいて正規表現regexに一致する部分をコールバック関数の返り値に置換します。

```
<?php
 function fun($str){ // コールバック関数
   return date("l",mktime(0,0,0,$str[2],$str[1],$str[0]));
 }

 $regex = "|(¥d{4})/(¥d{1,2})/(¥d{1,2})|";
 print preg_replace_callback($regex,'fun',"2002/2/1"); // Thursdayを出力
?>
```

■array preg_split(string regex, string str [,int lim, [int flags]])

Perl互換の正規表現により文字列を最大limの要素に分割し、配列として返します。

```
<?php
 $str = preg_split("/[¥s,]+/", "Hello, Taro Suzuki"); // 空白文字またはカンマで分割
 print_r($str); // $strはarray("Hello","Taro","Suzuki") となる
?>
```

2.3 文字列関数

PHPでは、Webアプリケーションにおいて有用な文字列処理を行なう多くの関数がサポートされています。文字列に含まれるHTMLや特殊文字をエスケープする関数（htmlentities()やaddslashes()など）は、セキュリティの観点からユーザ入力の前処理に使用されます。

■string addcslashes(string str, string range)

指定した範囲（先頭文字 … 終端文字）の文字をバックスラッシュでエスケープして返します。C言語と同様に「¥0, ¥a, ¥b, ¥f, ¥n, ¥r, ¥t, ¥v」はエスケープ文字と認識されます。rangeにはASCIIコードを8/16進数で指定でき、¥0 … ¥31のように範囲を指定することも可能です。

```
<?php
 $a = "!foo@¥n"; // 対象文字列
 print addcslashes($a,'a..z')."¥n";  // '!¥f¥o¥o@' を出力（文字指定）

 // ASCII非表示文字と!@をエスケープ
 print addcslashes($a,"¥0..¥37!@¥177..¥377")."¥n";  // '¥!foo¥@¥n'を出力（8進数指定）
 print addcslashes($a,"¥x0..¥x1f!@¥x7f..¥xff")."¥n";  // 同上（16進数指定）
?>
```

■string addslashes(string str)

データベースへのクエリの際にクォートが必要な文字（'、"、¥、¥0）をバックスラッシュ（¥）でエスケープした文字列を返します。設定オプション（php.iniで指定）でmagic_quotes_sybaseをOnをとした場合、' は ''でエスケープされ、" 、¥ はエスケープされません。

```
<?php
 $a = "'¥¥".chr(0x0)."¥""; // 対象文字列
 print addslashes($a)."¥n";
 // 「¥'¥¥¥0¥"」を出力 : magic_quotes_sybase = Off（デフォルト）の場合
 // 「''¥¥0"」を出力 : magic_quotes_sybase = Onの場合
?>
```

■string bin2hex(string str)

バイナリデータを16進数の文字列に変換して返します。

```
<?php
 $a = chr(0xde).chr(0x0).chr(0xdf); // バイナリデータ
 print bin2hex($a); // 'de00df' を出力
?>
```

■string chop(string str)

文字列から最後尾の空白文字を取り除いて返します。

```
<?php
 $line = "lemon appple  ";
 print chop($line); // "lemon apple" を出力
?>
```

■string chr(int ascii)

ASCIIコードで指定した1文字からなる文字列を返します。

```
<?php
 print chr(27)."B"; // ESC-Bを出力
?>
```

■string chunk_split(string str [, int len [, string end]])

文字列strにlen（デフォルトは76）バイトごとに文字列end（デフォルトは¥r¥n）を挿入して返します。*7

■mixed count_chars(string str [, int mode])

各バイト値（0 … 255）が存在する数を数え、**表3-12**に示すmodeに基づき値を返します。

*7
バイト単位で分割するため、日本語文字列の場合は文字化けする可能性があります。マルチバイト関数（mbstring）を使用してください。

表3-12　modeの設定値

値	説明
0	各バイト値をキー、各バイトの出現回数を値とする配列（デフォルト）
1	0と同じだが、出現回数>0のバイト値だけを抽出
2	0と同じだが、出現回数がゼロのバイト値だけを抽出
3	使用されているすべてのバイト値を有する文字列を返す
4	使用されていないすべてのバイト値を有する文字列を返す

```
<?php
 $str = "hello";
 $a = count_chars($str); print_r($a); // 各バイトを出現回数とする配列を出力
 $a = count_chars($str, 1); print_r($a); // 出現回数>0の要素のみの配列
 $a = count_chars($str, 2); print_r($a); // 出現回数=0の要素のみの配列
 print count_chars($str, 3)."¥n"; // 使用されているバイト値からなる文字列
 print count_chars($str, 4)."¥n"; // 使用されていないバイト値からなる文字列
?>
```

■string crc32(string str)

文字列からCRC32多項式を計算し、文字列として返します。

```
<?php
 print crc32("hello"); // 907060870を出力
?>
```

■string crypt(string str [, string salt])

crypt関数により一方向の暗号化を行ないます。暗合のsaltを指定可能（省略時はPHPが生成）です。saltの長さはシステムに依存し、定数CRYPT_SALT_LENGTHにより取得可能です（DESの場合：2、MD5の場合：12）。

```
<?php
 print "サポートされているアルゴリズム: ";
 print CRYPT_STD_DES ? "STD_DES " : ""; print CRYPT_EXT_DES ? "EXT_DES " : "";
 print CRYPT_MD5 ? "MD5 " : ""; print CRYPT_BLOWFISH ? "BLOWFISH¥n" : "¥n";
 print "saltの長さ: ".CRYPT_SALT_LENGTH."¥n";

 $password = "this is a password."; // パスワード文字列
 $crypted = crypt($password); // cryptにより暗合化
 $user_input = "this is a password."; // ユーザ入力が正しいと仮定
 if (crypt($user_input, $crypted) === $crypted){  // パスワードをチェック
   print "パスワードOK";
 } else {
   print "パスワードNG";
 }
?>
```

■echo(string arg1 [, string argn]...)

引数を出力します。echo()は関数ではなくPHPスクリプト言語の一部です。括弧は省略可能です。

■array explode(string sep, string str [, int lim])

文字列strを文字列sepにより分割し、結果を配列として返します。引数limにより要素数の上限を指定可能で、この場合、最後の要素に残りの文字列が含まれます。

```
<?php
 $a = "Yamada Taro";
 $b = explode(" ", $a);
 print_r($b); // array("Yamada", "Taro") を出力
?>
```

■void flush(void)

出力バッファをフラッシュし、ブラウザに出力します。*8

■string get_html_translation_table([int table [, int quote_table]])

HTMLにおける特殊文字やエンティティの変換テーブルを返します。引数tableにはHTML_SPECIALCHARS（デフォルト）、HTML_ENTITIESを指定します。

```
<?php
 $trans = get_html_translation_table(HTML_ENTITIES); // HTMLエンティティの変換データを取得
 $a = "Taro & Hanako";
 $b = strtr($a, $trans); // 「Taro &amp; Hanako」となる
?>
```

■array get_meta_tags(string file [, int path])

ファイル（またはURL）からmetaタグ<meta ……>を取得し、連想配列として返します。nameプロパティの値がキー、contentプロパティの値が値となります。*9 *10

例として以下のファイルtags.incを処理することを考えます。

●tags.incの内容

```
<head>
 <meta name="author" content="Taro">
 <meta name="tags" content="php document">
</head>
```

```
<?php
 $meta = get_meta_tags("tags.inc"); // metaタグを取得
 print $meta["author"]; // 'Taro'を出力
?>
```

■string htmlentities(string str [, int quote_style [, string charset]])

指定した文字セットにおいて変換可能な文字をすべてHTMLエンティティに変換して返します。引用符に関する動作を設定できます（**表3-13**）。デフォルトでISO-8859-1文字セットが使われます。

表3-13　quote_styleの設定値

quote_styleの値	動作
ENT_COMPAT	二重引用符（"）のみを変換し、一重引用符（'）はそのままとする（デフォルト）
ENT_QUOTES	二重引用符（"）と一重引用符（'）を両方とも変換する
ENT_NOQUOTES	二重引用符（"）と一重引用符（'）のどちらも変換しない

```
<?php
 print htmlentities("<nihongo>")."¥n"; // 出力: &lt;nihongo&gt;
 print htmlentities("<日本語>",ENT_NOQUOTES,"EUC-JP")."¥n"; // 出力: &lt;日本語&gt;
?>
```

■string htmlspecialchars(string str [, int quote_style [, string charset]])

HTMLにおける特殊文字（&、"、'、<、>）をマークアップ文字（&、"、'、<、>）に変換して返します。ユーザ入力をHTMLに変換する際に便利です。引用符に関する動作を設定できます（**表3-13**）。

■string implode(array arr, string str)　［エイリアス］join()

配列arrの要素を文字列strをはさんでひとつの文字列にして返します。

```
<?php
 $array = array('taro','hanako','jiro');
 print implode($array, ":"); // "taro:hanako:jiro"を出力
?>
```

■array localeconv(void)

カレントのロケールに基づき、数値フォーマット情報を返します。

*8
Webサーバの種類によっては使用できない場合があります。

*9
use_incに1を指定することにより、include_pathで指定したパスの検索も行なわれます。

*10
ファイル名がhttp://で始まっている場合はHTTP、ftp://で始まっている場合はFTPの接続がオープンされます（php.iniのディレクティブallow_url_fopen = Onの場合）。

string ltrim(string str [, string range])

文字列の先頭から空白文字（¥n、¥r、¥t、¥v、¥0）を取り除いたものを返します。*11

```
<?php
 print ltrim("¥t¥tHello")."¥n"; // 先頭のタブを削除して出力
 print ltrim(chr(0x1b)."Hello","¥x00..¥x1f")."¥n"; // 先頭のエスケープ文字を削除して出力
?>
```

string md5(string str)

文字列のMD5ハッシュ値を計算して返します。

string nl_langinfo(int item)

ロケールに関する設定値を返します。

```
<?php
 print nl_langinfo(DAY_1); // 'Sunday'を出力
 setlocale(LC_ALL,"ja_JP.eucJP");
 print nl_langinfo(DAY_1); // '日曜日'を出力
?>
```

string nl2br(string str)

文字列に含まれるすべての改行文字の前に
を挿入して返します。

```
<?php
 print nl2br("Hello,¥n Taro."); // Hello,<br />¥n Taro." を出力
?>
```

int ord(string str)

文字列の先頭の文字のASCIIコードを返します。

*11
rangeに削除する文字の範囲を指定することが可能です。addcslashes()のサンプルを参照してください。

```
<?php
 $str = "¥n Hello";
 print ord($str) == 10 ? "改行文字" : "それ以外"; // "改行文字"を表示
?>
```

■void parse_str(string str [, array result])

文字列をURL引数と同様に処理し、グローバル変数を作成します。

```
<?php
 $str = "a=hanako&b[]=yamada+taro&b[]=jiro";
 parse_str($str); // $strを処理
 print "$a¥n"; // "hanako"を出力
 print_r($b); // $bはarray("yamada taro","jiro")となる
?>
```

■print(string str)

文字列strを出力します。

■int printf(string format [, mixed args] ...)

フォーマットformatに基づき、引数を出力します。フォーマット指定はsprintf()と同じです。

■string quoted_printable_decode(string str)

quoted printableでエンコードされた文字列strをデコードし、8ビットバイナリ文字列として返します。

■string quotemeta(string str)

文字「. ¥ + * ? [^] ($) 」の前にバックスラッシュ(¥)を付加した文字列を返します。

```
<?php
 print quotemeta("http://foo/?c=b+d"); // http://foo/¥?c=b¥+d を出力します。
?>
```

■string rtrim(string str [, string range])

末尾の空白文字（¥n、¥r、¥t、¥v、¥0）を取り除いて返します。*11

```
<?php
 print rtrim("Hello.¥n")."¥n"; // 末尾の改行文字を削除して出力
 print rtrim("Hello.¥n".chr(0x1b),"¥x00..¥x1f")."¥n"; // 末尾の0x00～0x1fの文字を削除して出力
?>
```

■string setlocale(mixed category, string locale)

F

指定したカテゴリに関するロケールを設定し、ロケール値を返します。categoryにはの有効範囲（表3-14）を指定します。localeに0を指定した場合、現在のロケールを返します。

表3-14　categoryの設定値（有効範囲）

categoryの設定値（定数）	設定範囲
LC_ALL	すべてのロケール設定
LC_COLLATE	正規表現のマッチング
LC_CTYPE	文字の分類と変換（例：strtoupper()）
LC_MESSAGES	地域に依存するメッセージ
LC_MONETARY	貨幣単位
LC_NUMERIC	数字の区切り文字用
LC_TIME	日時（例：strftime()のフォーマット文字列）

```
<?php
 $s1 = "あ"; $s2 = "お";

 setlocale(LC_ALL, "ja_JP.eucJP");
 print strcoll($s1,$s2)."¥n"; // -8を出力
 print strcoll($s2,$s1)."¥n"; // 8を出力

 setlocale(LC_COLLATE, "C");
 print strcoll($s1,$s2)."¥n"; // -1を出力
 print strcoll($s2,$s1)."¥n"; // 1を出力
?>
```

■int similar_text(string str1, string str2 [, float percent])

二つの文字列の類似性を計算します。

```
<?php
 $str1 = "Is this apple.";
 $str2 = "What't up ?";
 print similar_text($str1, $str2); // 2を出力
?>
```

■string sprintf(string format, mixed [args] ...)

フォーマットを指定して引数argsを文字列に出力します。フォーマットには、%で始まるフォーマット指定子を指定可能です。詳細は、C言語の関数sprintf()のマニュアルを参照してください。

```
<?php
 $year = 2000; $month = 9; $day = 1;
 print sprintf("%04d/%02d/%02d", $year, $month, $day); // '2000/09/01' を出力
?>
```

■mixed sscanf(string str, string format [, string ...])

フォーマットを指定して文字列から変数の値を取得します。

```
<?php
 $str = "book 1000";
 sscanf($str, "%s %d", $product, $price);
 print $product . ":". $price; // "book:1000"を出力
?>
```

■string str_pad(string str, int pad_length [, string pad [,int pad_type]])

文字列strの左、右または両側を指定した長さで埋めます。padは埋める際に用いる文字列（デフォルト：空白）です。pad_typeに埋める側を以下のように指定可能です。

STR_PAD_RIGHT(右、デフォルト)、STR_PAD_LEFT(左)、STR_PAD_BOTH(両方)

■int strcasecmp(string str1, string str2)

大文字･小文字を区別せずに文字列比較を行ない、str1>str2の場合に正、str1<str2の場合に負、等しい場合に0を返します。

```
<?php
 print strcasecmp("Hello","HELLO") === 0 ? "等しい" : "異なる";
?>
```

int strcmp(string str1, string str2)

大文字･小文字を区別すること以外は、strcasecmp()と同じです。

int strcoll(string str1, string str2)

カレントのロケール設定値基づき文字列の比較を行ないます。[*12]

int strcspn(string str, string mask)

文字列においてマスク文字列maskの文字がどれも含まれていない最初のセグメントの長さを返します。

string strip_tags(string str [, string tags])

文字列からHTMLタグおよびPHPタグを取り除いて返します。引数tagsにより削除しないタグを指定可能です。

```
<?php
 $str = '<?php echo "Hello,";?><b>Taro</b>.';
 print strip_tags($str,"<b>"); // <b>Taro</b>. を出力
?>
```

string stripcslashes(string str)

文字列からバックスラッシュを取り除いて返します。C言語のエスケープ文字(¥n、¥r、…)はそのままとなります。

```
<?php
 $a = "fit";
 $b = addcslashes($a, "a..z")."¥n"; print $b; // "¥f¥i¥t"を出力
 print stripcslashes($b)."¥n"; // ¥f（LF）、i、¥t（TAB）を出力
?>
```

■string stripslashes(string str)

文字列からバックスラッシュを取り除いて返します。¥¥は、¥に変換されます。

```
<?php
 $a = "¥$foo¥¥¥¥";
 print stripslashes($a)."¥n"; // $foo¥を出力
?>
```

■string stristr(string str)

大文字･小文字を区別しないこと以外は、strstr()と同じです。

■int strlen(string str)

文字列の長さ（単位：バイト）で返します。バイナリデータにも使用可能です。

```
<?php
 print strlen("Hello!"); // 6を出力
?>
```

■int strnatcasecmp(string str1, string str2)

大文字･小文字を区別しないこと以外は、strnatcmp()と同じです。

■int strnatcmp(string str1, string str2)

「自然」アルゴリズムにより比較を行ないます。

```
<?php
 print strcmp("10.jpg","2.jpg"); // 通常のソート：-1を出力
 print strnatcmp("10.jpg","2.jpg"); // 自然順ソート：1を出力
?>
```

*12
setlocale() を参照。

■int strpos(string str, string key [, int offset])

文字列でキー文字列（key）が最初に見つかった位置（見つからない場合はFALSE）を返します。offsetにより検索開始位置を指定可能です。

■string strrchr(string str, string key)

文字列でキー文字（key）が最後に現れた位置からの部分文字列（見つからない場合はFALSE）を返します。keyは最初の文字のみが使用されます。

```
<?php
 print strrchr("Hello","l"); // "lo"を出力
?>
```

■string str_repeat(string str, int n)

文字列をn回反復した文字列を返します。

■string str_rot13(string str)

文字列のrot13変換を行ないます。

■string strrev(string str)

文字列を逆順にして返します。

■int strrpos(string str, string key)

キー（key）が最後に現れる位置（見つからない場合はFALSE）を返します。keyは最初の文字のみが使用されます。

```
<?php
 print strrpos("Hello","l"); // 3を出力
?>
```

■int strspn(string str, string mask)

すべて文字列maskの文字からなる最初のセグメントの長さを返します。

```
<?php
 print strspn("123 4567 890", "1234567890"); // 3を出力
?>
```

■string strstr(string str, string key)　［エイリアス］strchr()

キー（key）が最初に現れる場所から文字列の終わりまで（見つからない場合はFALSE）を返します。

```
<?php
 print strstr("Hello, Taro",","); // ", Taro"を出力
?>
```

■string strtok([string str,] string sep)

文字列をセパレータ（sep）によりトークンに分割して返します。2回目以降の分割ではセパレータのみを指定します。

```
<?php
 $a = "This is a pen";
 $tok = strtok($a," "); // 空白文字によりトークンに分割
 while($tok) {
   print "$tok:"; // 'This:is:a:pen'を出力
   $tok = strtok(" ");
 }
?>
```

■string strtolower(string str)

文字列のアルファベットをすべて小文字に変換して返します。

```
<?php
 print strtolower("Hello"); // helloを出力
?>
```

```
<?php
 print strtolower("Hello"); // helloを出力
?>
```

■string strtoupper(string str)

文字列のアルファベットをすべて大文字に変換して返します。

■mixed str_replace(mixed key, mixed rstr, mixed str)

文字列strの中の文字列keyをすべて文字列rstrに置換して返します。

```
<?php
 // "<body text=black>"を出力
 print str_replace("%body%", "black", "<body text=%body%>");
?>
```

■string strtr(string str, string from, string to)

文字列からfromの文字を検索し、toの対応する文字に置換して返します。fromとtoの長さが異なる場合、長い方の余分な文字は無視されます。

■string substr(string str, int pos [, int len])

文字列の位置pos（先頭文字の位置が0）からlenバイト分の文字列を返します。posに負の数を指定した場合、末尾からの位置となります。

```
<?php
 print substr("abcdef", 1)."¥n";     // "bcdef"を出力
 print substr("abcdef", 1, 3)."¥n";  // "bcd"を出力
 print substr("abcdef", -3, 1)."¥n"; // "d"を出力
?>
```

■int substr_count(string str, string rpos)

文字列の中に部分文字列rposが出現する回数を返します。

■string substr_replace(string str, string rpos, int pos [, int len])

文字列の位置pos（先頭文字の位置が0）からlenバイト分の文字列をrposで置換して返します。posに負の数を指定した場合、末尾からの位置となります。

```
<?php
 print substr_replace("abcdef","pp",3,2); // "abcppf"を出力
?>
```

■string trim(string str)

文字列の先頭および末尾にある空白文字（¥n、¥r、¥t、¥v、¥0、' '）を取り除いた文字列を返します。

```
<?php
 print trim("¥nfoo¥n"); // "foo"を出力
?>
```

■string ucfirst(string str)

文字列の最初の文字がアルファベットの場合、大文字に変換します。

```
<?php
 print ucfirst("japan"); // "Japan"を出力
?>
```

■string ucwords(string str)

文字列の各単語の最初の文字を大文字に変換します（アルファベットの場合のみ）。

```
<?php
 print ucwords("this is a pen."); // "This Is A Pen."を出力
?>
```

■string wordwrap(string str [,int width [, string break [, int cut]]])

F

指定した文字数（デフォルト：75）で分割文字（デフォルト："¥n"）を挿入して返します。英単語が途中で分割されないように分割文字が挿入されます。

```
<?php
 $str = "This is a long string.";
 print wordwrap($str, 15); // "This is a long string." を出力
?>
```

2.4 変数操作関数

PHPの変数は必要に応じて動的に作成され、型も整数、文字列などの中から動的に選択されます。

■mixed call_user_func(string func [, mixed param [, mixed ...]])

ユーザ定義関数funcをコールします。関数のパラメータparamを指定可能です。

```
<?php
 function sqr($x) {
   return $x*$x;
 }
 print call_user_func('sqr', 3); // ユーザ定義関数をコール
?>
```

クラス（オブジェクト）のメソッドをコールすることも可能です。この場合、最初の引数に配列で（インスタンス変数, メソッド名）を指定します。

```
<?php
 class Hello { // ユーザ定義クラス
   function show($name=""){
     print "Hello, $name¥n";
   }
 }

 $obj = new Hello();
 call_user_func(array(&$obj,"show"),"Taro"); // ユーザ定義メソッドをコールします。
?>
```

■mixed call_user_func_array(string func [, array param])

ユーザ定義関数funcをコールします。関数のパラメータparamを配列で指定可能です。

```
<?php
 function add($x,$y,$z) {
   return $x+$y+$z;
 }
 print call_user_func_array('add', array(1, 2, 3));
?>
```

■bool define(string name, mixed value [, bool icase])

定数nameを値valueで定義します。icaseにTRUEを指定した場合、定数の大文字・小文字は区別されません（デフォルトでは区別）。

```
<?php
 define("NAME", "Taro"); // 定数NAMEを定義
 print NAME."¥n"; // "Taro"を表示
 define("NAMEI", "Nakamura Taro", TRUE); // 大文字小文字を区別せずに定義
 print Namei."¥n"; // "Taro"を表示
?>
```

■bool defined(string name)

定数nameが定義されている場合にTRUE、それ以外の場合にはFALSEを返します。

```
<?php
 define("NAME", "Taro"); // 定数NAMEを定義
 if (defined("NAME")){ // NAMEが定義されている場合に表示
     print NAME;
 }
?>
```

■bool empty(mixed var)

変数が設定され、ゼロ以外の値を有する場合にFALSE、それ以外の場合にはTRUEを返します。*13

*13 empty()は関数ではなくPHPスクリプト言語の一部です。

■string gettype(mixed var)

変数の型を以下の文字列で返します。

boolean、integer、resource、double、string、array、object、NULL、unknown type

個々の型に関して確認を行なう際には、**表3-15**の関数を使用します。

表3-15　型の確認用関数

関数	動作および返り値
bool is_array(mixed var)	引数が配列の場合にTRUE、それ以外はFALSEを返す
bool is_bool(mixed var)	引数が論理値の場合にTRUE、それ以外はFALSEを返す
bool is_float(mixed var)	引数がfloatの場合にTRUE、それ以外はFALSEを返す
bool is_long(mixed var)	引数がlong整数の場合にTRUE、それ以外はFALSEを返す
bool is_null(mixed var)	引数がNULLの場合にTRUE、それ以外はFALSEを返す
bool is_numeric(mixed var)	引数が数値の場合にTRUE、それ以外はFALSEを返す
bool is_object(mixed var)	引数がオブジェクトの場合にTRUE、それ以外はFALSEを返す
bool is_resource(mixed var)	引数がリソースの場合にTRUE、それ以外はFALSEを返す
bool is_scalar(mixed var)	引数がスカラーの場合にTRUE、それ以外はFALSEを返す
bool is_string(mixed var)	引数が文字列の場合にTRUE、それ以外はFALSEを返す

■bool import_request_variables(string types [, string prefix])

ユーザ入力（POST/GET/Cookie変数）をグローバル変数にインポートします。

typesには、取得する変数入力の種類（"P"：POST、"G"：GET、"C"：Cookie）を指定します。

複数の指定を入力を指定することも可能ですが、同名の変数がある場合、あとから指定した入力パラメータで上書きされます。グローバル変数の前にprefixを付加することも可能です。

```
<?php  // 例：http://ホスト名/パス/スクリプト名.php?foo=10により10を出力
 import_request_variables("GP", "MY_"); // GET,POST変数を取得、接頭辞 $MY_ を付加
 print $MY_foo; // 変数の値を出力
?>
```

■int isset(mixed var)

変数が存在すればTRUE、そうでなければFALSEを返します。*14

*14
isset()は関数ではなくPHPスクリプト言語の一部です。

```
<?php
 $a = "test";
 print isset($a) ? "定義¥n" :  "未定義¥n"; // "定義"を表示
 unset($a);  // 変数 $aの割当を解除
 print isset($a) ? "定義¥n" :  "未定義¥n"; // "未定義"を表示
?>
```

■string pack(string format [, mixed args])

引数をフォーマットに基づきバイナリ文字列にパックし、バイナリデータとして返します。フォーマット指定用のコードも含めて動作はPerlの同名の関数と同じです。

フォーマットは特定の型を表す**パラメータ＋反復指定子**の形式で指定します。反復指定子には整数値または *****（最後まで反復）を指定します。[*15]

*15
unpack()のサンプルを参照してください。

■void print_r(mixed exp)

変数の値に関する情報を表示します。

```
<?php
 $a = array("taro","hanako");
 print_r($a); // Array ( [0] => taro [1] => hanako ) を出力
?>
```

■string serialize(mixed value)

値valueをシリアル化したバイト文字列を返します。オブジェクトや配列などの型や構造を失わずにPHPの値を保存したい場合に使用します。オブジェクトのメソッドはシリアル化されません。

```
<?php
 $a = array("taro","hanako");
 $str = serialize($a); // シリアル化
 echo $str; // a:2:{i:0;s:4:"taro";i:1;s:6:"hanako";} を出力
 $ar = unserialize($str);
 print_r($ar); // $arは$aと同じ配列となる
?>
```

■int settype(string var, string type)

変数の型を指定した値（下記）に変更します。

integer、double、string、array、object、boolean、resource

型を変更せずに指定した型で値を得るには、**表3-16**の関数を使用します。

表3-16　指定した型で値を得る関数

関数		動作および返り値
float	floatval(mixed var)	floatとして値を取得
int	intval(mixed var [,int base])	整数として値を取得。基数baseを指定可能
string	strval(mixed var)	文字列として値を取得

■string uniqid(string prefix [, bool lcg])

ミリ秒単位の現在時刻に基づいて、先頭にprefixがついたユニークなIDを返します。lcgにtrueを指定した場合、combined LCGが追加され、よりユニークな値となります。

■array unpack(string format, string data)

formatに基づきバイナリデータdataを解釈し、結果を連想配列として返します。

フォーマットの指定方法はpack()と同じです。分割する要素ごとにフォーマット文字列のうしろに連想配列のキーを指定し、スラッシュ（/）で区切ります。

```
<?php
 $bd = pack("c2n", 0x40, 0x50 , 1234);
 $a = unpack("c2chars/nint", $bd);
 print_r($a); // $aはarray("chars1"=>64,"chars2"=>80,"int"=>1234) となる
?>
```

■mixed unserialize(string str)

シリアル化された変数を元の型の変数として返します。[*16]

■void unset(mixed var)

変数を破棄します。*17

```
<?php
 $a = array('date'=>'9', 'month'=>'Jan');
 print_r($a);
 unset($a['date']); // 連想配列$aのキー'date'およびその値を破棄
 print_r($a);
 unset($a);         // 変数$aを破棄
?>
```

■void var_dump(mixed exp)

指定したPHPの式(exp)について型や値を含む構造化された情報を返します。

```
<?php
 $a = array("taro","hanako");
 var_dump($a);
 // array(2) { [0]=> string(4) "taro" [1]=> string(6) "hanako" } を出力
?>
```

■mixed var_export(mixed exp [, bool return])

指定したPHPの式(exp)の結果を表すPHPスクリプトコードを出力します。returnにTRUEを指定した場合は、結果を文字列で返します。

```
<?php
 $a = array("taro","hanako");
 $b = var_export($a, TRUE);
 print $b; // array(0=>'taro', 1=>'hanako')を出力
?>
```

*16 serialize()のサンプルを参照。

*17 unset()は関数ではなくPHPスクリプト言語の一部です。

2.5 オブジェクト・クラス関連

bool class_exists(string class_name)

指定したクラスが存在する場合にTRUE、それ以外の場合にFALSEを返します。

string get_class(object obj)

オブジェクトobjをインスタンスとするクラスの名前を返します。

array get_class_vars(string class_name)

指定したクラスのデフォルト値を有するプロパティを連想配列として返します。

```
<?php
 class Hello {
   var $str = "Hello, ";
   function show($name="") {
     print "$str $name";
   }
 }

 $obj = new Hello();
 $vars = get_class_vars(get_class($obj));
 print_r($vars); // Array([str] => Hello)を出力
?>
```

array get_class_methods(string class_name)

クラスメソッドの名前を配列として返します。

array get_declared_classes(void)

カレントのスクリプトで宣言されているクラスの名前を配列として返します。

■array get_object_vars(object obj)

オブジェクトのプロパティを配列として返します。

■string get_parent_class(mixed obj)

オブジェクトのインスタンスまたはクラス名を引数とし、親クラスの名前を返します。

■bool is_a(object obj, string class_name)

オブジェクトがclass_nameまたはサブクラスのインスタンスの場合にTRUE、それ以外の場合にFALSEを返します。

```
<?php
 class Hello {
   function Hello() {  return TRUE;  }
 }

 class Hello2 extends Hello {
   function Hello2() {  return TRUE;  }
 }

 $obj = new Hello();
 $obj2 = new Hello2(); // Helloのサブクラスのインスタンス

 print is_a($obj, "Hello") ? "Yes¥n": "No¥n"; // Yesを表示
 print is_a($obj, "Hello2") ? "Yes¥n": "No¥n"; // Noを表示
 print is_a($obj2, "Hello") ? "Yes¥n": "No¥n"; // Yesを表示
?>
```

■bool is_subclass_of(object obj, string class_name)

オブジェクトがclass_nameのサブクラスの場合にTRUE、それ以外の場合にFALSEを返します。

■bool method_exists(object obj, string method_name)

指定したメソッドが定義されている場合にTRUE、それ以外の場合にFALSEを返します。

Hypertext Preprocessor

PHP関数

Chapter-3 ファイル関連

3.1 ファイルシステム関数

PHPでは、ファイルの読み書き、作成といったファイルに関する処理が可能です。

▶php.ini設定オプション

パラメータ	デフォルト値	説明
allow_url_fopen	On	fopen()のファイル名でURLを使用可能とする
from	anonymous	fopen()でftpサーバに匿名アクセスする際のパスワード
include_path	NULL	ファイルを読み込むパス

▶注意

サンプルコードで用いるtest.txtの内容として以下を仮定します。

taro,1980,tokyo
hanako,1975,osaka

■string basename(string path)

パス名からファイル名を生成します。パス区切り文字は /（Microsoft Windowsでは¥も使用可）です。

```
<?php
 print basename("/home/httpd/html/index.html"); // index.htmlを出力します。
?>
```

■bool chgrp(string filename, mixed group)

F T

ファイルのグループをgroupに変更します。グループ変更の権限が必要です。

■bool chmod(string filename, int mode)

F T

ファイルのモードをmodeに変更します。modeは先頭に0をつけて8進数で指定します。

```
<?php
 print chmod("test.txt", 0666) ? "成功":"失敗";
?>
```

■int chown(string filename, mixed user)

ファイルfilenameの所有者をユーザuserに変更します。ユーザ変更の権限が必要です。

■void clearstatcache(void)

システムコールstat()、lstat()により得られるファイル情報のキャッシュをクリアします。

■bool copy(string src, string dest)

F T

srcからdestにファイルをコピーします。

■string dirname(string path)

パス名からディレクトリの名前を返します。

```
<?php
 print dirname("/home/httpd/html/index.html"); // index.html を出力します。
?>
```

■float diskfreespace(string dir)

指定したディレクトリで利用可能なバイト数を返します。

■bool fclose(resource fp)

T

ファイルを閉じます。

■bool feof(resource fp)

ファイルポインタがファイルの終端に達していればTRUEを返します。

■bool fflush(resource fp)

F T

出力をフラッシュします。

■string fgetc(resource fp)

ファイルから1文字読み出し、文字列として返します。

```
<?php
 $fp = fopen("test.txt","r");
 while(feof($fp) !== TRUE){ // ファイルの内容を読み出す
   print fgetc($fp);
 }
 fclose($fp);
?>
```

■array fgetcsv(resource fp,int len [, string delim])

CSVファイルから1行取得し、各レコードと要素とする配列として返します。ファイルの終端の場合にはfalseを返します。行の最大長をlenで指定します。delimによりデリミタを指定可能（デフォルトhは , ）です。

```
<?php
 $fp = fopen("test.txt","r") or die("test.txt couldn't open.");
 $a = fgetcsv($fp, 255);
 print_r($a); // $aはarray("taro",1980,"tokyo") となる
 fclose($fp);
?>
```

■string fgets(resource fp, int len)

ファイルから最大len-1バイト読み込み、文字列として返します。読み込みは、改行文字またはEOFを検出したか、len-1バイトを読み出した時点で終了します。

```
<?php
 $fp = fopen("test.txt","r");
 while($str = fgets($fp, 255)){
   print $str;
 }
 fclose($fp);
?>
```

■string fgetss(resource fp, int length [, string tags])

fgets()と機能は同じですが、HTMLおよびPHPのタグを取り除きます。削除したくないタグをtagsで指定可能です。

■array file(string filename [, int use_inc])

readfile()と機能は同じですが、各行を配列の要素として返します。*1 *2

■bool file_exists(string filename)

ファイルfilenameが存在する場合にTRUEを返します。

■bool flock(resource fp, int op)

F T

ファイルのロックを設定または解除します。opには**表3-17**の値を指定します。

表3-17 flock()関数のopの設定

設定値（定数）	説明
LOCK_SHまたは1	共有ロック　：読み込み時
LOCK_EXまたは2	排他的ロック：書き込み時
LOCK_UNまたは3	ロック解除

非ブロックモードを指定するにはopにLOCK_NBまたは4を加えます。

*1
use_incに1を指定することにより、include_pathで指定したパスの検索も行なわれます。

*2
ファイル名がhttp://で始まっている場合はHTTP、ftp://で始まっている場合はFTPの接続がオープンされます（php.iniのディレクティブ allow_url_fopen = Onの場合）。

```
<?php
 $cnt = file("count.txt");
 $cnt[0]++; // カウントアップ
 print $cnt[0]; // カウンタの値を出力

 $fp = fopen("count.txt","w") or die("count.txt couldn't open.");
 if (flock($fp, LOCK_EX)) {
   fwrite($fp, $cnt[0]); // カウンタの値を書き込む
   flock($fp, LOCK_UN);
 }
 fclose($fp);
?>
```

■resource fopen(string filename, string mode [, int use_inc])

F

ファイル（またはURL）をモードmodeでオープンし、ポインタを返します。*1 *2

modeには、r（読み込み）、r+（読み書き）、w（書き込み）、w+（読み書き）、a（追記）、a+（読み込みおよび追記）、b（バイナリモード、Windowsのみ）を指定します。

■int fpassthru(resource fp)

ファイルからデータをEOFまで読み込み、標準出力に書き出します。ファイルは読み込み終了時に自動的にクローズされます。出力したバイト数が返されます。

```
<?php
 $fp = fopen("test.txt","r");
 $size = fpassthru($fp); // ファイルの内容を出力
 print "¥n出力バイト数: $size¥n";
?>
```

■string fread(resource fp, int len)

ファイルから最高lenバイト読み込み、文字列として返します。読み込みはlenバイト読み込まれたか、EOFに達した時点で終了します。

```
<?php
 $fp = fopen("test.txt", "r");
 print fread($fp, 4096); // ファイルの内容を出力
 fclose($fp);
?>
```

■int fseek(resource fp, int offset [,int param])

ファイルポインタの位置をoffsetに設定します。成功時に0、それ以外の場合に-1を返します。paramには以下の値を指定可能です。

SEEK_SET ：位置をoffsetバイト目に設定する（デフォルト）
SEEK_CUR：現在の位置にoffsetを加えた位置に設定する
SEEK_END：ファイル終端にoffsetを加えた位置に設定する

■array fstat(resource fp)

ファイルの情報（**表3-2参照**）を配列として取得します。

```
<?php
 $fp = fopen("fstat.php", "r") or die("file can't open.");
 $st = fstat($fp); print_r($st); // ファイル情報を出力
 fclose($fp);
?>
```

■int ftell(resource fp)

ファイルポインタポインタの位置を返します。

■int ftruncate(resource fp, int size)

ファイルを指定した長さに丸めます。

■int fwrite(resource fp, string string [, int len]) ［エイリアス］fputs()

文字列をファイルに書き込みます。文字列の書き込みの完了時に処理が終了します。
lenを指定した場合、最大lenバイトまで書き込まれ、スラッシュ文字の削除は行なわれません。

■int set_file_buffer(resource fp, int buf)

ファイル書き込み用バッファをbufバイトに設定します。成功時に0、エラー時にEOFを返します。bufに0を指定した場合にはバッファリングされません。

■bool link(string file, string link)

F T

ファイルへのハードリンクlinkを作成します。

■bool mkdir(string path, int mode)

ディレクトリpathをモードmodeで作成します。引数modeにより（0を先頭につけた8進数で）アクセス権を指定可能です。

■int pclose(resource fp)

パイプを閉じます。パイプにより実行中のプロセスの終了ステータスを返します。

■resource popen(string command, string mode)

コマンドを実行し、そのプロセスへのパイプをオープン、ファイルポインタを返します。modeには、r（読み込み）またはw（書き込み）を指定します。

```
<?php
 $fp = popen("/bin/ls", "r");
 print fread($fp, 4096);
 pclose($fp);
?>
```

■int readfile(string filename)

ファイル（またはURL）の内容を標準出力に書き出し、読み込んだバイト数を返します。URLに関する処理はfopen()と同じです。

```
<?php
 $size = readfile("test.txt");
 print "¥n¥n読みこんだバイト数:  ".$size."¥n";
?>
```

■string readlink(string path)

指定したパスのシンボリックリンク先を返します。

```
<?php
 if (symlink("test.txt", "test1.txt") === TRUE){ // シンボリックリンクを作成
   print readlink("test1.txt"); // リンク先(test.txt)を出力
 }
?>
```

■bool rename(string oldname, string newname)

F T

oldnameをnewnameにリネームします。

■bool rewind(resource fp)

T

ファイルポインタの位置をファイルの先頭にセットします。

■bool rmdir(string dir)

F T

ディレクトリを削除します。削除できるのは、PHP実行ユーザにより削除可能な空のディレクトリのみです。

■array stat(string filename)

ファイルの情報をstatシステムコールにより取得し、stat構造体の値を**表3-18**に示す配列として返します。

また、個々のファイルの情報を取得するには**表3-19**に示す関数、個々のファイルの属性を確認するには**表3-20**に示す関数を使用します。

```
<?php
 $a = stat("test.txt");
 print "サイズ: ".$a['size']."¥n";
 print "最終更新時間: ".date("Y/m/d H:i:s", $a['mtime'])."¥n";
?>
```

表3-18 stat構造体の値(stat()、fstat()関数の返り値)

要素	名前	要素	名前
0	デバイス(dev)	7	サイズ(バイト数)(size)
1	iノード(ino)	8	最終アクセス時刻(atime)
2	iノードのモード(mode)	9	最終更新時刻(mtime)
3	リンク数(nlink)	10	最終変更時刻(ctime)
4	所有者のユーザID(uid)	11	ファイルI/Oのブロックサイズ*3(blksize)
5	所有者のグループID(gid)	12	割り当て済みブロック数*4(blocks)
6	デバイスタイプ(rdev)		

*3 st_blksize型をサポートするシステムでのみ有効(それ以外は-1を返す)

*4 st_blocks型をサポートするシステムでのみ有効(それ以外は-1を返す)

*5
エラー時はFALSEを返します。

*6
返される型は、fifo、char、dir、block、link、file、unknownのいずれかです。

表3-19 ファイル情報を取得する関数*5

関数		返り値
int	fileatime(string filename)	最終アクセス時刻
int	filectime(string filename)	最終更新時刻
int	filegroup(string filename)	所有者のグループID
int	fileinode(string filename)	i-node番号
int	filemtime(string filename)	最終更新時刻
int	fileowner(string filename)	所有者のユーザID
int	fileperms(string filename)	許可属性情報
int	filesize(string filename)	サイズ
string	filetype(string filename)	型*6
int	linkinfo(string filename)	リンク情報

表3-20 ファイル属性確認用関数

関数		返り値
bool	is_dir(string filename)	ディレクトリの場合にTRUE
bool	is_executable(string filename)	実行可能な場合にTRUE
bool	is_file(string filename)	通常ファイルの場合にTRUE
bool	is_link(string filename)	シンボリックリンクの場合にTRUE
bool	is_writeable(string filename)	書き込み可能な場合にTRUE
bool	is_readable(string filename)	読み込み可能な場合にTRUE

■array lstat(string filename)

ファイルの情報を得ます。動作はstat()と同じですが、シンボリックリンクの場合はリンク元の情報を返します。

■bool symlink(string filetarget, string link)

F T

ファイルへのシンボリックリンクlinkを作成します。

■string tempnam(string dir, string prefix)

F

prefixを先頭に付加したユニークなテンポラリファイル名をディレクトリdir（存在しない場合はシステムのテンポラリディレクトリ）以下に生成し、パス名を返します。

```
<?php
 $tempname = tempnam("/tmp", "foo"); // テンポラリファイル名を作成
 print $tempname;
 $fp = fopen($tempname, "w");
 fwrite($fp, "一事ファイルへの書き込み");
 fclose($fp);
 unlink($tempname); // テンポラリファイルを削除
?>
```

resource tmpfile()

F

テンポラリファイルを書き込みモードでオープンし、ファイルポインタを返します。テンポラリファイルは、fclose()コール時またはスクリプト実行終了時に削除されます。

```
<?php
 $fp = tmpfile() or die("failure in tmpfile()");
 $size = fwrite($fp, "テンポラリファイルに書き込み");
 print "$size バイト書き込みました。";
 fclose($fp); // ファイルを削除
?>
```

bool touch(string filename [, int time [, int atime]])

ファイルの最終更新時および最終アクセス時をtime（省略時は現在時刻）およびatimeに変更します。指定ファイルが存在しない場合には新規にファイルが作成されます。

```
<?php
 $time =  mktime(0,0,0,9,1,2002); // 2002/09/01 00:00:00を設定
 $file = "test.txt";
 touch($file, $time-3600, $time); // 最終修正/アクセス時刻を設定
 print date("Y/m/d H:i:s",fileatime($file))."¥n"; // 最終アクセス時刻を出力
 print date("Y/m/d H:i:s",filemtime($file))."¥n"; // 最終修正時刻を出力
?>
```

int umask([int mask])

umaskをmask & 0777に設定し、元のumask値を返します。maskを省略した場合は現在のumask値を返します。

bool unlink(string filename)

F T

ファイルを削除します。

3.2 ディレクトリ関数

■bool chdir(string dir)

F T

カレントディレクトリをdirに変更します。

■bool chroot(string dir)

F T

ルートディレクトリをdirに変更します。

■void closedir(resource dir_handle)

ディレクトリを閉じます。*7

■new dir(string dir)

ディレクトリをオープンし、オブジェクトインスタンスを生成します。プロパティhandle（ディレクトリへのポインタ）、path（ディレクトリへのパス）およびメソッドread、rewind、closeが使用可能です。

```
<?php
 $dir = dir("."); // カレントディレクトリをオープン
 print "パス: ". $dir->path."¥n";
 $dir->rewind();
 while($file = $dir->read()){ // ファイルリストを読み込んで表示
   print $file."¥n";
 }
 $dir->close();
?>
```

■resource opendir(string path)

F

ディレクトリをオープンし、ハンドルを返します。*7

*7 readdir()のサンプルを参照してください。

■string readdir(resource dir_handle)

F

ディレクトリから次のファイルのファイル名を返します。返されるファイル名の順序は不定です。

```
<?php
 $dir = opendir(".") or die("dir open failed."); // カレントディレクトリをオープン
 rewinddir($dir); // ディレクトリストリームを先頭にリセット
 while($file = readdir($dir)){ // ファイルリストを読み込んで表示
   print $file."¥n";
 }
 closedir($dir);
?>
```

■void rewinddir(resource dir_handle)

ディレクトリのストリームをディレクトリの先頭にリセットします。[*7]

3.3 イメージ関数

PHPでは、JPEG、GIF、PNG、SWF（Shockwave Flash Format）イメージの大きさを調べる関数がサポートされています。GDライブラリ（http://www.boutell.com/gd/）によりPNG、JPEG、WBMP、XBM型式などのイメージの作成と操作を行なったり、FreeTypeライブラリによりTrueTypeフォントを描画することができます。

日本語文字コードに対応したGDを使うか、または文字コードUTF-8（UNICODE）で文字を指定すると日本語TrueTypeフォントを表示することが可能になります。

▶PHP構築オプション

--with-gd[=DIR]	GDによるイメージ関数を有効にします。 （例： --with-gd=/usr/local）
--with-jpeg-dir=DIR	libjpegのインストール先を指定します。 （例： --with-jpeg-dir=/usr/local）
--with-png-dir=DIR	libpngのインストール先を指定します。 （例： --with-png-dir=/usr/local）
--with-xpm-dir=DIR	libXpmのインストール先を指定します。 （例： --with-xpm-dir=/usr/local）
--with-zlib-dir=DIR	zlibのインストール先を指定します。 （例： --with-zlib-dir=/usr/local）

--with-ttf[=DIR]	FreeType 1.xのサポートを有効にします。（例：--with-ttf=/usr/local）
--with-freetype-dir=DIR	FreeType2のインストール先を指定します。（例：--with-freetype-dir=/usr/local）
--with-t1lib[=DIR]	T1libのサポートを有効にします。（例：--with-t1lib=/usr/local）
--enable-gd-native-ttf	GDライブラリのTrueType文字列描画関数を有効にします。
--enable-exif	Exifサポート関数を有効にします。

例： **./configure --with-gd ---with-ttf --enable-gd-native-ttf ¥**
--enable-exif [そのほかのオプション**]**

▶ 参考URL

イメージライブラリGDの開発・配布元：**http://www.boutell.com/gd/**
libpngライブラリの開発・配布元：**http://www.libpng.org/pub/png/libpng.html**
Independent JPEG Group（libjpegの開発・配布元）：**http://www.ijg.org/**
FreeTypeの開発・配布元：**http://freetype.sourceforge.net/**

▶ 注意

・イメージ関数における描画の座標系は、左上が(0,0)と定義されています。
・ライセンス上の制約によりGIFの作成はサポートされていません。
・本稿執筆時点でイメージライブラリGDの最新版はgd-1.8.4ですが、αチャネルなどをサポートするgd-2.0.1も公開されています。gd-2.0.1はβ版であり、問題点も見つかっています。より安定して動作するまではgd-1.8.3またはgd-1.8.4を使用したほうがよいでしょう。

■array exif_read_data(string filename [, string sections [, bool thumbnail]])

F

JPEGファイルからExifヘッダを読み込み、連想配列として返します。sectionsに取得するタグの種類を指定することが可能です（**表3-21**）。thumbnailにTRUEを設定した場合、サムネイルのデータを読み込みます。

```
<?php
 $d = exif_read_data("demo0.jpg","EXIF"); // Exifデータを読み込みます。
 print_r($d);
?>
```

表3-21 オプションsectionsの設定値

文字列	返り値
FILE	FileName、FileSize、FileDateTime、SectionsFound（デフォルト）
COMPUTED	html、Width、Height、IsColorなど
ANY_TAG	IFD0、EXIDなどのタグを有する情報
IFD0	IFD0のタグつきデータ
THUMBNAIL	サムネイルに関する情報
COMMENT	JPEGイメージのコメントヘッダ
APP0	JPEGイメージのAPP0ヘッダ
EXIF	IFD0セクションのEXIFサブセクション
FPIX	FPIXセクション
GPS	GPSセクション
INTEROP	INTEROPセクション
APP12	APP12セクション

■string exif_thumbnail(string filename)

F

JPEGファイルからサムネイルを読み込みます。

```
<?php
 header("Content-type: image/jpeg");
 print exif_thumbnail("demo0.jpg"); // サムネイルを表示
?>
```

■array getimagesize(string filename [, array info])

F

画像ファイル（GIF、JPG、PNG、SWF、PSD、BMP、TIFFファイル）filenameの大きさに関する情報を取得し、配列として返します。返される配列$arrayは**表3-22**の要素からなります。infoパラメータを指定した場合、画像ファイルに関する情報（内容はイメージ形式に依存）が代入されます。

表3-22 getimagesize()から返される配列の要素

添字	説明
0	イメージの幅（単位：ピクセル）
1	イメージの高さ（単位：ピクセル）
2	イメージの種類 *8
3	HTML imgタグ型式の文字列（"height=xxx width=xxx"）
bits	ビット/ピクセル（SWFまたはJPEGのみ有効）
channels	チャンネル数（JPEGのみ有効）

*8
1:GIF、2:JPEG、3:PNG、4:SWF、5:PSD、6:BMP、7:TIFF（リトルインディアン）、8:TIFF（ビッグインディアン）

```
<?php
 $img = "demo0.jpg";
 $size = getimagesize($img);
 print_r($size); // サイズに関する情報を出力
 print "<img src=¥"$img¥" {$size[3]}>"; // HTMLイメージタグを出力
?>
```

int imagearc(resource im, int cx, int cy, int w, int h, int s, int e, int color)

座標(cx,cy)を中心とする部分楕円を描画します。引数には楕円の幅：w、高さ：h、始点角度：s、終点角度：eを指定します。*9

int imagechar(resource im, resource font, int x, int y, string c, int color)

座標 (x,y) に文字列cの最初の文字を描画します。引数にはカラーID (color) およびフォントID (font) を指定します。

フォントIDには、1～5 (組み込みフォント) またはimageloadfont()関数で取得したフォントIDを指定します。

*9 imagecreate()のサンプルを参照してください。

```
<?php
 header("Content-type: image/png");
 $im = imagecreate(100, 50); imagecolorallocate($im, 255, 255, 255);
 $blue = imagecolorallocate($im, 0, 0, 255);
 imagechar($im, 1, 40, 25, "P", $blue);
 imagechar($im, 2, 50, 25, "H", $blue);
 imagecharup($im, 3, 60, 25, "P", $blue);
 imagepng($im); // PNGイメージを出力
?>
```

int imagecharup(resource im, resource font, int x, int y, string c, int color)

文字を垂直に描画すること以外はimagechar()と同じです。

int imagecolorallocate(resource im, int red, int green, int blue)

指定したRGBを示すカラーIDを返します。最初に作成した色は背景色となります。

bool imagecolordeallocate(resource im, int color)

F T

指定したカラーIDを有する色を削除します。

■int imagecolorat(resource im, int x, int y)

座標 (x,y) 位置のピクセルのカラーIDを返します。

```
<?php
 $im = imagecreatefrompng("test1.png"); // PNGイメージファイルを読み込む
 $cid = imagecolorat($im, 50,170); // (50,170) のピクセルのカラーIDを取得
 $rgb = imagecolorsforindex($im, $cid); // カラーのRGB値を取得
 print_r($rgb); // array("red"=>255, "green"=>0, "blue"=>0)を出力
?>
```

■int imagecolorclosest(resource im, int red, int green, int blue)

指定したRGB値に最も近い画像パレット中の色のIDを取得します。

```
<?php
 $im = imagecreatefrompng("test1.png");    // PNGイメージファイルを読み込む

 $cid1 = imagecolorexact($im, 200, 0, 0); // RGB値（200,0,0）と等しいカラーIDを取得
 if ($cid1 == -1) {
  print "カラーが存在しません¥n"; // カラーのRGB値を取得
 } else {
  print_r(imagecolorsforindex($im, $cid1)); // カラーのRGB値を出力
 }
 $cid2 = imagecolorclosest($im, 200, 0, 0); // RGB値（200,0,0）に近いカラーIDを取得
 print_r(imagecolorsforindex($im, $cid2));  // カラーのRGB値（255,0,0）を出力

 $cid3 = imagecolorresolve($im, 200, 0, 0); // RGB値（200,0,0）に近いカラーIDを取得
                                            //  （ない場合は作成）
 print_r(imagecolorsforindex($im, $cid3));  // カラーのRGB値 (200,0,0)を出力
?>
```

■int imagecolorexact(resource im, int red, int green, int blue)

指定したRGB値に一致する色のID（存在しない場合は-1）を返します。*10

■int imagecolorresolve(resource im, int red, int green, int blue)

指定したRGB値の色または最も近い色のIDを返します。*10

*10
imagecolorclosest()のサンプルを参照。

■bool imagecolorset(resource im, int color, int red, int green, int blue)

カラーID（color）を指定したRGB値に設定します。この関数により塗りつぶしの操作を行なわずに同じ効果を得ることができます。

```
<?php
 $im = imagecreatefrompng("test1.png"); // PNGイメージファイルを読み込む
 $color = imagecolorat($im,50,170);  // (50,170)のピクセルのカラーIDを取得
 imagecolorset($im, $color, 127, 127, 0); // 赤色を茶色に変更
 imagepng($im, "test1a.png"); // PNGイメージをファイルに出力
?>
```

■array imagecolorsforindex(resource im, int color)

カラーIDを表すred、green、blueをキーとする連想配列を返します。[*11]

■int imagecolorstotal(resource im)

イメージパレットの色数を返します。

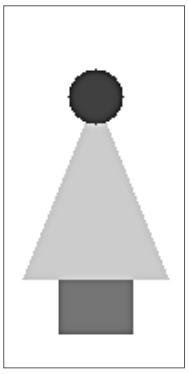
```
<?php
 $im = imagecreatefrompng("test1.png"); // PNGイメージファイルを読み込む
 print imagecolorstotal($im); // イメージパレットの色数（4）を出力
?>
```

■int imagecolortransparent(resource im [, int color])

透明色をcolorに設定します。指定した（指定しない場合は現在の）透明色のIDが返されます。

```
<?php
 $im = imagecreatefrompng("test1.png"); // PNGイメージファイルを読み込む
 $cid = imagecolorat($im, 10,10); // 背景色のカラーIDを取得
 imagecolortransparent($im, $cid); // 背景色を透過色に設定
 imagepng($im, "test2.png"); // PNGイメージをファイルに出力
?>
```

図3-1 imagecolortransparent()のサンプル出力



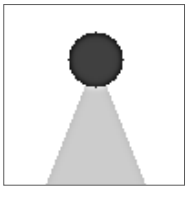
*11
imagecolorat()のサンプルを参照してください。

■int imagecopy(resource dst_im, resource src_im, int dx, int dy, int sx, int sy, int sw,int sh)

イメージ(src_im)の (sx,sy) を基準として幅sw、高さshの領域をイメージ(dst_im)の(dx,dy) を基準とし、同じ大きさの領域にコピーします。

```
<?php
 $src_im = imagecreatefrompng("test1.png"); // PNGイメージファイルを読み込む
 $dst_im = imagecreate(100,100); // PNGイメージファイルを作成
 imagecopy($dst_im, $src_im, 0, 0, 0, 20, 100, 100);
 imagepng($dst_im, "test3.png"); // PNGイメージをファイルに出力
?>
```

図3-2 imagecopy()のサンプル出力

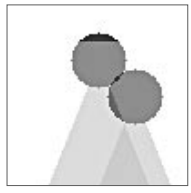


■int imagecopymerge(resource dst_im, resource src_im, int dx, int dy, int sx, int sy, int sw,int sh, int pct)

イメージ(src_im)の (sx,sy) を基準として幅sw、高さshの領域をイメージ(dst_im)の(dx,dy) を基準とした領域にコピーします。pctにイメージのマージ割合(0～100)を指定します。0の場合はコピーされず、100の場合はimagecopy()と同様の動作となります。

```
<?php
 $src_im = imagecreatefrompng("test1.png"); // PNGイメージファイルを読み込む
 $dst_im = imagecreatefrompng("test3.png"); // PNGイメージファイルを作成
 imagecopymerge($dst_im, $src_im, 20, 20, 0, 20, 100, 100, 50); // 50%ずつでマージ
 imagepng($dst_im, "test4.png"); // PNGイメージをファイルに出力
?>
```

図3-3 imagecopymerge()サンプル出力



■int imagecopymergegray(resource dst_im, resource src_im, int dx, int dy, int sx, int sy, int sw,int sh, int pct)

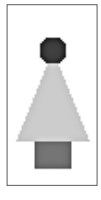
imagecopymerge()と動作は同じですが、マージ前にコピー先のピクセルをグレースケール変換します。これにより、コピーするイメージの色相が維持されます。

■int imagecopyresized(resource dst_im, resource src_im, int dx, int dy, int sx, int sy, int dw, int dh, int sw, int sh)

イメージ（src_im）の (sx,sy) を基準として幅sw、高さshの領域をイメージ（dst_im）の (dx,dy) を基準として幅dw、高さdhの領域にコピーします。

```
<?php
 $src = imagecreatefrompng("test1.png"); // PNGイメージファイルを読み込む
 $sx = imagesx($src); $sy = imagesy($src); // 幅と高さを取得
 $dst = imagecreate($sx/2,$sy/2); // PNGイメージファイルを作成
 imagecopyresized($dst, $src, 0, 0, 0, 0, $sx/2, $sy/2, $sx, $sy); // 1/2縮小コピー
 imagepng($dst, "test5.png"); // PNGイメージをファイルに出力（図3-4）
?>
```

図3-4 imagecopyresized()のサンプル出力

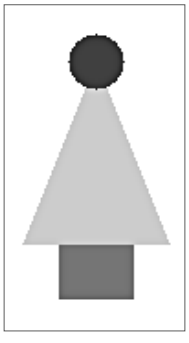


resource imagecreate(int w, int h)

幅w、高さhの空のイメージを作成し、イメージIDを返します。

```
<?php
 $im = imagecreate(100, 200); // 空のイメージを作成
 $white = imagecolorallocate($im, 255, 255, 255); // 最初に作成した色は背景色となる
 $red = imagecolorallocate($im, 255, 0, 0);
 $green = imagecolorallocate($im, 0, 255, 0);
 $blue = imagecolorallocate($im, 0, 0, 255);
 imagerectangle($im, 30, 180, 70, 150, $red); // (50,50),(150,100) を隅とする矩形を描画
 imagefilledpolygon($im, array(10,150,90,150,50,50),3,$green);
 imagearc($im, 50, 50, 30, 30, 0, 360, $blue); // (100,100) を中心とした円を描画
 imagefilltoborder($im, 50, 40, $blue, $blue); // (50,30) を含む領域を青で塗りつぶす
 imagefill($im, 50, 170, $red); // (50,30) を含む領域を青で塗りつぶす
 imagepng($im, "test1.png"); // PNGイメージをファイルに出力（図3-5）
?>
```

図3-5 imagecreate()および描画関数により作成したサンプルイメージ



resource imagecreatefromgd2part(string filename, int sx, int sy, int width, int height)

GD2イメージファイル（またはURL）の(sx, sy)を基準とした幅width、高さheightの領域から新規イメージを作成し、イメージIDを返します。

resource imagecreatefrompng(string filename)

PNG型式の既存のイメージファイル（またはURL）から新規イメージを作成し、イメージIDを返します。既存のイメージファイルから新規イメージを作成する関数として、**表3-23**に示す関数を使用できます。

```
<?php
 $im =  imagecreatefrompng("test1.png"); // 既存のPNGイメージを読み込む
 header("Content-type: image/png");
 imagepng($im); // PNGイメージを出力
?>
```

表3-23　既存のイメージファイルから新規イメージを作成する関数

関数名	サポートするイメージ型式
resource imagecreatefromgd(string filename)	GD
resource imagecreatefromgd2(string filename)	GD2
resource imagecreatefromjpeg(string filename)	JPEG
resource imagecreatefrompng(string filename)	PNG
resource imagecreatefromwbmp(string filename)	Windows Bitmap (bmp)
resource imagecreatefromxbm(string filename)	X Bitmap (xbm)

■resource imagecreatefromstring(string image)

文字列データとして格納したイメージデータから新規イメージを作成し、イメージIDを返します。

```
<?php
 $fd = fopen("test1.png", "rb"); // 既存のイメージファイルを読み込む
 $image = fread($fd, filesize("test1.png"));
 fclose($fd);

 $im =  imagecreatefromstring($image); // イメージデータからイメージを作成
 header("Content-type: image/png");
 imagepng($im); // PNGイメージを出力
?>
```

■bool imagedestroy(resource im)

T

イメージを保持するメモリを解放します。

■bool imagefill(resource im, int x, int y, int color)

T

座標(x,y)を含む領域を色colorで塗りつぶします。*9

■bool imagefilledpolygon(resource im, array points, int num, int color)

多角形を描画し、色colorで塗りつぶします。多角形の頂点は、配列pointsによりpoints[0] = x0、points[1] = y0、points[2] = x1、points[3] = y1のように指定します。numは頂点の総数です。*9

■bool imagefilledrectangle(resource im, int x1, int y1, int x2, int y2, int color)

左上の座標 (x1,y1)、右下の座 標 (x2,y2) で指定した領域に矩形を描画し、色colorで塗りつぶします。

■bool imagefilltoborder(resource im, int x, int y, int bid, int color)

座標 (x,y) を含む領域をbidを境界色として色colorで塗りつぶします。*9

■int imagefontheight(resource font)

フォントfontの高さをピクセル単位で返します。

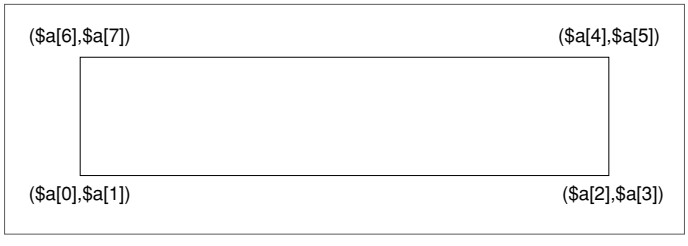
■int imagefontwidth(resource font)

フォントfontの幅をピクセル単位で返します。

■array imageftbbox(int size, int angle, string fontfile, string text [, array extrainfo])

FreeType2によりTrueTypeフォントでテキスト文字列を描画した場合のバウンディングボックスの大きさを配列（**図3-6参照**）として返します。*12

図3-6 配列要素とバウンディングボックスの関係



*12 この関数を使用するには、FreeType2ライブラリが必要です。

■array imagefttext(resource im, int size, int angle, int x, int y, int col, string fontfile, string text [, array extrainfo])

指定したフォントを用いてFreeType2によりテキストを描画します。*12 バウンディングボックスの大きさを示す配列（**図3-6参照**）を返します。

■int imagegammacorrect(resource im, float ig, float og)

入力γ値（ig）および出力γ値（og）に基づきγ補正を行ないます。

■int imageinterlace(int im [, int interlace])

インタレースビットinterlace（1：有効、0：無効）を設定し、現在のインタレースビットを返します。

■bool imageline(resource im, int x1, int y1, int x2, int y2, int color)

T

座標 (x1,y1) から (x2,y2) まで色colorで直線を描画します。

■int imageloadfont(string file)

F

ビットマップフォントをロードし、フォントIDを返します。

■int imagepalettecopy(resource dst_im, resource src_im)

イメージsrc_imからイメージdst_imにパレットの内容をコピーします。

■bool imagepng(resource im [, string filename])

F T

PNGイメージをブラウザ（またはファイルfilename）へ出力します。*9 *13
イメージデータを出力する関数として、**表3-24**に示す関数を使用できます。

*13 ブラウザに出力する場合、header()関数で適切なContent-Typeヘッダを送信しておく必要があります。

*14 JPEGイメージの品質(0～100)をqualityに指定することが可能です。0は最も圧縮率が高く、100は最も高品質となります。

*15 フォアグラウンド色をfgcolorに指定できます。

*16 imagepstext()のサンプルを参照してください。

表3-24 イメージデータを出力する関数

関数名	サポートするイメージ型式
bool imagegd(resource im [, string filename])	GD
bool imagegd2(resource im [, string filename])	GD2
bool imagejpeg(resource im [, string filename [, int quality]]) *14	JPEG
bool imagepng(resource im [, string filename])	PNG
bool imagewbmp(int im [, string filename [, int fgcolor]]) *16	Windows Bitmap

■bool imagepolygon(resource im, array points, int num, int color)

F T

color色で多角形を描画します。配列pointsは多角形の頂点で、points[0] = x0、points[1] = y0、points[2] = x1、points[3] = y1 …… のように指定します。numは頂点の総数です。

■array imagepsbbox(string text, resource font, int size [, int space [, int tight [, float angle]]])

F

Type1フォントを用いてテキスト文字列を描画した場合のバウンディングボックス（**図3-6参照**）を取得します。

■bool imagepsencodefont(resource font, string encodingfile)

F T

Type1フォントの文字エンコーディングベクトルを変更します。

■bool imagepsextendfont(resource font, float extend)]

F T

Type1フォントを圧縮または展開します。extendの値が1未満の場合、フォントの圧縮が行なわれます。

■bool imagepsfreefont(resource font)

T

ロード済みのType1フォント用メモリを解放します。

■resource imagepsloadfont(string filename)

F

Type1フォントをロードします。*16

■bool imagepsslantfont(resource font, float slant)

F

傾きを指定してスラントフォントを作成します。

■array imagepstext(resource im, string text, resource font, int size, int fg_color, int bg_color, int x, int y [, int space, int tight, float angle, int antialias])

F

Type1フォントにより文字列を描きます。**表3-25**に示すパラメータを引数とし、バウンディングボックスの大きさを示す配列（**図3-6参照**）を返します。

表3-25 imagepstext()の引数

引数	説明
im	イメージのリソース
text	描画するテキスト
font	Type1 フォントのリソース
size	大きさ（単位：ピクセル）
bg_color	背景色のカラーID
fg_color	描画色のカラーID
(x,y)	先頭文字の原点（ほぼ左下隅）の座標
antialias	アンチエイリアスで使用する色数（デフォルト：4）
angle	テキストの角度
space	フォントの空白のデフォルト値
tight	文字間の空白の量

```
<?php
 $im = imagecreate(350, 45);
 $black = imagecolorallocate($im, 0, 0, 0);
 $white = imagecolorallocate($im, 255, 255, 255);
 $t1font = "/usr/lib/X11/fonts/Type1/c0419bt_.pfb"; // 表示するType1フォント
 $font = imagepsloadfont($t1font);                  // Type1フォントをロード
 imagepstext($im, "PHP is Great!", $font, 32, $white, $black, 32, 32);
                                                    // テキスト描画
 imagejpeg($im, "test6.jpg", 70); // JPEGファイルとして出力（図3-7）
?>
```

図3-7 imagepstext()のサンプル出力

bool imagerectangle(resource im, int x1, int y1, int x2, int y2, int color)

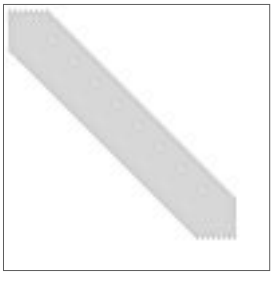
左上の座標 (x1,y1)、右下の座標 (x2,y2) を指定した領域に色colorの矩形を生成します。

bool imagesetbrush(resource im, resource brush)

線を描画する際のカスタムブラシを設定します。カスタムブラシを使用するには、線を描画する際にカラーIDとしてIMG_COLOR_BRUSHEDを指定します。

```
<?php
 $brush = imagecreatefrompng("brush1.png");
 $im = imagecreate(100,100); // 新規イメージを作成
 $w = imagecolorallocate($im, 255, 255, 255);
 imagesetbrush($im, $brush);
 imageline($im, 10, 10, 80,80, IMG_COLOR_BRUSHED); // 塗りつぶしを設定
 imagepng($im, "test7.png"); // PNGファイルとして出力
?>
```

図3-8 imagesetbrush()のサンプル出力



bool imagesetpixel(resource im, int x, int y, int color)

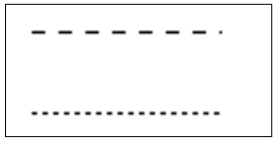
座標 (x,y) に色colorで点を描画します。

bool imagesetstyle(resource im, array style)

線のカスタムパターンを設定します。カスタムパターンを使用するには、線を描画する際にカラーIDとしてIMG_COLOR_STYLEDを指定します。

```
<?php
 $im = imagecreate(100,50); // 新規イメージを作成
 $w = imagecolorallocate($im, 255, 255, 255);
 $b = imagecolorallocate($im, 0, 0, 255);
 $style1 = array($b,$b,$b,$b,$b,$w,$w,$w,$w,$w);
 imagesetstyle($im, $style1); // 波線を設定
 imageline($im, 10, 10, 80,10, IMG_COLOR_STYLED);
 $style2 = array($b,$b,$w,$w);
 imagesetstyle($im, $style2); // 波線を設定
 imageline($im, 10, 40, 80,40, IMG_COLOR_STYLED);
 imagepng($im, "test8.png"); // PNGファイルとして出力
?>
```

図3-9 imagesetstyle()のサンプル出力



■bool imagesettile(resource im, resource tile)

T

塗りつぶし用のタイルイメージを設定します。塗りつぶし色としてIMG_COLOR_TILEDを指定します。

```
<?php
 $tile = imagecreatefrompng("php-logo.png"); // タイルイメージを取得
 $im = imagecreate(200,200); // 新規イメージを作成
 imagesettile($im,$tile); // 塗りつぶし用タイルを設定
 imagefill($im, 10, 10, IMG_COLOR_TILED); // 塗りつぶしを設定
 imagepng($im, "test9.png"); // PNGファイルとして出力
?>
```

図3-10 imagesettile()のサンプル出力

■bool imagestring(resource im, resource font, int x, int y, string s, int color)

T

座標 (x,y) から文字列sを色colorで描画します。

■bool imagestringup(resource im, resource font, int x, int y, string s, int color)

T

座標 (x,y) から垂直に文字列sを色colorで描画します。

■int imagesx(resource im)

イメージの幅を返します。*17

■int imagesy(resource im)

イメージの高さを返します。*17

■array imagettfbbox(int size, int angle, string fontfile, string text)

F

TrueTypeフォントを用いてテキスト文字列を描画した際のバウンディングボックスの大きさを配列（**図3-6参照**）として返します。*18

```
<?php
 $str = "日本語表示"; // 表示する文字列
 $ff = "/usr/lib/X11/fonts/TrueType/kochi-gothic.ttf"; // TrueTypeフォント
 $bbox = imagettfbbox(25, 0, $ff, $str); // Bounding Boxの座標を取得
 print_r($bbox);
?>
```

■array imagettftext(resource im, int size, int angle, int x, int y, int col, string fontfile, string text)

F

TrueTypeフォントを用いてテキスト文字列を描画します。*18 フォントファイル（font file）、サイズ（size）、角度（angle）、先頭文字の描画位置（x,y）、色（col）を引数として指定します。バウンディングボックスを示す配列（**図3-6を参照**）を返します。

*17
imagecopyresized()のサンプルを参照。

*18
この関数を使用するにはFree Typeライブラリが必要です。

```
<?php
 $im = imagecreate(200,50); // イメージ領域を新規に確保
 $blue = imagecolorallocate($im, 0, 0, 255); // 背景色（青色）を定義
 $white = imagecolorallocate($im, 255, 255,255); // 白色を定義
 $str = "日本語表示"; // 表示する文字列 *19
 $ff = "/usr/lib/X11/fonts/TrueType/kochi-gothic.ttf"; // TrueTypeフォント
 imagettftext($im,25,0,20,30,$white,$ff,$str); // 文字列を描画
 header("Content-type: image/png"); // HTTPヘッダにてPNGイメージを指定
 header("Cache-control: no-cache"); // キャッシュを無効にする
 imagepng($im); // 画像をブラウザに出力
?>
```

図3-11 imagettftext()によるサンプルイメージ

■int imagetypes()

使用しているPHPの環境でサポートされているイメージファイル形式を整数として返します。返り値は、PHP定数 IMG_GIF、IMG_JPG（またはIMG_JPEG）、IMG_PNG、IMG_WBMP、IMG_XPMのうち、サポートされているイメージ形式の論理和となります。

```
<?php // サポートされているイメージ型式を取得して表示
 $types = imagetypes();
 print ($types & IMG_PNG) ? "PNG " : "";
 print ($types & IMG_JPG) ? "JPEG " : "";
 print ($types & IMG_WBMP) ? "WBMP " : "";
 print ($types & IMG_XPM) ? "XPM¥n" : "¥n";
?>
```

*19
GDライブラリをJISX0208を有効にしてコンパイルしていない場合は、文字コードをUTF-8に変換する必要があります。

*20
Webブラウザがこの圧縮形式をサポートしている場合にかぎり使用できます。Webブラウザによりサポートされている圧縮形式は、PHP変数 $_SERVER["HTTP_ACCEPT_ENCODING"] の値により確認できます。

*21
levelに圧縮のレベル（0～9）を指定可能です。

3.4 圧縮関数

zlib（http://www.gzip.org/zlib/）を使用することにより、gzipで圧縮されたファイルを透過的に読み書きするための関数がサポートされます。

▶PHP構築オプション

--with-zlib　　圧縮機能を有効にします。

▶php.ini設定オプション

パラメータ	デフォルト値	説明
zlib.output_compression	Off	Onの場合、Webブラウザへの出力をgzipまたはdeflate圧縮する *18

■bool gzclose(resource fp)

F T

gzipファイルを閉じます。

■string gzcompress(string data [, int level])

F

文字列をZLIB圧縮（RFC 1950）したデータを返します。*21

```
<?php // 出力をgzip圧縮する例
 ob_start(); // 出力のバッファリングを開始
 ob_implicit_flush(0); // バッファのフラッシュを無効にする
?>
<html><body>
  ここでコンテンツを出力します。
</body></html>
<?php
 $contents = ob_get_contents(); // 出力を取得
 ob_end_clean(); // 出力バッファリングを終了
 header("Content-Encoding: gzip"); // gzip圧縮をHTTPヘッダで指定 *20
 echo "¥x1f¥x8b¥x08¥x00¥x00¥x00¥x00¥x00"; // gzipデータ用ヘッダ
 $size = strlen($contents); // 圧縮前のコンテンツのサイズ
 $crc = crc32($contents); // 圧縮前のコンテンツのCRC32チェックサム
 $contents = gzcompress($contents, 9); // コンテンツをgzip圧縮
 $contents = substr($contents, 0, strlen($contents)-4)
   . pack('V',$crc) . pack('V',$size); // フッタ部を置換
 echo $contents; // データ出力
?>
```

■string gzdeflate(string data [, int level])

F

文字列をDEFLATE圧縮（RFC 1951）したデータを返します。*21

```
<?php // 出力をdeflate圧縮する例
 ob_start(); // 出力のバッファリングを開始
 ob_implicit_flush(0); // バッファのフラッシュを無効にする
?>
<html><body>
  ここでコンテンツを出力します。
</body></html>
<?php
 $contents = ob_get_contents(); // 出力を取得
 ob_end_clean(); // 出力バッファリングを終了
 header("Content-Encoding: deflate"); // deflate圧縮をHTTPヘッダで指定 *20
 $contents = gzdeflate($contents, 9); // コンテンツをdeflate圧縮
 echo $contents; // データ出力
?>
```

■string gzencode(string data [, int level [, int encoding]])

F

文字列をgzip圧縮（RFC 1952）したデータを返します。*21

3番目の引数にFORCE_GZIP（デフォルト）またはFORCE_DEFLATEを指定することが可能です。

```
<?php // gzip圧縮されたファイルを作成する
 $str = str_repeat("PHP4徹底攻略¥n",1000); // 長い文字列（サンプルデータ）
 $gzdata = gzencode($str, 9, FORCE_GZIP); // 文字列を圧縮
 $fp = fopen("test.txt.gz", "w");
 fwrite($fp, $gzdata); // ファイルに書き込む
 fclose($fp);
?>
```

■bool gzeof(resource fp)

ファイルポインタがEOFに達したか、エラーの場合にTRUE、そのほかの場合にはFALSEを返します。

■array gzfile(string filename [, int use_inc])

gzip圧縮されたファイルfilenameを読み込み、配列として返します。*22

```
<?php
 $arr = gzfile("test.txt.gz"); // 圧縮されたデータを復元して配列に読み込む
 print $arr[0]; // 1行目だけ出力
?>
```

■string gzgetc(resource fp)

ファイルから（解凍した）1文字を読み込み、文字列として返します。EOFの場合にFALSEを返します。

■string gzgets(resource fp, int len)

F

ファイルから最大len-1バイトの文字を読み込み、（復元した）文字列として返します。len-1バイトを読み込むか、改行またはEOFに達した場合、読み込みを終了します。

```
<?php
 $fp = gzopen("test.txt.gz","r"); // gzip圧縮されたファイルをオープン
 print gzgets($fp,256); // 1行分のデータを出力
 gzclose($fp);
?>
```

■string gzgetss(resource fp, int len [, string allowable_tags])

動作はgzgets()と同じですが、読み込んだテキストからHTML/PHPタグをすべて削除します。使用可能なタグをオプションで指定可能です。

■string gzinflate(string data [, int len])

DEFLATE圧縮されたデータを復元して返します。復元後のデータの最大長をlenで指定できます。

■resource gzopen(string filename, string mode [,int use_inc])

F

gzipファイルをオープンし、ファイルポインタを返します。このポインタを指定して読み込むデータは自動的に解凍され、書き込むデータは圧縮されます。modeの指定方法はfopen()と同じですが、圧縮法の指定等のgzipで指定可能なオプションも指定可能です。*22

*22
use_incに1を指定した場合、include_pathで指定したパスも検索します。

```
<?php
 $str = str_repeat("PHP4徹底攻略¥n",1000); // 長い文字列（サンプルデータ）
 $fp = gzopen("test2.txt.gz","wb9"); // gzip圧縮するファイルをオープン
 gzwrite($fp,$str);
 gzclose($fp);
?>
```

■int gzpassthru(resource fp)

ファイルをEOFまで読み込み、標準出力に（復元した）結果を出力したあとでファイルを閉じます。出力したバイト数を返します。

```
<?php
 $fp = gzopen("test.txt.gz","r");
 $len = gzpassthru($fp); // 圧縮されたデータを復元して出力
 print "読み込んだバイト数: ".$len;
?>
```

■string gzread(resource fp, int len)

最大lenバイトのデータを読み込み、（復元された）lenバイトのデータが読み込まれたかEOFに達したときに読み込みは終了します。

■bool gzrewind(resource fp)

T

ファイルポインタの位置をファイルストリームの最初に設定します。

■int gzseek(resource fp, int offset)

ファイルポインタの位置をoffsetに設定します。成功時に0、それ以外の場合には-1を返します。

■int gztell(resource fp)

ファイルポインタの現在位置を返します。

■string gzuncompress(string data [,int len])

ZLIB圧縮されたデータを復元して返します。lenには出力の最大データ長を指定可能です。

■int gzwrite(resource fp, string str [,int len])　[エイリアス] gzputs()

文字列をファイルにgzip圧縮して書き込み、書き込んだ（圧縮前の）バイト数を返します。lenを指定した場合、lenバイトのデータを書き込んだ時点で書き込みを終了します。*23

■int readgzfile(string filename [, int use_inc])

F

gzip圧縮されたファイルを読み込み、復元して標準出力に出力します。読み込んだ（復元後の）データのバイト数を返します。*22

```
<?php
 $len = readgzfile("test.txt.gz"); // 圧縮されたデータを復元して出力
 print "読み込んだバイト数: ".$len;
?>
```

3.5 PDF関数

PHPでは、標準電子文書フォーマットであるPDF（Portable Document Format）形式のドキュメントを動的に作成することが可能です。

▶PHP構築オプション

--with-pdflib[=DIR]	PDFlibによるPDF関数を有効にします。 （例： --with-pdflib=/usr/local）
--with-jpeg-dir=DIR	libjpegのインストール先を指定します。 （例： --with-jpeg-dir=/usr/local）
--with-png-dir=DIR	libpngのインストール先を指定します。 （例： --with-png-dir=/usr/local）
--with-tiff-dir=DIR	libtiffのインストール先を指定します。 （例： --with-tiff-dir=/usr/local）
--with-zlib-dir=DIR	zlibのインストール先を指定します。 （例： --with-zlib-dir=/usr/local）

▶参考URL

PDFlibの開発･配布元　：**http://www.pdflib.com/pdflib/**

▶注意

・PDF関数を使用するには、PDFlib（バージョン3.0以降）が必要です。

*23
gzopen()のサンプルを参照してください。

・PDFlibを商用に使用する場合は、ライセンス（有償）が必要です。
・Webブラウザに直接PDFファイルを出力する際には、適切なHTTPヘッダを出力する必要があります。*24

*24 pdf_open_image_file()のサンプルを参照してください。

■int pdf_add_bookmark(resource pdf, string text [, int parent [, int open]])

階層構造のブックマークを上位のブックマークparent（省略時または0の場合は最上位）に追加し、ブックマークのIDを返します。*25

■bool pdf_add_launchlink(resource pdf, float x1, float y1, float x2, float y2, string filename)

T

左下隅を (x1,y1)、右上隅を (x2,y2) とするファイルへのリンクを追加します。

■bool pdf_add_locallink(resource pdf, float x1, float y1, float x2, float y2, int page, string dest)

T

左下隅を (x1,y1)、右上隅を (x2,y2) とする矩形領域にカレントのPDF文書内へのリンクを追加します。pageはページ番号、destはズームオプション（**表3-26**）です。

表3-26 ズームオプション文字列

オプション	説明
retain	ズーム倍率を保持する
fitpage	ウインドウにページ全体が表示されるように調整する
fitwidth	ウインドウにページの幅を合わせる
fitheight	ウインドウにページの高さを合わせる
fitbbox	ページのバウンディングボックスをウインドウに合わせる

■bool pdf_add_note(resource pdf, float x1, float y1, float x2, float y2, string contents, string title, string icon, int open)

T

左下隅を (x1,y1)、右上隅を (x2,y2) とする矩形領域に注記を追加します。titleは注記のタイトル、contentsは内容です。iconは、"comment"、"insert"、"note"、"paragraph"、"newparagraph"、"key"、"help"のいずれかです。*25

*25 openに1を設定すると注記がオープンされた状態となります。

■bool pdf_add_pdflink(resource pdf, float x1, float y1, float x2, float y2, string filename, int page, string dest)

T

左下隅を (x1,y1)、右上隅を (x2,y2) とする矩形領域にほかのPDF文書へのリンクを追加します。filenameはPDF文書ファイル、pageはページ、destはズームオプション（**表3-26**）です。

■bool pdf_add_thumbnail(resource pdf, int image)

T

既存のイメージをカレントのページのサムネイルとして追加します。サムネイルのサイズは、106*106ピクセル以下とする必要があります。

■bool pdf_add_weblink(resource pdf, float x1, float y1, float x2, float y2, string url)

T

左下隅を (x1,y1)、右上隅を (x2,y2) とする矩形領域に指定したURLへのリンクを設定します。

■bool pdf_arc(resource pdf, float x, float y, float r, float start, float end)

T

中心 (x,y)、半径r、開始角start、終了角endの円弧を反時計周りに描画します。

■bool pdf_arcn(resource pdf, float x, float y, float r, float start, float end)

T

中心 (x,y)、半径r、開始角start、終了角endの円弧を時計回りに描画します。*26

■int pdf_attach_file(resource pdf, float x1, float y1, float x2, float y2, string filename, string desc, string author, string mimetype, string icon)

T

左下隅を (x1,y1)、右上隅を (x2,y2) とする矩形領域にほかの文書を添付します。filenameは文書ファイル、descは説明、authorは作者、mimetypeはMIME型、iconは "graph"、"paperclip"、"pushpin"、"tag"のいずれかです。

*26 pdf_stroke()のサンプルを参照してください。

■bool pdf_begin_page(resource pdf, float w, float h)

T

高さh、幅w（単位：ポイント=1/72inch）で新規ページを開始します。*27

■int pdf_begin_pattern(resource pdf, float w, float h, float xstep, float ystep, int painttype)

高さh、幅wのバウンディングボックスに新規にパターンを作成し、パターンのハンドルを返します。xstep、ystepはそれぞれX、Y方向のパターン反復数です。painttype=1の場合は固有の色設定を使用し、2の場合は現在の色設定を使用します。*28

■int pdf_begin_template(resource pdf, float w, float h)

高さh、幅wを指定してテンプレートを開始します。

```
<?php
 $pdf = pdf_new();
 pdf_open_file($pdf, "test1.pdf"); // PDFファイルを作成
 $tpl = pdf_begin_template($pdf, 100, 20); // テンプレート定義を開始
 $font = pdf_findfont($pdf, "Times-Roman", "host", 0); // Times Romanを使用
 pdf_setfont($pdf, $font, 20);
 pdf_show_xy($pdf, "Hello!", 0, 0); // テキストを出力
 pdf_end_template($pdf);  // テンプレート定義を終了

 pdf_begin_page($pdf, 300, 150);
 for ($i=10; $i< 160; $i+=30){
    pdf_place_image($pdf, $tpl, 0, $i, 1.0); // テンプレートを配置
 }
 pdf_end_page($pdf);
 pdf_close($pdf);
?>
```

■bool pdf_circle(resource pdf, float x, float y, float r)

T

中心を(x,y)、半径をrとした円を描画します。*26

■bool pdf_clip(resource pdf)

カレントのパスを用いてイメージをクリッピングします。

■bool pdf_close(resource pdf)

T

PDFドキュメントを閉じます。

■void pdf_close_image(resource pdf, int image)

PDF_open_image()関数でオープンされたイメージを閉じます。

■bool pdf_close_pdi(resource pdf, int doc)

T

PDI機能によりオープンされたPDF文書を閉じます。*29

■bool pdf_close_pdi_page(resource pdf, int page)

T

PDI機能によりオープンされたPDF文書のページを閉じます。*29

■bool pdf_closepath(resource pdf)

T

カレントのパスを閉じます。

■bool pdf_closepath_fill_stroke(resource pdf)

T

カレントのパスの内部を閉じ、塗りつぶし、パスを描画します。

■bool pdf_closepath_stroke(resource pdf)

T

カレントのパスの内部を閉じ、パスを描画します。

■bool pdf_concat(resource pdf, float a, float b, float c, float d, float e, float f)

T

行列をカレントの変換行列（デフォルト[1,0,0,1,0,0]）に追加します。

*27
pdf_open_file()のサンプルを参照してください。

*28
pdf_setcolor()のサンプルを参照してください。

*29
PDI（PDF import library）機能は、ソースコード配布版のPDF libには含まれません。バイナリパッケージ版が必要です。また、商用ライセンスを取得しない場合は、背景に「www.pdflib.com」の文字が表示されます。

■bool pdf_continue_text(resource pdf, string text)

T

次の行にテキストを出力します。

■bool pdf_curveto(resource pdf, float x1, float y1, float x2, float y2, float x3, float y3)

T

ベジェ曲線をカレントの点から点 (x3,y3) まで (x1,y1) および (x2,y2) を制御点として描画します。*26

■bool pdf_delete(resource pdf)

T

PDF文書を閉じます。*24

■bool pdf_end_page(resource pdf)

T

ページを終了します。

■bool pdf_end_pattern(resource pdf)

T

パターンを終了します。

■bool pdf_end_template(resource pdf)

T

テンプレートを終了します。*30

■bool pdf_endpath(resource pdf)

カレントのパスを終了します。

■bool pdf_fill(resource pdf)

T

カレントのパスの内部を塗りつぶします。*26

*30
pdf_begin_template()のサンプルを参照してください。

*31
"builtin"、"macroman"、"winansi"、"host"またはユーザ定義の文字コード名です。

*32
modifier（不要の場合は0）によりパラメータを修飾します。

■bool pdf_fill_stroke(resource pdf)

T

カレントのパスの内部を塗りつぶし、パスを描画します。

■int pdf_findfont(resource pdf, string fontname, string encoding [, int embed])

F

フォントを検索し、見つかった場合にフォントIDを返します。フォントIDは、pdf_setfont()で使用されます。embedが0以外の場合、フォントの検索は行なわれますが使用されません。encodingは文字エンコーディング *31 です。*27

■string pdf_get_buffer(resource pdf)

PDF出力バッファの内容を取得します。この結果は、ほかのPDF関数をコールする前に使用する必要があります。*24

■int pdf_get_majorversion()

使用するPDFlibのメジャーバージョンを返します。

■int pdf_get_minorversion()

使用するPDFlibのマイナーバージョンを返します。

■string pdf_get_parameter(resource pdf, string key [, float modifier])

パラメータの値を文字列として取得します。*32

■string pdf_get_pdi_parameter(resource pdf, string key , int doc, int page, int index)

PDI文書内の文字列型パラメータの内容を取得します。*29

■float pdf_get_pdi_value(resource pdf, string key , int doc, int page, int index)

PDI文書内の数値型パラメータの内容を取得します。*29

float pdf_get_value(resource pdf, string key, float modifier)

PDF文書のパラメータの値を数値として取得します。*31 keyは、"imagewidth"、"image height"、"resx"、"resy"、"capheight"、"ascender"、"descender"、"font" などです。*33

bool pdf_initgraphics(resource pdf)

色およびグラフィックに関するパラメータをすべてデフォルト値にリセットします。

bool pdf_lineto(int fd, float x, float y)

T

カレントの点から座標 (x, y) の点まで線を描画します。*28

int pdf_makespotcolor(resource fd, string spotname)

カレントの色からスポットカラー（spotname）を作成し、色IDを返します。*28

bool pdf_moveto(resource pdf, float x, float y)

T

カレントの位置を (x,y) に設定します。*28

resource pdf_new()

新規にPDFオブジェクトを作成し、リソースIDを返します。*26

int pdf_open_ccitt(resource pdf, string filename, int w, int h, int bitreverse, int k, int blackls1)

幅wピクセル、高さhピクセルのCCITT G3/G4形式圧縮ビットマップのイメージファイルをオープンし、PDFイメージIDを返します。bitreverseが1の場合、ビットごとに反転されます。kは圧縮パラメータで、-1（G4エンコーディング）、0（一次元G3エンコーディング）、1（二次元G3エンコーディング）を指定します。blackls1に1を指定した場合、イメージの各ビットに関して1は黒、0は白と解釈されます。

bool pdf_open_file(resource pdf [, string filename])

T

新規にPDF文書をオープンします。filenameを指定しない場合、PDF文書はメモリ上に作成されます。

```
<?php // PDF出力の例
 $pdf = pdf_new();
 pdf_open_file($pdf, "test2.pdf"); // PDFファイルを作成
 pdf_begin_page($pdf, 595, 842); // A4 size
 $font = pdf_findfont($pdf, "Times-Roman", "host", 0); // Times Romanを使用
 $font_ja = pdf_findfont($pdf, "HeiseiMin-W3", "EUC-H", 0); // 平成明朝を使用
 pdf_setfont($pdf, $font, 30); // フォントをTimes Romanに設定
 pdf_show_xy($pdf, "Hello! PHP", 200, 800); // テキストを出力
 pdf_setfont($pdf, $font_ja, 30); // フォントを平成明朝に設定
 pdf_set_text_pos($pdf, 200, 750); // (200,750) からテキストを描画
 pdf_show($pdf, "こんにちは、ＰＨＰ");  // テキストを出力
 pdf_end_page($pdf);
 pdf_close($pdf);
?>
```

図3-12　pdf_open_file()のサンプル出力

*33
pdf_save()のサンプルを参照してください。

■bool pdf_open_image(resource pdf, string type, string source, string data, long len, int w, int h, int components, int bpc, string params)

指定した型式のイメージファイルをオープンし、PDFイメージIDを返します。typeにはイメージ形式（"jpeg"、"ccitt"、"raw"）、sourceにはイメージソース（"memory"、"fileref"、"url"）を指定します。componentsは色素の数であり、1（グレースケール）、3（RGB）、4（CMYK）のどれかを指定します。bpcは要素ごとのビット数であり、1、2、4、8のどれかを指定します。typeが"jpeg"の場合は8、"ccitt"の場合は1とします。lenは"raw"形式、paramsは"ccitt"形式のイメージのみで使用されます。

■int pdf_open_image_file(resource pdf, string type, string filename [, string strparam, int intparam])

F

指定した型（"png"、"gif"、"jpeg"、"tiff"）のイメージファイルをオープンし、PDFイメージIDを返します。

```
<?php
 $pdf = pdf_new();
 pdf_open_file($pdf);
 pdf_begin_page($pdf, 200, 300);
 $pim = pdf_open_image_file($pdf, "png", "php-big.png");
 pdf_place_image($pdf, $pim, 0, 250, 0.25);
 pdf_end_page($pdf);
 pdf_close($pdf);
 $buf = pdf_get_buffer($pdf); // PDFデータをバッファを取得
 $len = strlen($buf);
 header("Content-type: application/pdf"); // HTTPヘッダ出力
 header("Content-Length: $len");
 header("Content-Disposition: inline; filename=test3.pdf");
 echo $buf;
 pdf_delete($pdf);
?>
```

■int pdf_open_memory_image(resource pdf, resource image)

F

PHPのイメージ関数（第3章を参照）によるイメージをPDFドキュメントで利用できるようにし、PDFイメージIDを返します。

```
<?php
 $im = imagecreatefrompng("test1.png"); // PNGイメージファイルを読み込む
 $pdf = pdf_new();
 pdf_open_file($pdf, "test4.pdf"); // PDFファイルを作成
 $pim = pdf_open_memory_image($pdf, $im); // PDFイメージとして展開
 imagedestroy($im); // イメージを破棄
 pdf_begin_page($pdf, 300, 300);
 pdf_place_image($pdf, $pim, 100, 100, 1); //  PDFイメージをドキュメントに配置
 pdf_end_page($pdf);
 pdf_close_image($pdf, $pim); // PDFイメージをクローズ
 pdf_close($pdf);
?>
```

■int pdf_open_pdi(resource pdf, string filename, string strparam, int intparam)

既存のPDF文書ファイルをオープンし、文書IDを返します。strparam、intparamにはそれぞれNULL、0を指定します。[*29]

```
<?php
 $pdf = pdf_new();
 pdf_open_file($pdf, "test5.pdf"); // PDFファイルを作成
 $pdi = pdf_open_pdi($pdf, "test2.pdf", "", 0); // 既存のPDFファイルをインポート
 $page = pdf_open_pdi_page($pdf, $pdi, 1, NULL);
 $width = pdf_get_pdi_value($pdf, "width", $pdi, $page, 0);
 $height = pdf_get_pdi_value($pdf, "height", $pdi, $page, 0);
 pdf_begin_page($pdf, $width, $height);
 pdf_place_pdi_page($pdf, 1, 0, 0, 1.0, 1.0); // PDFページを配置
 pdf_close_pdi_page($pdf, $page);
 pdf_end_page($pdf);
 pdf_close($pdf);
 pdf_close_pdi($pdf, $pdi);
?>
```

■int pdf_open_pdi_page(resource pdf, int doc, int page, string label)

文書IDで指定したPDF文書の指定したページを取得し、イメージIDを返します。このイメージIDは、pdf_place_image()で使用します。labelにはNULLを指定します。*29

■bool pdf_place_image(resource pdf, resource image, float x, float y, float scale)

T

イメージを座標 (x,y) に置きます。scaleによりサイズ（1：等倍）を調整することが可能です。*34

■bool pdf_place_pdi_page(resource pdf, int page, float x, float y, float sx, float sy)

T

PDFページを左隅を座標 (x,y) とする位置に置きます。sx、syはそれぞれ縦および横方向の倍率です。*29

■bool pdf_rect(resource pdf, float x, float y, float r, float w, float h)

左下隅 (x,y)、幅wピクセル、高さhピクセルとなる矩形を描画します。*26

■bool pdf_restore(resource pdf)

T

pdf_save()により保存された環境を回復します。*33

■bool pdf_rotate(resource pdf, float angle)

T

カレントの座標系をangle度反時計回りに回転します。*33

■bool pdf_save(resource pdf)

T

カレントの環境を保存します。回転、変換を行なう場合に使用します。

```
<?php
 $pdf = pdf_new();
 pdf_open_file($pdf, "test6.pdf"); // PDFファイルを作成
 pdf_begin_page($pdf, 300, 150);
 $font = pdf_findfont($pdf, "Times-Roman", "host", 0); // Times Romanを使用
 pdf_setfont($pdf, $font, 30);
 pdf_show_xy($pdf, "Hello!", 0, 50); // テキストを出力
 $textx = pdf_get_value($pdf, "textx"); // 現在のテキスト位置を取得
 $texty = pdf_get_value($pdf, "texty");
 pdf_save($pdf); // 環境を保存
 pdf_translate($pdf, $textx,$texty); // 座標系の原点を (150,50) に設定
 pdf_rotate($pdf, 45);  // 座標系を45度回転
 pdf_show_xy($pdf, "PHP", 20, 0); // テキストを出力（45度回転）
 pdf_restore($pdf); // 環境を復元
 pdf_show_xy($pdf, "2002", 150, 50); // テキストを出力
 pdf_end_page($pdf);
 pdf_close($pdf);
?>
```

図3-13 pdf_save()のサンプル出力

*34
pdf_open_memory_image()のサンプルを参照してください。

■bool pdf_scale(resource pdf, float sx, float sy)

T

XおよびY方向の倍率としてsxおよびsyを設定します。

■bool pdf_set_border_color(resource pdf, float r, float g, float b)

T

RGB値により注記の境界色を設定します。

■bool pdf_set_border_dash(resource pdf, float b, float w)

T

白色および黒色の値により注記の波線のスタイルを設定します。

■bool pdf_set_border_style(resource pdf, string style, float w)

T

注記の境界のスタイルを設定します("solid"または"dashed")。

■bool pdf_set_info(resource pdf, string field, string value)

PDFドキュメントのフィールドの値をvalueに設定します。fieldには"Subject"、"Title"、"Creator"、"Author"、"Keywords"、"Trapped"またはユーザ定義のフィールドを指定可能です。

```
<?php
 $pdf = pdf_new();
 pdf_open_file($pdf, "test7.pdf");
 pdf_set_info($pdf, "Subject", "PDFlib test.");
 pdf_set_info($pdf, "Author", "Suzuki Taro");
 pdf_set_info($pdf, "Title", "Test for PHP wrapper of PDFlib");
 pdf_set_info($pdf, "Creator", "PHP with PDFlib");
 pdf_set_info($pdf, "Keywords", "PHP, PDFlib");
 pdf_begin_page($pdf, 595, 842); // A4 size
 pdf_end_page($pdf);
 pdf_close($pdf);
?>
```

■bool pdf_set_parameter(resource pdf, string name, string value)

T

文字列型のパラメータnameの値を設定します。

```
<?php
 $pdf = pdf_new();
 pdf_set_parameter($pdf, "resourcefile", "/usr/local/fonts/pdflib.upr");
                                                    // フォント設定ファイル
 pdf_set_parameter($pdf, "compatibility", "1.2");  // PDF 1.2 型式で出力（日本語使用不可）
 pdf_open_file($pdf, "test8.pdf");
 pdf_set_parameter($pdf, "openaction", "fitpage"); // ページ全体を表示
 pdf_begin_page($pdf, 595, 842); // A4 size
 pdf_set_parameter($pdf, "openaction", "fitpage"); // ページ全体を表示
 pdf_end_page($pdf);
 pdf_close($pdf);
?>
```

■bool pdf_set_text_pos(resource pdf, float x, float y)

T

テキストの描画開始位置を (x,y) に設定します。[*27]

■bool pdf_set_value(resource pdf, string name, float value)

T

数値型のパラメータnameの値を設定します。

```
<?php
 $pdf = pdf_new();
 pdf_open_file($pdf, "test9.pdf");
 $font_ja = pdf_findfont($pdf, "HeiseiKakuGo-W5", "EUC-H", 0); // 平成明朝を定義
 pdf_begin_page($pdf, 595, 842); // A4 size
 pdf_setfont($pdf, $font_ja, 48); // フォントを平成明朝に設定

 for ($i=0;$i<3;$i++){
   pdf_set_value($pdf, "textrendering", $i); // テキスト描画モードを設定
   pdf_set_text_pos($pdf, 200, 800-$i*50); // テキスト描画位置を設定
   pdf_show($pdf, "日本語"); // テキストを出力
 }
 pdf_end_page($pdf);
 pdf_close($pdf);
?>
```

■bool pdf_setcolor(resource pdf, string type, string color, float c1 [, float c2 [, float c3 [, float c4]]])

T

色空間と色を設定します。typeには"fill"（塗りつぶし）、"stroke"（輪郭描画）、"both"（輪郭描画および塗りつぶし）を指定します。colorには、色空間"gray"、"rgb"、"cmyk"、"spot"、

"pattern" を指定し、c1～c4にはその色空間のそれぞれの色要素を指定します。

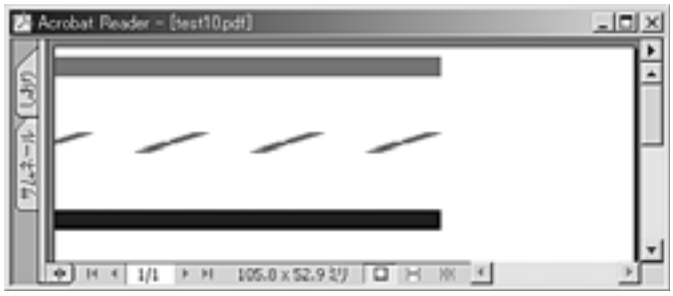
このほか、**表3-27**に示すPDFドキュメントのパラメータを設定する関数を使用できます。

```
<?php
 $pdf = pdf_new();
 pdf_open_file($pdf, "test10.pdf"); // PDFファイルを作成
 $ptn = pdf_begin_pattern($pdf, 60, 20, 60, 20, 1); // パターン定義を開始
 pdf_setcolor($pdf, "both", "rgb", 1, 0, 0); // 描画色を赤色に設定
 pdf_setlinewidth($pdf, 2);
 pdf_moveto($pdf, 0,0); pdf_lineto($pdf, 60,20);
 pdf_stroke($pdf); // パスを描画
 pdf_end_pattern($pdf);  // パターン定義を終了
 pdf_setcolor($pdf, "both", "rgb", 0, 0, 1); // 描画色を青色に設定
 $sp = pdf_makespotcolor($pdf, "blue"); // スポットカラーを作成

 pdf_begin_page($pdf, 300, 150);
 pdf_setlinewidth($pdf, 10); // 線幅を10ポイントに設定
 pdf_setcolor($pdf, "both", "rgb", 1, 0, 0); // 描画色を赤色に設定
 pdf_moveto($pdf, 0,140); pdf_lineto($pdf, 200,140); pdf_stroke($pdf); // 線を描画

 pdf_setcolor($pdf, "both", "pattern", $ptn); // 色としてパターンを指定
 pdf_moveto($pdf, 0,100); pdf_lineto($pdf, 200,100);  pdf_stroke($pdf); // 線を描画

 pdf_setcolor($pdf, "both", "spot", $sp, 0.5); // 色としてスポット色を指定
 pdf_moveto($pdf, 0,60); pdf_lineto($pdf, 200,60); pdf_stroke($pdf); // パスを描画

 pdf_end_page($pdf);
 pdf_close($pdf);
?>
```

表3-27　PDFドキュメントのパラメータを設定する関数

関数名	設定値
bool pdf_setdash(resource pdf, float w, float b)	破線パターン
bool pdf_setflat(resource pdf, float value)	平面度を0から100の値
bool pdf_setlinecap(resource pdf, int value)	linecapパラメータ
bool pdf_setlinejoin(resource pdf, int value)	linejoinパラメータ
bool pdf_setlinewidth(resource pdf, float width)	線幅
bool pdf_setmiterlimit(resource pdf, float value)	miterリミット
bool pdf_setpolydash(resource pdf, array pattern)	波線パターン
bool pdf_set_border_color(resource pdf, float r, float g, float b)	境界色
bool pdf_set_border_dash(resource pdf, float b, float w)	ダッシュのスタイル
bool pdf_set_border_style(resource pdf, float style, float w)	境界のスタイル

図3-14 pdf_setcolor()のサンプル出力



■bool pdf_setfont(resource pdf, int font, float size)

T

カレントのフォントを設定します。fontはpdf_findfont()により返されたフォントID、sizeはフォントのサイズです。*27

■bool pdf_setmatrix(resource pdf, float a, float b, float c, float d, float e, float f)

T

カレントの変換行列を設定します。

■bool pdf_show(resource pdf, string str)

T

カレントの位置に文字列を出力します。*27

■int pdf_show_boxed(resource pdf, string str, float x, float y, float w, float h, string mode [, string feature])

左下隅の座標 (x,y)、幅w、高さhのボックス内に文字列を出力します。modeには"left"、"right"、"center"、"justify"、"fulljustify"を指定可能です。

■bool pdf_show_xy(resource pdf, string str, float x, float y)

T

座標を (x,y) に文字列を出力します。*35

■bool pdf_skew(resource pdf, float alpha, float beta)

x、y方向の座標系をそれぞれalpha度、beta度歪ませます。

■float pdf_stringwidth(resource pdf, string text [, int font [, float size]])

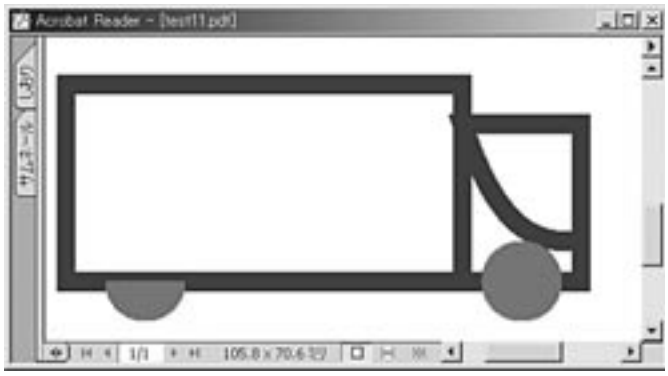
カレントフォントのテキストの幅を返します。

bool pdf_stroke(resource pdf)

カレントのパスに沿って線を描画します。

```
<?php
 $pdf = pdf_new(); pdf_open_file($pdf, "test11.pdf"); // PDFファイルを作成
 pdf_begin_page($pdf, 300, 200);
 pdf_setcolor($pdf, "both", "rgb", 0, 0, 1); // 描画色を青色に設定
 pdf_setlinewidth($pdf, 5); // 線幅を5に設定
 pdf_rect($pdf, 50, 50 , 100, 50); // 矩形を描画

 pdf_moveto($pdf, 150,50); // (150,50)に移動
 pdf_lineto($pdf, 180,50); pdf_lineto($pdf, 180,90); // 線を描画
 pdf_lineto($pdf, 150,90);// pdf_lineto($pdf, 200,200);
 pdf_curveto($pdf, 160, 60, 170, 60, 180, 60); // ベジェ曲線を描画
 pdf_stroke($pdf); // パスを描画
 pdf_setcolor($pdf, "both", "rgb", 1, 0, 0); // 描画色を赤色に設定
 pdf_circle($pdf, 165, 50 , 10); // 中心(165,50),半径10の円を描画
 pdf_fill($pdf); // パスを描画
 pdf_arcn($pdf, 70, 50 , 10, 0, 180); // 中心(70,50),半径10の円弧を180から0degまで描画
 pdf_fill($pdf); // パスを描画
 pdf_end_page($pdf);
 pdf_close($pdf);
?>
```

図3-15 pdf_stroke()のサンプル出力



*35
pdf_set_value()のサンプルを参照してください。

bool pdf_translate(resource pdf, float x, float y)

座標系の原点を (x,y) に設定します。*33

Hypertext Preprocessor

PHP関数

Chapter-4 システム関連

4.1 プログラム実行関数

*1
ユーザ入力をシェルコマンド実行関数（system()など）の引数として使用する際のセキュリティを確保するために使用されます。

*2
retを指定した場合、コマンドのステータスが書き込まれます。

`command`構文や本節の関数によりPHPスクリプトからシェル上で任意のコマンドを実行可能です。

シェルによるコマンドの実行はセキュリティホールになりやすいため注意が必要です。POST/GET/Cookieによるユーザからの入力をコマンドの一部として使用する場合には、escapeshellarg()、escapeshellcmd()といった関数により危険な入力をエスケープするといった対策が必要です。

シェルによるコマンドの実行はセーフモード（safe_mode）オプションを有効にした場合、safe_mode_exec_dir以下にあるコマンドのみが実行可能となります。

■string escapeshellarg(string arg)

文字列をシングルクオートでくくり、単一のシェルコマンド引数として返します。文字列に含まれるシングルクオートはすべてクオートでエスケープされます。*1 悪意のあるユーザ入力により複数のシェルコマンドが実行されることを防止することができます。

```
<?php
 $_POST['name'] = "Taro; cat /etc/passwd"; // 悪意のあるユーザ入力の例
 $arg = escapeshellarg($_POST['name']); // 'Taro; cat /etc/passwd' となる
 system("echo ".$arg); // Taro; cat /etc/passwd を出力
?>
```

■string escapeshellcmd(string cmd)

シェルコマンドとして実行する際にセキュリティ上問題を生じる可能性があるメタ文字（#、&、;、`、¥、"、|、*、?、~、<、>、^、(、)、[、]、{、}、$、¥、0x0a、0xff）をエスケープします。*1

```
<?php
 $_POST['name'] = "Taro; cat /etc/passwd"; // 悪意のあるユーザ入力の例
 $arg = escapeshellcmd($_POST['name']); // Taro ¥; cat /etc/passwd となる
 system("echo ".$arg); // Taro; cat /etc/passwd を出力
?>
```

■string exec(string cmd [, array &arr [, int &ret]])

コマンドを実行し、結果の最後の行を文字列として返します。標準出力への出力は行なわれません。arr を指定した場合、実行したコマンドの出力が行ごとに保存されます。*2

```
<?php
 exec('ls -l',$arr);
 print_r($arr); // 'ls -l'の結果を出力
?>
```

■void passthru(string cmd [, int &ret])

コマンドを実行し、コマンドの出力をバイナリデータとしてそのままブラウザ（標準出力）に出力します。*2

```
<?php
 passthru('ls -l'); // 'ls -l'の結果を出力
?>
```

■string shell_exec(string cmd)

シェルコマンドを実行し、結果を文字列として返します。

```
<?php
 $cmd = "ls ";
 print shell_exec($cmd); // ディレクトリリストを表示
?>
```

■string system(string cmd [, int &ret])

コマンドを実行し、結果を文字列として出力します。*2

```
<?php
 system('ls -l'); // 'ls -l'の結果を出力
?>
```

4.2 PHP実行関数

■int connection_aborted(void)

クライアントとの接続が断たれた場合にTRUEを返します。

■int connection_status(void)

接続ステータス（0：NORMAL、1：ABORTED、2：TIMEOUT）を返します。

■mixed eval(string str)

文字列をPHPコードとして評価し、その値を返します。動的にPHPコードを生成するといった用途に使用できます。適切にエスケープを行なう必要があります。

```
<?php
 $name = "Taro";
 eval("¥$a = strtoupper(¥"$name¥");"); // 大文字にして変数$aに代入
 print $a; // TAROを出力
?>
```

■void exit([mixed status])　［エイリアス］die()

スクリプトの実行を終了します。statusが文字列の場合は出力し、整数の場合はexitステータスとなります。*3

```
<?php
 $fp = fopen("exit.php","r") or exit("unable to open file.");
 print fgets($fp); // exit.phpの1行目を表示
 fclose($fp);
?>
```

■int ignore_user_abort([bool val])

クライアントとの接続が断たれた場合に、スクリプトの実行を中断するかどうかを設定します。引数を省略した場合、現在の設定が返されます。

```
<?php
 $old = ignore_user_abort(TRUE); // ユーザ側からの実行停止を無視する
 $fp = fopen("/tmp/tmpfile.txt","w") or die("file open failed.");
 for ($i=0; $i<10; $i++){
   // 時間のかかるループ処理の例:
   //  この間にSTOPボタンが押されてもファイルへの書き込みは継続される
   sleep(1); // 1秒間スリープ
   fwrite($fp, "count $i¥n");  // ファイルに書き込む
 }
 fclose($fp);
 ignore_user_abort($old);
?>
```

■void sleep(int sec)

sec秒の間、プログラムの実行を停止させます。

■void usleep(int micro_sec)

micro_secマイクロ秒の間、プログラムの実行を停止します。

*3
exit()/die()は関数ではなく、PHPスクリプト言語の一部です。

4.3 PHPオプションおよび情報

■bool dl(string library)

F T

指定したPHP拡張モジュールを動的にロードします。

```
<?php // 拡張モジュールgdがロードされていない場合にロードする
if (!extension_loaded('gd')) {
    if (!dl('gd.so')) {
        exit;
    }
}
?>
```

bool extension_loaded(string name)

指定したPHP拡張モジュールがロードされている場合にTRUE、それ以外の場合にFALSEを返します。

string get_cfg_var(string var)

PHP設定オプションvarの現在の設定値を返します。Apacheの設定ファイル（httpd.conf または .htaccess）でphp_valueなどにより設定した情報は返しません。PHPスクリプトの情報を取得するには**表3-28**の関数を使用します。また、PHPスクリプトの情報を設定するには**表3-29**の関数を使用します。なお、**表3-30**に示す関数により拡張モジュールや関数、インクルードされたファイル名に関する情報を得ることが可能です。

表3-28　スクリプトの情報を取得する関数

関数名	返り値
int getlastmod()	最終更新時刻（UNIX時間）*4
int getmyinode()	iノード番号 *4
int getmypid()	プロセスID *4
int getmyuid()	ユーザID *4
string get_current_user()	実行中のPHPスクリプトの所有者
int get_magic_quotes_gpc()	magic_quotes_gpcの値（0または1）
int get_magic_quotes_runtime()	magic_quotes_runtimeの値（0または1）

*4
エラーの場合はFALSEが返されます。

表3-29　スクリプトの情報を設定する関数

関数名	設定する値
void set_time_limit(int sec)	最大実行時間（デフォルトは30秒）
int set_magic_quotes_runtime(int val)	magic_quotes_runtimeの値（0または1）

表3-30　関数・モジュールに関する情報を取得する関数

関数名	返り値
array get_loaded_extensions()	コンパイル・ロード中のモジュール名
array get_extension_funcs(string module)	モジュールで使用可能な関数
array get_included_files()	include_once()でロードされたファイル
array get_required_files()	require_once()でロードされたファイル

string getenv(string var)

環境変数varの値を返します。

■array getrusage([int who])

現在のシステムリソースに関する情報をシステムコールgetrusage()から得て連想配列として返します。whoに1を指定した場合、getusage()にRUSAGE_CHILDRENが追加されます。

```
<?php
 $usage = getrusage(); // システムリソースを取得
 print $usage['ru_utime.tv_sec'].
   ".".$usage['ru_utime.tv_usec']; // 消費時間（ユーザ時間）を出力
?>
```

■mixed highlight_file(string filename [,bool return])　[エイリアス]how_source()

F T

ファイルの内容を使用してPHP構文をハイライト表示したHTML出力を行ないます。*5

■mixed highlight_string(string str [, bool return])

F T

文字列の内容を用いてPHP構文をハイライト表示したHTML出力を行ないます。*5

■bool phpinfo([int what])

T

現在のPHPの状態に関する情報（モジュール情報など）をHTML形式で出力します。**表3-31**に示すオプションを使用可能です。

```
<?php
 phpinfo(INFO_MODULES); // ロードされているモジュール名を表示
?>
```

表3-31　phpinfo()のオプション定数

定数	値	出力
INFO_GENERAL	1	configureオプション、実行環境（Webサーバ、システム）など
INFO_CREDITS	2	PHP4の開発者名など
IINFO_CONFIGURATION	4	PHPディレクティブのローカル/マスター値
INFO_MODULES	8	ロードされているモジュール名と設定値
INFO_ENVIRONMENT	16	環境変数
INFO_VARIABLES	32	環境変数、ユーザ入力（POST/GET/Cookie）変数、サーバ変数
INFO_LICENSE	64	PHPライセンス
INFO_ALL	-1	上記すべて（デフォルト）

*5
returnにTRUEを指定した場合、出力する代わりに文字列として返します。

■string phpversion(void)

現在動作中のPHPパーサのバージョンを表す文字列を返します。*6

■void putenv(string str)

環境変数を設定します。この設定はカレントのスクリプトを実行しているあいだのみ有効です。

*6 定数PHP_VERSIONからもバージョンを得ることができます。

```
<?php
 putenv("LANG=ja");
?>
```

■int version_compare(string ver1, string ver2 [, string op])

二つのバージョン文字列を比較し、最初の文字列が2番目よりも古い場合に-1、同じ場合に0、新しい場合に1を返します。

```
<?php
 print version_compare("4.2.2", "4.1.2")."¥n"; // 1を出力
?>
```

4.4 関数実行処理用の関数

■mixed func_get_arg(int i)

ユーザ定義関数のi番目（最初の引数が0）の引数を返します。引数の数を可変とする際に使用します。

```
<?php
 function foo() {
   for ($i=0; $i< func_num_args(); $i++){
     print func_get_arg($i); // 引数を表示
   }
 }

 foo(1, 2, 3); // 123を表示
?>
```

array func_get_args(void)

ユーザ定義関数の引数を配列として返します。

```
<?php
 function foo() {
  $args = func_get_args();
  print_r($args);
 }

 foo(1, 2, 3); // Array([0] => 1 [1] => 2 [2] => 3) を表示
?>
```

int func_num_args(void)

ユーザ定義関数の引数の数を返します。

bool function_exists(string fun)

関数funが存在するかどうかを確認します。関数が存在する場合にTRUE、存在しない場合にはFALSEを返します。

```
<?php
 if (function_exists('pg_connect')) {
   $conn = pg_connect("dbname=foo");
 } else {
   print "PostgreSQLはサポートされていません。";
 }
?>
```

array get_defined_functions()

定義されている関数の名前を二次元配列として返します。内部関数は配列$a["internal"]、ユーザ定義関数は$a["user"]として返されます。

```
<?php
 function myfun(){
   return TRUE;
 }

 $fun = get_defined_functions();
 print_r($fun["user"]); // ユーザ定義関数を表示
 print_r($fun["internal"]); // 内部関数を表示
?>
```

int register_shutdown_function(string func)

シャットダウン時に実行される関数を定義します。

```
<?php
 function fun() {
   print "Bye!";
 }

 register_shutdown_function('fun'); // シャットダウン関数を登録
 exit; // シャットダウン関数funがコールされ、"Bye!"と表示される
?>
```

void register_tick_function(string func [, mixed arg])

各tick(=N)で実行される関数を定義します。tickとはdeclareの実行中にパーサがN個の低レベル命令を実行することに発生するイベントです。

```
<?php
function foo() {
  print "tick¥n";
}

register_tick_function("foo");

declare (ticks=1) {
  for ($i = 1; $i < 5; $i++){

  }
} // "tick"を6回表示
?>
```

void unregister_tick_function(string func [, mixed arg])

各tickで実行される関数の登録を解除します。

4.5 エラー処理およびログ関数

■int assert(mixed assertion)

F

assertion式がFALSEの場合に指定された処理を行ないます。
処理の内容は、assert_options()により設定します。

```
<?php
 function assert_callback($file, $line, $code) { // コールバック関数
   print "File: $file, Line: $line, Code: $code¥n";
 }

 assert_options(ASSERT_ACTIVE, 1); // assertをアクティブに
 assert_options(ASSERT_WARNING, 0); // 警告出力を無効に
 assert_options(ASSERT_QUIET_EVAL, 1); // エラー出力を無効に
 assert_options(ASSERT_CALLBACK, 'assert_callback'); // コールバック関数を設定

 $x = 1;
 assert('$x != 1'); // FALSEとなるassersion式
 // File: assert.php, Line: 12, Code: $x != 1 を出力
?>
```

■mixed assert_options(int what [, mixed value])

表3-32に示すようなassertオプションを取得/設定します。

表3-32 assertのオプション

オプション	iniパラメータ	デフォルト値	説明
ASSERT_ACTIVE	assert_active	1	assert()の評価を有効にする
ASSERT_WARNING	assert_warning	1	assertion式がFALSEの場合にPHPの警告を発生する
ASSERT_BAIL	assert_bail	0	assertion式がFALSEの場合にPHPの実行を終了する
ASSERT_QUIET_EVAL	assert_quiet_eval	0	assertion式を評価するあいだ、通常のエラー出力を抑制する
ASSERT_CALLBACK	assert_callback	NULL	assertion式がFALSEの場合に実行されるコールバック関数*7を設定する

*7
コールバック関数 fun($file, $line, $code) となります。

■bool error_log(string message, int type [, string dest [, string extra]])

F T

エラーメッセージmessageをtypeで指定した先（表3-33）に送ります。

表3-33 error_log()ログ型式

typeの値	ログ記録先
0	設定ファイルphp.iniのerror_logディレクティブで指定した記録先（syslog、ファイルなど）
1	destパラメータで指定したメールアドレスにe-mailで送信extraパラメータがメールのヘッダとして使用される
3	destパラメータで指定したファイルに保存

```
<?php
 $con = pg_connect("host=localhost dbname=foo user=apache");
 if (!$con) { // 接続に失敗した場合
   $admin = "foo@hoge.net";
   $log = "/tmp/error_log";
   error_log("failure in database connection.", 0); // error_logの指定先に記録
   error_log("failure in database connection.", 1, $admin); // メールで通知
   error_log("failure in database connection.", 3, $log); // ファイルに記録
 }
?>
```

■int error_reporting([int level])

エラー出力レベルを設定し、元の設定値を返します。levelには表3-34の値の論理和を数値または定数で設定します（php.iniのerror_reportingの設定値がデフォルトです）。

```
<?php
 error_reporting(0); // エラー出力をすべてオフ
 error_reporting(E_ERROR | E_WARNING); // E_ERROR、E_WARNINGを指定
 error_reporting(1); // E_ERRORを指定
 error_reporting(E_ALL); // エラー出力をすべてオン
 print error_reporting(); // 現在の設定値（2047）を出力
?>
```

■bool restore_error_handler()

T

set_error_handler()により設定したユーザ定義のエラーハンドラ関数を元のハンドラ関数に戻します。

表3-34 エラー出力レベルの設定

数値	定数
1	E_ERROR
2	E_WARNING
4	E_PARSE
8	E_NOTICE
16	E_CORE_ERROR
32	E_CORE_WARNING
64	E_COMPILE_ERROR
128	E_COMPILE_ERROR
256	E_USER_ERROR
512	E_USER_WARNING
1024	E_USER_NOTICE
2047	E_ALL

■string set_error_handler(string error_handler)

F

ユーザ定義のエラーハンドラ関数を設定し、元のエラーハンドラ関数をします。ハンドラ関数の引数はエラーコードとメッセージです。また、エラーを発生したファイル名と発生行、エラー発生時のアクティブなシンボルテーブルをオプションで取得することができます。

```
<?php
 function my_handler($errno, $errstr, $errfile, $errline, $sym) { // エラーハンドラ関数
   switch ($errno){
   case E_ERROR:
   case E_CORE_ERROR:
   case E_USER_ERROR:
     print "致命的なエラー: $errstr  ($errfile の $errline行目) ¥n";
     exit -1;
     break;
   default:
     print "エラー: $errstr  ($errfile の $errline 行目) ¥n";
     print_r($sym); // エラー発生時のシンボルテーブル
     break;
   }
 }

 $old_handler = set_error_handler("my_handler"); // 定義したエラーハンドラを設定
 $a = 1/0; // 「エラー: Division by zero （... の17行目)」が出力される
?>
```

■bool trigger_error(string message [, int type])　[エイリアス] user_error()

T

ユーザレベルのエラー/警告/通知を発行します。エラーの型（デフォルト：E_USER_NOTICE）を指定できます。

```
<?php
 $num = 1; $den = 0;
 if ($den != 0){
   print $num/den;
 } else {
  trigger_error("divided by zero",E_USER_ERROR);
 }
?>
```

4.6 日付・時刻関数

■bool checkdate(int month, int day, int year)

指定した日付が有効な日付である場合にtrueを返します。dayがその月の日数の範囲内であることなどが確認されます。

```
<?php
 print checkdate(9,31,2002) ? "OK" : "NG"; // NGを出力
?>
```

■string date(string format [,int timestamp])

日付･時間をフォーマットして返します。フォーマットの指定は**表3-35**に基づきます。フォーマットとして認識されない文字はそのまま表示されます。

```
<?php
 $a = array("日","月","火","水","木","金","土");
 $w = date("w",mktime(0,0,0,1,1,2001)); // 2001年1月1日の曜日を取得
 print "21世紀の最初の曜日は{$a[$w]}曜日です。"; // 月曜日を出力
?>
```

表3-35 日付フォーマットの指定

記号	内容
a	"am"または"pm"
A	"AM"または"PM"
d	日付（2桁）。"01"～"31"
D	曜日（3文字）。"Mon"～"Sun"
F	月（英字）。"January"～"December"
h	12時間単位の時間（2桁）。"01"～"12"
H	24時間単位の時間（2桁）。"00"～"23"
g	12時間単位の時間（先頭に"0"をつけない）。"1"～"12"
G	24時間単位の時間（先頭に"0"をつけない）。"0"～"23"
i	分(2桁)。"00"～"59"
j	日付（先頭に"0"をつけない）。"1"～"31"
l	英字の曜日。"Monday"～"Sunday"
L	閏年の場合に"1"、それ以外に"0"
m	月（2桁）。"01"～"12"
n	月（先頭に"0"をつけない）。"1"～"12"
M	月（3文字）。"Jan"～"Dec"
s	秒（2桁）。"00"～"59"
S	英字形式の序数を表す接尾語。"th"や"nd"
t	指定した月の日数。"28"～"31"
U	UNIX時間（1970/1/1 からの秒数）
w	曜日（数値）。0（日曜）～6（土曜）
Y	年（4桁）。"2000"など
y	年（2桁）。"00"など
z	年間の積算日。"0"～"365"
Z	タイムゾーンのオフセット秒数。"-43200"～"43200"

■array getdate([int timestamp])

日付情報を**表3-36**に示すような連想配列として取得します。また、0番目の要素にUNIX時間が返されます。

```
<?php
 $time = getdate();
 print_r($time); // 日付情報を出力
?>
```

表3-36 getdate()関数が返す日付情報

キー	内容	キー	内容
seconds	秒	mon	月（数字）
minutes	分	year	年（数字）
hours	時間	yday	年間の積算日（数字）
mday	月単位の日付	weekday	曜日（文字列）"Friday" など
wday	曜日（数字）	month	月（文字列）"January" など

■array gettimeofday(void)

現在の時刻を連想配列として返します。キーおよび値は、sec（秒）、usec（マイクロ秒）、minuteswest（グリニッジ基準の分）、dsttime（夏時間補正）となります。

```
<?php
 $time = gettimeofday(); // 時間情報を取得
 print date("Y/m/d H:i:s",$time['sec'])."¥n"; // ローカル時間を出力
 print date("Y/m/d H:i:s",$time['sec']+$time['minuteswest']*60)."¥n"; // GMT
 print gmdate("Y/m/d H:i:s")."¥n"; // GMTを出力
?>
```

■string gmdate(string format [,int timestamp])

date()と同じですが、グリニッジ標準時（GMT）でフォーマットします。

```
<?php
 print gmdate("Y/m/d H:i:s"); // たとえば 2002/09/01 12:00:00 を出力する
?>
```

■int gmmktime(int hour, int min, int sec, int mon, int mday, int year [,int is_dst])

mkdate()と同じですが、グリニッジ標準時（GMT）を指定します。

■string gmstrftime(string format [,int timestamp])

strftime()と同じですが、グリニッジ標準時（GMT）を返します。

```
<?php
 print gmstrftime("%Y/%m/%d");  // たとえば 2002/09/01 を出力する
?>
```

array localtime([int timestamp [, bool is_assoc]])

ローカルな時刻を配列として返します。is_assocにTRUEを指定した場合は**表3-37**に示すような連想配列として返されます。

```
<?php
 $time = localtime();
 print "今日の積算日:{$time[7]}¥n";
 $time = localtime(mktime(0,0,0,9,1,2002),TRUE);
 print "2002/9/1の積算日:{$time['tm_yday']}¥n"; // 243を出力
?>
```

表3-37 localtime()関数が返す日付情報

キー	内容	キー	内容
tm_sec	秒	tm_year	1900年からの経過年
tm_min	分	tm_wday	週単位の日付
tm_hour	時間	tm_yday	年間の積算日
tm_mday	月単位の日付	tm_isdst	夏時間が有効な場合に"1"
tm_mon	月（数字）		

string microtime(void)

マイクロ秒単位の現在時間をUNIX時間基準（1970年1月1日 00:00:00）で返します。

int mktime(int hour, int min, int sec, int mon, int mday, int year [,int is_dst])

指定した日付・時間をUNIX時間として返します。夏時間の場合にis_dstを1に設定します。

```
<?php
 print date("Y/m/d", mktime(0,0,0,1,1,2002)); // 2002/01/01を表示
?>
```

string strftime(string format [,int timestamp])

ロケール設定に基づき、ローカルな時間をフォーマットします。formatには、**表3-38**に示す変換指定子を使用可能です。

```
<?php
 setlocale(LC_TIME, "ja_JP");
 print strftime("%A in Japan"); // "X曜日 in Japan"（Xは曜日）を出力
?>
```

表3-38 strftime()関数の変換指定子

指定子	返り値	指定子	返り値
%a	曜日の省略形	%r	amまたはpm表記の時間
%A	曜日	%R	時間（24時間表示）
%b	月の省略形	%S	秒
%B	月	%t	タブ文字
%c	日付と時間	%T	現在の時間（%H:%M:%S）
%C	世紀	%u	曜日（数値）1（月）～7（日）
%d	日付	%U	年間の積算週（日曜から開始）
%D	%m/%d/%yと等価	%V	ISO 8601:1988による週番号
%e	月単位の日付	%W	年間の積算週（月曜から開始）
%h	%bと同じ	%w	曜日（数値）0（日）～6（土）
%H	時間（24時間表示）	%x	時間を除いた日付
%I	時間（12時間表示）	%X	日付を除いた時間
%j	年間積算日	%y	年（2桁）"00"～"99"
%m	月	%Y	年（4桁）
%M	分	%Z	タイムゾーンまたはその省略形
%n	改行文字	%%	'%'
%p	"am"または"pm"		

■int strtotime(string time [,int timestamp])

英文形式の日付をUNIX時間に変換します。

```
<?php
 print strtotime("1 may 2001"); // 2001/5/1のUNIX時間（988642800）を出力
?>
```

■int time(void)

現在の時間をUNIX時間で返します。

```
<?php
 print date("Y/n/j H:i:s",time()); // 現在の日時を表示
?>
```

4.7 セッション関数

PHPでは、セッション機能がサポートされています。セッション機能によりHTTPリクエスト間でユーザ情報を保持することなどが可能になります。

セッションは、プログラムの先頭でsession_start()（またはsession_register()）をコールすることにより有効となります。*8

セッションデータは、セッションIDにより管理されます。セッションIDはCookie、URLパラメータ、HTMLフォームのパラメータによりクライアント/サーバ間で保持されます。

セッション機能が有効な場合、PHPはセッションIDがリクエストに含まれているかを確認し、含まれている場合には関連するセッションデータをグローバル配列変数$_SESSIONとして復元します。各セッションデータは、$_SESSIONの要素として取得または設定することができます。

セッション機能の詳細については『**PHP4 徹底攻略 実戦編**』を参照してください。

▶PHP構築オプション

`--with-mm[=DIR]` セッションデータの保存先としてmmライブラリを使用可能とします。

▶php.ini設定オプション

パラメータ	デフォルト値	説明
session.auto_start	Off	セッションを自動開始する場合にOn
session.cache_expire	180	Webブラウザのキャッシュの有効期限（単位：分）
session.cache_limiter	nocache	Webブラウザ等でのキャッシュの動作を制御するパラメータで、nocache、private、private_noexpire、publicの中から指定する
session.cookie_domain	NULL	Cookieを有効とするドメイン名
session.cookie_lifetime	Off	Cookieの有効期限（単位：秒）。Offの場合はセッションCookie（Webブラウザ終了まで）となる
session.cookie_path	/	セッションCookieを有効とするパス
session.cookie_secure	0	1（TRUE）の場合、Cookieはセキュアな接続（SSL/TLS）でのみ送信される
session.entropy_file	NULL	セッションIDを作成時に使用する乱数生成ファイル
session.entropy_length	0	ファイルからの読み込みバイト数
session.gc_maxlifetime	1440	データがゴミとみなされるまでの秒数
session.gc_probability	1	ごみ集めルーチンの起動確率
session.name	PHPSESSID	セッション名（Cookie名として使用）
session.referer_check	NULL	HTTP_REFERERに指定した文字が現れる場合のみセッションを有効とする

*8 設定ファイルphp.iniでsession.auto_startに1を指定している場合は自動的にセッションが開始されます。

▶「php.ini設定オプション」(続き)

パラメータ	デフォルト値	説明
session.save_handler	files	セッションデータの取得/保存用ハンドラ
session.save_path	/tmp	セッションデータが保存されるディレクトリ *9
session.serialize_handler	php	シリアル化用のハンドラ
session.use_cookies	On	Cookieを使用する場合にOn
session.use_trans_sid	1	1の場合、セッションIDを自動的に付加する *10
url_rewriter.tag	"a=href,area=href,frame=src,input=src,form=fakeentry"	URL Rewritingで書き変えるタグを指定

*9 デフォルトの設定(/tmp)の場合、セッションデータをほかのユーザに見られる可能性があります。

*10 --enable-trans-sidをつけてPHPをコンパイルした場合のみ動作します。

セッションのパラメータの一部は、php.iniで設定する以外に**表3-39**の関数によりPHPスクリプトで指定可能です。

表3-39 セッション機能のパラメータを設定または取得する関数 *11

関数名	設定(または取得)するパラメータ
int session_cache_expire([int expire])	session.cache_expire
string session_cache_limiter([string limiter])	session.cache_limiter
string session_module_name([string name])	session.save_handler
string session_name([string name])	session.name
string session_save_path([string path])	session.save_path
void session_set_cookie_params(int lifetime [, string path [, string domain [, bool secure]]])	session.cookie_lifetime,session.cookie_path, session.cookie_domain,session.cookie_secure

*11 引数を省略すると現在の設定値を返します。設定した値は実行中のスクリプトでのみ有効です。

*12 session_start()のサンプルを参照してください。

■bool session_decode(string data)

シリアル化されたセッションデータを復元し、セッションに保存する変数を設定します。クラス変数を復元する場合、事前にクラス定義を行なう必要があります。

```
<?php
 class foo { // クラス定義
   var $s = "test";
   function method() {print $this->cnt++;}
 }

 session_start(); // セッション開始
 session_decode('v|O:3:"foo":1:{s:1:"s";s:4:"test";}'); // セッションデータをデコード
 print $_SESSION['v']->s; // 出力："test"
?>
```

■bool session_destroy()

F T

セッションに登録されたデータをすべて破棄します。

■string session_encode()

カレントのセッションの内容をシリアル化（エンコード）した文字列を返します。

```
<?php
 class foo {
   var $s = "test";
   function method() {print $this->cnt++;}
 }

 $v = new foo; // クラスfooのインスタンスを作成
 session_register("v"); // セッション変数として$vを登録
 print session_encode(); // v|O:3:"foo":1:{s:1:"s";s:4:"test";} が出力される
?>
```

■array session_get_cookie_params()

セッションID保持用のCookieに関するパラメータを**表3-40**に示す連想配列として返します。

表3-40　session_get_cookie_params()の返り値

要素タグ	取得するパラメータ
"lifetime"	session.cookie_lifetime
"path"	session.cookie_path
"domain"	session.cookie_domain
"secure"	session.cookie_secure

■string session_id([string id])

セッションIDを返します。idを指定した場合、セッションIDを更新します。

```
<?php
 $id = md5(uniqid("secret")); // セッションIDを生成
 print $id."¥n";
 session_id($id); // セッションIDを設定
 session_start(); // セッションを開始
 print session_id(); // セッションIDを出力
?>
```

■bool session_is_registered(string varname)

変数が現在のセッションに登録されている場合にTRUEを返します。[*12]

■bool session_register(mixed name [, mixed ...])

セッションに変数を登録します。引数の数は可変です。

■void session_set_save_handler(string open, string close, string read, string write, string destroy, string gc)

セッション機能のハンドラとしてユーザ関数を設定します。open、close、read、write、destroy、gcの処理を行なう関数を定義し、関数名を指定します。RDBMSなどによりセッションデータを保持する場合に使用します。

■bool session_start()

セッションデータを作成し、TRUEを返します。GETまたはクッキーにより指定されたセッションIDに基づき現在のセッションを復帰します。セッションIDを受け渡す手段としてCookieを使用している場合、session_start()をコールする前に出力を行なうことはできません。

```
<?php // 簡易カウンタ
 session_start(); // セッションを開始
 if(!session_is_registered('cnt')){
   $_SESSION['cnt'] = 0;
 }
 print $_SESSION['cnt']++; // カウンタを出力したあとで加算
?>
```

■bool session_unregister(string name)

セッションからセッション変数を削除します。

■void session_unset()

現在登録されているすべてのセッション変数を解放します。

■void session_write_close()

セッションデータを書き込んで、セッションを終了します。

Chapter-5 WWW/ネットワーク関連

5.1 HTTP関連の関数

■object get_browser([string user_agent])

ユーザのブラウザの機能を検索し、オブジェクトとして返します。設定ファイルのbrowscapにてbrowscap.ini（別途入手する必要あり）の位置を指定する必要があります。user_agentを指定した場合は、その文字列について機能を検索します（デフォルトは$HTTP_USER_AGENTを使用）。

```
<?php
 $obj = get_browser("Mozilla/4.0");
 print_r($obj); // Netscape 4.0に関する情報を出力
?>
```

■void header(string string [, bool replace])

HTTPヘッダを送信します。オプションreplaceにより同じHTTPヘッダを複数指定した場合に、あとの値で置換する（TRUE：デフォルト）か、二重に指定する（FALSE）かを指定できます。この関数は、ほかのデータを出力する前にコールする必要があります。

```
<?php
 header("Location: http://www.php.net");   // ブラウザをリダイレクト
 header("Pragma: no-cache");               // HTTP/1.0でキャッシュ無効を指定
?>
```

■bool headers_sent()

HTTPヘッダを送信済みの場合にTRUEを返します。

```
<?php
 setcookie("foo","1234"); // Cookieを設定
 ob_end_flush(); // 出力バッファをフラッシュ
 print headers_sent() ? "送信済み" : "送信前"; // 「送信済み」を出力
 if (isset($_COOKIE['foo'])){
   print $_COOKIE['foo']; // 出力:1234
 }
?>
```

■bool setcookie(string name [, string value [, int expire [, string path [, string domain [, bool secure]]]]])

F T

Cookie nameの値を設定します。CookieはHTTPヘッダにより送信され、セッション管理などに使用されます。nameのみを指定した場合、その名前のCookieがクライアントから削除されます。name以外の引数はオプションで**表3-41**のように設定します。

表3-41 関数setcookie()のオプション*1

引数	内容
value	Cookieの設定値。自動的にURLエンコードされる
expire	有効期間をUNIX時間（整数、単位：秒）で指定する
path	Cookieを有効とするパス
domain	Cookieを有効とするドメイン
secure	セキュアな接続（SSL/TLS）によりCookieを送信する場合にTRUEとする

```
<?php
 // 適用するパスおよびドメインを指定、有効期間は1分
 setcookie("user","taro",time()+60,"/~foo/",".phpserv.jp");
 if (isset($_COOKIE['user'])){
   print $_COOKIE['user']; // taroを出力
 }
?>
```

*1 オプションの引数は "" または 0により指定を省略できます。

5.2 URL/エンコード関数

■string base64_encode(string data)

文字列をBASE64でエンコードして返します。

```
<?php
 $a = base64_encode("日本語"); // BASE64でエンコード
 print $a."¥n"; // 出力: xvzL3Ljs
 print base64_decode($a)."¥n"; // 出力: 日本語
?>
```

■string base64_decode(string encoded_data)

BASE64エンコードされた文字列をデコードし、文字列として返します。結果がバイナリデータとなることもあります。

■array parse_url(string url)

URLを要素（scheme、host、port、user、pass、path、query、fragment）に分割し、連想配列として返します。

```
<?php
 $url = "ftp://taro:secret@ftp.phpserv.jp/pub/test.txt";
 $a = parse_url($url); // URL文字列をパース
 print_r($a); // array("scheme"=>"ftp","host"=>"ftp.phpserv.jp",
              //  "user"=>"taro", "pass"=>"secret", "path"=>"/pub/test.txt") を出力
?>
```

■string urldecode(string str)

URLエンコードされた文字列をデコードし、文字列として返します。

■string urlencode(string str)

-、_、. 以外の非英数文字をURLエンコードし、文字列として返します。rawurlencode()と異なり、空白を%20ではなく + に置換します。

```
<?php
 $url = "test.php?foo=" . urlencode("日本語 Japan");
 print "<A HREF=¥"$url¥">リンク</A>";
 //  「<A HREF="test.php?foo=%C6%FC%CB%DC%B8%EC+Japan">リンク</A>」を出力
?>
```

■string rawurldecode(string str)

URLエンコードされた文字列strをデコードして結果を返します。

```
<?php
 $url = 'name%20taro%40tokyo';
 print rawurldecode($url);  // 'name taro@tokyo' を出力
?>
```

■string rawurlencode(string str)

文字列をRFC1738に基づきURLエンコードして返します。日本語などを含むパラメータをURLに指定する場合には、かならずURLエンコードする必要があります。

```
<?php
 $url = "test.php?foo=" . rawurlencode("日本語 Japan");
 print "<A HREF=¥"$url¥">リンク</A>";
 //  「<A HREF="test.php?foo=%C6%FC%CB%DC%B8%EC%20Japan">リンク</A>」を出力
?>
```

5.3 メール関連の関数

PHPスクリプトからメールを送信することが可能です。日本語メールを送信する場合は、mb_send_mail()を使用します。

▶php.ini設定オプション

パラメータ	デフォルト値	説明
SMTP	localhost	SMTPサーバのアドレス（Win32環境のみ）
sendmail_from	me@localhost.com	送信者アドレス（Win32環境のみ）
sendmail_path	sendmail -t -i	sendmailコマンド（UNIX環境のみ）

■bool mail(string to, string subject, string message [, string headers [, string params]])

F T

メッセージを宛先to（複数アドレス指定可能）にメールします。headersによりヘッダ文字列、paramsによりsendmailのパラメータを追加指定できます。

```
<?php
 $subject = "this is subject.";
 $message = "Hello!";
 $headers = "From: webmaster@phpserv.jp";
 $opt = "-fnobody@localhost";
 mail("foo@phpserv.jp", $subject, $message); // メール送信
 mail("foo@phpserv.jp", $subject, $message, $headers,$opt); // メール送信（追加ヘッダつき）
?>
```

■int ezmlm_hash(string addr)

メーリングリストezmlm用のハッシュ値を計算します。

5.4 ネットワーク関数

■bool checkdnsrr(string host [, string type])

ホストのDNSレコードを検索し、見つかった場合にTRUE、それ以外の場合にFALSEを返します。typeにはレコードの種類（A、MX、NS、SOA、PTR、CNAME、ANY）を指定します（デフォルトはMX）。

```
<?php
 print checkdnsrr("www.php.net") === TRUE ? "OK":"NG";
?>
```

■int closelog(void)

システムログへの接続を閉じます。通常、closelog()をコールする必要はありません。

resource fsockopen(string host, int port [, int &errno [, string &errstr [, float timeout]]])

F

ホスト名とポートを指定してインターネットドメイン（AF_INET）またはUNIXドメイン（AF_UNIX）のソケットをオープンし、ポインタを返します。引数は**表3-42**のように設定します。

*2 データグラム型の場合はudp://を先頭に付加（デフォルトはストリーム型）する。

*3 小数点以下（マイクロ秒）も指定可能です。

*4 optionおよびfacilityについては、システムのopenlog()のマニュアルを参照してください。

表3-42 関数fsockopen()の設定オプション

引数	内容
host	ホスト名（UNIXドメインの場合はソケットのパス）*2
port	ポート番号（UNIXドメインの場合は0）
errno	システムコールconnect()のエラー番号
errstr	システムコールconnect()のエラーメッセージ
timeout	処理のタイムアウトまでの時間（秒）*3

```
<?php
 // localhostのポート80にソケット接続をオープン
 $fp = fsockopen("localhost", 80, $errno, $errstr);
 if(!$fp) { // 接続に失敗した場合
   print "$errstr ($errno)¥n"; // エラーの内容を表示
 } else {
   fputs($fp,"GET / HTTP/1.0¥n¥n");   // HTTPによりデータを取得
   while(!feof($fp)) { // ページを表示
     print fgets($fp,128);
   }
   fclose($fp);
 }
?>
```

string gethostbyaddr(string ip)

IPアドレスipを有するホスト名（取得できない場合はip）を返します。

```
<?php
 print gethostbyaddr("127.0.0.1"); // ローカルホストの名前を出力
?>
```

string gethostbyname(string host)

ホストのIPアドレスを返します。

■array gethostbynamel(string host)

ホストのIPアドレスのリストを配列として返します。

```
<?php
 $a = gethostbynamel("www.microsoft.com"); // サーバ名からIPアドレスのリストを取得
 print_r($a); // IPアドレスのリストを出力
?>
```

■bool getmxrr(string host, array &mxhosts [, array &weight])

ホストをDNSのMXレコードから検索し、見つかった場合にTRUE、それ以外の場合にFALSEを返します。検索結果が配列mxhostsに、重み情報が配列weightに代入されます。

```
<?php
 $host = "mail.foo.net";
 getmxrr($host, $hosts, $weight) or die("mx records not found.");
 for ($i=0; $i<count($hosts);$i++){
   print $hosts[$i] . ":" . $weight[$i] . "¥n";
 }
?>
```

■bool openlog(string ident, int option, int facility)

T

システムログへの接続をオープンします。文字列identが各メッセージに追加されます。通常はsyslog()をコールした際に自動的にオープンされるため不要です。*4

■resource pfsockopen(string host, int port [, int &errno [, string &errstr [, int timeout]]])

F

持続的接続としてソケットをオープンします。オプションおよび動作はfsockopen()と同じです。

■array socket_get_status(resource sid)

ソケットのステータスを連想配列（表3-43）として取得します。

表3-43 socket_get_status()から返される連想配列の要素

要素名	型	説明
timed_out	bool	タイムアウトした場合に1
blocked	bool	ソケットがブロッキングモードの場合に1
eof	bool	EOFに達した場合に1
unread_bytes	int	ソケットバッファの残りのバイト数

```
<?php
 $fp = fsockopen("localhost", 80) or die("socket open failure.");
 fputs($fp,"GET / HTTP/1.0¥n¥n"); // HTTPによりデータを取得
 socket_set_timeout($fp, 2);
 print fread($fp, 4096); // データを読み込んで出力
 print_r(socket_get_status($fp)); // ステータスを表示
 fclose($fp);
?>
```

■bool socket_set_blocking(resource sid, int mode)

F T

ソケットのブロックモードを設定します。modeには0（非ブロックモード）または1（ブロックモード）を指定します。

■bool socket_set_timeout(resource sid, int seconds [, int microsecs])

F T

ソケットのタイムアウト時間を秒+マイクロ秒に設定します。

■bool syslog(int priority, string message)

T

システムログにメッセージを出力します。メッセージの中で%mはerrnoの値に置換されます。priorityは**表3-44**に示す値を指定します。

```
<?php
 openlog("userlog", LOG_PID | LOG_PERROR, LOG_LOCAL0);
 syslog(LOG_WARNING,"Warning"); // 出力：userlog[15495]: Warning
?>
```

表3-44 syslog()のpriorityパラメータの設定

定数	説明
LOG_EMERG	システムは使用不可
LOG_ALERT	緊急性のある警告
LOG_CRIT	致命的なエラー
LOG_ERR	エラー
LOG_WARNING	警告
LOG_NOTICE	通知
LOG_INFO	情報を記録するメッセージ
LOG_DEBUG	デバッグ用のメッセージ

5.5 出力バッファリング関連の関数

■bool ob_clean()

F T

出力用バッファの内容をクリアします。

■string ob_end_clean()

F T

出力用バッファの内容を破棄し、出力のバッファリングをオフにします。

■bool ob_end_flush()

F T

出力用バッファの内容をフラッシュしたあとで破棄します。

■bool ob_flush()

F T

出力用バッファの内容をフラッシュします。

■string ob_get_contents()

出力用バッファの内容を返します。出力内容を変数に取得したい場合に有用です。出力バッファリングが有効でない場合には、FALSEを返します。

```
<?php
 ob_start(); // 出力バッファリングを開始
 print "バッファリングされます。"; // 出力
 $info = ob_get_contents(); // 出力バッファの内容を取得
 ob_end_clean(); // 出力バッファリングを終了
 print "バッファ：".$info; // "バッファ：バッファリングされます。"を出力
?>
```

■int ob_get_length()

F

出力バッファの長さを返します。

```
<?php
 ob_start(); // 出力バッファリングを開始
 print "This is string."; // 出力（15バイト）
 $length = ob_get_length(); // バッファ内にあるデータのバイト数を取得
 ob_end_clean(); // 出力バッファリングを終了
 print $length; // 15を出力
?>
```

■int ob_get_level()

出力バッファのネストレベルを返します。

■array ob_get_status([bool full_status])

出力バッファに関する情報を連想配列として返します。

```
<?php
 ob_start("mb_output_handler"); // 出力バッファリングを開始
 $info = ob_get_status(); // 出力バッファのステータスを取得
 ob_end_clean(); // 出力バッファリングを終了
  print_r($info); // array("level"=>2, "type"=>1, "status"=>0, "name"=>mb_
output_handler, "del"=>1)を出力する
?>
```

■void ob_implicit_flush([int flag])

バッファの自動フラッシュをOn（デフォルト）またはOffに設定します。自動フラッシュをOnにすると出力のバッファリングは無効となり、現在の出力バッファの内容が出力されます。

■void ob_start([string callback [, int chunk_size [, bool erase]]])

出力バッファリングを有効にします。オプションで出力バッファリングのコールバック関数を指定することが可能です。

```
<?php
 ob_start("mb_output_handler"); // ハンドラ関数を指定してバッファリング開始
 print "日本語"; // mbstring.http_output で指定した文字コードで出力
?>
```

Hypertext Preprocessor

PHP関数

Chapter-6 数値処理/乱数関連

6.1 数学関数

PHPではC言語などでサポートされる数学関数を使用可能です。サポートされる基本的な関数を**表3-45**に示します。また、**表3-46**に示す任意精度関数（BC関数）も使用可能です。このほか、乱数、基底の変換などの関数がサポートされています。

表3-45 基本的な数値関数

関数名	返り値
mixed abs(mixed arg)	絶対値 *1
float acos(float arg)	逆余弦
float acosh(float arg)	逆双曲線余弦
float asin(float arg)	逆正弦
float asinh(float arg)	逆双曲線正弦
float atan(float arg)	逆正接
float atan2(float y, float x)	y/xの4象元アークタンジェント（[-π,π]）
float atanh(float arg)	逆双曲線正接
float ceil(float arg)	arg<nとなる最小の整数
float cos(float arg)	余弦
float cosh(float arg)	双曲線余弦
float deg2rad(float arg)	degをradに変換した値
float exp(float arg)	eのarg乗
float expm1(float arg)	exp(arg)-1
float floor(float arg)	arg>=nとなる最大の整数
float hypot(float x, float y)	sqrt(x*x+y*y)
float log(float arg)	自然対数
float log10(float arg)	常用対数
float log1p(float arg)	og(1+arg)
float pi(void)	円周率の近似値
mixed pow(mixed base, mixed exp)	baseのexp乗
float rad2deg(float arg)	radをdegに変換した値
float sin(float arg)	正弦
float sinh(float arg)	双曲線正弦
float sqrt(float arg)	平方根
float tan(float arg)	正接
float tanh(float arg)	双曲線正接

*1
入力がfloatの場合はfloat、それ以外はint。

*2
sclにより結果の小数点以下の精度の桁数（デフォルト：0）を指定可能

表3-46 任意精度関数（--enable-bcmathをPHP構築オプションに指定した場合のみ使用可能）*2

関数名	返り値
string bcadd(string a,string b [,int scl])	a+b
string bccomp(string a,string b [,int scl])	a>b：1,a=b：0,a<b：-1
string bcdiv(string a,string b [,int scl])	a/b
string bcmod(string a,string b)	aをbで割った余り
string bcmul(string a,string b [,int scl])	a*b
string bcpow(string a,string b [,int scl])	aのb乗
string bcscale(int scl)	精度のデフォルト値をscl桁に設定
string bcsqrt(string a [,int scl])	平方根
string bcsub(string a,string b [,int scl])	a-b

```
<?php // 任意精度計算のサンプル
 $a = 1.23; $b = pi();
 print bcadd($a,$b)."¥n"; // 4を出力
 print bcadd($a,$b,5)."¥n"; // 4.37159を出力
 bcscale(3); // 桁数を3桁に変更
 print bcadd($a,$b)."¥n"; // 4.371を出力
?>
```

■string base_convert(string num, int from, int to)

数値numの基底をfromからtoに変換し、文字列として返します。fromおよびtoはともに2～36の範囲内でなければなりません。10以上の基底を表す数は、文字a～zで表されます。基底の変換には、**表3-47**に示す関数も使用可能です。

表3-47 基底変換用関数

関数名	動作
int bindec(string num)	2進数を10進数に変換
string decbin(int num)	10進数を2進数に変換
string dechex(int num)	10進数を16進数に変換
string decoct(int num)	10進数を8進数に変換
int hexdec(string num)	16進数を10進数に変換
int octdec(string num)	8進数を10進数に変換

```
<?php
 print base_convert("ff",16,10)."¥n"; // 255を出力
 print hexdec("ff")."¥n"; // 255を出力
 print dechex(255)."¥n"; // ffを出力
?>
```

■bool is_finite(float num)

floatの範囲内にある場合にTRUEを返します。

■bool is_infinite(float num)

正または負の無限大の場合にTRUEを返します。

■bool is_nan(float num)

NaNの場合にTRUEを返します。

■mixed max(mixed arg1, mixed arg2, ... ,mixed argn)

引数（スカラーまたは配列）の最大値を返します。

■mixed min(mixed arg1, mixed arg2, ... ,mixed argn)

引数（スカラーまたは配列）の最小値を返します。

■string number_format(float num [, int dec[, string point[, string sep]]])

数値を1000の位ごとにフォーマットした文字列を返します。デフォルトでは小数点以下は無視されます。

decにより小数点以下の表示桁数、pointにより小数点文字（デフォルトは.）、sepにより1000単位の区切り文字（デフォルトは,）を指定可能です。

```
<?php
 $num = 12345.678;
 print number_format($num)."¥n"; // "12,346" を出力
 print number_format($num,2)."¥n"; // "12,345.68" を出力
 print number_format($num,2,"#","-")."¥n"; // "12-345#68" を出力
?>
```

■float round(float val [, int precision])

valを指定した精度で整数値に丸めた数値を返します。丸め誤差によりかならずしも四捨五入とはならないことに注意してください。

```
<?php
 print round(11.5)."¥n"; // 12を出力
 print round(pi(),4)."¥n"; // 3.1416を出力
?>
```

6.2 乱数関連の関数

PHPではシステムコールrand()を使用する乱数（**表3-48**）により高品質なMT乱数（**表3-49**）をサポートしています。

Mersenne Twister（MT）法による乱数は、通常のライブラリ関数に実装されている乱数発生器よりも高品質で高速です。

Mersenne Twisterのホームページは以下の場所にあります。

http://www.math.keio.ac.jp/~matumoto/emt.html

表3-48 乱数をサポートする関数

関数名	動作
int getrandmax(void)	乱数の最大値を返す
int rand([int min][,int max])	[min,max]の範囲の乱数を返す*3
void srand(int seed)	乱数をシードseedで初期化する

*3 デフォルトは[0,RAND_MAX]。

*4 デフォルトは[0,RAND_MAX]

```
<?php
 // 現在の時間をマイクロ秒で表した値をシードとして初期化する
 srand((double)microtime()*1000000);
 $randval = rand(); // 乱数を求める
?>
```

表3-49 MT乱数をサポートする関数

関数名	動作
int mt_getrandmax(void)	MT乱数の最大値を返す
int mt_rand([int min][,int max])	[min,max]の範囲のMT乱数を返す*4
void mt_srand(int seed)	MT乱数をシードseedで初期化する

```
<?php
 // マイクロ秒単位の現在の時間をシードとして初期化します。
 mt_srand((double)microtime()*1000000);
 $randval = mt_rand(); // 乱数を発生します。
?>
```

Hypertext Preprocessor

PHP関数

Chapter-7 マルチバイト文字(日本語)関連

7.1 マルチバイト関数

PHPでは日本語を含むマルチバイト文字列を処理するための機能がサポートされています。この機能では、日本語の文字コード変換や文字列処理関数などがサポートされています。

また、ユーザからのHTTP入力の文字コードを自動的に検出し、内部文字コードに変換することが可能です。*1

主な文字コードとしては以下がサポートされています。

ASCII、EUC-JP、SJIS、ISO-2022-JP(JIS)、ISO-8859-1、UTF-8、UTF-16、UCS-2、SJIS-win、eucJP-win

▶PHP構築オプション

--enable-mbstring	マルチバイト関数を有効にします。
--enable-mbstr-enc-trans	ユーザ入力の文字コード自動検出/変換機能を有効にします。

▶php.ini設定オプション

パラメータ	デフォルト値	説明
mbstring.detect_order	auto *2	(HTTP入力以外の)文字コードの検出順
mbstring.func_overload	0	PHP関数をマルチバイト対応関数で置換(**表7-1参照**) *3
mbstring.internal_encoding	EUC-JP	PHP内部で使用する文字コード
mbstring.http_output	pass	出力の文字コード
mbstring.http_input	NULL	HTTP入力(POST/GET/Cookie)文字コードの検出順 *4
mbstring.substitute_character	'?'	文字コード変換ができない場合に置換する文字 *5

パラメータの設定の取得または変更は、**表3-50**に示す関数で行なうことも可能です。

*1 --enable-mbstr-enc-trans をconfigureに指定してPHPを構築した場合。

*2 autoは、ASCII、JIS、UTF-8、EUC-JP、SJISの順に自動検出することを意味します。

*3 設定値(1、2、4)の論理和で指定します。

*4 autoの場合はASCII、JIS、UTF-8、EUC-JP、SJISの順に自動検出、passの場合は無変換。

*5 置換文字のUnicode値、"none"(出力なし)、"long"(16進数の文字コード)のいずれかを指定します。

表3-50 mbstring.func_overloadオプションにより置換される関数

設定値	置換される関数	置換後の関数
0	なし	なし
1	mail()	mb_send_mail()
2	strlen(),strpos(),strrpos substr()	mb_strlen(),mb_strpos(),mb_strrpos mb_substr()
4	ereg(),eregi(),ereg_replace() eregi_replace(),split()	mb_ereg(),mb_eregi(),mb_ereg_replace() mb_eregi_replace(),mb_split()

表3-51 マルチバイト関数のパラメータを設定または取得する関数[*6]

関数名	設定（または取得）するパラメータ
rray mb_detect_order([mixed encoding-list])	mbstring.detect_order
string mb_http_input([string type])	mbstring.http_input
mixed mb_http_output([string encoding])	mbstring.http_output
string mb_internal_encoding([string encoding])	mbstring.internal_encoding
string mb_language([string language])	mbstring.language [*7]
mixed mb_substitute_characters([mixed c])	mbstring.substitute_character

▶ **注意**

・文字コードの自動検出が常にうまくいくわけではありません。特にHTMLフォームからの入力を検出する場合、ユーザ入力の長さが短いと自動検出に失敗することがあります。この場合、HTMLフォーム内に以下のような日本語文字列を含むhidden要素を指定する必要があります。[*8]

```
<input type=hidden value="適当な日本語文字列">
```

・eucJP-win、SJIS-winは、EUC-JP、SJISに対してベンダ拡張文字領域およびユーザ定義文字領域（CP932相当）の拡張を行なった文字コードです。iモードの絵文字などのSJIS拡張文字を使用する場合には、SJIS-winを指定します。[*8]

*6 引数を省略すると現在の設定値を返します。設定した値は実行中のスクリプトでのみ有効です。

*7 PHP 4.3.0以降でサポートされています。

*8 詳細については『PHP4徹底攻略 実戦編』を参照してください。

■string mb_convert_encoding(string str, string to_encoding [, string from_encoding])

F

文字列をfrom_encoding（省略時には内部文字コード）からto_encodingに変換し、文字列として返します。文字コードとしては、EUC-JP、SJIS、JIS、UTF-8、auto（自動検出）を指定します。

```
<?php
 print mb_convert_encoding("日本語","SJIS","auto");
?>
```

■string mb_convert_kana(string str [,string option[, string code]])

F

文字列strを全角から半角、または半角から全角に変換します。optionに**表3-52**に示す変換モードを指定可能です（デフォルトは "KV"）。

```
<?php
 print mb_convert_kana("ﾎﾟﾙﾄｶﾞﾙ","KV")."¥n";  // 出力：ポルトガル
 print mb_convert_kana("渋谷区神南1-2-3","A"); // 出力：渋谷区神南１－２－３
?>
```

表3-52　変換モード

設定値	動作モード
r	英字を全角から半角に変換
R	英字を半角から全角に変換
n	数字を全角から半角に変換
N	数字を半角から全角に変換
a	英数字(21H～7EH)を全角から半角に変換
A	英数字(21H～7EH)を半角から全角に変換
k	全角カタカナを半角カナに変換
K	半角カナを全角カタカナに変換
h	全角ひらがなを半角カナに変換
H	半角カナを全角ひらがなに変換
c	全角カタカナを全角ひらがなに変換
C	全角ひらがなを全角カタカナに変換
V	濁音記号を前の文字と組み合わせて1文字に変換（K、Hと組み合わせて使用）

■string mb_convert_variables(string to_encoding , mixed from_encoding, mixed vars [, mixed vars ...])

F

変数の文字コードを変換し、変換前の文字コードを返します。

```
<?php
 $a = array("日本語","文字列");
 $b = "日本語";
 $c = "文字列";
 mb_convert_variables("SJIS","auto",$a); // 配列変数の文字コードを変換
 $old = mb_convert_variables("SJIS","auto",$b,$c); // 文字列変数の文字コードを変換
 print $old; // 変換前の文字コード（EUC-JP）を出力
 print_r($a); // SJISに変換された配列データを出力
 print $b.$c."¥n"; // SJISに変換された文字列データを出力
?>
```

■string mb_decode_mimeheader(string str)

F

MIMEエンコードされた文字列をデコードして返します。

■string mb_decode_numericentity(string str, array convmap [, string encoding])

数値エンティティでエンコードされた文字列をデコードして返します。変換用マップを配列として指定します。変換用マップは以下の形式の配列で指定します。

```
array(開始コード1,終了コード1,オフセット1,マスク1,
      開始コード2,終了コード2,オフセット2,マスク2, ...)
```

```
<?php
 $str = "&#63647;のち&#63648;";  // iモードの絵文字を含む文字列
 // iモード絵文字用変換テーブル
  $convmap  =  array(0xe63e,0xe69b,0x1261,0xffff,  0xe69c,0xe6a5,0x12a4,0xffff,
0xe6ce,0xe6da,0x12a4,0xffff,  0xe6db,0xe70b,0x12a5,0xffff);
 $str_new = mb_decode_numericentity($str, $convmap, "sjis-win"); // 文字列に変換
 print $str_new;  // 出力：[image] のち[image]
?>
```

■string mb_detect_encoding(string str [, mixed encoding-list])

文字列の文字コードを検出して返します。

```
<?php
 $str = "日本語文字列";
 $enc = mb_detect_encoding($str, "auto"); // 自動検出
 print $enc."¥n"; // 出力:EUC-JP
 $enc = mb_detect_encoding($str, "ASCII,eucJP-win,SJIS-win"); // 文字列で候補指定
 print $enc."¥n"; // 出力:eucJP-win
 $enc = mb_detect_encoding($str, array("ASCII","eucJP-win","SJIS-win")); // 配列で候補指定
 print $enc."¥n"; // 出力:eucJP-win
?>
```

■array mb_detect_order([mixed encoding-list])

(HTTP入力以外の)文字コードの検出順を設定します。引数を省略した場合、現在の設定値を返します。

```
<?php
 $a = mb_detect_order();  print_r($a); // デフォルトの設定値を出力
 mb_detect_order("ASCII,ISO-2022-JP,eucJP-win,SJIS-win"); // 文字列で指定
 $a = mb_detect_order();  print_r($a); // 配列("ASCII","ISO-2022-JP","eucJP-
win","SJIS-win")を出力
 mb_detect_order(array("ASCII","JIS","UTF-8","SJIS-win"));  // 配列で指定
 $a = mb_detect_order();  print_r($a); // 配列（"ASCII","JIS","UTF-8","SJIS-
win"）を出力
?>
```

■string mb_encode_mimeheader(string str [, string charset [, string transfer-encoding [, string linefeed]]])

指定した文字列をMIMEヘッダ用の文字列としてエンコードして返します。charsetに使用する文字セット（デフォルト：ISO-2022-JP）、transfer-encodingに"B"：Base64（デフォルト）または"Q"：Quoted-Printableを指定します。また、linefeedには改行文字（デフォルト：CRLF）を指定します。

```
<?php
 $str = mb_encode_mimeheader("鈴木"); // MIMEヘッダ変換（ISO-2022-JP,Base64）
 print $str."¥n"; // 出力： =?ISO-2022-JP?B?GyRCTmtMWhsoQg==?=
 $str = mb_encode_mimeheader("鈴木", "UTF-7", "Q"); // MIMEヘッダ変換（UTF-
7,Quoted-Printable）
 print $str."¥n"; // 出力： =?UTF-7?Q?+kjRnKA-?=
 print mb_decode_mimeheader($str)."¥n"; // 出力： 鈴木
?>
```

■string mb_encode_numericentity(string str, array convmap[, string encoding])

文字を数値エンティティにエンコードします。変換用マップの形式は、mb_decode_numericentity()と同じです。

```
<?php
 $str = "  のち  ";  // iモードの絵文字を含む文字列
 // iモード絵文字用変換テーブル
 $convmap = array(0xe63e,0xe69b,0x1261,0xffff, 0xe69c,0xe6a5,0x12a4,0xffff,
                  0xe6ce,0xe6da,0x12a4,0xffff, 0xe6db,0xe70b,0x12a5,0xffff);
 $str_new = mb_encode_numericentity($str, $convmap, "sjis-win"); // 数値エンティティに変換
 print $str_new;  // 出力：&#63647;のち&#63648;
?>
```

■mixed mb_get_info(string type)

マルチバイト関数モジュールの設定値を取得します。typeには取得するオプションを "all"、"http_output"、"http_input"、"internal_encoding"、"func_overload" を指定します。"all" を指定した場合にはオプション名をキーとする配列が返されます。

```
<?php
 $a = mb_get_info("all"); print_r($a); // カレントの設定値を出力
?>
```

■string mb_http_input([string type])

F

検出されたHTTP入力の文字コードを返します。引数に下記の入力の型（省略時は直近に処理された入力）を指定できます。

"G"：GET
"P"：POST
"C"：Cookie
"S"：mb_parse_str()で検出した文字コード
"I"：mbstring.http_inputの設定値

```
<?php
 print mb_http_input()."¥n"; // 出力：直近に検出したHTTP入力の文字コード
 mb_parse_str("v=日本語"); // mb_parse_str()を実行
 print mb_http_input("S")."¥n"; // 出力：mb_parse_str()で検出した文字コード
 $a = mb_http_input("I"); print_r($a); // 出力：mbstring.http_inputの設定値
?>
```

■mixed mb_http_output([string encoding])

F T

HTTP出力の文字コードを設定します。引数を省略した場合、現在の設定値を返します。

■mixed mb_internal_encoding([string encoding])

F T

内部文字コードを設定します。引数を省略した場合、現在の設定値を返します。

■mixed mb_language([string language])

F T

カレントの言語を設定します。引数を省略した場合、現在の設定値を返します。mb_send_mail()はこの設定値に基づきMIMEヘッダを生成します。

■string mb_output_handler(string contents, int status)

F T

内部文字コードから出力文字コードへの変換を行ないます。出力バッファリング機能のコールバック関数として設定します。コールバック関数の設定は、ob_start()関数またはphp.iniのoutput_handlerにより行ないます。*9

```
<?php
 mb_http_output("SJIS"); // 出力文字コードをSJISに設定
 ob_start('mb_output_handler');
 print "日本語文字列"; // 文字列がSJISで出力される
?>
```

■bool mb_parse_str(string encoded_string, [, array &result])

F T

URLエンコードされたユーザ入力データ（GET/POST/Cookie）を引数とし、パースしたあとで文字コードを検出し、内部文字コードに変換します。変換されたデータは、グローバル変数に格納されます。2番目の引数を指定した場合は、配列として結果が代入されます。

```
<?php
 $str = "jstr=".urlencode("日本語"); // URLエンコードされた文字列を作成
 print $str."¥n"; // 出力：jstr=%C6%FC%CB%DC%B8%EC
 if(mb_parse_str($str, $_USER) === TRUE){ // 文字列をパースし、配列変数に代入
   print $_USER['jstr']; // 出力： 日本語
 }
?>
```

Column mb_output_handlerとContent-Typeヘッダについて

mb_output_handlerによる出力文字コードへの変換機能は、header()関数でContent-Typeを送信すると（バイナリデータが送信されたとみなして）自動的に無効となります。例えば、auの携帯メールを送信する際にContent-Typeにtext/hdmlを指定する必要がある時は、スクリプトの中に以下のように記述し、default_mimetypeを変更します。

```
ini_set('default_mimetype','text/hdml')
```

PHP 4.2.3以降では、Content-Typeにtext/*を指定した場合には、出力文字コード変換が有効となるように変更される予定です。

*9 イメージデータなどのバイナリデータを出力する場合は、出力文字コードを"pass"とし、変換を無効とする必要があります。

■string mb_preferred_mime_name(string encoding)

指定した文字コード(encoding)に適したMIME charset文字列を返します。

```
<?php
 header("Content-Type: text/html; charset=".mb_preferred_mime_name("SJIS"));
 // "Content-Type: text/html; charset=Shift_JIS"をHTTPヘッダとして出力
?>
```

■bool mb_send_mail(string to, string subject, string message [, string headers[, string parameters]])

F T

メッセージを指定してemailを送信します。ヘッダはmb_language()の設定に基づき設定され、subjectおよびmessageの文字コードは、メール送信用の文字コードに変換/エンコードされます。オプションで追加ヘッダ(headers)およびメール送信プログラム(sendmail)のパラメータ(parameters)を指定することが可能です。

なお、送信先(to)および追加ヘッダ(headers)の文字コード変換は行なわれないため、必要に応じて変換を行なう必要があります。

```
<?php
 $extra = "From: ".mb_encode_mimeheader("管理者")."<admin@phpserv.jp>"; // From: ヘッダアドレス
 mb_send_mail("taro@phpserv.jp","題名","こんにちは",$extra); // メール送信
?>
```

● 送信されるメールのサンプル

```
To: taro@phpserv.jp
Subject: =?ISO-2022-JP?B?GyRCQmpMPhsoQg==?=
From: =?ISO-2022-JP?B?GyRCNElNfTxUGyhC?= <admin@phpserv.jp>
Mime-Version: 1.0
Content-Type: text/plain; charset=ISO-2022-JP
Content-Transfer-Encoding: 7bit
Message-Id: <20020721062449.779F59426B@phpserv.jp>
Date: Sun, 21 Jul 2002 15:24:49 +0900 (JST)

こんにちは
```

■string mb_strcut(string str, int start, int len [, string encoding])

F

文字列のstartバイト目（先頭：0）から最大lenバイト分の文字列を切り出して返します。*10 *11

```
<?php
 $str = "abcＤＥＦghi"; // 文字列（EUC-JP）
 print mb_strcut($str,3,5)."¥n"; // 出力:ＤＥ
 print mb_strcut($str,4,5)."¥n"; // 出力:ＤＥ
 print mb_strcut($str,4,6)."¥n"; // 出力:ＤＥＦ
?>
```

■string mb_strimwidth(string str, int start, int width [, string marker [, string encoding]])

F

文字列のstart文字目（先頭：0）*12 から（終端マーカを含んで）最大widthバイト分の文字列を切り出して返します。終端がマルチバイト文字の境界となる場合、その文字は除かれます。*10 オプション（marker）により文字列の丸めを生じた場合の終端マーカを指定できます。

```
<?php
 $str = "abcＤＥＦghi"; // 文字列(EUC-JP)
 print mb_strimwidth($str,3,5)."¥n"; // 出力:ＤＥ
 print mb_strimwidth($str,3,6)."¥n"; // 出力:ＤＥＦ
 print mb_strimwidth($str,4,5)."¥n"; // 出力:ＥＦg
 print mb_strimwidth($str,4,6,"..")."¥n"; // 出力:ＥＦ..
?>
```

■int mb_strlen(string str [,string encoding])

文字列の文字数を返します。*11 *12

```
<?php
 print mb_strlen("あいうえお"); // 出力：5
?>
```

*10
開始位置がマルチバイト文字の境界となる場合、その文字は含まれます。最後の文字がマルチバイト文字の境界となる場合、その文字は除かれます。

*11
処理に使用する文字コード（encoding）を指定することが可能です（デフォルト：内部文字コード）。

*12
マルチバイト文字も1文字として数えられます。

■int mb_strpos(string str, string key [, int offset [, string encoding]])

文字列の中でキーが最初に現れる位置（先頭文字：0）を返します。offsetにより検索開始位置を指定可能です。*11

```
<?php
 print mb_strpos("あいうえお","あ")."¥n";// 出力：0
 print mb_strpos("あいうえお","うえ")."¥n"; // 出力：2
?>
```

■int mb_strrpos(string str, string key [, string encoding])

文字列の中でキーが最後に現れる位置を返します。*11

```
<?php
 print mb_strrpos("ＡＢＣＡＢＣ","Ａ")."¥n";// 出力：3
?>
```

■string mb_strwidth(string str [, string encoding])

文字列の表示幅を返します。*11

```
<?php
 $str = "abcＤＥＦghi"; // 文字列（EUC-JP）
 print mb_strwidth($str)."¥n"; // 出力：12
 $str_utf8 = mb_convert_encoding($str,"UTF-8","EUC-JP"); // 文字列（UTF-8）
 print strlen($str_utf8)."¥n"; // 出力：15 (UTF-8の日本語文字は3バイト/文字)
 print mb_strwidth($str_utf8,"UTF-8")."¥n"; // 出力：12
?>
```

■mixed mb_substitute_characters([mixed c])

文字コード変換時に適当な文字コードが存在しない場合に使用される代替文字を指定します。Unicode値または"none"（出力しない）、"long"（文字コードを文字列で出力）のどれかを指定します。引数を省略した場合は現在の設定値を返します。

■string mb_substr(string str, int start, int len [, string encoding])

F

start文字目（先頭文字：0）からlen文字分の文字列を返します。*11 *12

```
<?php
 $str = "abcＤＥＦghi"; // 文字列（EUC-JP）
 print mb_substr($str,3,5)."¥n"; // 出力:ＤＥＦgh
 print mb_substr($str,4,5)."¥n"; // 出力:ＥＦghi
?>
```

7.2 マルチバイト対応正規表現関数

PHPでは、日本語のマルチバイト文字に対応した正規表現関数がサポートされています。

▶PHP構築オプション

--enable-mbregex　　マルチバイト対応の正規表現関数を有効にします。

▶php.ini設定オプション

パラメータ	デフォルト値	説明
mbstring.mb_regex_encoding	EUC-JP	正規表現の文字エンコーディング （SJIS、EUC-JP、UTF-8、ASCII）

■int mb_ereg(string pattern, string str [, array &regs])

F

正規表現にマッチした場合（空文字を含む）に1を返します。3番目の引数を指定した場合は一致した部分のバイト数を返します。マッチしないかエラーの場合にFALSEを返します。

```
<?php
 if(mb_ereg("日本(.)","日本語",$regs)){
   print_r($regs); // $regsはarray("日本語","語") となる
 }
?>
```

bool mb_ereg_match(string pattern, string str [, string option])

正規表現に一致する場合にTRUE、それ以外の場合にFALSEを返します。optionには、表3-53に示す文字を指定します。

表3-53 正規表現のオプション

オプション	説明
i	大文字・小文字を区別しない
x	拡張正規表現を使用する。空白を無視し、コメントをつけられる（デフォルトで有効）
m	マルチラインモード。改行文字も "." に含まれる
s	シングルラインモード。 "^" および "$" は改行を無視する
p	POSIXモード。改行も通常文字と見なす
l	POSIX正規表現に基づき、最長一致を行なう
e	置換文字列をPHPの式として評価する（nb_ereg*_replace()のみ使用可）

```
<?php
 print mb_ereg_match("¥w+","ＡＢＣ") ? "一致":"不一致";
?>
```

bool mb_ereg_search([string pattern [, string option]])

正規表現に一致する場合にTRUE、それ以外の場合にFALSEを返します。*13 *14 *15

int mb_ereg_search_getpos()

検索の開始位置を取得します。

array mb_ereg_search_getregs()

直近の検索に一致した部分文字列を取得します。

bool mb_ereg_search_init(string str[, string pattern[, string option]])

FT

検索対象の文字列と正規表現を設定します。*14 *15

*13
検索対象の文字列は、mb_ereg_search_init()により設定します。

*14
正規表現を省略した場合は前回のものを使用します。

*15
optionには、表3-53に示す文字を指定します。

■array mb_ereg_search_pos([string pattern[, string option]])

F

正規表現に一致する部分文字列の位置を配列として返します。*13 *14 *15

■array mb_ereg_search_regs([string pattern[, string option]])

正規表現に一致する部分文字列を配列として返します。*13 *14 *15

■bool mb_ereg_search_setpos(int pos)

F T

検索の開始位置を設定します。

■bool mb_eregi(string pattern, string str [, array regs])

mb_ereg()と同じですが、大文字小文字を区別しません。

■string mb_ereg_replace(string pattern, string rstr , string str [, string option])

正規表現に一致する部分文字列を置換します。*15

```
<?php
 print mb_ereg_replace("語","人","日本語と中国語"); // "日本人と中国人"を出力
?>
```

■string mb_eregi_replace(string pattern, string rstr , string str)

mb_ereg_replace()と同じですが、大文字小文字を区別しません。

■string mb_regex_encoding([string encoding])

正規表現用の文字コード（SJIS、EUC-JP、UTF-8、ASCII）を設定または取得します。

```
<?php
 mb_regex_encoding("SJIS"); // 正規表現用文字コードをSJISに設定
 print mb_regex_encoding(); // 現在の設定値（SJIS）を出力
?>
```

■array mb_split(string pattern, string str [, int limit])

マルチバイト文字列を正規表現により分割し、結果を配列として返します。

```
<?php
 $str = mb_split("[あ-ん]+","日本語は難しい");
 print_r($str); // array("日本語","難","")を出力
?>
```

7.3 Namazu（全文検索）関数

Namazu拡張モジュールにより、Namazuによる全文検索をPHPスクリプトから行なうことができます。Namazu 2.0以降が必要です。

Namazu拡張モジュールはPEAR/PECLに含まれており、PHPのソースコードに含まれていないため、以下のようにCVSコマンドにより入手します。

```
-> cvs -d:pserver:cvsread@cvs.php.net:/repository login
   （パスワードとして'phpfi'を入力）
-> cvs -d:pserver:cvsread@cvs.php.net:/repository co pear
-> cp -r pear/PECL/namazu php-4.2.2/ext/
-> cd php-4.2.2; ./buildconf
```

▶PHP構築オプション

--with-namazu[=DIR]　Namazu関数サポートを有効にします。

▶php.ini設定オプション

パラメータ	デフォルト値	説明
namazu.lang	ja	検索に使用する言語
namazu.loggingmode	Off	NMZ.slogへログを記録する場合にOn
namazu.sortmethod	score	検索結果のソート方法（score ¦ date ¦ field:{フィールド名}）
namazu.sortorder	descending	ソート順（昇順、降順）の指定(ascending ¦ descending)

▶参考URL

Namazuのページ：**http://www.namazu.org/**

■bool nmz_close(int nmzid)

T

インデックスを閉じます。

■string nmz_codeconv_query(string query)

F

言語が日本語の場合、クエリ文字列の文字コードを内部文字コード（EUC-JP）に変換します。入力文字コードとしてShift_JIS、ISO-2022-JP、EUC-JPをサポートしています。

```
<?php
 $query = mb_convert_encoding("日本語","SJIS");
 print nmz_codeconv_query($query); // 出力：日本語（EUC-JP）
?>
```

■array nmz_fetch_date(resource result [, int limit [, int offset]])

F

各文書の日付を配列として返します。*16

■array nmz_fetch_field(resource result, string field [, int limit [, int offset]])

F

各文書の指定フィールドを配列として返します。*16

```
<?php
 $query = "PHP"; // 検索文字列
 $index = "/usr/share/namazu/index/JF"; // Namazuのindexを指定（JF）
 $id = nmz_open($index) or die("Can't open namazu index.");
 $result = nmz_search($id, $query) or die("query failed."); // 検索を実行
 $subject = nmz_fetch_field($result, "subject"); // 題名を取得
 $uri = nmz_fetch_field($result, "uri"); // URIを取得
 $score = nmz_fetch_score($result); // スコア情報を取得
 for ($i=0; $i< nmz_num_hits($result); $i++){ // 検索結果を出力
   printf("%d %s %s¥n",$score[$i],$subject[$i],$uri[$i]);
 }
?>
```

■array nmz_fetch_score(resource result [, int limit [, int offset]])

F

各文書のスコアを配列として返します。*16 *17

bool nmz_free_result(resource result)

F T

検索結果を解放します。

string nmz_get_idxname(int index)

インデックスの名前を取得します。

```
<?php
 $index = array("/usr/share/namazu/index/JF",
                "/usr/share/namazu/index/JMAN") ; // Namazuのindexを指定（JF、JMAN）
 nmz_open($index) or die("Can't open namazu index.");
 for ($i=0; $i<nmz_get_idxnum(); $i++){
   print nmz_get_idxname($i)."¥n";
 }
?>
```

int nmz_get_idxnum()

インデックスの数を返します。[*18]

int nmz_get_idx_totalhitnum(int index)

指定したインデックスの全ヒット数を返します。

```
<?php
 $query = "PHP"; // 検索文字列
 $index = array("/usr/share/namazu/index/JF",
                "/usr/share/namazu/index/JMAN"); // Namazuのindexを指定（JF）
 $id = nmz_open($index) or die("Can't open namazu index.");
 $result = nmz_search($id, $query) or die("query failed."); // 検索を実行
 print "合計ヒット数: ". nmz_num_hits($result) . "¥n"; // ヒット数を出力
 for ($i=0; $i<nmz_get_idxnum(); $i++){
   print "$i 番目のインデックスのヒット数: ". nmz_get_idx_totalhitnum($i) . "¥n";
                                                          // ヒット数を出力
 }
?>
```

*16 取得数の上限（limit）およびオフセット（offset）を指定可能です。

*17 nmz_fetch_field()のサンプルを参照してください。

*18 nmz_get_idxname()のサンプルを参照してください。

■array nmz_get_idx_hitnumlist(int index)

F

指定したインデックスのトークンを配列として取得します。フレイズ検索の場合、フレイズに関するヒット数が配列のキー"phrase"に代入されます。

```
<?php
 $id = nmz_open("/usr/share/namazu/index/JF") or die("Can't open namazu index.");
 $result = nmz_search($id, "{PostgreSQL 7.1}") or die("query failed."); // フレーズ検索を実行
 for ($i=0; $i<nmz_get_idxnum(); $i++){
   $a = nmz_get_idx_hitnumlist($i); // $aは（"phrase"=>1,"postgresql"=>8,"7.1"=>100）となる
   print_r($a);
 }
?>
```

■string nmz_get_lang()

F

検索に用いる言語を取得します。

```
<?php
 print nmz_get_lang(); // 出力：ja_JP.eucJP
?>
```

■string nmz_get_lang_ctype()

F

検索に用いる言語名を表す文字列（CTYPE）を取得します。

```
<?php
 print nmz_get_lang_ctype(); // 出力：ja_JP.eucJP
?>
```

■array nmz_get_querytoken(resource result)

F

クエリ中の単語を配列として取得します。

```
<?php
 $id = nmz_open("/usr/share/namazu/index/JF") or die("Can't open namazu index.");
 $result = nmz_search($id, "GNU Emacs") or die("query failed."); // フレーズ検索を実行
 $a = nmz_get_querytoken($result); // array("GNU","Emacs") となる
 print_r($a);
?>
```

■bool nmz_info()

T

ソートの方法および順番の設定値に関する情報を出力します。

```
<?php
 nmz_set_lang("ja"); // 日本語に設定
 nmz_set_loggingmode(TRUE); // NMZ.slogへログを記録
 nmz_set_sortmethod("score"); // スコアでソート
 nmz_set_sortorder("descending"); // 降順ソート
 nmz_info();  // 出力：language,sort method,sort orderの設定値
?>
```

■bool nmz_is_lang_ja()

言語の設定が日本語の場合にTRUE、それ以外の場合にFALSEを返します。

■int nmz_num_hits(resource result)

F

検索結果の文書数を取得します。*19

■int nmz_open(mixed index)

F

インデックス（配列も指定可）をオープンし、インデックスのIDを返します。*20

■int nmz_result_date(resource result, int i)

F

i番目の文書の日付をUNIX時間で取得します。

```
<?php
 $id = nmz_open("/usr/share/namazu/index/JF") or die("Can't open namazu index.");
 $result = nmz_search($id, "PHP") or die("query failed."); // フレーズ検索を実行
 for ($i=0; $i<nmz_num_hits($result);$i++){
   print date('Y/m/d',nmz_result_date($result,$i))."¥t"; // 日付を出力
   print nmz_result_field($result,$i,"date")."¥n"; // 日付を出力（nmz_result_field()）
 }
?>
```

*19
nmz_search()のサンプルを参照してください。

*20
nmz_get_idxname()、nmz_search()などのサンプルを参照してください。

■string nmz_result_field(resource result, int i, string field)

F

i番目の文書の指定したフィールドを取得します。*21

■int nmz_result_score(resource result, int i)

F

i番目の文書の点数（スコア）を取得します。*21

■int nmz_search(int nmzid, string query)

F

クエリを実行し、検索結果リストへのハンドルを返します。

```
<?php
 $query = "PHP"; // 検索文字列
 $index = "/usr/share/namazu/index/JF"; // Namazuのindexを指定（JF）
 $id = nmz_open($index) or die("Can't open namazu index.");
 $result = nmz_search($id, $query) or die("query failed."); // 検索を実行
 print "ヒット数: ". nmz_num_hits($result) . "¥n"; // ヒット数を出力
?>
```

■bool nmz_set_lang(string lang)

F T

検索に用いる言語を設定します。*22

■bool nmz_set_loggingmode(bool mode)

T

NMZ.slogへの出力を有効とする場合にTRUEを指定します。*22

■bool nmz_set_sortmethod(string method)

F T

ソートの方法を指定します。*22

■bool nmz_set_sortorder(string order)

F T

ソートの順番を指定します。*22

*21
nmz_result_date()のサンプルを参照してください。

*22
nmz_info()のサンプルを参照してください。

Chapter-8 PHP関数 XML関連

XML（eXtensible Markup Language）は、次世代のデータ記述言語として注目されており、さまざまな用途に使用されるようになりつつあります。

PHPではXMLパーサAPIとして標準的な以下のものがサポートされています。

・SAX（Simple API for XML）パーサ

各要素についてコールバック関数を定義し、シーケンシャルにXMLデータを処理します。シンプルで処理が軽く、Webでの使用に向いています。PHPでは、SAXパーサとしてexpat（http://sourceforge.net/projects/expat/）がサポートされています。

・DOM（Document Object Model）パーサ

XMLデータのツリー構造を解釈し、任意の要素にランダムアクセスが可能です。データ構造の一部を利用したい場合に使用します。PHPでは、GNOME-XML（libxml2）がサポートされています。

8.1 SAX関数 − SAXパーサ

SAXパーサはXMLデータをシーケンシャルに処理するため、使用メモリ量が少なく、Webアプリケーションに適しています。

▶PHP構築オプション

--with-xml[=DIR]	expatによるXML関数（SAX）を有効（デフォルト）にします。DIRを省略するとPHP付属のexpatが使用されます。
--without-xml	expatによるXML関数を無効にします。

▶参考URL

expatの開発・配布元：**http://www.libexpat.org/**

▶注意

本節では以下のサンプルXMLデータ（ns-sample.xml）を使用します。

```
<?xml version="1.0" encoding="UTF-8"?>
<members xmlns:phpj="http://www.php.gr.jp/project/i18n/1.0"
         xmlns="http://www.php.net/4.0">
 <person>
  <name>鈴木太郎</name>
  <email>suzuki@php.net</email>
  <phpj:email>taro@php.gr.jp</phpj:email>
 </person>
 <person>
  <name>佐藤花子</name>
  <email>sato@php.net</email>
  <phpj:email>hanako@php.gr.jp</phpj:email>
 </person>
</members>
```

*1
xml_parse()のサンプルを参照してください。

■string utf8_decode(string data)

UTF-8エンコードの文字列をISO-8859-1に変換します。

■string utf8_encode(string data)

ISO-8859-1エンコードの文字列をUTF-8に変換します。

■string xml_error_string(int code)

エラーコードを指定してエラー文字列を取得します。*1

■int xml_get_current_byte_index(resource parser)

現在の処理位置をバイト単位で取得します。*1

■int xml_get_current_column_number(resource parser)

現在処理中のカラム番号を取得します。*1

int xml_get_current_line_number(resource parser)

現在処理中の行番号を取得します。*1

int xml_get_error_code(resource parser)

XMLパーサのエラーコードを取得します。*1

int xml_parse(resource parser, string data [, int isfinal])

XMLドキュメントの処理を開始します。dataには処理するデータを指定します。isfinalにTRUEを指定することにより、最後のデータであることを指示します。処理を行なう際には、必要に応じて設定されたイベントのハンドラがコールされます。エラーの場合は0を返します。

```
<?php // XMLデータをテキスト出力する例
 function start_element($parser, $name, $attrs){ print "start: $name¥n";}
 function end_element($parser, $name){ print "end: $name¥n";}
 function character_data($parser, $data){ print "data: $data¥n";}

 $parser = xml_parser_create("UTF-8"); // XMLパーサを生成
 xml_parser_set_option($parser, XML_OPTION_CASE_FOLDING, TRUE); // 要素名を大文字に変換
 xml_set_element_handler($parser, "start_element", "end_element"); // 要素タグ用ハンドラを登録
 xml_set_character_data_handler($parser, "character_data");  // 文字データ用ハンドラを登録

 $fp = fopen("ns-sample.xml","r"); // XMLファイルをオープン
 while ($data = fread($fp, 4096)) { // XMLデータをパース
  if (!xml_parse($parser, $data, feof($fp))){ // パーサエラーが発生した場合
    die(sprintf("XML error: %s at line %d col %d byte %d",
                xml_error_string(xml_get_error_code($parser)), // エラー文字列
                xml_get_current_line_number($parser), // カレント行
                xml_get_current_column_number($parser), // カレントカラム
                xml_get_current_byte_index($parser))); // 処理位置（バイト）
  }
 }
 xml_parser_free($parser); // XMLパーサを解放
?>
```

■bool xml_parse_into_struct(resource parser, string data, array struct, array index)

F T

XMLデータをパースし、配列に格納します。パース時には、xml_parse()と同様に設定されたハンドラがコールされます。

```
<?php
 $data = implode("¥n",file("ns-sample.xml")); // 処理するXMLデータ
 $parser = xml_parser_create("UTF-8");
 xml_parse_into_struct($parser, $data, $val, $index); // パースしたデータを配列に格納
 print_r($val[$index['EMAIL'][0]]);  // 出力： array('tag', 'type', 'level', 'value')
?>
```

■resource xml_parser_create([string encoding])

F

XMLパーサを作成し、XMLパーサへのハンドルを返します。encodingによりパーサのエンコード(以下)を指定可能です。*1

ISO-8859-1 (デフォルト)、US-ASCII、UTF-8

■resource xml_parser_create_ns([string encoding [, string sep]])

F

名前空間を指定してXMLパーサを作成し、XMLパーサへのハンドルを返します。要素名/属性名は、**名前空間URI＋セパレータ(デフォルト':')＋要素名/属性名**の形式でハンドラに渡されます。

```
<?php
 function start_element($parser, $name, $attrs){ print "start: $name¥n";}
 function end_element($parser, $name){ print "end: $name¥n";}
 function ns_start($parser, $prefix, $uri){ print "ns-start: $prefix $uri¥n";}
 function ns_end($parser, $prefix){ print "ns-end: $prefix¥n";}

 $parser = xml_parser_create_ns("UTF-8"); // XMLパーサを生成
 xml_set_element_handler($parser, "start_element", "end_element"); // 要素タグ用ハンドラ
 xml_set_start_namespace_decl_handler($parser, "ns_start"); // 名前空間開始ハンドラ
 xml_set_end_namespace_decl_handler($parser, "ns_end"); // 名前空間終了ハンドラ

 $data = implode("¥n",file("ns-sample.xml")); // 処理するXMLデータ
 xml_parse($parser, $data); // XMLデータをパース
?>
```

■bool xml_parser_free(resource parser)

F T

XMLパーサ用のリソースを解放します。*1

■mixed xml_parser_get_option(resource parser, int option)

XMLパーサからオプションを取得します。取得可能なオプションは、xml_parser_set_option()を参照してください。

```
<?php
 $parser = xml_parser_create("UTF-8"); // XMLパーサを生成
 print xml_parser_get_option($parser, XML_OPTION_TARGET_ENCODING); // 出力：UTF-8
?>
```

■bool xml_parser_set_option(resource parser, int option, mixed value)

F T

以下のオプションの値を設定します。*1

XML_OPTION_CASE_FOLDING　：タグの大文字変換を有効にする場合にtrue（デフォルト）

XML_OPTION_TARGET_ENCODING：ターゲットエンコード（ISO-8859-1、US-ASCII、UTF-8。デフォルトはソースエンコードと同じ）

■bool xml_set_character_data_handler(resource parser, string handler)

F T

文字データ用ハンドラ関数を設定します。ハンドラ関数は以下のように定義します。*1

```
handler(resource parser, string data)
```

dataには文字データが指定されます。

<member age="31">Sasaki</member>を処理する場合、ハンドラ関数は以下のようにコールされます。

```
handle($parser, "Sasaki");
```

■bool xml_set_default_handler(resource parser, string handler)

F T

デフォルトのハンドラを設定します。デフォルトのハンドラは、通常のハンドラが設定されていない場合にコールされます。ハンドラ関数の仕様は、xml_set_character_data_handler()と同じです。

<member age="31">Sasaki</member>を処理する場合、ハンドラ関数は以下のようにコールされます（要素ハンドラ、文字データハンドラが未定義の場合）。

```
handler($parser, "<member age=¥"31¥">Sasaki</member>");
```

■bool xml_set_element_handler(resource parser, string h_start, string h_end)

F T

startおよびend要素のハンドラ関数をそれぞれ以下のh_start、h_endに設定します。*1

```
h_start(resource parser, string name, array attribs)
```

nameにはコールされた要素の名前、連想配列attribsにはすべての属性が指定されます。

```
h_end(resource parser, string name)
```

nameにはコールされた要素の名前が指定されます。

<member age="31">Sasaki</member>を処理する場合、ハンドラ関数はそれぞれ以下のようにコールされます。

```
h_start($parser, "member", array("age">="31"));
h_end($parser, "member");
```

■bool xml_set_end_namespace_decl_handler(resource parser, string handler)

F T

名前空間宣言終了時にコールされるハンドラ関数を設定します。*2

■bool xml_set_external_entity_ref_handler(resource parser, string handler)

F T

外部エンティティ参照ハンドラ関数を設定します。ハンドラ関数は以下のように定義します。PHPのXMLパーサ（expat）は外部エンティティを直接解釈せず、このハンドラをコールします。*2

```
handler(resource parser, string open, string base, string sid, string pid)
```

*2
xml_parser_create_ns()のサンプルを参照してください。

引数は以下のように定義します。

open：空白で区切ったエンティティのリスト
base：使用せず（常に空文字列）
sid　：システムID
pid　：パブリックID

<!ENTITY tag_ext SYSTEM "ext.xml">を宣言したあとで &tag_ext; を使用した場合、以下のようにハンドラがコールされます。

```
handler($parser,"tag_ext","","ext.xml","");
```

■bool xml_set_notation_decl_handler(resource parser, string handler)

F T

NOTATION命令用のハンドラ関数を設定します。

```
<!NOTATION name {PUBLIC pid|SYSTEM sid}>
```

ハンドラ関数は以下のように定義します。

```
handler(resource parser, string notation, string base, string sid, string pid)
```

引数の内容は、xml_set_unparsed_entity_decl_handler()のハンドラ関数と同じです。

■bool xml_set_object(resource parser, object obj)

F T

オブジェクトの内部で指定したパーサを使用可能にします。必要なハンドラ関数はメソッドとして実装します。

```
<?php
 class XML {
  var $parser;
  function XML() { // コンストラクタ
    $this->parser = xml_parser_create("UTF-8");
    xml_set_object($this->parser, $this); // オブジェクトを設定
    xml_set_element_handler($this->parser, "h_start", "h_end");
    xml_set_character_data_handler($this->parser, "h_char");
  }

  function parse($data) { xml_parse($this->parser, $data);}
  function h_start($parser, $name, $attrs){ var_dump($name, $attrs);}
  function h_end($parser, $name){ var_dump($name);}
  function h_char($parser, $data){ var_dump($data);}
 }

 $xml_parser = new XML();
 $xml_parser->parse("<name id='12'>Taro</name>");
?>
```

■bool xml_set_processing_instruction_handler(resource parser, string handler)

F T

XMLパーサの処理命令(PI)用ハンドラ関数を設定します。ハンドラ関数は以下のように定義します。処理命令は <? … ?> と記述します。

```
handler(resource parser, string target, string data)
```

targetにはPIのターゲット、dataにはPIデータが指定されます。

<?php print "Hello!"?> を処理する場合、ハンドラ関数は以下のようにコールされます。

```
handler($parser,"php","Hello!");
```

■bool xml_set_start_namespace_decl_handler(resource parser, string handler)

F T

名前空間宣言開始時にコールされるハンドラ関数を設定します。*2

■bool xml_set_unparsed_entity_decl_handler(resource parser, string handler)

F T

以下のような外部エンティティのハンドラ関数を設定します。

```
<!ENTITY name {PUBLIC pid|SYSTEM sid} NDATA notation>
```

ハンドラ関数は以下のように定義します。

```
handler(resource parser, string name, string base, string sid,
string pid, string notation)
```

name	：エンティティ名
base	：現在は使用せず(常に空文字列)
sid	：システムID
pid	：パブリックID
notation	：表記法(NOTATION命令で事前に指定)

<!ENTITY png123 SYSTEM "123.png" NDATA png> を処理した場合、ハンドラ関数は以下のようにコールされます。

```
handler($parser,"png123","","123.png","","png");
```

8.2 DOM関数 − DOMパーサ

DOMパーサはXMLデータ全体を読み込み、XMLデータ全体をオブジェクトとして処理します。PHPでは、GNOME xmlライブラリ（libxml2）により、DOMレベル2相当の処理を行なうことができます。

▶PHP構築オプション

--with-dom[=DIR]	libxml2 *3 によるDOM関数を有効にします。
--with-dom-xslt[=DIR]	libxslt *4 によるXSLT関数を有効にします。
--with-dom-exslt[=DIR]	libxslt *4 によるEXSLT関数を有効にします。
--with-zlib-dir	zlibのインストール先を指定します（必須）。

▶ 参考URL

GNOME xmlライブラリの開発・配布元：**http://www.xmlsoft.org/**

▶DOM関数によるXMLデータの生成

DOM関数によりXMLデータを生成する際の手順は以下のようになります。

①domxml_new_doc()関数によりXMLオブジェクトのインスタンスを生成する。
②create_element()メソッドによりオブジェクトツリーのルートノードを生成する。
③create_element()メソッドによりオブジェクトツリーの子ノードを生成する。
④append_child()メソッドにより作成したノードを配置する。

各クラスの主なメソッドを**表3-54、3-55**に示します。

表3-54　メソッドと属性（DomDocumentクラスのメソッド）

メソッド名	説明
doctype()	オブジェクトDomDocumentTypeを取得
document_element()	ルートノードを生成
create_element(name)	要素を生成
create_text_node(name)	テキストノードを生成
create_comment(name)	コメントを生成
create_cdata_section(name)	CDATAセクションを生成
create_processing_instruction(name)	処理命令を生成
create_attribute(name)	属性を作成
create_entity_reference(name)	エンティティ参照を作成
get_elements_by_tagname(name)	指定したタグ名に一致するDOM要素を配列として返す
dump_mem()	XMLドキュメントを文字列にダンプ
dump_file(file)	XMLドキュメントをファイルにダンプ
html_dump_mem()	XMLドキュメントをHTMLとして文字列にダンプ

*3
libxmlバージョン2.4.14以降が必要です。

*4
libxsltバージョン1.0.3以降が必要です。

表3-55 メソッドと属性（DomNodeクラスのメソッド）

メソッド名	説明
append_child(node)	ノードを親ノードに追加
append_sibling(node)	ノードを子ノードに追加
attributes()	属性のリストを配列として返す
child_nodes()	子ノードのリストを返す
insert_before(node,refnode)	基準ノードの前にノードを追加
node_content()	ノードの内容を取得
node_name()	ノードの名前を取得
node_type()	ノードの型を取得
owner_document()	ノードが属しているドキュメントを返す
parent_node()	親ノードを返す
set_content(content)	ノードの内容を指定
set_name(name)	ノードの名前を指定
unlink_node()	ノードを削除する

■object domxml_new_doc(string version)

F

新規にDOMオブジェクトを作成し、Dom Document型のオブジェクトを返します。

```
<?php // HTML文書を出力する例
 $dom = domxml_new_doc("1.0");
 $html = $dom->create_element("html");
 $html->set_attribute("lang", "ja");
 $root = $dom->append_child($html);
 // ノードの作成
 $head = $dom->create_element("head");
 $title = $dom->create_element("title");
 $body = $dom->create_element("body");
 $title_text = $dom->create_text_node("PHP News");
 $body_text = $dom->create_text_node("PHP 4.2.2 was released in July.");
 // 作成したノードを配置
 $title->append_child($title_text);
 $head->append_child($title);
 $root->append_child($head);
 $body->append_child($body_text);
 $root->append_child($body);

 print $dom->dump_file("/tmp/test1.xml",FALSE,TRUE); // XML文書を出力
?>
```

■object domxml_open_file(string file)

F

XMLドキュメントからDOMオブジェクトを作成し、Dom Document型のオブジェクトを返します。

```
<?php
 $dom = domxml_open_file("ns-sample.xml"); // XMLドキュメントからDOMオブジェクト作成
 $root = $dom->document_element(); // ルートノードを取得
 print_r($root);
?>
```

■object domxml_open_mem(string xmldata)

F

XMLデータからDOMオブジェクトを作成し、Dom Document型のオブジェクトを返します。

```
<?php
 $xmlstr = join("", file("ns-sample.xml")); // パースするXMLデータ
 $dom = domxml_open_mem($xmlstr); // XMLデータをパースし、DOM Documentクラスを返す
 $root = $dom->document_element(); // ルートノードを取得
 print_r($root);
?>
```

■string domxml_version()

GNOME XMLのバージョンを文字列として返します。

```
<?php
 print domxml_version()."¥n";
?>
```

Hypertext Preprocessor

PHP

Chapter-9 オプション

PHPでは、動作設定を行なう種々の設定パラメータを設定ファイル（php.ini）により行ないます。本章では、一般的な設定オプションを示します。なお、データベース接続用の設定パラメータなどの各機能ごとの設定オプションについては、第3部の対応する章に示しています。

なお、PHPの設定オプションは、Apacheの設定ディレクティブにおいても指定可能です。設定オプションには、一般ユーザがPHPスクリプトの内部で自由に設定変更が可能なユーザパラメータとphp.iniまたはApacheの設定ファイル（通常はhttpd.conf）でのみ設定可能なシステムパラメータの2種類があります。これらは、各々、httpd.conf/.htaccessで以下のように設定します。*1

1. ユーザパラメータ(.htaccessまたはhttpd.confで設定可能)

php_value パラメータ名 値
php_flag パラメータ名 **on|off**

2. システムパラメータ(httpd.confでのみ設定可能)

php_admin_value パラメータ名 値
php_admin_flag パラメータ名 **on|off**

*1
.htaccessで設定を変更するためには、"AllowOverride Options"をhttpd.confの対応するディレクトリ権限設定で指定する必要があります。

*2
将来的にこの機能は廃止される予定です。

*3
複数の文字をセパレータとして指定することが可能です（例：";&"）。

*4
EGPCS: Environment（環境変数）、GET、POST、Cookie、Server（サーバ変数）。

9.1 PHP言語/スクリプトエンジンに関するオプション

パラメータ変数	引数の型	説明
allow_call_time_pass_reference	bool	関数コール時に引数を参照渡しと指定可能とする場合にOn *2
asp_tags	bool	ASP型式のタグ<% ... %>を使用可能とする場合にOn
engine	bool	ApacheでPHPスクリプトエンジンを有効とする場合にOn
expose_php	bool	Webサーバの出力ヘッダにPHPのバージョン情報を出力する場合にOn
ignore_user_abort	bool	ユーザによる接続が閉じられたときに実行を継続する場合にOn
implicit_flush	bool	各出力ブロックで自動的に出力をフラッシュする場合にOn
max_execution_time	int	スクリプトの最大実行時間
memory_limit	int	スクリプトが使用できる最大メモリ量
output_buffering	int	出力バッファのサイズ（バイト数）
output_handler	string	出力バッファリング用ハンドラ関数
precision	int	float数の表示桁数
short_open_tag	bool	短縮型のタグ（<? ... ?>）を使用可能とする場合にOn

9.2 エラー処理およびログ設定に関するオプション

パラメータ変数	引数の型	説明
display_errors	bool	標準出力にエラー出力を行なう場合にOn
error_log	string	エラーメッセージの記録先
error_reporting	int	エラー表示レベル（設定値については表4-7を参照）
log_errors	bool	サーバーのエラーログにエラーメッセージを記録する場合にOn
track_errors	bool	直近のエラーを$php_errormsgで参照可能とする場合にOn
html_errors	bool	offの場合、エラー出力にHTMLタグを含めない
warn_plus_overloading	bool	文字列の結合に'+'を使用した際に警告を発生する場合にOn

9.3 データ処理に関するオプション

パラメータ変数	引数の型	説明
always_populate_raw_post_data	bool	$HTTP_RAW_POST_DATAにPOSTデータを代入する場合にOn
auto_append_file	string	スクリプトの最後に付加するファイル
auto_prepend_file	string	スクリプトの最初に付加するファイル
arg_separator.output	string	PHPが生成するURLで使用される引数セパレータ（デフォルト：&）
arg_separator.input	string	PHPがパースするURLで使用される引数セパレータ（デフォルト：&）*3
default_charset	string	charsetのデフォルト値
default_mimetype	string	MIMEタイプのデフォルト値
magic_quotes_gpc	bool	GET/POST/Cookieのパース時に',",¥,NULLを¥でエスケープする場合にOn
magic_quotes_runtime	bool	外部ソースでの実行されるコマンド文字列のエスケープを行なう場合にOn
magic_quotes_sybase	bool	'を'でエスケープする場合にOn
post_max_size	int	POSTデータの最大サイズ（バイト数）
register_argc_argv	bool	コマンドライン引数（argc、argv）をグローバル変数として登録する場合にOn
register_globals	bool	EGPCSをパース後にグローバル変数に登録する場合にOn *4
variables_order	string	EGPCSをパースする順番を指定 *4

9.4 ファイル関連のオプション

パラメータ変数	引数の型	説明
browscap	string	Webブラウザに関する情報ファイルbrowscap.iniのパス
doc_root	string	PHP用のルートディレクトリ
file_uploads	bool	ファイルアップロード機能を有効とする場合にOn
upload_tmp_dir	string	アップロードしたファイルを一時的に保存するディレクトリ
upload_max_filesize	int	アップロードされるファイルの最大サイズ
user_dir	string	ユーザディレクトリ

9.5 セーフモード関連のオプション

パラメータ変数	引数の型	説明
safe_mode	bool	セーフモードを有効にする場合にOnとする
safe_mode_gid	bool	Onを指定すると、同じグループID（GID）を有するファイルもオープンできるようセーフモードの制約が緩和される
safe_mode_include_dir	string	指定したディレクトリに関しては、セーフモードのUID/GIDチェックの対象から除外される
safe_mode_exec_dir	string	セーフモードが有効な場合、関数exec()などで実行可能なコマンドは、指定したディレクトリ以下にあるものに制限される
safe_mode_allowed_env_vars	string	セーフモードでは、ユーザが変更可能な環境変数は、指定した接頭辞を有するもののみに制限される
safe_mode_protected_env_vars	string	セキュリティ上の理由によりputenv()による変更を禁止する環境変数を指定する。この制約は、safe_mode_allowed_env_varsよりも優先する