

やさしい
PHP
入門

日向俊二●著



■ サンプルファイルのダウンロードについて

本書掲載のサンプルファイルは、下記 URL からダウンロードできます。

<https://----->

- 本書の内容についてのご意見、ご質問は、お名前、ご連絡先を明記のうえ、小社出版部宛文書（郵送または E-mail）でお送りください。
- 電話によるお問い合わせはお受けできません。
- 本書の解説範囲を越える内容のご質問や、本書の内容と無関係なご質問にはお答えできません。
- 匿名のフリーメールアドレスからのお問い合わせには返信しかねます。

本書で取り上げられているシステム名／製品名は、一般に開発各社の登録商標／商品名です。本書では、™ および ® マークは明記していません。本書に掲載されている団体／商品に対して、その商標権を侵害する意図は一切ありません。本書で紹介している URL や各サイトの内容は変更される場合があります。

はじめに

PHP（ピー・エイチ・ピー）はオープンソースの汎用プログラミング言語とその公式の処理系を指します。本書では、主に PHP のプログラミング言語としての側面に焦点を当てます。

PHP は、Web サーバーで実行されて Web ページを生成するために使われることが多いですが、プログラミング言語としての PHP の用途はそれだけにとどまりません。

本書では、最初に PHP 全体について解説したあと、前半ではプログラミング言語としての PHP について解説し、後半では Web サーバーで実行される PHP プログラムについて解説します。そのため、Web サーバーで PHP を利用するときに必要な、Web サーバーのことや HTML とその関連技術について、最初はまったく知らなくても PHP の学習を始めることができます。そして後半でサーバーで実行する PHP プログラムファイルや HTML に埋め込んでサーバーで実行する PHP スクリプトについて解説します。このときには HTML の知識が必要になるので、PHP のプログラミングに必要な範囲で HTML についても説明します。このように段階を追って学習することで、一度に多くの事項に惑わされることなく、楽しく容易に PHP を習得することができます。

本書を活用して PHP プログラミングの基礎を楽しく学んでください。

2021 年春
著者しるす

本書の表記

- ⌞ 本書の紙面上でプログラムリストやコマンド入力の長い行を折り返して記載していることを示します。実際に入力する際には、この記号およびその前の改行や空白は無視してください。
- abc 斜体の表記は、そこに具体的な文字や数値が入ることを表します。たとえば「PHP 7.*X*」は、*X*に数値が入り、「PHP 7.3」や「PHP 7.4」となることを表します。
- [...] 書式の説明において角括弧で囲った部分は省略可能であることを示します。
- > Windows のコマンドプロンプトを表します。
- \$ Linux など UNIX 系 OS のコマンドプロンプトを表します。
- php > PHP を対話シェル（対話型インタプリタ）として起動したときのプロンプトです。
- > abc 対話シェルのプロンプトや OS のプロンプトに続く太字の表記は、読者が入力する部分です。



.....
本文を補足するような説明や、知っておくとよい話題です。
.....

ご注意

- 本書では、PHP のバージョン *8.m.n* を PHP 8 と呼び、PHP のバージョン *7.m.n* を PHP 7 と呼ぶことがあります。
- 本書の内容は本書執筆時の状態で記述しています。執筆時の PHP の最新のバージョンは 8.0.2 です。ただし、執筆時点で PHP 8 をサポートしていないサーバーが多数あります。また、将来、PHP のバージョンが変わるなど、何らかの理由で記述と実際とが異なる結果となる可能性があります。
- 本書は PHP のすべてのことについて完全に解説するものではありません。必要に応じて PHP のドキュメントなどを参照してください。
- 本書のサンプルは、プログラミングを理解するために掲載するものです。実用的なアプリとして提供するものではありませんので、ユーザーのエラーへの対処や安全

確保、その他の面で省略してあるところがあります。

動作を確認した PHP のバージョン

- 8.0.2 (Windows 10)
- 7.4.9 (Windows 10)
- 7.4.3 (Linux/Ubuntu)

書に関するお問い合わせについて

本書の内容に関するご質問、お問い合わせは、**下記の事項を明記の上**、巻末記載の出版社所在地宛郵送もしくは sales@cutt.co.jp 宛メールにて**文書でお送りください**。

- 氏名、連絡先（住所またはメールアドレス）
- 書名、記載ページ
- 問い合わせ内容
- 実行環境

なお、本書の記載内容から外れたご質問、特に、OS、PHP のバージョン、Web サーバーなどの特定の組み合わせに関するご質問などにはお答えできません。あらかじめご了承ください。

はじめに iii

第 1 章 PHP の概要 1

1.1 PHP について 2
 ◆ プログラミング言語2 ◆ PHP 処理系3 ◆ PHP の用途3

1.2 PHP の実行方法 4
 ◆ コマンドラインで直接実行する4 ◆ 対話シェルで実行する4
 ◆ スクリプトファイルを実行する4 ◆ Web サーバーで実行する5

1.3 PHP のバージョン 6
 ◆ バージョンの呼称6 ◆ バージョンによる違い6
 ◆ 最近のバージョン7 ◆ PHP のバージョン確認7

1.4 PHP と関連言語 9
 ◆ HTML9 ◆ JavaScript9 ◆ 役割の分担10

第 2 章 はじめての PHP 11

2.1 単純な PHP プログラムの実行 12
 ◆ コマンドラインで直接実行する12 ◆ 対話シェルで実行する16

2.2 スクリプトの実行 18
 ◆ スクリプトファイル18 ◆ ファイルの作成19 ◆ ファイルの保存20
 ◆ スクリプトの実行20 ◆ PHP コードの視覚的な分離22
 ◆ 複数のコードの記述22 ◆ 短縮形式23 ◆ PHP 以外の要素23

2.3 PHP の日本語対応 24
 ◆ 日本語の文字列24 ◆ 実行時の日本語対応25 ◆ 日本語の名前25

練習問題 26

第 3 章 PHP の基本的な要素 27

3.1 文 28
 ◆ 文28 ◆ インデント30 ◆ コメント30

3.2 名前と文字 31
 ◆ キーワード31 ◆ 名前の文字32 ◆ エスケープシーケンス33
 ◆ 空白34

3.3	型.....	35
	◆ データ型35 ◆ 論理値型35 ◆ 整数型36 ◆ 浮動小数点数型36	
	◆ 文字列型37 ◆ 配列38 ◆ NULL41 ◆ 型の変換42	
3.4	変数と定数.....	43
	◆ 変数43 ◆ 定数44	
3.5	演算子.....	45
	◆ 単項演算子45 ◆ 加算子/減算子45 ◆ 二項演算子46	
	◆ 代入演算子47 ◆ 参照の代入48 ◆ ビット演算子48	
	◆ 比較演算子49 ◆ 論理演算子50 ◆ 配列演算子51	
	◆ 演算子の結合順序52 ◆ 実行演算子53	
	練習問題	54

第 4 章 入出力.....55

4.1	コンソール出力.....	56
	◆ echo と print56 ◆ printf()57 ◆ 書式指定文字列59	
	◆ print_r()61 ◆ 省略形出力62	
4.2	コンソール入力.....	63
	◆ キーボードからの入力63 ◆ fgets(STDIN)64	
	◆ 数値の入力66 ◆ fscanf(STDIN)66	
4.3	コマンド引数.....	68
	◆ コマンドラインと引数68 ◆ コマンドライン引数の処理68	
4.4	ファイルの読み書き.....	70
	◆ ファイルへのテキストの出力70 ◆ ファイルへの追加73	
	◆ 単純なファイルからの読み込み74 ◆ while 文を使った読み込み76	
	◆ エラー処理77	
	練習問題	78

第 5 章 制御構造.....79

5.1	条件分岐.....	80
	◆ if 文80 ◆ switch 文83 ◆ match 文88	
5.2	繰り返し.....	90
	◆ for 文90 ◆ foreach 文93 ◆ while 文94 ◆ do-while 文96	
	◆ break 文96 ◆ return97 ◆ continue97	
5.3	無条件分岐.....	98
	◆ goto98	
	練習問題	100

第 6 章 関数…… 101

6.1	関数	102
	◆ 関数 ……102 ◆ 文字列処理関数 ……107	
	◆ マルチバイト文字列関数 ……112 ◆ 日時に関する関数 ……115	
6.2	関数の定義	119
	◆ 関数の定義 ……119 ◆ 複数の値を返す関数 ……120 ◆ デフォルト引数値 ……122	
	◆ 可変長引数リスト ……123 ◆ 値渡しと参照渡し ……124 ◆ 関数と変数 ……125	
6.3	さまざまな関数	126
	◆ 変数関数 ……126 ◆ 無名関数 ……127 ◆ 再帰関数 ……127	
6.4	モジュールと名前空間	128
	◆ モジュール ……128 ◆ 名前空間 ……129	
	練習問題	132

第 7 章 クラスとオブジェクト…… 133

7.1	クラスとオブジェクト	134
	◆ オブジェクト ……134	
7.2	クラス	135
	◆ 既存のクラスの例 ……135 ◆ 新しいクラスの定義 ……137	
7.3	継承	143
	◆ ベースクラスとサブクラス ……143 ◆ ベースクラスの定義 ……144	
	◆ サブクラスの定義 ……145	
7.4	インターフェイス	148
	◆ インターフェイス ……148 ◆ インターフェイスの定義と利用 ……148	
	練習問題	152

第 8 章 Web サーバーと HTML…… 153

8.1	ホームページの仕組み	154
	◆ Web サーバーとクライアント ……154 ◆ HTML と PHP ……156	
8.2	Web サーバー	157
	◆ Web サーバーについて ……157 ◆ ローカルサーバー ……158	
	◆ レンタルサーバー ……158 ◆ 自前の公開サーバー ……159	
	◆ ビルトインサーバー ……159	
8.3	HTML	159
	◆ タグ ……160 ◆ ファイルの配置 ……164 ◆ スタイル ……168	
	◆ コメントと特殊タグ ……168	
	練習問題	170

第 9 章 HTML と PHP プログラム 171

9.1	PHP を含む HTML ファイル.....	172
	◆ Web サーバーの働き172 ◆ 単純な例172 ◆ ファイルの配置173	
	◆ Web ブラウザ単独の処理174 ◆ 日本語に対応させた例175	
	◆ 関数を呼び出す例176 ◆ Web サーバー v.s. Web ブラウザ178	
9.2	サーバー上の PHP スクリプトファイル.....	181
	◆ PHP スクリプトの実行181 ◆ PHP スクリプトの中の HTML182	
	練習問題	184

第 10 章 フォームと PHP 185

10.1	フォーム.....	186
	◆ HTML のフォーム186 ◆ HTML の input 要素187	
	◆ データの受け取り190	
10.2	より複雑なフォーム.....	192
	◆ さまざまなフォームの入力要素192 ◆ さまざまなデータの処理195	
	◆ メールの送信199	
10.3	GET メソッド.....	203
	◆ GET メソッドのフォーム203 ◆ PHP ファイル205	
	練習問題	208

第 11 章 データベースとの連携 211

11.1	データベースの基礎.....	212
	◆ データベース212 ◆ データベース管理システム213	
	◆ データベースとデータベースアプリ214 ◆ データベースの構造214	
	◆ カレントレコード215 ◆ キー215 ◆ リレーショナルデータベース216	
	◆ SQL217 ◆ トランザクションとコミット220	
11.2	MySQL.....	221
	◆ MySQL221 ◆ データベースの作成221 ◆ テーブルの作成222	
	◆ データの登録223 ◆ データの取得224	
11.3	PHP でのデータベース操作.....	226
	◆ PDO226 ◆ 接続とデータ取得の流れ226	
	◆ データベースの新規作成と使用230 ◆ テーブル作成230	
	◆ データの登録232 ◆ データの検索233 ◆ データの更新235	
	◆ データの削除236 ◆ テーブルの削除236	
11.4	データベースとフォーム.....	237
	◆ データベースの準備237 ◆ レコードの検索と表示237	
	◆ レコードの登録と更新241	
	練習問題	246

第 12 章 さまざまな話題.....247

12.1 エラー対策とデバッグ	248
◆ 例外処理248	◆ デバッグ249
◆ デバッグの手順250	◆ 最良のデバッグ方法251
12.2 安全確保.....	251
◆ php.ini による設定25	◆ 暗号化254
12.3 応用プログラミング	255
◆ 対話シェルの活用255	◆ コマンドラインツール255
◆ GUI アプリ255	
12.4 パフォーマンス.....	256
◆ PHP のパフォーマンス256	◆ パフォーマンスの改善256
12.5 今後の進み方	257
◆ PHP のさまざまな要素257	◆ HTML と CSS258
◆ データベース258	

付 録.....259

付録 A インストールと設定のヒント	260
付録 B トラブル対策	267
付録 C 練習問題解答例.....	273
付録 D 参考リソース	295
索引.....	297

第 1 章

PHP の概要

ここでは PHP とその使われ方の概要を説明します。

1.1 PHPについて

PHPはオープンソースの汎用プログラミング言語とその公式の処理系を指します。

◆ プログラミング言語

まず、汎用プログラミング言語の側面からPHPについて説明しましょう。

PHPは、Webサーバーで実行されてWebページを生成するために使われることが多いですが、決してWebプログラミング専用の言語というわけではありません。プログラミング言語としてのPHPの用途は多岐に渡ります。他のプログラミング言語と比較して得手不得手はありますが、PHPは厳密には汎用プログラミング言語に分類されます。

PHPが汎用プログラミング言語であるという点は見逃されがちですが重要です。PHPのプログラムを実行するためにWebサーバーが必要不可欠ということはありませんし、実行結果はWebブラウザ上でしか確認できないということもありません。PHPは、HTMLなど他の言語やWebサーバーなどの問題と完全に切り離して、それ単体でプログラミング言語として学習することができます。

PHPは、命令を1つずつ解釈しては実行するインタプリタ言語です。そのため、例えばC言語やGo言語のようなコンパイラ言語と呼ぶプログラミング言語で作成したプログラムよりも実行時の速度が遅いといえますが、特に速度を要求されない一般的な用途には十分実用的な速度で実行されます。

インタプリタ言語は、実行する前にコンピューターが実行できる情報に変換するためのコンパイルという作業がないので、比較的手軽に利用できます。しかし、その反面、実行してみないとわからない間違いが紛れ込むリスクが増すという問題点もあります。ですから、手軽に利用できる言語だからといって安易に取り組んでよいわけではありません。

PHPは、高レベルのプログラムを比較的容易に記述して実行するためのプログラミング言語という観点から、スクリプト言語という種類にも属します。



高レベルのプログラムとは一般ユーザーが通常使うようなプログラムであるのに対して、低レベルのプログラムは機械や装置などに組み込まれていて一般ユーザーが直接使うことがないようなプログラムです。

◆ PHP 処理系

PHP のもう 1 つの側面は、プログラミング言語としての PHP で書かれたプログラムを実行する処理系としての側面です。また、PHP のインタプリタは Web サーバーと連携して、PHP だけでなく HTML や JavaScript、テキストなども使って作成されたドキュメントを適切に処理することができます。

◆ PHP の用途

PHP にはさまざまな用途があります。

現在、最も多く使われているのは、いわゆるホームページ（Web ページ）を作成するという用途です。単に情報を表示するだけの静的な Web ページを作成するには HTML という記述言語だけでもできますが、状況に応じて変わる動的な Web ページを Web サーバーで生成するときにはそのための言語が必要になります。その目的のためによく使われるのが PHP です。

また、PHP はさまざまなコマンドラインツールやなんらかの作業を行うプログラムを作成するときにも使うことができます。

さらに、ウィンドウを利用する GUI アプリケーションを PHP で作成することもできます（本書では取り上げません）。

1.2 PHPの実行方法

PHPはさまざまな方法で実行することができます。具体的な例は第2章と第9章で説明します。

◆ コマンドラインで直接実行する

PHPのプログラムを実行する最も単純な方法は、OSのコマンドラインでプログラムコードを指定してPHPインタプリタを起動して実行する方法です。この方法で実行できるプログラムは単純なものに限られます。

◆ 対話シェルで実行する

PHPの対話型インタプリタを起動して、その中でプログラムコードを入力しながら実行することもできます。この対話モードで実行する環境を「対話シェル (Interactive shell)」といいます。

この方法は、ちょっとした実験をしたり、関数と呼ぶプログラムの部品を試してみたいようなときに便利です。

実際の開発においても、PHPのコードをチェックしたりデバッグする際に、ページ全体をWebサーバーに保存してクライアントから実行をリクエストして調べるよりも、対話シェルでPHPのプログラム部分だけを実行して検証するほうが効率的であることがあります。

◆ スクリプトファイルを実行する

ある程度まとまったPHPのプログラムコードをファイルに保存して、コマンドラインから一気に実行することができます。

この方法を使って、やや複雑な処理を行うツールなどを作成して、必要なときに実行することができます。また、WebでPHPを使っている場合でも、この方法は関数を作成してテストしたり、データベースに直接アクセスするプログラムを作りたいようなと

きなどに便利です。

ここまでの3種類の方法は第2章で具体的に説明します。

◆ Web サーバーで実行する

PHPのプログラム（スクリプトともいう）は、HTMLで記述するWebページの中に埋め込むことができます。また、PHPのプログラムだけを含むWebサーバーで実行されるPHPのスクリプトを作ることができます。これらのプログラムはWebサーバーで実行されます。

この形式のプログラムを実行するには、単にPHPが埋め込まれたWebページをWebサーバーへのリクエストによって表示するか、PHPのスクリプトファイルにアクセスします。ただし、ページの中のPHPプログラムが意図したように実行されるように、Webサーバーに必要なものをインストールして設定しなければなりません。

HTMLで記述するWebページの中に埋め込んだPHPのプログラム（スクリプト）をWebサーバーで実行するためには、HTMLの知識が必要になります。また、Webページをサーバーで表示できるようにするための知識も必要になります。さらに、PHPのプログラムを含むWebページを意図通りに扱えるWebサーバーを用意するか、あるいは、そのように準備されたWebサーバーを借りるなどの必要があります。

このプログラムの実行方法は第9章以降で具体的に説明します。

1.3 PHPのバージョン

PHPは1990年代に作成され、いままでバージョンアップが繰り返されています。

◆ バージョンの呼称

PHPのバージョンは、正確には次の形式で表記します。

メジャーバージョン・マイナーバージョン・リリースバージョン

例えば、「8.0.2」のように表記できますが、一般的には、8.0や7.4あるいは5.6のようにリリースバージョンを省略して表記することがよくあります。

さらにマイナーバージョンを省略して、PHPのバージョン8.mをPHP 8と呼び、PHPのバージョン7.mをPHP 7と呼ぶことがあります。

PHP 5より前の古いバージョンのPHPは現在ではほとんど使われません。また、PHP 5.6のセキュリティ対応が2018年12月で終了しているため、PHP 5もやむをえない場合だけ使うようにして、新しいプログラムではPHP 7以降を使うのが良いでしょう。

◆ バージョンによる違い

バージョンアップされると、機能や関数などが追加されます。既存の部分は引き継がれることが多いので、基本的には前のバージョンと互換性があります。バージョンごとの変更の主な点は次のドキュメントに記載されています。

<https://www.php.net/manual/ja/appendices.php>

また、詳細の変更については次のドキュメントで確認できます。

<https://www.php.net/manual/ja/index.php>

次の2点には特に注意が必要です。

- いくつかの点で前のバージョンで使えたものが使えない、あるいは詳細が変わっていることがあります。前のバージョンのソースを新しいバージョン用に書き換えるときには変更された部分をチェックする必要があります。
- PHP 5 以前など古いバージョンと PHP 7 や PHP 8 では互換性がない部分が多くなります。安全確保のためにも古いバージョンを使うことは推奨しません。



本書の内容は PHP 5.6 より古いバージョンには適用できない部分があります。

◆ 最近のバージョン ◆

2020年11月にPHP 8.0がリリースされ、本書執筆時の現在の最新バージョンは8.0.2です。最新のバージョンはPHP 8であるともいえます。ただし、現時点では多くのレンタルサーバーはPHP 8をサポートしておらず、PHP 7やPHP 5をサポートしているでしょう。

ローカルで（自分のマシンで）PHPのプログラムを実行したい場合や自分でWebサーバーを準備する場合は、PHP 8でもPHP 7でも好みのものをインストールして実行することができます。しかし、サーバーを借りてWebページを公開する際にPHPを利用するときには、サーバーの管理者に利用可能なPHPのバージョンを確認しておく必要があります。

◆ PHPのバージョン確認 ◆

ローカルなPHPのバージョンはPHPをインストールしたマシンで「php -v」を実行して確認することができます。

```
>php -v
PHP 7.4.15 (cli) (built: Feb  2 2021 20:47:45) ( ZTS Visual C++ 2017 x64 )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies

>
```

あるいは次のように表示されます。

```
>php -v
PHP 8.0.2 (cli) (built: Feb  3 2021 18:36:40) ( ZTS Visual C++ 2019 x64 )
Copyright (c) The PHP Group
Zend Engine v4.0.2, Copyright (c) Zend Technologies

>
```

Linux の場合は例えば次のように出力されます。

```
$ php -v
PHP 7.4.3 (cli) (built: Oct  6 2020 15:47:56) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
$
```

PHP のバージョンは `phpversion()` や `phpinfo()` という PHP の関数と呼ぶものを使って調べることもできます。ただし、サーバーを提供している組織によっては、サーバー保護のためにこれらの関数を実行できないように設定している場合があります。

1.4 PHP と関連言語

特に Web サーバーで PHP を活用するときには、PHP 単独ではなく、HTML と JavaScript という言語を活用することがあります。

◆ HTML

HTML (HyperText Markup Language) は主に Web ページ (ホームページともいう) を記述するために使われます。HTML は、Web ページの見出しやテキスト、入力フィールドなどの要素や、そのページの属性を示す情報などを記述するために使います。

PHP に対応した Web サーバーは、HTML 中の PHP コードの出力を HTML 中に埋め込んで Web ブラウザなどの Web クライアントに送ります。Web ページで PHP を使うときには、HTML の知識も必須になります。

なお、HTML の表現力を高めるために CSS (Cascading Style Sheets) というものをよく使います。CSS は、ページに表示する要素の配置や見栄えなどを指定するために使われます。HTML を使うときにはよほど単純な場合を除いて一般的には CSS も併用するので、事実上、CSS は HTML の一部であるとみなすことができます。そのため、Web ページで PHP を使うときには、HTML と同様に CSS の知識も必要になります。

◆ JavaScript

HTML は原則として静的なページ (変化のない) を表現する言語です。Web ブラウザなどの Web クライアントで、例えば表示されたそのときの Web クライアントの時刻を表示するというように、そのときの状況に合わせて表示を変える場合には、一般的には JavaScript が使われます。

JavaScript のごく単純な例を第 9 章で示しますが、JavaScript については本書ではそれ以上は扱いません。

◆ 役割の分担

Web サーバーで PHP を活用するときに、それぞれの言語は次のように役割を分担しています。

- HTML

Web ブラウザのような Web クライアントに表示する内容を記述します。また、PHP や JavaScript のスクリプト（プログラム）を内部に記述するためにも使われます。

- PHP

Web サーバーで PHP を活用するときに、PHP はサーバーで実行すべき処理を行います。例えば、Web ページの所定の場所にユーザーが入力した情報をもとに、Web サーバーにあるデータベースから情報を検索して表示するときに PHP を使います。

- JavaScript

Web クライアントで実行すべき処理を行います。例えば、Web ページの所定の場所にユーザーが情報を入力したかどうか確認するときに JavaScript を使います。