

# PHP

## 逆引きハンドブック

蒲生 睦男◆著

WebサイトやWebアプリの開発によく使われるPHPを網羅的に解説!

Web開発者の  
必携書!

PHP  
5.3/5.4  
対応

「やりたいこと」からPHPの機能を探せる!



24時間無料でサンプルデータをダウンロードできます。



C&R研究所

---

# PHP

## 逆引きハンドブック

---

蒲生 睦男◆著

PHP  
5.3/5.4  
対応

## ■権利について

- 本書に記述されている社名・製品名などは、一般に各社の商標または登録商標です。
- 本書では™、©、®は割愛しています。

## ■本書の内容について

- 本書は著者・編集者が実際に操作した結果を慎重に検討し、著述・編集しています。ただし、本書の記述内容に関わる運用結果にまつわるあらゆる損害・障害につきましては、責任を負いませんのであらかじめご了承ください。
- 本書で紹介している操作やコードの実行結果の画面は、Windows 7(日本語版)とGoogle Chromeを基本にしています。他の環境では、画面のデザインや操作が異なる場合がございますので、あらかじめご了承ください。

## ■サンプルについて

- 本書で紹介しているサンプルは、C&R研究所のホームページ(<http://www.c-r.com>)からダウンロードすることができます。ダウンロード方法については、5ページを参照してください。
- サンプルデータの動作などについては、著者・編集者が慎重に確認しております。ただし、サンプルデータの運用結果にまつわるあらゆる損害・障害につきましては、責任を負いませんのであらかじめご了承ください。
- サンプルデータの著作権は、著者及びC&R研究所が所有します。許可なく配布・販売することは堅く禁止します。

### ●本書の内容についてのお問い合わせについて

この度はC&R研究所の書籍をお買いあげいただきましてありがとうございます。本書の内容に関するお問い合わせは、FAXまたは郵送で「書名」「該当するページ番号」「返信先」を必ず明記の上、次の宛先までお送りください。お電話や電子メール、または本書の内容とは直接的に関係のない事柄に関するご質問にはお答えできませんので、あらかじめご了承ください。

〒950-3122 新潟県新潟市北区西名目所4083-6 株式会社 C&R研究所 編集部  
FAX 025-258-2801  
「PHP逆引きハンドブック」サポート係

---

## III PROLOGUE

PHP (PHP: Hypertext Preprocessor) は、Webアプリケーションの開発用として人気の高いプログラミング (スクリプト) 言語です。言語体系がシンプルで自由度が高く、比較的簡単に習得することができます。また、本格的なオブジェクト指向プログラミングもサポートしているので、初心者から上級者まで幅広いユーザーに支持されています。

本書は、PHPを使ってWebアプリケーションのプログラミングを行う人のためのリファレンス&テクニック集です。PHPのステートメント、関数などを目的別に分類し、やりたいことから探していただけるように、逆引き形式を採用しています。手元に置いていつでもお使いいただけるようにA5サイズを採用していますので、何度でも繰り返しページをめくってみてください。

本書では、PHPの標準的な環境で利用できる関数を中心に、プログラミング事例を豊富に網羅し、各機能を初心者でもわかりやすいシンプルな事例で、かつ、実務にもすぐに応用できる事例で紹介しています。事例のサンプルコード中には、プログラムの解説を記述していますので、コードを読むことでプログラムの動作について理解を深めることができます。さらに、ONEPOINTではサンプルで利用したポイントとなる命令語や関数についてわかりやすく解説するとともに、COLUMNには、読者のみなさんの理解を深める手助けとなるように、補足情報や応用的な事例を豊富に盛り込んでいます。

また、本書は、PHP4、PHP5 (PHP5.3、PHP5.4) に対応しており、基礎知識や基本文法、オブジェクト指向プログラムやセキュリティ対策も解説しているので、初心者から中級者以上まで、幅広く使える内容になっています。

本書で紹介したサンプルファイルは、無料でダウンロードすることができます。ダウンロードしたサンプルを実際に実行して動作を確認してみてください。また、プログラムを修正して動作を変えるなどして、自由にカスタマイズしてみてください。プログラムを読んだり、プログラムをカスタマイズしたりすることで、PHPの理解が深まり、読者の皆様のスキルアップにつながると思います。

最後に、本書を執筆するにあたってお世話になったスタッフの皆様に心から感謝を申し上げます。そして、本書がPHPでWebアプリケーションの開発を行う読者の皆様に少しでもお役に立てれば、これ以上の幸せはありません。

2012年6月

C&R研究所ライティングスタッフ

蒲生 睦男

# 本書について

## III 動作確認の環境について

本書では、次のような環境で動作を確認しています。

- Windows : PHP4.4.9/5.3.8/5.4.3
- Mac OS X : PHP5.3.1
- Linux(Ubuntu) : PHP5.3.8

## III 本書の表記方法

本書の表記についての注意点は、次のようになります。

### ▶ サンプルコードの中の「\」（バックスラッシュ）と「¥」（円記号）について

Windows環境では、「\」（バックスラッシュ）は「¥」（円記号）で表示されます。本書のサンプルコードでは「\」で表記していますので、Windows環境をお使いの方は読み替えてください。なお、Ubuntuでもアプリケーションによっては「¥」で表示されます。

### ▶ サンプルコードの中の▼について

本書に記載したサンプルコードは、誌面の都合上、1つのサンプルコードがページをまたがって記載されていることがあります。その場合は▼の記号で、1つのコードであることを表しています。

### ▶ バージョンの対応について

本書では項目のタイトル部分に、対応バージョンのマークを記載しています。基本的には記載しているバージョン以降でその項目（関数など）が対応となりますが、一部、例外があります。その場合はHINTに記載しています。なお、薄く表示されている場合は、非対応となります。

### ▶ 動作確認時のURLについて

主にCHAPTER 12の項目で、動作確認のため、URLを直接、入力している箇所があります。入力するURLについては、お使いの環境に合わせて変更してください。

### III サンプルファイルのダウンロードについて

本書のサンプルデータは、C&R研究所のホームページからダウンロードすることができます。  
本書のサンプルを入手するには、次のように操作します。

- ① 「<http://www.c-r.com/>」にアクセスします。
- ② トップページ左上の「商品検索」欄に「109-2」と入力し、[検索] ボタンをクリックします。
- ③ 検索結果が表示されるので、本書の書名のリンクをクリックします。
- ④ 書籍詳細ページが表示されるので、[サンプルデータダウンロード] ボタンをクリックします。
- ⑤ 下記の「ユーザー名」と「パスワード」を入力し、ダウンロードページにアクセスします。
- ⑥ 「サンプルデータ」のリンク先のファイルをダウンロードし、保存します。

#### サンプルのダウンロードに必要な ユーザー名とパスワード

ユーザー名	phpgh
パスワード	9i2h5

※ユーザー名・パスワードは、半角英数字で入力してください。また、「J」と「j」や「K」と「k」などの大文字と小文字の違いもありますので、よく確認して入力してください。

### III サンプルコードの利用方法

サンプルファイルは、CHAPTERごとのフォルダの中に、項目番号のフォルダに分かれています。サンプルはZIP形式で圧縮してありますので、解凍してお使いください。

それぞれのフォルダ内には、ソースファイル(拡張子「.php」)が保存されています。そのファイル(サンプルによってはフォルダごと)をWebサーバーのルートディレクトリに保存し、ブラウザで表示すると、スクリプトが実行されます。環境の構築や実行方法については、CHAPTER 01を参照してください。

## CHAPTER 01 PHPの基礎知識

001	PHPとは .....	38
	COLUMN ■ PHPで作られている代表的なアプリケーション	
002	実行環境について .....	40
	COLUMN ■ Apacheが起動しないときの対処方法	
003	統合開発環境のインストール .....	45
004	コードの記述方法について .....	50
005	PHPマニュアルについて .....	51
	COLUMN ■ HTML版のヘルプ	

## CHAPTER 02 PHPの基本文法

006	型について .....	54
	COLUMN ■ 擬似的な型について	
007	変数について .....	56
	COLUMN ■ シンボルテーブルについて	
008	定数について .....	58
009	演算子について .....	59
	COLUMN ■ 特別な演算子について	
010	コメントについて .....	63
011	条件分岐について .....	64
	COLUMN ■ 「elseif」の別の書き方と注意点	
012	ループ(繰り返し処理)について .....	67
013	関数について .....	70
014	関数の引数について .....	72
015	例外処理について .....	74
016	インクルードファイルについて .....	77
	COLUMN ■ ファイルを探すディレクトリを指定する	

## CHAPTER 03 文字列・マルチバイト文字列

017	文字列について .....	80
	COLUMN ■ 文字列への文字単位のアクセスと修正	
018	正規表現とパターンマッチについて .....	84
	COLUMN ■ PCRE関数の種類	
019	文字列を出力する .....	87
	ONEPOINT ■ 文字列を出力する際は「print」文や「echo」文を使用する	
	COLUMN ■ 関数の定義	

020	文字列をフォーマットして出力する	88
	ONEPOINT ■ 文字列をフォーマットして出力するには 「printf」関数や「vprintf」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ フォーマット文字列における引数の番号付けと交換について	
021	文字列を指定書式でフォーマットする	90
	ONEPOINT ■ 文字列をフォーマットするには 「sprintf」関数や「vsprintf」関数を使用する	
	COLUMN ■ フォーマット文字列の指定子について	
	COLUMN ■ 関数の定義	
022	文字列を指定書式に従って分解する	94
	ONEPOINT ■ 文字列を指定書式に従って分解するには「sscanf」関数を使用する	
	COLUMN ■ 「sscanf」関数で分解した値を直接、変数に格納する	
	COLUMN ■ 関数の定義	
023	文字列をエスケープする	96
	ONEPOINT ■ 文字列をエスケープするにはエスケープ関数を使用する	
	COLUMN ■ 「addslashes」関数の第2引数について	
	COLUMN ■ 関数の定義	
024	文字列をアンエスケープする	98
	ONEPOINT ■ 文字列をアンエスケープするにはアンエスケープ関数を使用する	
	COLUMN ■ 関数の定義	
025	アスキーコードを文字列に変換する	99
	ONEPOINT ■ アスキーコードを文字列に変換するには「chr」関数を使用する	
	COLUMN ■ 関数の定義	
026	文字列をアスキーコードに変換する	100
	ONEPOINT ■ 文字列をアスキーコードに変換するには「ord」関数を使用する	
	COLUMN ■ 関数の定義	
027	文字列を指定文字数で改行する	101
	ONEPOINT ■ 指定文字数で改行するには「chunk_split」関数や 「wordwrap」関数を使用する	
	COLUMN ■ 関数の定義	
028	文字や文字列の出現回数を取得する	103
	ONEPOINT ■ 文字や文字列の出現回数を取得するには 文字をカウントする関数を使用する	
	COLUMN ■ 関数の定義	
029	指定したマルチバイト文字列の出現回数を数える	105
	ONEPOINT ■ マルチバイト文字列の出現回数を数えるには 「mb_substr_count」関数を使用する	
	COLUMN ■ 関数の定義	
030	文字列を分割して配列に格納する	106
	ONEPOINT ■ 文字列を分割して配列に格納するには「explode」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 第3引数「limit」に負の数を指定した場合の動作	



□ 3 1	配列要素を指定した区切り文字で連結する	108
	ONEPOINT ■ 配列要素を指定した区切り文字で連結するには「implode」関数を使用する	
	COLUMN ■ 関数の定義	
□ 3 2	特定の区切り文字で文字列を分割する	109
	ONEPOINT ■ 特定の区切り文字で文字列を分割するには「strtok」関数を使用する	
	COLUMN ■ 関数の定義	
□ 3 3	CSV文字列を分解して配列に格納する	111
	ONEPOINT ■ CSV文字列を分解して配列に格納するには「str_getcsv」関数を使用する	
	COLUMN ■ 関数の定義	
□ 3 4	文字列を配列に変換する	112
	ONEPOINT ■ 文字列を配列に変換するには「str_split」関数を使用する	
	COLUMN ■ 関数の定義	
□ 3 5	文字列から空白を取り除く	114
	ONEPOINT ■ 文字列から空白を取り除くにはトリミング関数を使用する	
	COLUMN ■ 削除する文字を指定する	
	COLUMN ■ 関数の定義	
□ 3 6	文字列を固定長の他の文字列で埋める	116
	ONEPOINT ■ 文字列を固定長の他の文字列で埋めるには「str_pad」関数を使用する	
	COLUMN ■ 関数の定義	
□ 3 7	回数を指定して文字列を繰り返す	118
	ONEPOINT ■ 回数を指定して文字列を繰り返すには「str_repeat」関数を使用する	
	COLUMN ■ 関数の定義	
□ 3 8	文字列を置換する	119
	ONEPOINT ■ 文字列を置換するには文字列置換関数を使用する	
	COLUMN ■ 「str_replace」「str_ireplace」関数で複数の検索・置換文字列を指定する	
	COLUMN ■ 関数の定義	
□ 3 9	文字列を比較する	122
	ONEPOINT ■ 文字列を比較するには文字列比較関数を使用する	
	COLUMN ■ 関数の定義	
□ 4 0	文字列の一部分を比較する	125
	ONEPOINT ■ 文字列の一部分を比較するには「substr_compare」関数を使用する	
	COLUMN ■ 関数の定義	
□ 4 1	文字列の一部分を取得する	127
	ONEPOINT ■ 文字列の一部分を取得するには「substr」関数を使用する	
	COLUMN ■ 関数の定義	
□ 4 2	特定の文字列の前後の文字列を取得する	129
	ONEPOINT ■ 文字列の前後の文字列を取得するには「strstr」関数や「strchr」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 第2引数に文字列以外を指定した場合の動作	

□ 43	マルチバイト文字列中の特定の文字列の前後を取得する	131
	ONEPOINT ■ マルチバイト文字列中の特定文字列の前後を取得するには「mb_strstr」関数や「mb_strchr」関数を使用する	
	COLUMN ■ 関数の定義	
□ 44	指定した文字列が現れる位置を調べる	133
	ONEPOINT ■ 指定した文字列が現れる位置を調べるには「strpos」関数や「strrpos」関数を使用する	
	COLUMN ■ 検索開始位置を指定する	
	COLUMN ■ 関数の定義	
□ 45	マルチバイト文字列中で特定の文字列が現れる位置を調べる	136
	ONEPOINT ■ マルチバイト文字列中で特定の文字列が現れる位置を調べるには「mb_strpos」「mb_strrpos」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「mb_strrpos」関数の引数「encoding」について	
□ 46	文字列の中から任意の文字を探す	138
	ONEPOINT ■ 文字列の中から任意の文字を探すには「strpbrk」関数を使用する	
	COLUMN ■ 関数の定義	
□ 47	文字列の長さを取得する	140
	ONEPOINT ■ 文字列の長さを取得するには「strlen」関数や「mb_strlen」関数を使用する	
	COLUMN ■ 全角文字のバイト数について	
	COLUMN ■ UTF-8の全角文字を2バイトで扱う方法	
	COLUMN ■ 関数の定義	
□ 48	文字列を逆順にする	143
	ONEPOINT ■ 文字列を逆順にするには「strrev」関数を使用する	
	COLUMN ■ 関数の定義	
□ 49	文字列をランダムにシャッフルする	144
	ONEPOINT ■ 文字列をランダムにシャッフルするには「str_shuffle」関数を使用する	
	COLUMN ■ 関数の定義	
□ 50	文字列を16進表現に変換する	145
	ONEPOINT ■ 文字列を16進表現に変換するには「bin2hex」関数を使用する	
	COLUMN ■ 関数の定義	
□ 51	文字列を暗号化する	146
	ONEPOINT ■ 文字列を暗号化するにはハッシュ関数を使用する	
	COLUMN ■ サポートされているハッシュアルゴリズムの一覧を取得する	
	COLUMN ■ 「crypt」関数のハッシュ方式を指定する	
	COLUMN ■ 関数の定義	
	COLUMN ■ ハッシュについて	
□ 52	文字列の大文字と小文字を変換する	150
	ONEPOINT ■ 文字列の大文字と小文字を変換するには「strtolower」「strtoupper」関数を使用する	
	COLUMN ■ 関数の定義	

053	文字列や単語の最初の文字を大文字にする	151
	<b>ONEPOINT</b> ■ 文字列の最初の文字を大文字にするには「ucfirst」、 単語最初の文字を大文字にするには「ucwords」関数を使用する	
	<b>COLUMN</b> ■ 関数の定義	
054	マルチバイト文字列のアルファベットの大文字と小文字を変換する	152
	<b>ONEPOINT</b> ■ マルチバイト文字列を変換するにはマルチバイト文字列変換関数を使用する	
	<b>COLUMN</b> ■ 関数の定義	
	<b>COLUMN</b> ■ マルチバイト文字列関数の文字エンコーディングの設定について	
055	カナを変換する	154
	<b>ONEPOINT</b> ■ カナを変換するには「mb_convert_kana」関数を使用する	
	<b>COLUMN</b> ■ 関数の定義	
056	文字コードを変換する	156
	<b>ONEPOINT</b> ■ 文字コードを変換するには「mb_convert_encoding」関数や 「mb_convert_variables」関数を使用する	
	<b>COLUMN</b> ■ 関数の定義	
	<b>COLUMN</b> ■ 変換前の文字エンコーディングを自動検出する	
057	文字列が特定のエンコーディングで有効かどうかを調べる	158
	<b>ONEPOINT</b> ■ 文字列が特定のエンコーディングで有効かどうかを調べるには 「mb_check_encoding」関数を使用する	
	<b>COLUMN</b> ■ 関数の定義	
058	文字列の文字エンコーディングを調べる	160
	<b>ONEPOINT</b> ■ 文字列の文字エンコーディングを調べるには 「mb_detect_encoding」関数を使用する	
	<b>COLUMN</b> ■ 関数の定義	
059	サポートされているエンコーディングの一覧を取得する	161
	<b>ONEPOINT</b> ■ サポートされているエンコーディングの一覧を取得するには 「mb_list_encodings」関数を使用する	
	<b>COLUMN</b> ■ 関数の定義	
060	文字エンコーディング用の「MIME charset」文字列を取得する	163
	<b>ONEPOINT</b> ■ 「MIME charset」文字列を取得するには 「mb_preferred_mime_name」関数を使用する	
	<b>COLUMN</b> ■ 関数の定義	
061	「mbstring」モジュールの内部設定値を取得する	164
	<b>ONEPOINT</b> ■ 「mbstring」モジュールの内部設定値を取得するには 「mb_get_info」関数を使用する	
	<b>COLUMN</b> ■ 関数の定義	
	<b>COLUMN</b> ■ 「mb_get_info」関数が返す内部設定値	
062	「mbstring」モジュールの基本設定を取得・設定する	166
	<b>ONEPOINT</b> ■ 「mbstring」モジュールの基本設定を取得・設定するには 「mbstring」基本設定関数を使用する	
	<b>COLUMN</b> ■ 関数の定義	

- 63 マルチバイト正規表現の文字エンコーディングを設定・取得する … 168  
**ONEPOINT** ■ マルチバイト正規表現用の文字エンコーディングを設定・取得するには  
「mb\_regex\_encoding」関数を使用する  
**COLUMN** ■ 関数の定義
- 64 マルチバイト正規表現関数のオプションを取得・設定する … 169  
**ONEPOINT** ■ マルチバイト正規表現関数のオプションを取得・設定するには  
「mb\_regex\_set\_options」関数を使用する  
**COLUMN** ■ マルチバイト正規表現関数のオプションについて  
**COLUMN** ■ 関数の定義
- 65 マルチバイト文字列が正規表現に一致するか調べる … 171  
**ONEPOINT** ■ マルチバイト文字列が正規表現に一致するか調べるには  
「mb\_ereg\_match」関数を使用する  
**COLUMN** ■ マルチバイト正規表現関数のオプション設定する  
**COLUMN** ■ 関数の定義
- 66 マルチバイト文字列で正規表現に一致する部分文字列を取得する … 173  
**ONEPOINT** ■ マルチバイト文字列で正規表現に一致する部分文字列を取得するには  
「mb\_ereg」関数を使用する  
**COLUMN** ■ 関数の定義
- 67 マルチバイト文字列を正規表現で置換する … 175  
**ONEPOINT** ■ マルチバイト文字列を正規表現で置換するには  
「mb\_ereg\_replace」関数を使用する  
**COLUMN** ■ 関数の定義
- 68 マルチバイト文字列で正規表現に一致する部分を検索する … 177  
**ONEPOINT** ■ マルチバイト文字列で正規表現による検索を行うには  
「mb\_ereg\_search」関数を使用する  
**COLUMN** ■ 関数の定義
- 69 マルチバイト文字列を正規表現で分割する … 180  
**ONEPOINT** ■ マルチバイト文字列を正規表現で分割するには  
「mb\_split」関数を使用する  
**COLUMN** ■ 関数の定義
- 70 マルチバイト文字列の一部を取得する … 181  
**ONEPOINT** ■ マルチバイト文字列の一部を取得するには「mb\_strcut」関数や  
「mb\_substar」関数を使用する  
**COLUMN** ■ 関数の定義
- 71 マルチバイト文字列を含むクエリ文字列を展開する … 182  
**ONEPOINT** ■ マルチバイト文字列を含むクエリ文字列を展開するには  
「mb\_parse\_str」関数を使用する  
**COLUMN** ■ 関数の定義
- 72 指定した幅でマルチバイト文字列を丸める … 184  
**ONEPOINT** ■ 指定した幅でマルチバイト文字列を丸めるには  
「mb\_strimwidth」関数を使用する  
**COLUMN** ■ 関数の定義

## 073 マルチバイト文字列の幅を取得する ..... 185

**ONEPOINT** ■マルチバイト文字列の幅を取得するには  
「mb\_strwidth」関数を使用する

**COLUMN** ■関数の定義

## CHAPTER 04 数学・日付・時刻

## 074 数値の絶対値を取得する ..... 188

**ONEPOINT** ■数値の絶対値を取得するには「abs」関数を使用する  
**COLUMN** ■関数の定義

## 075 円周率の値を取得する ..... 189

**ONEPOINT** ■円周率の値を取得するには「pi」関数を使用する  
**COLUMN** ■関数の定義

## 076 三角関数を取得する ..... 190

**ONEPOINT** ■三角関数を取得するには「Math」関数の三角関数を使用する  
**COLUMN** ■関数の定義  
**COLUMN** ■双曲線関数について

## 077 度単位とラジアン単位を相互に変換する ..... 192

**ONEPOINT** ■度単位をラジアン単位に変換するには「deg2rad」関数、  
ラジアン単位を度単位に変換するには「rad2deg」関数を使用する  
**COLUMN** ■関数の定義  
**COLUMN** ■定義済みの数学定数について

## 078 直角三角形の斜辺の長さを計算する ..... 194

**ONEPOINT** ■直角三角形の斜辺の長さを計算するには「hypot」関数を使用する  
**COLUMN** ■関数の定義

## 079 べき乗や平方根を計算する ..... 195

**ONEPOINT** ■べき乗を計算するには「pow」関数、  
平方根を計算するには「sqrt」関数を使用する  
**COLUMN** ■関数の定義

## 080 数値の基数を変換する ..... 196

**ONEPOINT** ■数値の基数を変換するには基数変換関数を使用する  
**COLUMN** ■関数の定義

## 081 端数の切り上げ/切り捨てを行う ..... 198

**ONEPOINT** ■端数を切り上げるには「ceil」関数、  
切り捨てるには「floor」関数を使用する  
**COLUMN** ■関数の定義

## 082 浮動小数点数を丸める ..... 199

**ONEPOINT** ■浮動小数点数を丸めるには「round」関数を使用する  
**COLUMN** ■関数の定義

## 083 eの累乗を計算する ..... 201

**ONEPOINT** ■eの累乗を計算するには「exp」関数を使用する  
**COLUMN** ■関数の定義

084	対数を計算する	202
	ONEPOINT ■ 対数を計算するには「log」関数や「log10」関数を使用する	
	COLUMN ■ 関数の定義	
085	割り算の余りを求める	203
	ONEPOINT ■ 割り算の余りを求めるには「fmod」関数を使用する	
	COLUMN ■ 関数の定義	
086	乱数を生成する	204
	ONEPOINT ■ 乱数を生成するには「rand」「mt_rand」「lcg_value」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 乱数生成器にシードを与える	
087	乱数の最大値を取得する	206
	ONEPOINT ■ 乱数の最大値を取得するには「getrandmax」関数や「mt_getrandmax」関数を使用する	
	COLUMN ■ 関数の定義	
088	最大値/最小値を求める	207
	ONEPOINT ■ 最大値を求めるには「max」関数、	
	ONEPOINT ■ 最小値を求めるには「min」関数を使用する	
	COLUMN ■ 関数の定義	
089	任意の精度で数値の演算を行う	208
	ONEPOINT ■ 任意の精度で数値の演算を行うには「BC Math」関数を使用する	
	COLUMN ■ 関数の定義	
090	数値を判定する	211
	ONEPOINT ■ 数値を判定するには「is_」で始まる判定用の関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ フォームの入力値が数値かどうかをチェックする	
091	日付や時刻をフォーマットして取得する	213
	ONEPOINT ■ 日付や時刻をフォーマットするには日付文字列フォーマット関数を使用する	
	COLUMN ■ 「date」関数と「gmdate」関数のフォーマット文字	
	COLUMN ■ 「strftime」関数と「gmstrftime」関数のフォーマット文字	
	COLUMN ■ 関数の定義	
092	日付/時刻情報を取得する	218
	ONEPOINT ■ 日付/時刻情報を取得するには「getdate」関数や「localtime」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「getdate」関数や「localtime」関数が返す連想配列の内容	
093	現在のUnixタイムスタンプを取得する	220
	ONEPOINT ■ 現在のUnixタイムスタンプを取得するには「time」「microtime」「gettimeofday」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「gettimeofday」関数が返す配列の内容	

094	日付/時刻をUnixタイムスタンプに変換する	222
	ONEPOINT ■ 日付/時刻をUnixタイムスタンプに変換するには「mktime」「gmtime」関数を使用する	
	COLUMN ■ 関数の定義	
095	英文形式の日付をUnixタイムスタンプに変換する	224
	ONEPOINT ■ 英文形式の日付をUnixタイムスタンプに変換するには「strptime」関数を使用する	
	COLUMN ■ 関数の定義	
096	指定した日付が有効かどうかを調べる	225
	ONEPOINT ■ 指定した日付が有効かどうかを調べるには「checkdate」関数を使用する	
	COLUMN ■ 関数の定義	
097	処理を一時停止する	226
	ONEPOINT ■ 処理を一時停止するには遅延関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ スクリプトが実行可能な秒数を設定する	

## CHAPTER 05 変数・配列

098	変数を指定の型に変換する	230
	ONEPOINT ■ 変数を指定の型に変換するには型変換関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 型キャストで型を変換する	
099	変数の型を設定する	232
	ONEPOINT ■ 変数の型を設定するには「settype」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 型を変換する方法について	
100	変数が設定されているかどうかを調べる	234
	ONEPOINT ■ 変数が設定されているかを調べるには「empty」関数や「isset」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 定数が設定されているかどうかを調べる	
101	変数を破棄する	236
	ONEPOINT ■ 変数を破棄するには「unset」関数を使用する	
	COLUMN ■ 「unset」関数の内部動作について	
	COLUMN ■ 関数の定義	
102	変数の型を検査する	240
	ONEPOINT ■ 変数の型を検査するには「is_」で始まる判定用の関数を使用する	
	COLUMN ■ 関数の定義	
103	リソース型を調べる	242
	ONEPOINT ■ リソース型を調べるには「get_resource_type」関数を使用する	
	COLUMN ■ 関数の定義	

1 0 4	すべての定義済みの変数を配列で取得する	243
	ONEPOINT ■ すべての定義済みの変数を配列で取得するには 「get_defined_vars」関数を使用する	
	COLUMN ■ 関数の定義	
1 0 5	配列やオブジェクトの中身をわかりやすく出力する	245
	ONEPOINT ■ 配列やオブジェクトの中身をわかりやすく出力するには 「print_r」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ オブジェクトのプロパティの表示について	
1 0 6	変数に関する情報をダンプする	247
	ONEPOINT ■ 変数に関する情報をダンプするには「var_dump」関数を使用する	
	COLUMN ■ 関数の定義	
1 0 7	変数に関する情報を文字列表現で出力する	249
	ONEPOINT ■ 変数に関する情報を文字列表現で出力するには 「var_export」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ クラスを含む配列やクラスをエクスポートする	
1 0 8	配列を作成する	251
	ONEPOINT ■ 配列を作成するには「array」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ インデックスが1から始まる配列を作成する	
1 0 9	変数名とその値から配列を作成する	253
	ONEPOINT ■ 変数名とその値から配列を作成するには「compact」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「compact」関数の注意点	
1 1 0	指定した範囲の整数を有する配列を作成する	255
	ONEPOINT ■ 指定した範囲の整数を有する配列を作成するには 「range」関数を使用する	
	COLUMN ■ 関数の定義	
1 1 1	指定した値で埋めた配列を作成する	257
	ONEPOINT ■ 指定した値で埋めた配列を作成するには「array_fill」関数を使う	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「array_fill」関数の第1引数に負の数を指定した場合の挙動	
1 1 2	指定した長さ・値で配列を埋める	259
	ONEPOINT ■ 指定した長さ・値で配列を埋めるには「array_pad」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「array_pad」関数で一度に追加できる要素の最大数	
1 1 3	2つの配列を結合して配列を作成する	261
	ONEPOINT ■ 2つの配列を結合して配列を作成するには 「array_combine」関数を使用する	
	COLUMN ■ 関数の定義	



1 1 4	配列を統合する	262
	ONEPOINT ■ 配列を統合するには 「array_merge」「array_merge_recursive」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 他の型を配列にマージする	
1 1 5	配列の要素数を数える	264
	ONEPOINT ■ 配列の要素数を数えるには「count」関数を使用する	
	COLUMN ■ 関数の定義	
1 1 6	配列の値の数を数える	265
	ONEPOINT ■ 配列の値の数を数えるには「array_count_values」関数を使用する	
	COLUMN ■ 関数の定義	
1 1 7	配列の内部ポインタを操作する	267
	ONEPOINT ■ 配列の内部ポインタを操作するには 「next」「prev」「reset」「end」関数を使用する	
	COLUMN ■ 関数の定義	
1 1 8	配列の値を別々の変数に代入する	269
	ONEPOINT ■ 配列の値を別々の変数に代入するには「list」関数を使用する	
	COLUMN ■ 関数の定義	
1 1 9	配列のキーと値のペアを取得する	270
	ONEPOINT ■ 配列のキーと値のペアを取得するには「each」関数や「foreach」文を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「foreach」文の定義	
1 2 0	配列からキーを取り出す	272
	ONEPOINT ■ 配列からキーを取り出すには「key」関数を使用する	
	COLUMN ■ 関数の定義	
1 2 1	配列を変数に展開する	273
	ONEPOINT ■ 配列を変数に展開するには「extract」関数を使用する	
	COLUMN ■ 関数の定義	
1 2 2	配列をキーで並べ替える	275
	ONEPOINT ■ 配列をキーで並べ替えるには「ksort」関数や「krsort」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 配列ソート関数の特性	
1 2 3	配列を並べ替える	277
	ONEPOINT ■ 配列を並べ替えるには配列ソート関数を使用する	
	COLUMN ■ 「sort_flags」引数について	
	COLUMN ■ 関数の定義	
1 2 4	複数の配列を関連付けて並べ替える	280
	ONEPOINT ■ 複数の配列を関連付けて並べ替えるには 「array_multisort」関数を使用する	
	COLUMN ■ 多次元の配列をソートする	
	COLUMN ■ 関数の定義	

1 2 5	配列をシャッフルする	284
	ONEPOINT ■ 配列をシャッフルするには「shuffle」関数を使用する	
	COLUMN ■ 関数の定義	
1 2 6	配列の要素をランダムに取得する	285
	ONEPOINT ■ 配列の要素をランダムに取得するには「array_rand」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 乱数生成器にシードを与える	
1 2 7	配列の値の合計・積を計算する	287
	ONEPOINT ■ 配列の値の合計を計算するには「array_sum」関数、 積を計算するには「array_product」関数を使用する	
	COLUMN ■ 関数の定義	
1 2 8	配列のキーを大文字/小文字に変換する	288
	ONEPOINT ■ 配列のキーを大文字/小文字に変換するには 「array_change_key_case」関数を使用する	
	COLUMN ■ 関数の定義	
1 2 9	配列のキーと値を入れ替える	289
	ONEPOINT ■ 配列のキーと値を入れ替えるには「array_flip」関数を使用する	
	COLUMN ■ 関数の定義	
1 3 0	配列の要素を逆順にする	291
	ONEPOINT ■ 配列の要素を逆順にするには「array_reverse」関数を使用する	
	COLUMN ■ 関数の定義	
1 3 1	配列を分割する	293
	ONEPOINT ■ 配列を分割するには「array_chunk」関数を使用する	
	COLUMN ■ 関数の定義	
1 3 2	配列の一部を展開する	295
	ONEPOINT ■ 配列の一部を展開するには「array_slice」関数を使用する	
	COLUMN ■ 関数の定義	
1 3 3	配列に値があるかどうかを調べる	297
	ONEPOINT ■ 配列に値があるかどうかを調べるには「in_array」関数を使用する	
	COLUMN ■ 関数の定義	
1 3 4	指定したキー・添字が配列にあるかを調べる	299
	ONEPOINT ■ 指定したキー・添字が配列にあるかを調べるには 「array_key_exists」関数を使用する	
	COLUMN ■ 関数の定義	
1 3 5	配列を検索する	300
	ONEPOINT ■ 配列を検索するには「array_search」関数を使用する	
	COLUMN ■ 関数の定義	
1 3 6	配列のキー・値をすべて取得する	301
	ONEPOINT ■ 配列のキーをすべて取得するには「array_keys」関数、 値をすべてを取得するには「array_values」関数を使用する	
	COLUMN ■ 指定した値を含むキーを取得する	
	COLUMN ■ 関数の定義	

1 3 7	配列の差や共通項を取得する	303
	ONEPOINT ■ 配列の差や共通項を取得するには 「array_diff」「array_intersect」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ ユーザー定義のコールバック関数で配列を差や共通項を取得する関数	
1 3 8	配列の先頭/末尾の要素を取り出す	306
	ONEPOINT ■ 配列の先頭の要素を取り出すには「array_shift」関数、 末尾の要素を取り出すには「array_pop」関数を使用する	
	COLUMN ■ 関数の定義	
1 3 9	配列の先頭/最後に要素を追加する	308
	ONEPOINT ■ 配列の先頭に要素を追加するには「array_unshift」関数、 最後に要素を追加するには「array_push」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 代入式で配列の要素を追加する	
1 4 0	配列の要素を置換する	310
	ONEPOINT ■ 配列の要素を置換するには「array_replace」関数を使用する	
	COLUMN ■ 関数の定義	
1 4 1	配列の一部を削除して他の要素で置換する	312
	ONEPOINT ■ 配列の一部を削除して他の要素で置換するには 「array_splice」関数を使用する	
	COLUMN ■ 関数の定義	
1 4 2	配列から重複した値を削除する	314
	ONEPOINT ■ 配列から重複した値を削除するには「array_unique」関数を使用する	
	COLUMN ■ 関数の定義	
1 4 3	指定した配列の要素にコールバック関数を適用する	315
	ONEPOINT ■ 指定した配列の要素にコールバック関数を適用するには 「array_map」関数を使用する	
	COLUMN ■ 「array_map」関数で2つ以上の配列を使用する	
	COLUMN ■ 関数の定義	
1 4 4	配列の全要素にユーザー定義関数を適用する	317
	ONEPOINT ■ 配列の全要素にユーザー定義関数を適用するには 「array_walk」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ コールバック関数によって配列の値そのものを変更する方法	

## CHAPTER 06 クラス・オブジェクト

1 4 5	オブジェクト指向について	320
1 4 6	クラスを定義する	321
	ONEPOINT ■ クラスを定義するには「class」キーワードを使用する	
	COLUMN ■ クラスの定義方法	
	COLUMN ■ PHP4でのクラス定義	

1 4 7	クラスのオブジェクトを生成する	323
	ONEPOINT ■ クラスのオブジェクトを生成するには「new」キーワードを使用する	
	COLUMN ■ キーワードの定義	
	COLUMN ■ 可変変数を使ってオブジェクトを作成する	
1 4 8	プロパティを定義する	325
	ONEPOINT ■ プロパティを定義するにはアクセス指定子の後ろに変数を宣言する	
	COLUMN ■ アクセス権について	
	COLUMN ■ プロパティの定義	
1 4 9	メソッドを定義する	329
	ONEPOINT ■ メソッドを定義するにはアクセス指定子の後に関数を宣言する	
	COLUMN ■ メソッドのアクセス権を確認する	
	COLUMN ■ メソッドの定義	
1 5 0	オブジェクトのプロパティやメソッドにアクセスする	332
	ONEPOINT ■ プロパティやメソッドにアクセスするにはアロー演算子「->」を使用する	
	COLUMN ■ クラスメソッドからプロパティにアクセスする	
1 5 1	静的なプロパティ・メソッドを定義する	334
	ONEPOINT ■ 静的なプロパティ・メソッドを定義するには「static」キーワードを使用する	
	COLUMN ■ 「self」と「parent」キーワードについて	
	COLUMN ■ 静的なプロパティ・メソッドの定義	
1 5 2	コンストラクタ・デストラクタを定義する	338
	ONEPOINT ■ コンストラクタを定義するには「__construct」メソッド、デストラクタを定義するには「__destruct」メソッドを使用する	
	COLUMN ■ メソッドの定義	
	COLUMN ■ 古い形式のコンストラクタを定義する	
1 5 3	オブジェクト定数を定義する	340
	ONEPOINT ■ オブジェクト定数を定義するには「const」キーワードを使用する	
	COLUMN ■ インターフェイス定数を定義する	
1 5 4	クラスを継承する	342
	ONEPOINT ■ クラスを継承するには「extends」キーワードを使用する	
	COLUMN ■ 継承クラスに新しいメンバーを追加する	
	COLUMN ■ 派生クラスの定義方法	
1 5 5	プロパティやメソッドをオーバーライドする	345
	ONEPOINT ■ プロパティやメソッドをオーバーライドするには派生クラスで再定義する	
	COLUMN ■ メソッドをオーバーライドする際に引数を追加する際の注意点	
	COLUMN ■ 派生クラスから基底クラスのメソッドにアクセスする	
	COLUMN ■ クラスやメソッドのオーバーライドを防止する	
1 5 6	プロパティをオーバーロードする	348
	ONEPOINT ■ プロパティをオーバーロードするにはマジックオーバーロードメソッドを使用する	
	COLUMN ■ プロパティのオーバーロードを防止する	
	COLUMN ■ メソッドの定義	

157	メソッドをオーバーロードする	352
	ONEPOINT ■メソッドをオーバーロードするには「__call」メソッドを使用する	
	COLUMN ■未定義のメソッドを静的コンテキストで実行したときにオーバーロードする	
	COLUMN ■メソッドの定義	
158	抽象クラス・抽象メソッドを定義する	355
	ONEPOINT ■抽象クラス・抽象メソッドを定義するには「abstract」キーワードを使用する	
	COLUMN ■抽象クラス・抽象メソッドの定義方法	
159	インターフェイスを定義する	357
	ONEPOINT ■インターフェイスを定義するには「interface」キーワードを使用する	
	COLUMN ■1つのクラスに複数のインターフェイスを実装する	
	COLUMN ■インターフェイスを継承する	
	COLUMN ■インターフェイスの定義方法	
160	オブジェクトのコピーを作成する	363
	ONEPOINT ■オブジェクトのコピーを作成するには「clone」キーワードを使用する	
	COLUMN ■プロパティに必要な変更を実行する	
	COLUMN ■メソッドの定義	
161	クラスのエイリアスを作成する	366
	ONEPOINT ■クラスのエイリアスを作成するには「class_alias」関数を使用する	
	COLUMN ■関数の定義	
162	クラスが定義済みかどうかを確認する	368
	ONEPOINT ■クラスが定義済みかどうかを確認するには「class_exists」関数を使用する	
	COLUMN ■関数の定義	
163	インターフェイスが定義済みかどうかを確認する	369
	ONEPOINT ■インターフェイスが定義済みかどうかを確認するには「interface_exists」関数を使用する	
	COLUMN ■関数の定義	
164	クラスメソッドの存在をチェックする	370
	ONEPOINT ■クラスメソッドの存在をチェックするには「method_exists」関数を使用する	
	COLUMN ■関数の定義	
165	プロパティの存在をチェックする	371
	ONEPOINT ■プロパティの存在をチェックするには「property_exists」関数を使用する	
	COLUMN ■関数の定義	
166	クラスメソッドの名前を取得する	372
	ONEPOINT ■クラスメソッドの名前を取得するには「get_class_methods」関数を使用する	
	COLUMN ■関数の定義	
167	クラスのプロパティを取得する	374
	ONEPOINT ■クラスのプロパティを取得するには「get_class_vars」関数を使用する	
	COLUMN ■関数の定義	

1 6 8	オブジェクトのプロパティを取得する	376
	ONEPOINT ■ オブジェクトのプロパティを取得するには 「get_object_vars」関数を使用する	
	COLUMN ■ 関数の定義	
1 6 9	オブジェクトのクラス名を取得する	378
	ONEPOINT ■ オブジェクトのクラス名を取得するには「get_class」関数を使用する	
	COLUMN ■ 関数の定義	
1 7 0	オブジェクトの親クラスの名前を取得する	379
	ONEPOINT ■ オブジェクトの親クラスの名前を取得するには 「get_parent_class」関数を使用する	
	COLUMN ■ 関数の定義	
1 7 1	オブジェクトを検査する	381
	ONEPOINT ■ オブジェクトを検査するには「is_a」関数や「instanceof」演算子を使用する	
	COLUMN ■ 関数の定義	
1 7 2	オブジェクトをシリアル化/アンシリアル化する	383
	ONEPOINT ■ オブジェクトをシリアル化するには「serialize」関数、 アンシリアル化するには「unserialize」関数を使用する	
	COLUMN ■ 関数の定義	

## CHAPTER 07 ファイル・ディレクトリ

1 7 3	カレントディレクトリを変更/取得する	386
	ONEPOINT ■ カレントディレクトリを変更するには「chdir」関数、取得するには 「getcwd」関数を使用する	
	COLUMN ■ 関数の定義	
1 7 4	ファイルやディレクトリのリストを取得する	388
	ONEPOINT ■ ファイルやディレクトリのリストを取得するには「dir」クラスを使用する	
	COLUMN ■ 手続き型の関数でファイルやディレクトリのリストを取得する	
	COLUMN ■ 「scandir」関数でファイルやディレクトリのリストを取得する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「glob」関数でファイルやディレクトリのリストを取得する	
1 7 5	パスの最後にある名前の部分を取得する	392
	ONEPOINT ■ パスの最後にある名前の部分を取得するには 「basename」関数を使用する	
	COLUMN ■ 関数の定義	
1 7 6	親ディレクトリのパスを取得する	393
	ONEPOINT ■ 親ディレクトリのパスを取得するには「dirname」関数を使用する	
	COLUMN ■ 関数の定義	
1 7 7	ファイルパスに関する情報を取得する	394
	ONEPOINT ■ ファイルパスに関する情報を取得するには「pathinfo」関数を使用する	
	COLUMN ■ 関数の定義	

178	絶対パスを取得する	396
	ONEPOINT ■ 絶対パスを取得するには「realpath」関数を使用する	
	COLUMN ■ 関数の定義	
179	ディスク容量を取得する	397
	ONEPOINT ■ ディスク容量を取得するには「disk_free_space」関数や「disk_total_space」関数を使用する	
	COLUMN ■ 関数の定義	
180	ファイルのステータスのキャッシュをクリアする	398
	ONEPOINT ■ ファイルのステータスのキャッシュをクリアするには「clearstatcache」関数を使用する	
	COLUMN ■ 関数の定義	
181	ファイルのパーミッションを設定・取得する	400
	ONEPOINT ■ ファイルのパーミッションを設定するには「chmod」関数、取得するには「fileperms」関数を使用する	
	COLUMN ■ ファイルのパーミッションをわかりやすい完全な文字列表示形式に変換する	
	COLUMN ■ 関数の定義	
182	ファイルをコピーする	403
	ONEPOINT ■ ファイルをコピーするには「copy」関数を使用する	
	COLUMN ■ 関数の定義	
183	ファイル名を変更する	405
	ONEPOINT ■ ファイル名を変更するには「rename」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ ファイルを移動する	
184	ファイルを削除する	407
	ONEPOINT ■ ファイルを削除するには「unlink」関数を使用する	
	COLUMN ■ 関数の定義	
185	ファイルやURLをオープンする	408
	ONEPOINT ■ ファイルやURLをオープンするには「fopen」関数を使用する	
	COLUMN ■ 「fopen」関数の第2引数の設定値	
	COLUMN ■ 関数の定義	
186	開いているファイルを閉じる	410
	ONEPOINT ■ 開いているファイルを閉じるには「fclose」関数を使用する	
	COLUMN ■ 関数の定義	
187	ファイル終端かどうかを調べる	412
	ONEPOINT ■ ファイル終端かどうかを調べるには「feof」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「feof」関数を使用する際の注意点	
188	ファイルを読み込む	414
	ONEPOINT ■ ファイルを読み込むには「fgetc」関数や「fgets」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「fgets」関数の第2引数「length」について	
	COLUMN ■ Mac上で作成されたファイルで行末が認識できない場合の対処方法	

189	バイナリファイルを読み込む .....	416
	ONEPOINT ■ バイナリファイルを読み込むには「fread」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「fread」関数を使用する際の注意点	
190	フォーマットに従ってファイルを読み込む .....	418
	ONEPOINT ■ フォーマットに従ってファイルを読み込むには「fscanf」関数を使用する	
	COLUMN ■ 「fscanf」関数の処理値を変数に直接代入する	
	COLUMN ■ 関数の定義	
191	ファイルの全体を読み込む .....	420
	ONEPOINT ■ ファイルの全体を読み込むには「file」関数や「file_get_contents」関数を使用する	
	COLUMN ■ 関数の定義	
192	ファイルの内容をそのまま出力する .....	422
	ONEPOINT ■ ファイルの内容をそのまま出力するには「readfile」関数を使用する	
	COLUMN ■ 関数の定義	
193	ファイルの読み残しデータをすべて出力する .....	423
	ONEPOINT ■ ファイルの読み残しデータをすべて出力するには「fpassthru」関数を使用する	
	COLUMN ■ 画像をブラウザに出力する	
	COLUMN ■ 関数の定義	
194	ファイルに書き込む .....	425
	ONEPOINT ■ ファイルに書き込むには「fwrite」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「fwrite」関数を使用する際の注意点	
195	フォーマットされた文字列をファイルに書き込む .....	427
	ONEPOINT ■ フォーマットされた文字列をファイルに書き込むには「fprintf」「vfprintf」関数を使用する	
	COLUMN ■ 関数の定義	
196	簡単な手順で文字列をファイルに書き込む .....	429
	ONEPOINT ■ 簡単な手順でファイルに書き込むには「file_put_contents」関数を使用する	
	COLUMN ■ ファイルに文字列を追記する	
	COLUMN ■ 関数の定義	
197	ファイルをロックする .....	431
	ONEPOINT ■ ファイルをロックするには「flock」関数を使用する	
	COLUMN ■ ロック中に「flock」関数でブロックを行わないようにする	
	COLUMN ■ 関数の定義	
	COLUMN ■ ファイルのロックについて	
198	ファイルポインタの位置を移動する .....	434
	ONEPOINT ■ ファイルポインタを移動するには「fseek」関数を使用する	
	COLUMN ■ ファイルポインタをファイルの終端から数えた位置に移動する	
	COLUMN ■ 関数の定義	



199	ファイルを指定したサイズに丸める	437
	ONEPOINT ■ ファイルを指定したサイズに丸めるには「ftruncate」関数を使用する	
	COLUMN ■ 関数の定義	
200	一意な名前のテンポラリファイルを作成する	439
	ONEPOINT ■ テンポラリファイルを作成するには「tempnam」関数を使用する	
	COLUMN ■ 関数の定義	
201	自動削除されるテンポラリファイルを作成する	440
	ONEPOINT ■ 自動削除されるテンポラリファイルを作成するには「tmpfile」関数を使用する	
	COLUMN ■ 関数の定義	
202	ディレクトリを作成する	442
	ONEPOINT ■ ディレクトリを作成するには「mkdir」関数を使用する	
	COLUMN ■ 関数の定義	
203	ディレクトリを削除する	444
	ONEPOINT ■ ディレクトリを削除するには「rmdir」関数を使用する	
	COLUMN ■ 関数の定義	
204	ファイルやディレクトリの存在をチェックする	446
	ONEPOINT ■ ファイルやディレクトリの存在をチェックするには「file_exists」関数を使用する	
	COLUMN ■ 関数の定義	
205	ファイルやディレクトリを調べる	447
	ONEPOINT ■ ファイルやディレクトリを調べるには「is」で始まる判別用関数を使用する	
	COLUMN ■ 関数の定義	
206	ファイルの情報を取得する	449
	ONEPOINT ■ ファイルの情報を取得するにはファイル統計情報取得関数を使用する	
	COLUMN ■ 関数の定義	
207	ファイルの最終アクセス時刻・更新日を設定する	451
	ONEPOINT ■ ファイルの最終アクセス時刻・更新日を設定するには「touch」関数を使用する	
	COLUMN ■ 関数の定義	
208	ファイルに関する統計情報を取得する	453
	ONEPOINT ■ ファイルの統計情報を取得するには「stat」関数や「fststat」関数を使用する	
	COLUMN ■ 「stat」関数の結果フォーマット	
	COLUMN ■ 関数の定義	
209	ワイルドカードでファイルを探す	456
	ONEPOINT ■ ワイルドカードでファイルを探すには「fnmatch」関数を使用する	
	COLUMN ■ 関数の定義	
210	パターンにマッチするファイルやディレクトリを探す	458
	ONEPOINT ■ パターンにマッチするファイルやディレクトリを探すには「glob」関数を使用する	
	COLUMN ■ 関数の定義	

## CHAPTER 08 画像

- 2 1 1 GDライブラリに関する情報を取得する** ..... 460  
**ONEPOINT** ■ GDライブラリに関する情報を取得するには「gd\_info」関数を使用する  
**COLUMN** ■ 関数の定義
- 2 1 2 サポートしている画像形式を取得する** ..... 462  
**ONEPOINT** ■ サポートしている画像形式を取得するには「imagetypes」関数を使用する  
**COLUMN** ■ 関数の定義  
**COLUMN** ■ 画像形式を表す定数について
- 2 1 3 画像の大きさを取得する** ..... 464  
**ONEPOINT** ■ 画像の大きさを取得するには「getimagesize」関数を使用する  
**COLUMN** ■ 関数の定義  
**COLUMN** ■ 「getimagesize」関数の返り値の「mime」キーの使用方法
- 2 1 4 画像の幅と高さを取得する** ..... 466  
**ONEPOINT** ■ 画像の幅を取得する「imagesx」関数、  
高さを取得するには「imagesy」関数を使用する  
**COLUMN** ■ 関数の定義
- 2 1 5 指定した画像形式に対応する拡張子を取得する** ..... 467  
**ONEPOINT** ■ 画像形式に対応する拡張子を取得するには  
「image\_type\_to\_extension」関数を使用する  
**COLUMN** ■ 関数の定義
- 2 1 6 指定した画像形式に対応するMIMEタイプを取得する** ..... 469  
**ONEPOINT** ■ 指定の画像形式に対応するMIMEタイプを取得するには  
「image\_type\_to\_mime\_type」関数を使用する  
**COLUMN** ■ 関数の定義
- 2 1 7 新規画像を作成する** ..... 471  
**ONEPOINT** ■ 新規画像を作成するには「imagecreatetruecolor」「imagecreate」  
関数を使用する  
**COLUMN** ■ 関数の定義
- 2 1 8 ファイルやURLから新規画像を作成する** ..... 473  
**ONEPOINT** ■ ファイルやURLから新規画像を作成するには  
「imagecreatefrom」関数を使用する  
**COLUMN** ■ 関数の定義
- 2 1 9 画像データを含む文字列から新規画像を作成する** ..... 475  
**ONEPOINT** ■ 画像データを含む文字列から画像を作成するには  
「imagecreatefromstring」関数を使用する  
**COLUMN** ■ 関数の定義
- 2 2 0 画像をブラウザ/ファイルに出力する** ..... 477  
**ONEPOINT** ■ 画像をブラウザ/ファイルに出力するには画像出力関数を使用する  
**COLUMN** ■ 関数の定義
- 2 2 1 画像を破棄する** ..... 479  
**ONEPOINT** ■ 画像を破棄するには「imagedestroy」関数を使用する  
**COLUMN** ■ 関数の定義

2 2 2	画像で使用する色を作成する	480
	ONEPOINT ■ 画像で使用する色を作成するには 「imagecolorallocate」関数を使用する	
	COLUMN ■ 関数の定義	
2 2 3	透明色を設定する	482
	ONEPOINT ■ 透明色を設定するには「imagecolortransparent」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 透明色を設定する際の注意点	
2 2 4	イメージの色リソースを開放する	484
	ONEPOINT ■ イメージの色リソースを開放するには 「imagecolordeallocate」関数を使用する	
	COLUMN ■ 関数の定義	
2 2 5	線を描画する	486
	ONEPOINT ■ 線を描画するには「imageline」関数や「imagedashedline」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「imageline」関数と「imagesetstyle」の組み合わせで破線を描画する	
2 2 6	多角形や矩形を描画する	488
	ONEPOINT ■ 多角形を描画するには「imagepolygon」関数、 矩形を描画するには「imagerectangle」関数を使用する	
	COLUMN ■ 関数の定義	
2 2 7	アンチエイリアス機能を有効にする	490
	ONEPOINT ■ アンチエイリアス機能を有効にするには 「imageantialias」関数を使用する	
	COLUMN ■ 関数の定義	
2 2 8	円や円弧を描画する	492
	ONEPOINT ■ 円や円弧を描画するには「imagearc」関数や「imageellipse」関数を使用する	
	COLUMN ■ 「imagefilledarc」関数の「style」引数について	
	COLUMN ■ 関数の定義	
2 2 9	文字を描画する	496
	ONEPOINT ■ 文字を描画するには「imagechar」関数や「imagecharup」関数を使用する	
	COLUMN ■ 関数の定義	
2 3 0	点を描画する	498
	ONEPOINT ■ 点を描画するには「imagesetpixel」関数を使用する	
	COLUMN ■ 関数の定義	
2 3 1	線描画用にブラシイメージを設定する	500
	ONEPOINT ■ 線描画用にブラシイメージを設定するには「imagesetbrush」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ ブラシを使用する際の注意点	

232	線描画用のスタイルを設定する	502
	ONEPOINT ■ 線描画用のスタイルを設定するには「imagesetstyle」関数を使用する	
	COLUMN ■ スタイルに透明なピクセルを追加する	
	COLUMN ■ 関数の定義	
233	線描画用の線幅を設定する	505
	ONEPOINT ■ 線描画用の線幅を設定するには「imagesetthickness」関数を使用する	
	COLUMN ■ 関数の定義	
234	図形の内部を塗りつぶす	507
	ONEPOINT ■ 図形内部を塗りつぶすには「imagefill」関数や「imagefilltoborder」関数を使用する	
	COLUMN ■ 関数の定義	
235	塗りつぶし用のタイルを設定する	509
	ONEPOINT ■ 塗りつぶし用のタイルを設定するには「imagesettile」関数を使用する	
	COLUMN ■ 背景が透けて見えるタイルを作成する	
	COLUMN ■ 関数の定義	
	COLUMN ■ タイルを使用する際の注意点	
236	特定位置にあるピクセルのカラーインデックスを取得する	512
	ONEPOINT ■ ピクセルのカラーインデックスを取得するには「imagecolorat」関数を使用する	
	COLUMN ■ 関数の定義	
237	指定したカラーインデックスの色を設定する	514
	ONEPOINT ■ 指定したカラーインデックスの色を設定するには「imagecolorset」関数を使用する	
	COLUMN ■ 関数の定義	
238	カラーインデックスからRGB値を取得する	516
	ONEPOINT ■ カラーインデックスからRGB値を取得するには「imagecolorsforindex」関数を使用する	
	COLUMN ■ 関数の定義	
239	画像パレットの色数を取得する	517
	ONEPOINT ■ 画像パレットの色数を取得するには「imagecolorstotal」関数を使用する	
	COLUMN ■ 関数の定義	
240	パレットをコピーする	518
	ONEPOINT ■ パレットをコピーするには「imagepalettecopy」関数を使用する	
	COLUMN ■ 関数の定義	
241	画像の一部をコピーする	520
	ONEPOINT ■ 画像の一部をコピーするには「imagecopy」関数を使用する	
	COLUMN ■ 関数の定義	
242	画像をコピーしてマージする	522
	ONEPOINT ■ 画像をコピーしてマージするには「imagecopymerge」関数や「imagecopymergegray」関数を使用する	
	COLUMN ■ 関数の定義	

2 4 3	画像のサムネイルを作成する	524
	ONEPOINT ■ 画像のサムネイルを作成するには「imagecopyresized」関数や「imagecopyresampled」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「imagecopyresized」関数と「imagecopyresampled」関数の引数について	
	COLUMN ■ 「imagecopymerge」関数と「imagecopymergegray」関数の引数について	
2 4 4	画像を回転する	527
	ONEPOINT ■ 画像を回転するには「imagerotate」関数を使用する	
	COLUMN ■ 関数の定義	
2 4 5	インターレース画像を作成する	529
	ONEPOINT ■ インターレース画像を作成するには「imageinterlace」関数を使用する	
	COLUMN ■ 関数の定義	
2 4 6	画像にガンマ補正を適用する	531
	ONEPOINT ■ 画像にガンマ補正を適用するには「imagegammacorrect」関数を使用する	
	COLUMN ■ 関数の定義	
2 4 7	TrueColorイメージをパレットイメージに変換する	532
	ONEPOINT ■ TrueColorイメージをパレットイメージに変換するには「imagetruecolortopalette」関数を使用する	
	COLUMN ■ 関数の定義	
2 4 8	画像に畳み込みフィルタを適用する	534
	ONEPOINT ■ 画像に畳み込みフィルタを適用するには「imageconvolution」関数を使用する	
	COLUMN ■ 畳み込みフィルタでシャープネスやぼかしを適用する	
	COLUMN ■ 関数の定義	
2 4 9	画像にフィルタを適用する	537
	ONEPOINT ■ 画像にフィルタを適用するには「imagefilter」関数を使用する	
	COLUMN ■ 関数の定義	

## CHAPTER 09 データ

2 5 0	CSV形式のファイルを読み込む	540
	ONEPOINT ■ CSV形式のファイルを読み込むには「fgetcsv」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「fgetcsv」関数を使用する際の注意点	
2 5 1	CSV形式でファイルに書き込む	542
	ONEPOINT ■ CSV形式でファイルに書き込むには「fputcsv」関数を使用する	
	COLUMN ■ 関数の定義	
2 5 2	配列や連想配列の値をJSON形式にエンコードする	544
	ONEPOINT ■ 配列や連想配列の値をJSONエンコードするには「json_encode」関数を使用する	
	COLUMN ■ 関数の定義	

253	JSON形式の文字列をデコードする .....	546
	ONEPOINT ■ JSON形式の文字列をデコードするには 「json_decode」関数を使用する	
	COLUMN ■ 数の定義	
254	JSONエンコード/デコード処理中に発生したエラーを取得する ...	548
	ONEPOINT ■ JSONエンコード/デコード中に発生したエラーを取得するには 「json_last_error」関数を使用する	
	COLUMN ■ 関数の定義	
255	データをパック/アンパックする .....	550
	ONEPOINT ■ データをパックするには「pack」関数、アンパックするには 「unpack」関数を使用する	
	COLUMN ■ 「pack」関数と「unpack」関数のフォーマットの書式コード	
	COLUMN ■ 関数の定義	
256	XMLファイルを読み込む .....	552
	ONEPOINT ■ XMLファイルを読み込むには「simplexml_load_file」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ SimpleXMLについて	
257	DOMノードからSimpleXMLノードを作成する .....	554
	ONEPOINT ■ DOMノードからSimpleXMLノードを作成するには 「simplexml_import_dom」関数を使用する	
	COLUMN ■ 関数の定義	
258	XML文字列をオブジェクトに代入する .....	556
	ONEPOINT ■ XML文字列をオブジェクトに代入するには 「simplexml_load_string」関数を使用する	
	COLUMN ■ メソッドの定義	
259	ファイルからSimpleXMLのオブジェクトを作成する .....	558
	ONEPOINT ■ SimpleXMLのオブジェクトを作成するには 「SimpleXMLElement」クラスを使用する	
	COLUMN ■ 関数の定義	
260	XPathクエリで要素を取得する .....	560
	ONEPOINT ■ XPathクエリで要素を取得するには「xpath」メソッドを使用する	
	COLUMN ■ メソッドの定義	
261	XPathクエリ用の名前空間プレフィックスを作成する .....	562
	ONEPOINT ■ XPathクエリ用の名前空間プレフィックスを作成するには 「registerXPathNamespace」メソッドを使用する	
	COLUMN ■ 関数の定義	
262	XMLタグの中で定義された属性とその値を取得する .....	564
	ONEPOINT ■ XMLタグ中の属性と値を取得するには 「attributes」メソッドを使用する	
	COLUMN ■ メソッドの定義	
263	XML要素に属性を追加する .....	565
	ONEPOINT ■ XML要素に属性を追加するには「addAttribute」メソッドを使用する	
	COLUMN ■ メソッドの定義	

264	XMLノードに子要素を追加する	567
	ONEPOINT ■ XMLノードに子要素を追加する「addChild」メソッドを使用する	
	COLUMN ■ メソッドの定義	
265	指定した要素の子ノードを見つける	569
	ONEPOINT ■ 指定した要素の子ノードを見つけるには「children」メソッドを使用する	
	COLUMN ■ 名前空間やプレフィックスを指定して子ノードを見つける	
	COLUMN ■ メソッドの定義	
266	子要素の数を取得する	572
	ONEPOINT ■ 子要素の数を取得するには「count」メソッドを使用する	
	COLUMN ■ メソッドの定義	
267	XML要素の名前を取得する	574
	ONEPOINT ■ XML要素の名前を取得するには「getName」メソッドを使用する	
	COLUMN ■ メソッドの定義	
268	名前空間を取得する	576
	ONEPOINT ■ 名前空間を取得するには 「getDocNamespaces」「getNamespaces」メソッドを使用する	
	COLUMN ■ 関数の定義	
269	XMLオブジェクトをXML文字列で出力する	578
	ONEPOINT ■ XMLオブジェクトをXML文字列で出力するには 「asXML」メソッドを使用する	
	COLUMN ■ メソッドの定義	

## CHAPTER 10 Web・ネットワーク

270	URIを解析する	582
	ONEPOINT ■ URIを解析するには「parse_url」関数と「parse_str」関数を使用する	
	COLUMN ■ 関数の定義	
271	URLエンコード/デコードする	584
	ONEPOINT ■ URLエンコードするには「urlencode」関数、 URLデコードするには「urldecode」関数を使用する	
	COLUMN ■ 関数の定義	
272	URLエンコードされたクエリ文字列を生成する	586
	ONEPOINT ■ URLエンコードされたクエリ文字列を生成するには 「http_build_query」関数を使用する	
	COLUMN ■ URLエンコード形式を指定する	
	COLUMN ■ 関数の定義	
273	base64エンコード/デコードする	588
	ONEPOINT ■ base64エンコードするには「base64_encode」関数、 base64デコードするには「base64_decode」関数を使用する	
	COLUMN ■ 関数の定義	

274	特殊文字とHTMLエンティティを相互に変換する	589
	ONEPOINT ■ 特殊文字とHTMLエンティティを相互変換するには「htmlspecialchars」関数と「htmlspecialchars_decode」関数を使用する	
	COLUMN ■ 「htmlspecialchars」関数と「htmlspecialchars_decode」関数の第2引数について	
	COLUMN ■ 「htmlspecialchars」関数で変換に使用される文字セットを指定する	
	COLUMN ■ 変換済みのHTMLエンティティを二重エンコードしないように設定する	
	COLUMN ■ 関数の定義	
275	ファイルから1行取り出してHTMLタグを取り除く	593
	ONEPOINT ■ ファイルから1行取り出してHTMLタグを取り除くには「fgets」関数を使用する	
	COLUMN ■ 関数の定義	
	COLUMN ■ Mac上で作成されたファイルで行末が認識できない場合の対処方法	
276	文字列からHTMLタグやPHPタグを取り除く	595
	ONEPOINT ■ 文字列からHTMLタグやPHPタグを取り除くには「strip_tags」関数を使用する	
	COLUMN ■ 関数の定義	
277	改行文字の前にHTMLの改行タグを挿入する	596
	ONEPOINT ■ 改行文字の前にHTMLの改行タグを挿入するには「nl2br」関数を使用する	
	COLUMN ■ 関数の定義	
278	HTTPヘッダーを送信する	597
	ONEPOINT ■ HTTPヘッダーを送信するには「header」関数を使用する	
	COLUMN ■ 関数の定義	
279	HTTP通信のレスポンスヘッダーを取得する	598
	ONEPOINT ■ HTTP通信のレスポンスヘッダーを取得するには「get_headers」関数を使用する	
	COLUMN ■ 関数の定義	
280	Webページのメタタグ情報を取得する	599
	ONEPOINT ■ Webページのメタタグ情報を取得するには「get_meta_tags」関数を使用する	
	COLUMN ■ 関数の定義	
281	ホスト名やIPアドレスを取得する	600
	ONEPOINT ■ ホスト名やIPアドレスを取得するには「gethost」で始まる関数を使用する	
	COLUMN ■ 関数の定義	
282	サーバー情報を取得する	602
	ONEPOINT ■ サーバー情報を取得するには「\$_SERVER」変数を使用する	
	COLUMN ■ 「\$_SERVER」変数のエントリ	
283	フォームの送信データを取得する	605
	ONEPOINT ■ フォームの送信データを取得するには「\$_GET」「\$_POST」「\$_REQUEST」変数を使用する	



284	GET/POST/Cookie変数をインポートする	607
	ONEPOINT ■ GET/POST/Cookie変数をインポートするには「import_request_variables」関数を使用する	
	COLUMN ■ 変数名に接頭辞を付加する	
	COLUMN ■ 関数の定義	
285	クッキーを取得・設定する	609
	ONEPOINT ■ クッキーを取得するには「\$_COOKIE」変数、クッキーを設定するには「setcookie」関数を使用する	
	COLUMN ■ 「php.ini」の出力バッファリングの設定	
	COLUMN ■ 関数の定義	
	COLUMN ■ 「setcookie」関数の引数について	
286	ファイルをアップロードする	612
	ONEPOINT ■ ファイルをアップロードするには「\$_FILES」変数と「move_uploaded_file」関数を使用する	
	COLUMN ■ 複数のファイルをアップロードする	
	COLUMN ■ 関数の定義	
287	Webページの内容を取り込む	616
	ONEPOINT ■ Webページの内容を取り込むには「file_get_contents」を使用する	
	COLUMN ■ 関数の定義	
288	「cURL」関数でHTTPリクエストを送信する	618
	ONEPOINT ■ 「cURL」関数でリクエストを送信するには初期化・設定・実行・クローズの順に処理を実行する	
	COLUMN ■ XAMPPで「cURL」関数を使用できるようにする	
	COLUMN ■ 「cURL」関数で「HTTP POST」リクエストを送信する	
	COLUMN ■ 関数の定義	
289	ストリームについて	621
290	「http」ストリームを作成する	623
	ONEPOINT ■ 「http」ストリームを作成するには「stream_context_create」関数を使用する	
	COLUMN ■ 関数の定義	
291	セッションについて	625
292	セッションを開始する	627
	ONEPOINT ■ セッションを開始するには「session_start」関数を使用する	
	COLUMN ■ 関数の定義	
293	セッション変数を設定・取得する	629
	ONEPOINT ■ セッション変数を設定・取得するには「\$_SESSION」変数を使用する	
294	セッションの登録データをすべて破棄する	630
	ONEPOINT ■ セッションの登録データをすべて破棄するには「session_destroy」関数を使用する	
	COLUMN ■ 関数の定義	
295	セッション名を設定・取得する	632
	ONEPOINT ■ セッション名を設定・取得するには「session_name」関数を使用する	
	COLUMN ■ 関数の定義	

296	セッションIDを設定・取得する	633
	ONEPOINT ■ セッションIDを設定・取得するには「session_id」関数を使用する	
	COLUMN ■ 関数の定義	
297	セッションIDを置き換える	634
	ONEPOINT ■ セッションIDを置き換えるには「session_regenerate_id」関数を使用する	
	COLUMN ■ 関数の定義	
298	セッションデータをエンコード/デコードする	635
	ONEPOINT ■ セッションデータをエンコードするには「session_encode」関数、デコードするには「session_decode」関数を使用する	
	COLUMN ■ 関数の定義	
299	セッションクッキーのパラメータを設定・取得する	637
	ONEPOINT ■ セッションクッキーのパラメータを設定するには「session_set_cookie_params」関数、取得するには「session_get_cookie_params」関数を使用する	
	COLUMN ■ 関数の定義	

## CHAPTER 11 データベース

300	テストデータベースを作成する	640
	COLUMN ■ 「mysql」関数について	
	COLUMN ■ データベースへのユーザー登録	
301	MySQLデータベースに接続する	643
	ONEPOINT ■ MySQLデータベースに接続するには「mysql_connect」関数を使用する	
	COLUMN ■ 関数の定義	
302	直前に発生した接続エラーの情報を取得する	645
	ONEPOINT ■ 接続エラーの番号を取得するには「mysql_connect_errno」関数、エラーの内容を取得するには「mysql_connect_error」関数を使用する	
	COLUMN ■ 関数の定義	
303	MySQLに関する情報を取得する	646
	ONEPOINT ■ MySQLに関する情報を取得するには「mysql_get_xxx_info」関数を使用する	
	COLUMN ■ 関数の定義	
304	現在の接続のスレッドIDを取得する	648
	ONEPOINT ■ 現在の接続のスレッドIDを取得するには「mysql_thread_id」関数を使用する	
	COLUMN ■ 関数の定義	
305	MySQLサーバーとの接続状況を調べる	650
	ONEPOINT ■ MySQLサーバーとの接続状況を調べるには「mysql_ping」関数を使用する	
	COLUMN ■ 関数の定義	

306	デフォルトのデータベースを選択する	652
	ONEPOINT ■ デフォルトのデータベースを選択するには「mysql_select_db」関数を使用する	
	COLUMN ■ 関数の定義	
307	直前に発生したエラーの番号と内容を取得する	654
	ONEPOINT ■ 直前に発生したエラーの番号を取得するには「mysql_errno」関数、エラーの内容を取得するには「mysql_error」関数を使用する	
	COLUMN ■ 関数の定義	
308	SQLクエリを実行する	656
	ONEPOINT ■ SQLクエリを実行するには「mysql_query」関数を使用する	
	COLUMN ■ 関数の定義	
309	SQLの実行結果を数値添字配列と連想配列で取得する	658
	ONEPOINT ■ SQLの実行結果を数値添字配列と連想配列で取得するには「mysql_fetch_array」関数を使用する	
	COLUMN ■ 「mysql_fetch_row」関数と「mysql_fetch_assoc」関数について	
	COLUMN ■ 関数の定義	
310	SQLの実行結果をオブジェクトで取得する	661
	ONEPOINT ■ SQLの実行結果をオブジェクトで取得するには「mysql_fetch_object」関数を使用する	
	COLUMN ■ 関数の定義	
311	指定した行に内部ポインタを移動する	663
	ONEPOINT ■ 指定した行に内部ポインタを移動するには「mysql_data_seek」関数を使用する	
	COLUMN ■ 関数の定義	
312	クライアントの文字セットを設定する	665
	ONEPOINT ■ クライアントの文字セットを設定するには「mysql_set_charset」関数を使用する	
	COLUMN ■ 関数の定義	
313	SQL文中の特殊文字をエスケープする	667
	ONEPOINT ■ SQL文中の特殊文字をエスケープするには「mysql_real_escape_string」関数を使用する	
	COLUMN ■ 関数の定義	
314	直近のクエリで更新された自動生成IDを取得する	669
	ONEPOINT ■ 直近のクエリで更新された自動生成IDを取得するには「mysql_insert_id」関数を使用する	
	COLUMN ■ 関数の定義	
315	クエリ結果のフィールドとレコードの数を取得する	671
	ONEPOINT ■ 結果の列の数を取得するには「mysql_num_fields」関数、行の数を取得するには「mysql_num_rows」関数を使用する	
	COLUMN ■ 関数の定義	

3 1 6	フィールドの情報を取得する .....	673
	ONEPOINT ■ フィールドの情報を取得するには 「mysqli_fetch_field」関数を使用する	
	COLUMN ■ 複数のフィールドの情報をまとめて取得する	
	COLUMN ■ 特定のフィールドの情報を取得する	
	COLUMN ■ 関数の定義	
3 1 7	現在のレコードの各フィールドの長さを取得する .....	677
	ONEPOINT ■ 現在のレコードの各フィールドの長さを取得するには 「mysqli_fetch_lengths」関数を使用する	
	COLUMN ■ 関数の定義	
3 1 8	結果保持用メモリを開放する .....	679
	ONEPOINT ■ 結果保持用メモリを開放するには「mysqli_free_result」関数を使用する	
	COLUMN ■ 関数の定義	
3 1 9	テーブルにレコードを追加する .....	681
	ONEPOINT ■ テーブルにレコードを追加するには「INSERT INTO」文を使用する	
	COLUMN ■ SQLコマンドの定義	
3 2 0	レコードのデータを更新する .....	683
	ONEPOINT ■ レコードのデータを更新するには「UPDATE」文を使用する	
	COLUMN ■ SQLコマンドの定義	
3 2 1	テーブルからレコードを削除する .....	685
	ONEPOINT ■ テーブルからレコードを削除するには「DELETE」文を使用する	
	COLUMN ■ SQLコマンドの定義	
3 2 2	テーブルを作成する .....	687
	ONEPOINT ■ テーブルを作成するには「CREATE TABLE」文を使用する	
	COLUMN ■ SQLコマンドの定義	
3 2 3	新しいデータベースを作成する .....	689
	ONEPOINT ■ 新しいデータベースを作成するには 「CREATE DATABASE」文を使用する	
	COLUMN ■ SQLコマンドの定義	
3 2 4	データベースを削除する .....	691
	ONEPOINT ■ データベースを削除するには「DROP DATABASE」文を使用する	
	COLUMN ■ SQLコマンドの定義	
3 2 5	テーブルの一覧を取得する .....	693
	ONEPOINT ■ テーブルの一覧を取得するには「SHOW TABLES」文を使用する	
	COLUMN ■ SQLコマンドの定義	

## CHAPTER 12 セキュリティ・エラー処理

3 2 6	セキュリティ対策について .....	696
	COLUMN ■ PHPのセーフモードについて	
3 2 7	XSS攻撃対策を行う .....	698
	ONEPOINT ■ XSS攻撃対策を行うには出力文字列を「htmlspecialchars」関数で エスケープする	

328	CSRF攻撃対策を行う	701
	ONEPOINT ■ CSRF攻撃対策を行うにはトークンを使用する	
329	NULLバイト攻撃対策を行う	705
	ONEPOINT ■ NULLバイト攻撃対策を行うには入力値をチェックする	
	COLUMN ■ 「バイナリセーフな関数」と「バイナリセーフでない関数」について	
330	セッション固定攻撃対策を行う	708
	ONEPOINT ■ セッション固定攻撃対策を行うには認証後にセッションIDを変更する	
	COLUMN ■ セッション/固定攻撃以外のセッションハイジャック対策を行う	
331	ディレクトリトラバーサル対策を行う	712
	ONEPOINT ■ ディレクトリトラバーサル対策を行うには NULLバイトとディレクトリ名を取り除く	
332	SQLインジェクション対策を行う	714
	ONEPOINT ■ SQLインジェクション対策を行うには SQL文中の特殊文字をエスケープする	
	COLUMN ■ プリペアドステートメントを使用する	
333	OSコマンドインジェクション対策を行う	720
	ONEPOINT ■ OSコマンドインジェクション対策を行うには 「escapeshellarg」関数を使用する	
	COLUMN ■ 関数の定義	
334	メールヘッダーインジェクション対策を行う	721
	ONEPOINT ■ メールヘッダーインジェクション対策を行うには メールアドレスの形式をチェックする	
335	HTTPヘッダーインジェクション対策を行う	723
	ONEPOINT ■ HTTPヘッダーインジェクション対策を行うには 外部からの入力値をHTTPヘッダーとして出力しないようにする	
336	エラーメッセージの出力を抑制する	724
	ONEPOINT ■ エラーメッセージの出力を抑制するには「@」演算子を使用する	
337	エラー発生時にスクリプトを終了する	726
	ONEPOINT ■ エラー発生時にスクリプトを終了するには「exit」関数を使用する	
	COLUMN ■ 関数の定義	
338	エラーレベルを設定する	727
	ONEPOINT ■ エラーレベルを設定するには「error_reporting」関数を使用する	
	COLUMN ■ 関数の定義	
339	ユーザーエラーを発生させる	729
	ONEPOINT ■ ユーザーエラーを発生させるには「trigger_error」関数を使用する	
	COLUMN ■ 関数の定義	
340	エラーハンドラを設定する	731
	ONEPOINT ■ エラーハンドラを設定するには「set_error_handler」関数を使用する	
	COLUMN ■ エラーハンドラ関数の定義	
	COLUMN ■ 関数の定義	
	●索引	733



# CHAPTER 01

---

## PHPの基礎知識

# PHPとは

## III PHPの概要

PHP(PHP:Hypertext Preprocessor)は、Webアプリケーションを開発するためのプログラミング(スクリプト)言語です。平易な文法で初心者でも習得しやすく、大規模なWebアプリケーションの開発にも広く用いられています。

### ▶ HTMLに埋め込むことができる

PHPは、HTMLに処理内容を記述したスクリプトを埋め込み、実行結果に応じて動的にHTMLデータを生成することができます。たとえば、ページのアクセス時間に応じて異なるメッセージを動的に生成して出力することができます。

### ▶ サーバーサイド・スクリプト

PHPは、Webサーバー上で動作するサーバーサイド・スクリプトです。クライアント(ブラウザ)から、PHPスクリプトにアクセスすると、そのコードがサーバー上で実行され、結果がクライアントに送信されます。クライアントは、スクリプトを実行した結果を受け取りますが、その出力を作成したコードの内容を知ることはいけません。

### ▶ 強力なテキスト処理機能

PHPには、便利なテキスト処理機能が用意されています。たとえば、Perl互換の正規表現(PCRE)や、XML文書のパースやXML文書へのアクセスのための数多くの拡張モジュール・ツールなどがあります。

### ▶ データベースへの容易なアクセス

PHPは、標準でMySQLデータベースをサポートしており、データベース機能を用いたWebアプリケーションを簡単に作成することができます(CHAPTER 11参照)。また、データベースごとに用意された拡張モジュールや「PDO」のような抽象化レイヤーを使用することで、多くのデータベースに接続することができます。

### ▶ クラスベースオブジェクト指向のサポート

PHPは、クラスベースオブジェクト指向をサポートしており、オブジェクト指向でプログラミングすることができます。PHP5以降では、完全なオブジェクトモデルが搭載され、本格的なオブジェクト指向プログラミングが可能です(CHAPTER 06参照)。

### ▶ マルチプラットフォーム

PHPは、Linux、多くのUnix系システム(HP-UX、Solaris、OpenBSDなど)、Windows、Mac OS X、RISC OS、その他、すべての有名なOSで動作します。また、ApacheやIIS、その他のサーバーなど、現在、使用されているほとんどすべてのWebサーバーをサポートしています。

### ▶ 多彩なファイルの生成機能

PHPは、HTMLだけでなく、イメージやPDFファイルを生成する機能を備えています。

「libswf」や「Ming」を使用して、Flashムービーを生成することもできます。また、XHTMLやXMLというその他の文書も自動的に生成することができ、ファイルシステムに保存したり、印刷したり、サーバー側でキャッシュしたりすることもできます。

▶ **多くのオープンソースのフレームワークやライブラリが利用可能**

多くのオープンソースのフレームワークやライブラリが利用可能で、開発効率をさらに高めることができます。

▶ **豊富な拡張モジュール**

「PEAR」というオンラインライブラリ集で、さまざまな付加機能がモジュールとして提供されています。モジュールを組み込みことで機能を拡張することができます。

▶ **多くのプロトコルに対応**

PHPは、「IMAP」「SNMP」「NNTP」「POP3」「HTTP」「COM」(Windowsのみ)など、多くのプロトコルを用いる他のサービスの状態を追跡する機能もサポートしています。低レベルのネットワークソケットをオープンし、他のプロトコルを用いて通信を行うことも可能です。また、PHPはWDDXをサポートし、基本的にすべてのWebプログラミング言語間で複雑なデータ交換を行うことができます。

**COLUMN**

**PHPで作られている代表的なアプリケーション**

PHPで作られている代表的なWebアプリケーションには、次のようなアプリケーションがあります。

アプリケーション名	説明
IBM WebSphere sMash	Webアプリケーションのプラットフォーム
MediaWiki	Wikiソフトウェア
phpBB	電子掲示板
phpMyAdmin	MySQLデータベースをGUIで管理するためのツール
PukiWiki	Wikiソフトウェア
Serendipity Weblog System	CMS(コンテンツマネジメントシステム)
SilverStripe CMS	CMS(コンテンツマネジメントシステム)
WordPress	CMS(コンテンツマネジメントシステム)
XOOPS	CMS(コンテンツマネジメントシステム)
OpenPNE	SNS運営用のサーバサイドプログラム群

●WordPress





# 実行環境について

## PHPの実行環境

PHPの開発を行うには、PHPが動作するWebサーバーが必要になります。自宅のパソコンにPHPの実行環境を構築する場合、XAMPP(ザンプ)を使うと便利です。XAMPPをインストールすると、Webサーバー(Apache)、PHP、MySQLなど、PHPの開発に必要なソフトウェア一式がインストールされ、インターネット上のWebサーバーとほぼ同じ環境を簡単に構築することができます。

## XAMPPのインストール

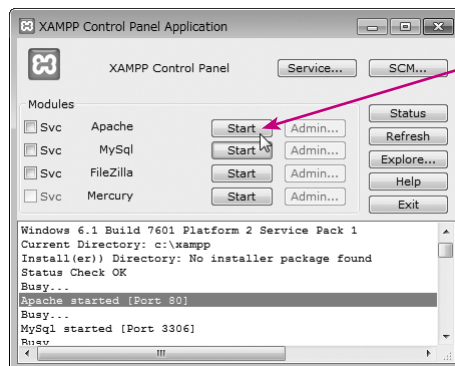
XAMPPは、Webアプリケーションの実行に必要なフリーソフトを集めたパッケージです。OSによって、XAMPPのバージョン、および、含まれるPHPのバージョンが異なります。2012年6月現在の最新バージョンは、次の通りです。なお、Linux用のXAMPPは、64ビットシステムでは動作しません。

OS	XAMPPのバージョン	PHPのバージョン
Windows	1.7.7	5.3.8
Linux(x86用)	1.7.7	5.3.8
Mac OS X	1.7.3	5.3.1

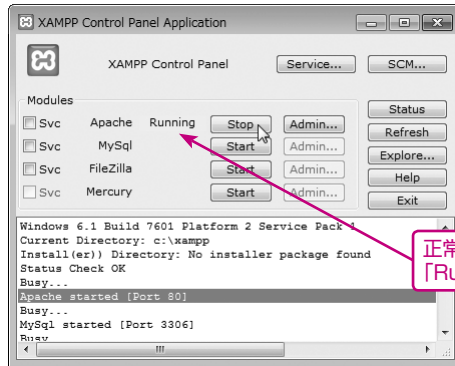
### Windowsへのインストール

「<http://www.apachefriends.org/jp/xampp-windows.html>」から、インストーラ(xampp-win32-1.7.7-VC9-installer.exe)をダウンロードします。インストーラを実行し、画面の指示に従って操作します。

インストールが完了したら、XAMPPコントロールパネルを使って、Webサーバー(Apache)とMySQLサーバーをスタートさせます。



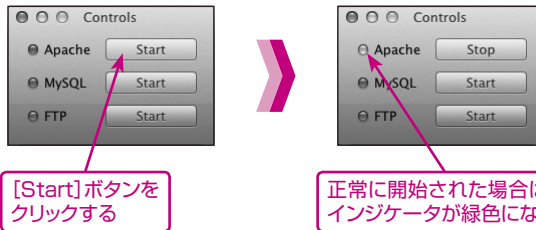
[Start] ボタンをクリックする



### ▶ Macへのインストール

「<http://www.apachefriends.org/jp/xampp-macosx.html>」から、「XAMPP Mac OS X 1.7.3」(xampp-macosx-1.7.3.dmg)をダウンロードして開きます。自分の「Applications」フォルダへ「XAMPP」フォルダをドラッグ&ドロップします。

インストールが完了したら、XAMPP Controlを使って、Webサーバー(Apache)とMySQLサーバーをスタートさせます。



### ▶ Linux(Ubuntu)へのインストール

Ubuntuにインストールするには、次のように操作します。

- ① 「<http://www.apachefriends.org/jp/xampp-linux.html>」から、「XAMPP Linux 1.7.7」(xampp-linux-1.7.7.tar.gz)をダウンロードします。
- ② 端末を起動し、「cd ダウンロード」と入力します。
- ③ 「sudo tar xvfz xampp-linux-1.7.7.tar.gz -C /opt」と入力します。
- ④ rootユーザーのパスワードを入力します。この後、ダウンロードしたファイルが「/opt」に展開されます。
- ⑤ 「sudo chmod 777 /opt/lampp/htdocs」とコマンドを入力し、Webサーバーのルートディレクトリのパーミッションを変更します。
- ⑥ 「sudo chmod 777 /opt/lampp/etc/php.ini」とコマンドを入力し、「php.ini」のパーミッションを変更します。
- ⑦ 「sudo /opt/lampp/lampp start」とコマンドを入力し、Webサーバー(Apache)とMySQLサーバーをスタートさせます。

### III PHPの日本語環境の設定

「php.ini」を編集して、PHPの日本語環境を設定します。本書では、文字エンコーディングを「UTF-8」に統一しています。

#### ▶ 「php.ini」の保存場所

XAMPPをデフォルトの設定でインストールした場合の「php.ini」の保存場所は、次の通りです。

OS	保存場所
Windows	C:\xampp\php
Linux (x86版)	/opt/lampp/etc
Mac OS X	/Applications/XAMPP/xamppfiles/etc

#### ▶ 日本語環境の設定

「php.ini」をテキストエディタで開き、次の表の通りに変更します。なお、行の先頭に「;」がある場合は、「;」を削除します。

ディレクティブ	設定値
date.timezone	Asia/Tokyo
default_charset	"UTF-8"
mbstring.language	Japanese
mbstring.internal_encoding	UTF-8
mbstring.substitute_character	none;

### III MySQLの「root」パスワードの設定

phpMyAdminでMySQLの「root」のパスワードを設定し、そのパスワードを「config.inc.php」に記述します。

- ① ブラウザで「http://localhost/phpmyadmin/」にアクセスし、phpMyAdminを開きます。  
なお、言語が日本語になっていない場合は、日本語に変更してください。
- ② 「特権」タブをクリックし、ホストが「localhost」の「root」の「特権を編集」をクリックします。
- ③ 「パスワードを変更する」の「パスワード」と「もう一度入力してください」にパスワードを入力し、  
「実行する」ボタンをクリックします。
- ④ 「config.inc.php」をテキストエディタで開きます。
- ⑤ 「\$cfg['Servers'][\$i]['password'] = ";」という記述を検索し、「"」内に④で設定したパスワードを記述し、上書き保存します。

なお、「config.inc.php」の保存場所は、次の通りです。

OS	保存場所
Windows	C:\xampp\phpMyAdmin
Linux (x86版)	/opt/lampp/phpmyadmin
Mac OS X	/Applications/XAMPP/xamppfiles/phpmyadmin

### III 実行環境の確認

構築した実行環境の動作を確認するためのファイル「test.php」を作成し、WWWサーバーのルートディレクトリに保存します。ブラウザで「http://localhost/test.php」にアクセスし、PHPの設定情報が出力されることを確認します。

● test.php

```
<?
phpinfo();
?>
```

それぞれのOSのWWWサーバーのルートディレクトリは、次の通りです。

OS	WWWサーバーのルートディレクトリ
Windows	C:\xampp\htdocs
Linux(x86版)	/opt/lampp/htdocs
Mac OS X	/Applications/XAMPP/xamppfiles/htdocs

「http://localhost/test.php」にアクセスすると、次のように表示されます。

<b>System</b>	Windows NT GAMO-PCX 6.1 build 7601 (Windows 7 Business Edition Service Pack 1) i586
<b>Build Date</b>	Aug 23 2011 11:47:20
<b>Compiler</b>	MSVC9 (Visual C++ 2008)
<b>Architecture</b>	x86
<b>Configure Command</b>	gscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--disable-isapi" "--without-mssql" "--without-pdo-mssql" "--without-pike" "--with-pdo-oci=D:\php-sdk\oracle\instantclient_101\sdk\shared" "--with-oci8=D:\php-sdk\oracle" "#instantclient_101\sdk\shared" "--with-oci8-11g=D:\php-sdk\oracle\instantclient_11g\sdk\shared" "--enable-object-out-dir=/obj/" "--enable-com-dotnet" "--with-mcryptstatic" "--disable-static-analyze"
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	enabled
<b>Configuration File (php.ini) Path</b>	C:\Windows
<b>Loaded Configuration File</b>	C:\xampp\php\php.ini
<b>Scan this dir for additional ini files</b>	(none)
<b>Additional ini files parsed</b>	(none)
<b>PHP API</b>	20090626
<b>PHP Extension</b>	20090626
<b>Zend Extension</b>	220090626
<b>Zend Extension Build</b>	API220090626,TS,VC9
<b>PHP Extension</b>	API220090626,TS,VC9

COLUMN

Apacheが起動しないときの対処方法

TCP/IPを使ったネットワーク・サービスでは、ある特定のポートでリッスン(待ち受け)することにより、クライアントからの要求を受け付けるようになっています。Apacheではポート番号「80」をリッスンしていますが、他のサービス(SkypeやIISなど)と競合することがあり、Apacheが起動しない場合があります。この場合、次のように操作して、Apacheのリッスンするポートを変更します。

- ❶ 「httpd.conf」をテキストエディタで開きます。
- ❷ 54行目の「Listen 80」を「Listen 8080」(他のサービスと重複しない番号を指定する)と書き換え、上書き保存します。

なお、この変更を行うと、Webサーバーにアクセスする際に「http://localhost:8080/」のように、URLの末尾にポート番号を指定する必要があります。「httpd.conf」の保存場所は、次の通りです。

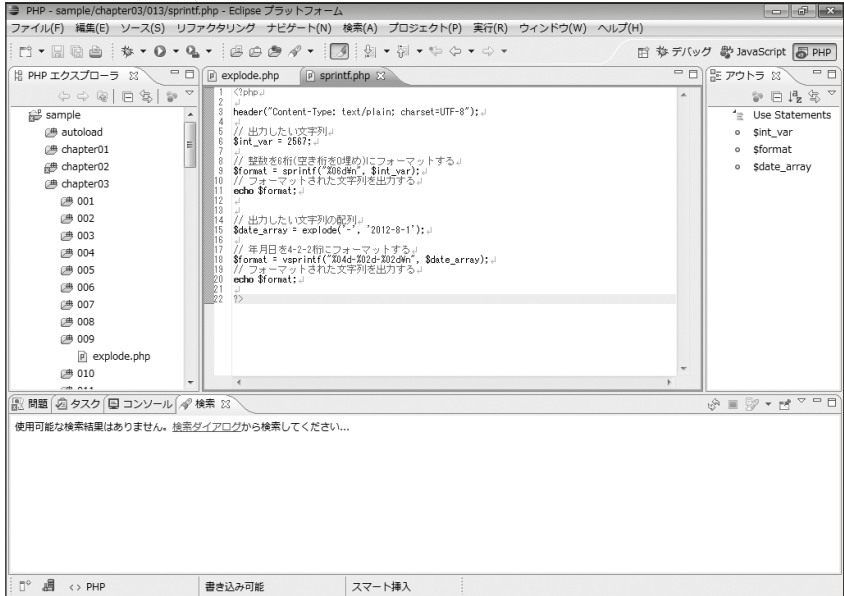
OS	保存場所
Windows	C:\xampp\apache\conf
Linux(x86版)	/opt/lampp/etc
Mac OS X	/Applications/XAMPP/xamppfiles/etc

# 統合開発環境のインストール

## III Eclipse PDTについて

Eclipse PDTは、PHP開発者向けのEclipseです。Eclipseは、IBMによって開発された統合開発環境で、プラグインでさまざまな機能を組み込むことができます。

### ●Eclipse PDT



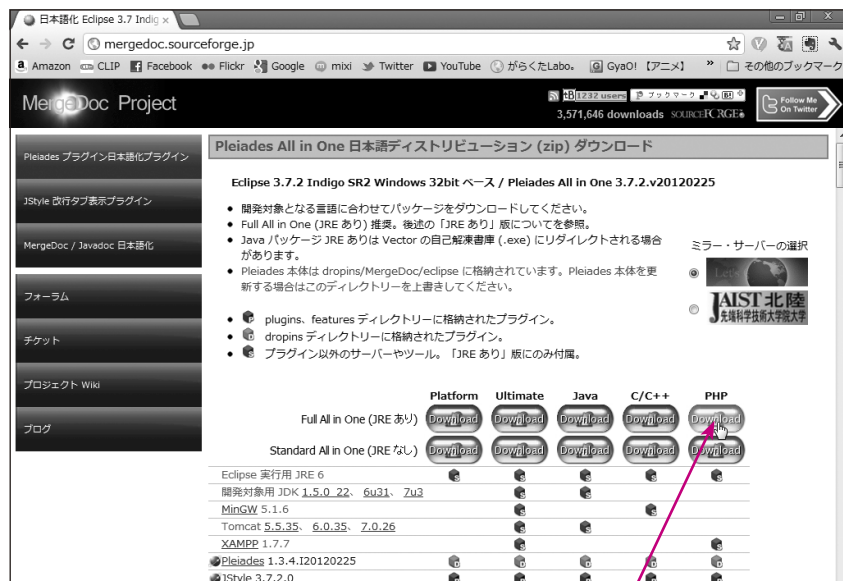
## III Windowsへのインストール

WindowsにEclipse PDTをインストールするには、「Pleiades All in One PHP日本語ディストリビューション」を使うと簡単です。

### ▶「Pleiades All in One PHP日本語ディストリビューション」のインストール

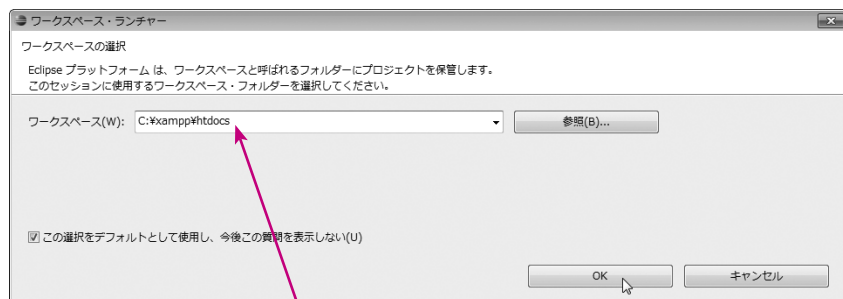
「Pleiades All in One PHP日本語ディストリビューション」をインストールするには、次のように操作します。

- ① ブラウザで「<http://mergedoc.sourceforge.jp/>」のページを表示します。
- ② [Eclipse 3.7 Indigo Pleiades All in One] ボタンをクリックします。
- ③ 「PHP」の下の[Download] ボタン(青いボタン)をクリックします。



[Download] ボタン(青いボタン)  
をクリックする

- ④ ダウンロードした圧縮ファイルの「pleiades-e3.7-php-jre\_YYYYMMDD.zip」(YYYYMMDD はバージョンによって異なる)を開き、「eclipse」フォルダを「C:」ドライブにコピー(または移動)します。
- ⑤ 「eclipse」フォルダを開き、「eclipse.exe」をダブルクリックしてEclipseを起動し、[ワークスペース(W)]を「C:\xampp\htdocs」に設定して[OK] ボタンをクリックします。



ワークスペースを  
設定する

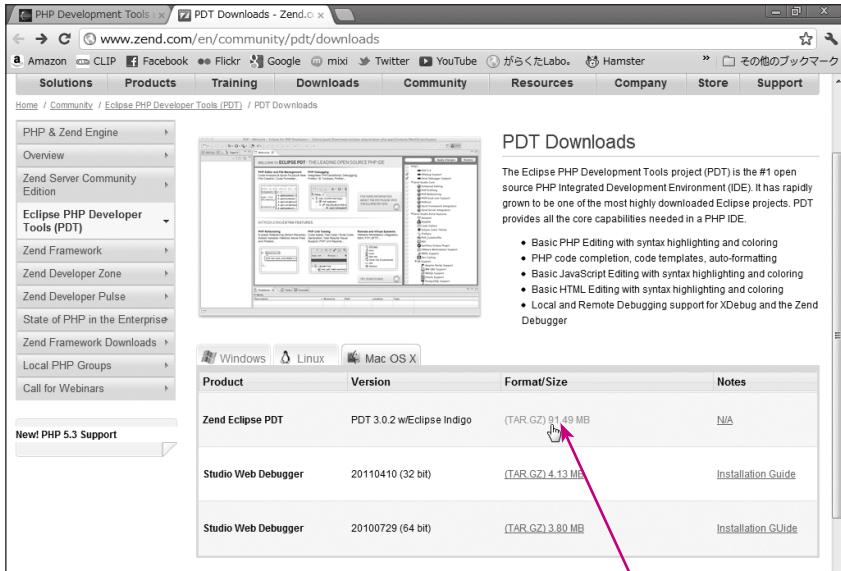
### Macへのインストール

MacにEclipse PDTをインストールするには、Eclipse PDTとPleiadesを使います。Pleiades は、日本語化するために使います。

## ▶ Eclipse PDTのインストール

Eclipse PDT (2012年6月現在のバージョン)をインストールするには、次のように操作します。

- 1 ブラウザで「<http://eclipse.org/pdt/downloads>」のページを表示します。
- 2 「Download All-In-One package including Eclipse 3.7.2 and PDT 3.0.2 from Zend」(表示内容はバージョンにより異なる)をクリックします。
- 3 「Mac OS X」タブを選択し、「Zend Eclipse PDT」のダウンロードリンク「(TAR.GZ) 91.49 MB」(サイズの数値はバージョンにより異なる)をクリックします。



- 4 ダウンロードした「eclipse-php-3.0.2.v20120511142-Mac.tar.gz」(「v」以降の数値はバージョンにより異なる)を「アプリケーション」フォルダに展開します。展開すると「eclipse-php」フォルダができます。

## ▶ Eclipse PDTの日本語化

Eclipse PDTを日本語化するには、次のように操作します。

- 1 ブラウザで「<http://mergedoc.sourceforge.jp/>」にアクセスします。
- 2 「Pleiades 本体ダウンロード」の「1.3.3」(2012年6月時点の安定版)をクリックします。
- 3 ダウンロードした「Pleiade\_1.3.3.zip」を「ダウンロード」フォルダに展開します。展開すると、「Pleiades\_1.3.3」フォルダができます。
- 4 「Pleiades\_1.3.3」フォルダの「features」フォルダの中にあるフォルダをすべて、「eclipse-php」フォルダの「features」フォルダ内にコピー(または移動)します。
- 5 「Pleiades\_1.3.3」フォルダの「plugins」フォルダの中にあるフォルダをすべて、「eclipse-php」フォルダの「plugins」フォルダにコピー(または移動)します。



- ⑥ 「eclipse-php」フォルダの「eclipse-php」をControlキーを押しながらクリックし、[パッケージの内容を表示]を選択します。
- ⑦ 「Contents」フォルダを開いて、「MacOS」フォルダを開きます。
- ⑧ 「eclipse-php.ini」をテキストエディタで開き、最終行に次の記述を追加して、上書き保存します。

```
-javaagent:/Applications/eclipse-php/plugins/jp.sourceforge.mergedoc.pleiades/pleiades.jar
```

- ⑨ 初回のみターミナルから、次のコマンドを入力して、Eclipse PDTを起動し、プラグインを初期化します。

```
/Applications/eclipse-php/eclipse-php.app/Contents/MacOS/eclipse-php -clean
```

- ⑩ [ワークスペース]を[/Applications/XAMPP/xamppfiles/htdocs]に設定して[OK]ボタンをクリックします。

### Linux (Ubuntu) へのインストール

UbuntuにEclipse PDTをインストールするには、Eclipse PDTとPleiadesを使います。Pleiadesは、日本語化するために使います。また、Ubuntuの場合、JDK、または、JREをインストールする必要があります。

#### ▶ JDKのインストール

JDK (Java SE Development Kit 7)をインストールするには、次のように操作します。

※URL、ファイル名はバージョンにより異なります。この例は2012年6月現在のURL、ファイル名です。

- ① ブラウザで「<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1637583.html>」のページを表示します。
- ② 64ビット版の場合は「jdk-7u5-linux-x64.tar.gz」、32ビット版の場合は「jdk-7u5-linux-i586.tar.gz」をダウンロードします。
- ③ ダウンロードした圧縮ファイルを「ホーム」フォルダに展開します。
- ④ テキストエディタで「~/profile」を開き、次の記述を追加します。「jdk」の後ろに続く数値はバージョン番号で、バージョンによって変わるので注意してください。「<ユーザー名>」には、ログイン時のユーザー名を入力します。

```
export JAVA_HOME=/home/<ユーザー名>/jdk1.7.0_05
export PATH=/home/<ユーザー名>/jdk1.7.0_05/bin:$PATH
export CLASSPATH=.:home/<ユーザー名>/
```

- ⑤ システムを再起動します。

#### ▶ Eclipse PDTのインストール

Eclipse PDT ((2012年6月現在のバージョン)をインストールするには、次のように操作します。

- ① ブラウザで「<http://eclipse.org/pdt/downloads/>」のページを表示します。
- ② 「Download All-In-One package including Eclipse 3.7.2 and PDT 3.0.2 from Zend」(表示内容はバージョンにより異なる)をクリックします。

- ③ 「Linux」タブを選択し、64ビット版の場合は「Version」が「PDT 3.0.2 w/Eclipse Indigo (64 bit)」のダウンロードリンク「(TAR.GZ) 91.86 MB」を、32ビット版の場合は「Version」が「PDT 3.0.2 w/Eclipse Indigo (32 bit)」のダウンロードリンク「(TAR.GZ) 91.68 MB」をクリックします。なお、ダウンロードリンクのサイズの数値はバージョンにより異なります。
- ④ ダウンロードした圧縮ファイルを「ホーム」フォルダに展開します。展開すると「eclipse-php」フォルダができます。

#### ▶ Eclipse PDTの日本語化

Eclipse PDTを日本語化するには、次のように操作します。

- ① ブラウザで「<http://mergedoc.sourceforge.jp/>」にアクセスします。
- ② 「Pleiades 本体ダウンロード」の「1.3.3」(2012年6月時点の安定板)をクリックします。
- ③ 「ダウンロード」フォルダを開いて、ダウンロードした「Pleiades\_1.3.3.zip」を右クリックし、「ここに展開する」を選択します。展開すると「Pleiades\_1.3.3」フォルダができます。
- ④ 「Pleiades\_1.3.3」フォルダの「features」フォルダの中にあるフォルダをすべて、「eclipse-php」フォルダの「features」フォルダ内にコピー(または移動)します。
- ⑤ 「Pleiades\_1.3.3」フォルダの「plugins」フォルダの中にあるフォルダをすべて、「eclipse-php」フォルダの「plugins」フォルダにコピー(または移動)します。
- ⑥ 「eclipse-php」フォルダの「eclipse-php.ini」をテキストエディタで開き、最終行に次の記述を追加して、上書き保存します。

```
-javaagent:plugins/jp.sourceforge.mergedoc.pleiades/pleiades.jar
```

- ⑦ Eclipseを起動し、「ワークスペース」を「/opt/lampp/htdocs」に設定して[OK] ボタンをクリックします。

# コードの記述方法について

## III ソースファイルの作成方法

コードを記述するソースファイルは、テキストエディタや、Eclipseなどの統合開発環境 (IDE) で記述します。統合開発環境は、関数名や変数名の補完や、デバッガなど、効率的に開発するための機能を利用できるのでお勧めです。Eclipseのインストール方法については、45ページを参照してください。

## III コード記述の規則

PHPのコードを記述する際の規則は、次の通りです。

- PHPのソースファイルの拡張子は「.php」
- PHPのコードは「<?php」(開始タグ)で始まり「?>」(終了タグ)で終わる
- 行(ステートメント)の最後には「;」を付ける

## III HTMLへの埋め込みと脱出

PHPのコードは、HTMLファイルの任意の場所に埋め込むことができます。PHPコードを埋め込んだHTMLファイルの拡張子は、「.php」にする必要があります。

PHPは、HTML中の「<?php」から「?>」の範囲だけをパースして実行します。その他の箇所は無視され、そのまま出力されます。

```
<p>この部分は無視されます。</p>
<?php echo '一方、この部分はパースされます。'; ?>
<p>この部分も無視されます。</p>
```

「if」文などを使用して、特定の条件のときにHTMLから脱出することもできます。PHPは「?>」を見つけると、それ以降新たに「<?php」を見つかるまでの内容をすべて出力します。

```
<?php if ($expression) { ?>
    <strong>「$expression」が「真」ならここが出力されます。</strong>
<?php } else { ?>
    <strong>「$expression」が「偽」ならここが出力されます。</strong>
<?php } ?>
```

## III 文字エンコーディングと改行コード

PHPのソースファイルは、文字エンコーディング(文字コード)には「UTF-8(BOMなし)」の使用をお勧めします。改行コードについては、「LF」または「CR+LF」のどちらでもかまいません。なお、Windows環境では、マルチバイト文字列が含まれていないファイルは、「UTF-8」で保存することができないため、コメントなどの形で全角文字を1文字でも記述しておくようにします。

# PHPマニュアルについて

## III PHPマニュアルの種類

PHPマニュアルには、オンラインマニュアルとchm版(Windowsのヘルプファイル)のオンラインマニュアルがあります。

### ▶ オンラインマニュアル

オンラインマニュアルを利用するには、次のURLにアクセスします。

**URL** <http://php.net/manual/ja/>



特定の関数のマニュアルを表示するには、「<http://php.net/>」の後に関数名を指定します。たとえば、「empty」関数のマニュアルを表示するには、次のURLにアクセスします。

**URL** <http://php.net/empty>

### ▶ chm版のマニュアル(Windows)

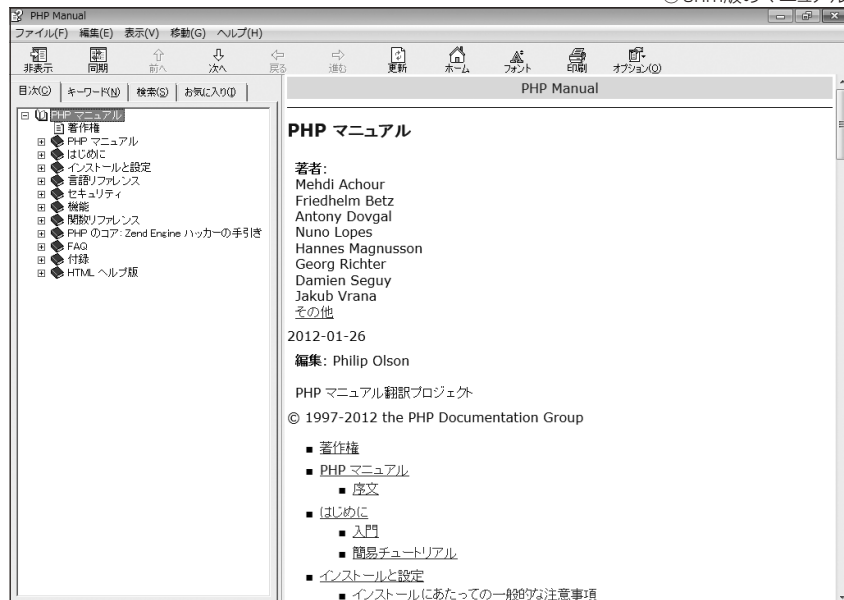
chm版のマニュアルには、ユーザーノートを含まない「HTML Help file」と、ユーザーノートを含む「HTML Help file (with user notes)」の2種類があります。ユーザーノートには、関数の使用例や、その他役に立つ情報が記述されています。

たとえば、chm版のマニュアルをダウンロードして、オフラインで見られるようにするには、次のように操作します。

- ① <http://www.php.net/download-docs.php>にアクセスします。

- ② 一覧表の「Japanese」の行の「HTML Help file」列、または、「HTML Help file (with user notes)」列の「chm」をクリックします。
- ③ 任意のミラーサーバーのリンクを選択してダウンロードします。
- ④ ダウンロードしたファイル「php\_manual\_ja.chm」を右クリックし、[プロパティ(R)]を選択します。
- ⑤ [ブロックの解除(K)]をクリックし、[OK]ボタンをクリックします。

●chm版のマニュアル



## COLUMN

### HTML版のヘルプ

PHPにヘルプには、HTML版もあります。HTML版ヘルプには、「Single HTML file」と「Many HTML files」の2種類があります。前者は1つのHTMLファイルに、後者は複数のHTMLファイルにヘルプの内容が記述されています。HTML版ヘルプは「gz」形式で圧縮されているため、利用するには「7-zip」(<http://www.7-zip.org/>)など、「gz」形式の圧縮ファイルを解凍するためのツールが必要になります。



# CHAPTER 02

---

## PHPの基本文法

# 型について

## PHPのサポートする型

PHPは、4種類のスカラー型、2種類の複合型、2種類の特別な型の合計8種類の基本型をサポートします。スカラー型は、常に1つの値を保持している変数です。スカラー型には、「論理値」「整数」「浮動小数点」「文字列」の4種類があります。複合型は、複数の値を格納することができる変数です。複合型には、「配列」と「オブジェクト」の2種類があります。特別な型には、「リソース」と「NULL」の2種類があります。PHPでは、変数の型は自動認識されるので、明示的に型の宣言をする必要はありません。

### ▶ 論理値(Boolean)

真偽の値を表します。値は「TRUE」(true)か「FALSE」(false)です。

### ▶ 整数(integer)

整数は、10進数、16進数、8進数、2進数表記で使用できます。8進数表記は「0」を、16進数表記は「0x」を、2進数表記は「0b」を数の前に付けます。なお、2進数の整数リテラルは、PHP5.4.0から利用可能です。

```
$a = 1234; // 10進整数
$a = -123; // 負の数
$a = 0123; // 8進数(10進数の83と等価)
$a = 0x1A; // 16進数(10進数の26と等価)
```

### ▶ 浮動小数点数(float)

浮動小数点数は、小数点以下の値を持つ数値(実数)です。科学記法で表した数値は、浮動小数点数として扱われます。

```
$a = 1.234;
$b = 1.2e3; // 1200
$c = 7E-10; // 0.0000000007
```

### ▶ 文字列(string)

文字列は、「」(シングルクォーテーション)、または、「"」(ダブルクォーテーション)で囲んで表現します。文字列の詳細については、80ページを参照してください。

### ▶ 配列(array)

配列は、「array」関数で作成します。具体的な作成方法について、CHAPTER 05を参照してください。PHPの配列は、実際には順番付けられたマップです。マップは型の一種で、値をキーに関連付けます。

### ▶ オブジェクト(object)

オブジェクトは、クラスをインスタンス化して作成します。詳細については、CHAPTER 06を参照してください。

## ▶ リソース(resource)

リソースは、外部リソースへのリファレンスを保持している特別な変数です。リソースは、特別な関数によって作成され、使用されます。

## ▶ NULL

ある変数が値を持たないことを表します。値は「NULL」(null)です。

## III 型の変換

PHPでは、変数の型はその変数が使用される文により定義されます。オペランドのどれかが「float」の場合、すべてのオペランドは「float」として評価され、結果は「float」になります。その他の場合、オペランドは「int」として解釈され、結果も「int」になります。

```
$foo = 2; // 「foo」は整数
$foo = $foo + 1.3; // ここでは「foo」はfloat(3.3)
$foo = 5 + "10 Little Piggies"; // 「foo」は整数(15)
```

ある変数を強制的にある特定の型として評価させたい場合には、型キャストを行います。PHPの型キャストは、変換しようとする型をカッコで囲み、キャストする変数の前に置きます。

```
$foo = 10; // 「foo」は整数
$bar = (bool) $foo; // 「bar」はbool
```

使用可能なキャストは、次の通りです。

キャスト	説明
(int)、(integer)	整数にキャスト
(bool)、(boolean)	論理値にキャスト
(float)、(double)、(real)	floatにキャスト
(string)	文字列にキャスト
(array)	配列にキャスト
(object)	オブジェクトにキャスト
(unset)	NULLにキャスト(PHP5以降)
(binary)	バイナリ文字列にキャスト(PHP5.2.1以降)

### COLUMN 擬似的な型について

PHPマニュアル、および、本書のCOLUMNの関数の定義では、可読性を向上させるために、次の擬似的な型も使用します。

型	説明
mixed	引数に多様な型(すべてである必要はない)を使うことができることを示す
number	引数がintまたはfloatのどちらでも良いことを示す
callback	ユーザーが定義するコールバック関数を引数として受け入れる

### 関連項目 ▶▶▶

- 文字列について ..... p.80



# 変数について

## PHPの変数

変数は、値を格納しておくことができる場所です。PHPの変数は、ドル記号「\$」の後ろに「変数名」が続く形式で表します。変数名は、大文字と小文字を区別します。有効な変数名は、文字、または、「\_」（アンダースコア）から始まり、任意の数の文字、数字、「\_」（アンダースコア）が続きます。

```
$var = 1; // 初期値「1」の変数「var」を宣言する
$5foo = 2; // 無効な変数名
```

### ▶ 代入

変数への値の代入は、「=」演算子を使用します。PHPでは変数を初期化する必要がないため、通常、変数の宣言と代入は同時に行われます。

```
$var = 100; // 変数「var」に「100」を代入する
$foo = $var; // 変数「foo」に変数「var」の値を代入する
```

### ▶ 参照による代入

参照による代入を行うには、代入する変数の前に「&」（アンパサンド）を付けます。この場合、2つ変数は同じ内容を指します。参照による代入は「参照渡し」や、単に「参照」といいます。また、「リファレンス」という場合もあります。

```
$foo = 'Bob'; // 「Bob」を変数「foo」に代入する
$bar = &$foo; // 変数「foo」を変数「bar」により参照する
$bar = "My name is $bar"; // 変数「bar」を変更すると変数「foo」も変更される
// 結果、以下は同じ内容「My name is Bob」が出力される
echo $bar;
echo $foo;
```

## 変数のスコープ

スコープとは、変数の有効範囲のことです。PHPのスコープには、スクリプト内のどこでも有効な「グローバルスコープ」と、関数内だけで有効な「ローカルスコープ」の2種類があります。関数の中で宣言された変数は「ローカルスコープ」、それ以外で宣言された変数は「グローバルスコープ」になります。グローバルスコープな変数を「グローバル変数」、ローカルスコープな変数を「ローカル変数」といいます。PHPのグローバル変数は、関数内部では使用できません。グローバル変数を関数の内部で使用する場合、関数の内部でグローバルとして宣言する必要があります。

```
$a = 'global'; // グローバルスコープ
```



```
function test(){
    $b = "local"; // ローカルスコープ
    echo $a; // 関数内で変数「a」は宣言されていないのでエラーとなる
    echo $b; // 「local」を出力する
}
test();
```

変数を関数の内部でグローバルとして宣言するには、「global」キーワードを使います。

```
$a = 'global'; // グローバルスコープ

function test(){
    global $a; // 変数「a」をグローバル宣言
    $b = "local"; // ローカルスコープ
    echo $a; // 「global」を出力する
    echo $b; // 「local」を出力する
}
test();
```

## COLUMN シンボルテーブルについて

PHPでは、変数はシンボルテーブルによって管理されます。シンボルテーブルは、変数名と、その値の実際の格納場所を対応付けする表です。たとえば、「\$a = 'hello';」と宣言した場合、次のようなシンボルテーブルが作成されます。

●シンボルテーブル

変数名	格納庫番号
a	#001

●格納場所

番号	値	型
#001	hello	string

参照による代入は、同じ格納場所に複数の変数名を割り当てます。たとえば、「\$b = &\$a」と参照代入した場合、シンボルテーブルは次のようになります。

●シンボルテーブル

変数名	格納庫番号
a	#001
b	#001

●格納場所

番号	値	型
#001	hello	string

### 関連項目 ▶▶▶

- 関数の引数について ..... p.72

## 定数について

### III PHPの定数

定数は、識別しやすい名前が付けられた値です。PHPの定数は、「define」関数を使用して定義します。引数には、定数名と値を指定します。定数の名前は、変数と同じ規則に従います。定数は一度、定義すると、変更、または、未定義にすることはできません。また、定数に指定できるのは、スカラー値（「bool(boolean)」 「int(integer)」 「float(double)」 「string」）のみです。定数の値を取得するには、単に定数の名前は指定します。変数とは異なり、名前の前に「\$」は不要です。

```
// 定数「CONSTANT」の値を「Hello world.」として定義する
```

```
define("CONSTANT", "Hello world.");  
echo CONSTANT; // 「Hello world」を出力する
```

定数名はデフォルトで大文字と小文字を区別します。大文字と小文字を区別しないようにするには、「define」関数の第3引数に「true」を指定します。

```
// 定数名の大文字小文字を区別しない
```

```
define("CONSTANT", "Hello world.", "true");  
echo constant; // 「Hello world」を出力する
```

PHP5.3.0以降では、クラス定義の外部で「const」キーワードを使って定数を定義することもできます。ただし、この場合はトップレベルで宣言しなければなりません。関数やループ、「if」文の内部では宣言することはできません。クラス内の定数（オブジェクト定数）の宣言については、340ページを参照してください。

```
// 「const」キーワードによる定数の定義
```

```
// PHP 5.3.0 以降  
const CONSTANT = 'Hello World';  
echo CONSTANT; // 「Hello world」を出力する
```

特定の定数が定義されているかどうかを調べるには、「defined」関数を使用します。

```
// 「CONSTANT」が定義されているかどうか確認する
```

```
if (defined('CONSTANT')) {  
    echo CONSTANT;  
}
```

#### 関連項目 ▶▶▶

- オブジェクト定数を定義する ..... p.340

# 演算子について

## III PHPの演算子

演算子は、演算を行う記号です。演算子には、優先順位があります。優先順位を変更するには、優先させたい部分を「()」で囲みます。PHPの演算子には、次のような種類があります。

### ▶ 加算子/減算子

加算子(++)は、変数の値をインクリメント(「1」増やす)します。減算子(--)は、変数の値をデクリメント(「1」減らす)します。前置の場合は演算後に結果を返し、後置の場合は演算前の値を返した後に演算を実行します。

演算子	説明
++	インクリメント
--	デクリメント

```
$a = 1; $b = 1; $c = 1; $d = 1;
```

```
// 変数「a」の値に「1」を加えてからを返す
```

```
echo ++$a; // 2
```

```
// 変数「b」の値を返してから「1」を加える
```

```
echo $b++; // 1
```

```
// 変数「c」の値から「1」を引いてからを返す
```

```
echo --$c; // 0;
```

```
// 変数「d」の値を返してから「1」を引く
```

```
echo $d--; // 1
```

### ▶ 論理演算子

論理演算子は、2つの値についての論理値を求めます。「!」(否定)は単項演算子で、後ろに続けて指定された1つの値だけを評価します。結果は「true(TRUE)」か「false(FALSE)」で返されます。この演算子は、条件分岐などに使用します。

演算子	説明
!	値が「false」の場合に「true」、「true」の場合に「false」を返す(否定)
&&	2つの値が共に「true」の場合に「true」を返す(論理積)
	2つの値のどちらかが「true」の場合に「true」を返す(論理和)
and	2つの値が共に「true」の場合に「true」を返す(論理積)
xor	2つの値が共に「true」の場合に「false」を、それ以外の場合に「true」を返す(排他的論理和)
or	2つの値のどちらかが「true」の場合に「true」を返す(論理和)

```
$a = true; $b = true;
```

```
var_dump(!$a); // false
```

```
var_dump($a && $b); // true
```

```
var_dump($a || $b); // true
var_dump($a and $b); // true
var_dump($a xor $b); // false
var_dump($a or $b); // true
```

▶ 代数演算子

代数演算子は、四則演算を行います。一般的な基礎代数と同様に、「乗算」「除算」「剰余」は、「加算」「減算」より優先されます。なお、「-」演算子を変数名や値、および、式の手前に付けると符号が反転します。

演算子	説明
*	乗算
/	除算
%	剰余
+	加算
-	減算

```
$a = 10; $b = 3;

echo $a + 2 * $b; // 16
echo ($a + 2) / $b; // 4
echo -$a; // -10
echo $a * (-$b); // -30
echo ($a + -2) * $b; // 24
echo -( $a + 2) / $b; // -4
```

▶ ビット演算子

ビット演算子は、値を2進数にしてビット単位での演算を行います。特定のビットを評価したり、操作したりするために使います。

演算子	説明
~	NOT(否定)
<<	左ビットシフト
>>	右ビットシフト
&	AND(ビット積)
^	XOR(排他的論理和)
	OR(ビット和)

```
$a = 3; // 0011
$b = 5; // 0101

echo decbin(~$a); // 11111111111111111111111111111100
echo decbin($a << 2); // 1100
echo decbin($b >> 1); // 0010
echo decbin($a & $b); // 0001
echo decbin($a ^ $b); // 0101
echo decbin($a | $b); // 0111
```

## ▶ 比較演算子

比較演算子は、2つの値を比較します。結果は「true (TRUE)」か「false (FALSE)」で返されます。この演算子は、条件分岐などに使用します。論理演算子と組み合わせることで複雑な条件を指定することができます。

演算子	説明
<	左辺値が右辺値より小さい場合に「true」を返す
<=	左辺値が右辺値以下の場合に「true」を返す
>	左辺値が右辺値より大きい場合に「true」を返す
>=	左辺値が右辺値以上の場合に「true」を返す
<>	左辺値が右辺値に等しくない場合に「true」を返す
==	左辺値が右辺値に等しい場合に「true」を返す
===	左辺値が右辺値に等しく、同じ型である場合に「true」を返す
!=	左辺値が右辺値に等しくない場合に「true」を返す
!==	左辺値が右辺値に等しくないか、同じ型でない場合に「true」を返す

## ▶ 三項演算子

三項演算子は、条件式の真偽に応じた値を設定するための演算子です。三項演算子の構文は、次の通りです。

(式1) ? (式2) : (式3)

「式1」が「true (TRUE)」の場合に「式2」を、「式1」が「false (FALSE)」の場合は「式3」を値とします。なお、PHP5.3以降では、三項演算子の真ん中の部分を省略することもできます。その場合は、「式1」が「true」の場合は「式1」を、それ以外の場合は「式3」を値とします。

(式1) ? : (式3)

## ▶ 配列演算子

配列演算子は、2つの配列の結合と代入、比較を行います。比較結果は「true (TRUE)」か「false (FALSE)」で返されます。

演算子	説明
+	2つの配列を結合する
==	2つの配列をキーと値のペアが等しい場合に「true」を返す
===	2つの配列をキーと値のペアが等しく、その並び順が等しく、かつ、型も等しい場合に「true」を返す
!=	2つの配列が等しくない場合に「true」を返す
<>	2つの配列が等しくない場合に「true」を返す
!==	2つの配列が同一でない場合に「true」を返す
=>	左辺で指定したキー名の要素の値に右辺値を代入する

## ▶ エラー制御演算子

エラー制御演算子「@」は、エラーメッセージの出力を抑制します。「@」をPHPの関数や式の前に付けた場合、その関数や式によって生成されたエラーメッセージは無視されます(724ページ参照)。

▶ 実行演算子

実行演算子は、シェルコマンドを実行するための演算子です。「`」(バッククォート)で囲んだシェルコマンドを実行し、出力を返します。この演算子は、「shell\_exec」関数と等価です。

▶ 型演算子

型演算子「instanceof」は、変数が特定のクラスのオブジェクトのインスタンスであるかどうかを調べます(381ページ参照)。

▶ 複合演算子

複合演算子は、演算と代入の式を省略形で表すための演算子です。式の中の値を使用し、その値をその式の結果とすることができます。複合演算子は、すべての演算子に関して使用することができます。複合演算子の主な例は、次の通りです。

演算子	説明	例	省略前の式
+=	加算代入	\$a += \$b	\$a = \$a + \$b
-=	減算代入	\$a -= \$b	\$a = \$a - \$b
&=	AND代入	\$a &= \$b	\$a = \$a & \$b
!=	OR代入	\$a  = \$b	\$a = \$a   \$b
.=	文字列結合代入	\$a .= \$b	\$a = \$a . \$b

COLUMN

特別な演算子について

PHPには、特別な演算子として、「&」演算子、「new」演算子(キーワード)、「clone」演算子(キーワード)、「[]」演算子などがあります。

▶ 「&」演算子

「&」演算子は、参照による代入を行うための演算子です(56ページ参照)。

▶ 「new」演算子

「new」演算子は、クラスのオブジェクトを作成するための演算子です(323ページ参照)。

▶ 「clone」演算子

「clone」演算子はオブジェクトのクローンを作成するための演算子です(363ページ参照)。

▶ 「[]」演算子

「[]」演算子は、配列の要素を参照するための演算子です。たとえば、「\$myarray」配列の2番目の要素を参照するには、「myarray[2]」と記述します。

関連項目 ▶▶▶

- オブジェクトを検査する ..... p.381
- エラーメッセージの出力を抑制する ..... p.724

# コメントについて

## PHPのコメント

コメントは、プログラムの実行中に無視される部分です。ユーザー定義の関数や特定の命令についての説明をコメントとして記述しておくと、プログラムの可読性が向上し、デバッグやメンテナンスが容易になります。PHPは、3つの形式のコメントをサポートしています。

形式	説明
Unixシェル型	「#」から行末をコメントとする(単一行用)
C/C++	「//」から行末までをコメントとする(単一行用)
C/C++	「/*」から「*/」までをコメントとする(複数行用)

```
$a = 'Hello'; // C/C++型の単一行用のコメント

$b = 'World'; # Unixシェル型の単一行用のコメント

echo $a . "World";

/*
C/C++型の複数行用のコメント

この中の行はすべてコメントとして無視される

*/

/*
 * C/C++型の複数行用のコメント
 *
 * 行頭を揃えるために、「*」を付けて記述する場合もある
 *
 */
```



# 条件分岐について

## III PHPでの条件分岐の種類

条件分岐は、すべてのプログラミング言語において最も重要な機能の1つです。PHPには、「if」と「switch」を使った条件分岐があります。

### ▶ 「if」による条件分岐

「if」は、条件の真偽によって実行する処理を変化させる条件分岐です。次のような構文で使います。

```
if (条件式1) {  
    条件式1が「真」(true)のときに実行されるコード  
} elseif (条件式2) {  
    条件式2が「真」(true)のときに実行されるコード  
} else {  
    条件式1と条件式2のどちらも「偽」(false)のときに実行されるコード  
}
```

「elseif」や「else」は省略することができます。「elseif」は複数、使用することができます。また、開き波カッコを「:」(コロン)、閉じ波カッコを「endif;」に変更した構文を使用することもできます。

```
if (条件式1) :  
    条件式1が「真」(true)のときに実行されるコード  
elseif (条件式2) :  
    条件式2が「真」(true)のときに実行されるコード  
else :  
    条件式1と条件式2のどちらも「偽」(false)のときに実行されるコード  
endif;
```

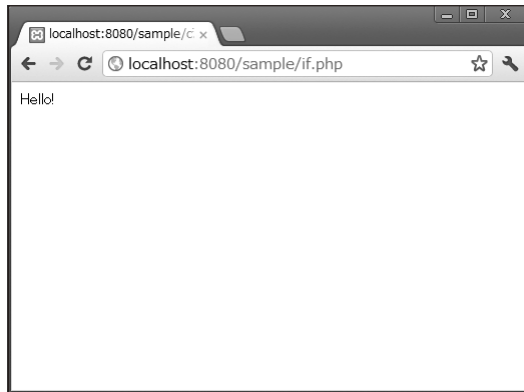
次のコードは「if」構文を使って、時間ごとに異なるメッセージを出力しています。

```
<?php  
  
$h =date('G');  
  
if ($h > 18) {  
    echo "Good evening!";  
} elseif ($h > 9 ) {  
    echo "Hello!";  
} elseif ($h > 5 ) {  
    echo "Good morning!";  
} else {  
    echo "sleeping...";  
}
```



```
?>
```

このコードを10時から18時の間に実行した場合は、次のように表示されます。



### ▶ 「switch」による条件分岐

「switch」は、変数または式の結果の値に応じて、異なる処理を実行させる条件分岐です。次のような構文で使います。

```
switch (変数) {
    case 値1:
        変数が値1に等しいときに実行されるコード
        break;
    case 値2:
        変数が値2に等しいときに実行されるコード
        break;
    case 値3:
        変数が値3に等しいときに実行されるコード
        break;
    default:
        変数が「case」に指定したどの値にも一致しないときに実行されるコード
}
```

最後の「default」節は、省略することができます。条件は上から順に評価され、変数(または式)の値と一致する値を持つ「case」節の処理を実行し、最初の「break」文まで実行を続けます。このため、「case」節の終わりに「break」文を書かなかった場合、次の「case」節が実行されるので注意が必要です。

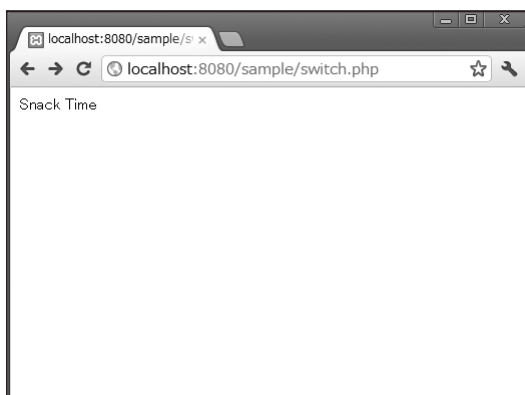
次のコードは「switch」構文を使って、時間ごとに異なるメッセージを出力しています。

```
<?php

$h = date('G');
```

```
switch ($h) {  
    case 7:  
        echo "Breakfast Time";  
        break;  
    case 12:  
        echo "Lunch Time";  
        break;  
    case 15:  
        echo "Snack Time";  
        break;  
    default:  
        echo "not break time.";  
}  
  
?>
```

このコードを15時に実行した場合は、次のように表示されます。



#### COLUMN 「elseif」の別の書き方と注意点

PHPでは「elseif」を、「else」と「if」を半角スペースで区切って「else if」と書くこともできます。ただし、「:」（コロン）を使って「if/elseif」条件を指定する場合は、「else if」のように分割してはいけません。分割すると、パースエラーとなります。

# ループ(繰り返し処理)について

## PHPのループの種類

PHPのループには、「while」「do-while」「for」「foreach」の4種類の構文があります。

### ▶「while」ループ

「while」ループは、条件式が真(true)である間、入れ子の文を繰り返し実行します。このループは、繰り返す回数が不明な場合に使います。

```
// 波カッコ構文
while (条件式) {
    繰り返すコード
}
```

```
// コロン構文
while (条件式) :
    繰り返すコード
endwhile;
```

### ▶「do-while」ループ

「do-while」ループは、条件式の評価を最後に行うこと以外は、「while」ループと同じです。「do-while」ループは、最低1回の実行が保証されています。一方、「while」ループは、最初に条件式を評価するため、繰り返し処理を一度も実行せずにループを終了する場合があります。「do-while」ループには、コロン構文はありません。

```
do {
    繰り返すコード
} while (条件式);
```

### ▶「for」ループ

「for」ループは、初期化式、条件式、増減式の3つを一度に指定できるループです。このループは、繰り返す回数が決まっている場合に使います。

```
// 波カッコ構文
for (初期化式; 条件式; 増減式) {
    繰り返すコード
}
```

```
// コロン構文
for (初期化式; 条件式; 増減式) :
    繰り返すコード
endfor;
```

たとえば、変数「i」をカウンタとして、10回、繰り返す場合には、次のように記述します。

```
for ($i = 0; $i < 10; $i++) {
    繰り返すコード
}
```

### ▶ 「foreach」ループ

「foreach」ループは、配列の要素に対して繰り返し処理を実行します。「foreach」ループは、配列とオブジェクトだけに使用できます。

```
foreach (配列 as 値変数) {
    繰り返すコード
}

foreach (配列 as キー変数 => 値変数) {
    繰り返すコード
}
```

最初の形式は、指定した配列に関してループ処理を行い、配列の要素を1つずつ順番に取り出し、現在の要素の値を値変数に代入します。2番目の形式は、さらに各反復で現在の要素のキーをキー変数に代入します(270ページ参照)。

## III ループからの脱出とスキップ

ループから脱出するには「break」を、ループの途中の処理をスキップするには「continue」を使います。

### ▶ 「break」を使ったループからの脱出

「break」は、現在実行中の「for」「foreach」「while」「do-while」「switch」構造の実行を終了します。オプションの引数で、ネストしたループ構造を抜ける数を指定することもできます。

```
$arr = array('1', '2', '3', '4', '5');
while (list(, $val) = each($arr)) {
    if ($val == '3') {
        // 「val=3」のときにループを終了する
        break;
    }
    echo "$val<br />";
}
```

// オプション引数を使用する

```
$i = 0;
while (++$i) {
    switch ($i) {
        case 5:
```



```

    echo "At 5 <br />";
    break 1; // 「switch」構造のみを抜ける
case 10:
    echo "At 10; quitting <br />";
    break 2; // 「switch」と「while」抜ける
default:
    break;
}
}

```

### ▶ 「continue」を使ったループ処理のスキップ

「continue」は、現在の繰り返しループの残りの処理をスキップし、次の繰り返しループの処理に進みます。オプションの引数で、処理をスキップするループ構造のレベルの数を指定することもできます。

```

$arr = array('1', '2', '3', '4', '5');
while (list($key, $value) = each($arr)) {
    if (!$key % 2) {
        // キーが偶数の要素をスキップする
        continue;
    }
    echo "$value<br />";
}

```

// オプション引数を使用する

```

$i = 0;
while ($i++ < 3) {
    echo "Outer<br />";
    while (1) {
        echo "&nbsp;&nbsp;&nbsp;Middle<br />";
        while (1) {
            echo "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Inner<br />";
            continue 3; // 3レベルスキップする
        }
        echo "This never gets output.<br />";
    }
    echo "Neither does this.<br />";
}

```

### 関連項目 ▶▶▶

- 配列のキーと値のペアを取得する ..... p.270

## 関数について

### III PHPの関数の定義について

PHPでは、他のプログラミング言語を同様に、自分で関数を定義することができます。自分で定義した関数を「ユーザー定義関数」といいます。

#### ▶ 定義方法

関数は、次のような構文で定義します。

```
// 値を返さない関数
function 関数名 (引数1, 引数2, … 引数n) {
    処理コード
}

// 値を返す関数
function 関数名 (引数1, 引数2, … 引数n) {
    処理コード
    return 返り値;
}
```

引数は、必要な数だけ「,」(カンマ)で区切って指定します。不要な場合は、省略することもできます。また、関数は、必要に応じて「return」により値を返すことができます。

#### ▶ 関数の呼び出し

関数の呼び出しは、次のような構文で行います。

```
// 値を返さない関数
関数名(引数1, 引数2, … 引数n);

// 値を返す関数
変数 = 関数名(引数1, 引数2, … 引数n);
```

PHPは可変関数の概念をサポートしているので、次のように変数を使用して関数を呼び出すこともできます。

```
function foo($name) {
    echo "Hello! $name <br />\n";
}

// 「echo」のラッパー関数(「echo」の機能を内包した関数)
function echoit($string) {
    echo $string . "<br />\n";
}
```



// 関数名「foo」を変数「func」に代入する

```
$func = 'foo';
```

// 「foo」関数を呼び出す

```
$func('Tom'); // Hello! Tom
```

// 「echoit」関数を呼び出す

```
$func = 'echoit';
```

```
$func('Hello!'); // Hello!
```

なお、可変関数は、「echo」「unset」「isset」「empty」「include」「print」のような言語構造と組み合わせて使用することはできません。これらの言語構造を可変変数として使うには、独自のラッパー関数を使う必要があります。

### III 無名関数について

無名関数は、関数名も持たない関数で、「クロージャ」とも呼ばれます。無名関数は、PHP 5.3.0以降で使用可能です。

#### ▶ 定義方法

無名関数の呼び出しは、次のような構文で行います。

// コールバックパラメータに使用する場合

```
function (引数1, 引数2,...引数n) {
    処理コード
    return 返回值;
}
```

// 変数へ代入する場合

```
変数 = function(引数1, 引数2,...引数n){
    処理コード
};
```

// コールバック関数として使用する

```
echo preg_replace_callback('~([a-z])~',
    function ($match) {
        return strtoupper($match[1]);
    }, 'hello-world'), "<br />\n";
// helloWorld
```

// 変数への代入する

```
$greet = function($name) {
    echo "Hello $name <br />\n";
};
$greet('World'); // Hello World
```



# 関数の引数について

## III 引数について

引数は、関数を呼び出すときに渡す値です。関数の中では、その値を使って処理を行い、必要に応じて結果を返します。PHPの関数は引数のリストにより、関数へ情報を渡すことができます。このリストは、「,」（カンマ）で区切られた式のリストです。引数の評価は、左から右の順番で行われます。

### ▶ 「値渡し」と「参照渡し」について

引数には、「値渡し」と「参照渡し」があります。「値渡し」は、変数のコピーを渡す方法で、関数側で変数の内容を変更しても、元の変数には影響がありません。「参照渡し」は、変数への参照(56ページ)を渡す方法で、関数側で変数の内容を変更すると、呼び出し側の変数も同様に更新されます。

PHPのデフォルトは「値渡し」です。「参照渡し」するには、関数の定義において、アンバサンド「&」を引数名の前に付加します。

```
// 引数「string」を「参照渡し」する関数
function add_str(&$string) {
    $string = 'Hello '. $string;
}

$str = 'World';
add_str($str);
echo $str;    // 「Hello World」を出力する
```

### ▶ デフォルト引数値

デフォルト引数値は、引数の既定値です。デフォルト引数値を指定するには、関数の定義において、代入演算子の「=」を使って、引数にデフォルト値を指定します。PHPでは、配列やNULLをデフォルト値にすることもできます。また、PHP5以降では、デフォルト引数値を「参照渡し」にすることもできます。

```
// デフォルト引数値「world」を指定する
function hello($str = "world") {
    return "Hello $str! <br />\n";
}

echo hello();    // Hello world!
echo hello(null); // Hello !
echo hello("Tom"); // Hello Tom!
```

デフォルト引数値を持つ関数を、引数を省略して呼び出した場合は、デフォルト引数値が使われます。引数を指定して呼び出した場合は、指定した引数の値が使われます。

なお、デフォルト引数値は定数式である必要があります。変数やクラスのメンバーは指定することはできません。また、デフォルト引数値を有する引数は、デフォルト引数値がない引数の右側にすべてある必要があります。

```
// 間違った用例
function greet1($str = "Hello", $name) {
    return "$str $name ! <br />\n";
}

// 第2引数を省略して関数をコールすることはできない
echo greet1("Tom");

// 正しい用例
function greet2($name, $str = "Hello") {
    return "$str $name ! <br />\n";
}

// 第2引数を省略して関数をコールすることができる
echo greet2("Tom");
```

### ▶ 可変長引数リスト

可変長引数リストは、引数の数が変わる関数を定義するためにサポートされている機能です。可変長引数リストと使用するには、「func\_num\_args」関数と「func\_get\_args」関数(または「func\_get\_arg」関数)を使います。

「func\_num\_args」関数は関数に渡された引数の数を、「func\_get\_args」関数は関数の引数リストを配列として返します。なお、「func\_get\_arg」関数は、引数のリストから要素を1つ返します。

次のコードは、引数で指定された値の平均値を出力します。

```
<?php
// 平均を求める関数
function avg(){
    $sum = 0;
    // 引数の数だけ処理を繰り返す
    for($i = 0; $i < func_num_args(); $i++){
        // 引数の値の合計を求める
        $sum += func_get_arg($i);
    }
    // 平均値を求める
    $avg = $sum / func_num_args();
    return $avg;
}
echo avg(2,1,2,1,3,4,5,1,3,6); // 2.8
?>
```

### 関連項目 ▶▶▶

● 変数について ..... p.56