

ちゃんと使える力を身につける

Webと プログラミングの きほんのきほん

大澤文孝 著

解消

Webプログラムが
なぜ動くのか、
よくわからないんです...



言語に限定されない
応用が効く基礎力をつける!

- ✓ インターネットのしくみがわかる!
- ✓ Webプログラムがどのように動いているかを学べる!
- ✓ 正規表現やテンプレート、ライブラリ、セキュリティまでカバー!

ちゃんと使える力を身につける

Webと プログラミングの きほんのきほん

大澤文孝 著



■本書のサポートサイト

<http://book.mynavi.jp/support/pc/5409/>

本書に掲載しているサンプルファイルをダウンロードできます。また、訂正情報や補足情報などを掲載していきます。

■本書に記載された内容は、情報の提供のみを目的としております。したがって、本書を用いての運用はすべてお客様自身の責任と判断において行ってください。

■本書の制作にあたっては正確な記述につとめました。著者や出版社のいずれも、本書の内容に関して何らかの保証をするものではなく、内容に関するいかなる運用結果についても一切の責任を負いません。あらかじめご了承ください。

■本書は2015年2月段階での情報に基づいて執筆されています。本書に登場するサービス情報、URL、商品情報などは、すべて原稿執筆時点でのものです。執筆以降に変更されている可能性がありますので、ご了承ください。

■本書の内容は著作物です。著作権者の許可を得ずに無断で複写・複製・転載することは禁じられています。

■本書中の会社名や商品名は、該当する各社の商標または登録商標です。本書中では™および®は省略させていただいております。

これからWebプログラムを学ぶ人へ

Webプログラムは、Webブラウザから実行されるプログラムの総称です。

「キーワードを入力すると結果が表示される検索エンジン」「ボタンをクリックするとモノが買えるショッピングサイト」など、Webブラウザを使って操作するさまざまな場面で、Webプログラムが使われています。

本書は、インターネットやWebの仕組みを探りながら、Webプログラムの基礎を習得することを目指した書です。



インターネットやWebの仕組みを知る重要性

本書は、Webプログラムの書です。ですから、実際にプログラムを作ります。しかし、それは本書の目的の半分に過ぎません。

本書では、最初に、インターネットやWebの仕組みから話を進めます。本書の半ばまで、Webプログラムの実際の作り方は、登場しません。

すぐにWebプログラムを始めたい人にとって、これは、じれったい構成かも知れません。でも、挫折しないためには、仕組みを知ることが大事です。



データの規定

インターネットに限らず、コンピュータ上では、あらゆるものが、文字や数値が並んだ「データ」として表現されます。

Webブラウザでは、レイアウトされた美しいページが表示されたり、動画が再生されたり、ドラッグ&ドロップで操作できたりしますが、これらも、すべて、文字や数値が並んだデータとして表現されているものに過ぎません。

具体的には、Chapter1で説明しますが、たとえば、Webページで大見出しを示すには、「<h1>見出し</h1>」のように、全体をh1で括ったデータを渡すという規定があります。

このように規定された書式のデータをWebブラウザに与えることが、思い通りに操るときの基本です。

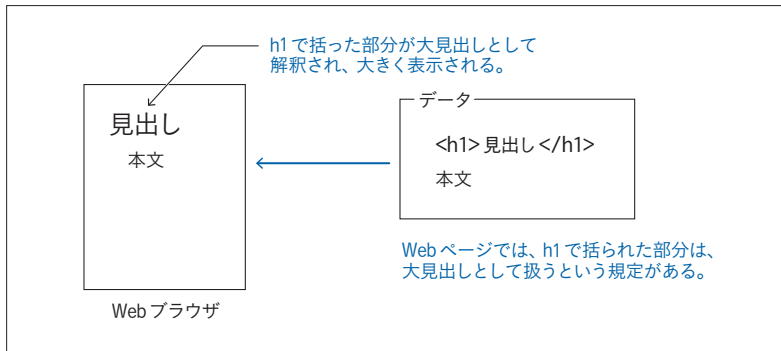


図 1-1
Web ブラウザを操る基本



Web プログラムはデータを作るためのもの

Web プログラムは、規定されたデータを作るものに過ぎません。

つまり、大見出しを表示したいなら、Web プログラムから「<h1>見出し</h1>」という文字を表示するようにプログラムを作ります。

ここで注目したいのは、Web プログラムには、「大見出しを作る」という命令は、基本的には、存在しないということです（なぜ「基本的に」と断っているのかは、すぐあとにフレームワークについて説明するときに明らかになります）。Web プログラムから、「<h1>見出し</h1>」と表示すれば、それが Web ブラウザによって、大見出しとして解釈されるのに過ぎないのです。

このように考えると、Web プログラムの世界では、どのような仕組みで、どのような規定のデータがやりとりされているかを理解するのが、とても大事です。

「<h1>で囲まれていたら大見出しにする」という Web の取り決めを知らずして、Web プログラムを作れるわけがありません。

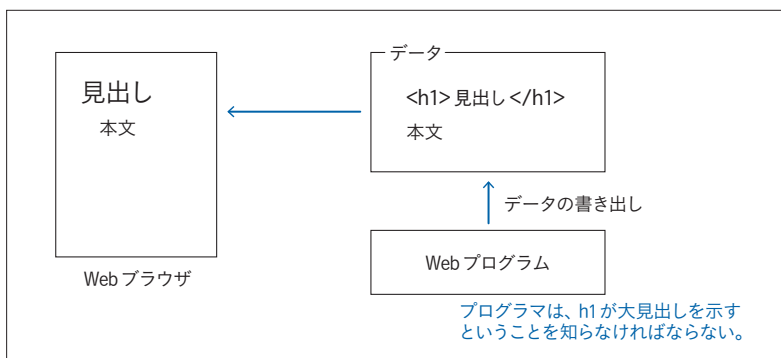


図 1-2
Web プログラムはデータを作るもの



プログラミング言語による手法の違い

Web プログラムは、何らかのプログラミング言語を使って記述します。プログラミング言語には、たとえば、PHP、Perl、Java、Python、Ruby など、さまざまなものがあり、文法が違います。

どのプログラミング言語で記述しても、同じことを実現できます。たとえば、「<h1>見出し</h1>」という文字を表示する場合、次のように文法が異なります。しかし、ブラウザで表示される結果は、どれも同じです。

PHP

```
echo "<h1>大見出し</h1>";
```

Perl

```
echo "<h1>大見出し</h1>";
```

Java

```
System.out.print("<h1>大見出し</h1>");
```

Python

```
print "<h1>大見出し</h1>"
```

Ruby

```
print("<h1>大見出し</h1>");
```



「〇〇は、こういう意味」ではなく 「〇〇したいから〇〇する」という考え方をする

プログラムを生業としている人の多くは、1つではなく、いくつかのプログラミング言語を扱えます。たくさんのプログラミング言語を習得するのは大変と思うかも知れませんが、実は、そんなに難しくありません。

その理由は、どのようなプログラミング言語でも、基本的な構造や概念は同じであるからです。

たくさんのプログラミング言語を扱うプログラマは、「〇〇は、こういう意味」ではなく「〇〇したいから〇〇する」という考え方を身につけています。

たとえば、

```
echo "<h1>見出し</h1>";
```

を、「見出しを表示する命令」という理解はしていません。そうではなくて、

- ①見出しを表示したい
- ②そのためには「<h1> 見出し</h1>」と表示する必要がある
- ③そのプログラミング言語で、文字を表示する命令を調べる
- ④プログラミング言語のリファレンスマニュアルを見たら、echo という命令が文字を表示するということがわかった
- ⑤よって、「echo "<h1> 見出し</h1>";」とすればよい

という考え方をします。

プログラミング言語によって違うのは、④と⑤だけです。

ですから、プログラマは、リファレンスマニュアルを調べて、基本的な構文だけ分かりさえすれば、どんなプログラミング言語でも扱えるのです。

MEMO

リファレンス（Reference）マニュアルとは、プログラミング言語の機能がまとめられた仕様書のことです。機能別やアルファベット順などで、プログラミング言語がもつ機能の一覧が記載されています。



隠蔽化するフレームワークの弊害

ところで、実際には、「見出しを表示する」という命令が、あらかじめ用意されているプログラミング環境もあります。そのような環境は、「フレームワーク（Framework）」と呼ばれます。

フレームワークは、基本的な命令の集合体です。たとえば、「見出しを表示する」という機能は、内部で「<h1>」を表示する処理に置き換えられます。

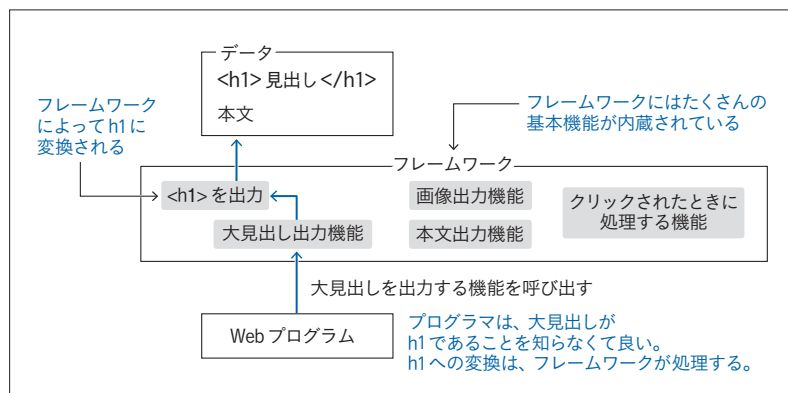


図 1-3
フレームワークを使ったプログラミング

フレームワークには、2つの利点があります。

①プログラマが細部を知らなくて済む

プログラマは、フレームワークに用意された命令さえ知っていれば良く、インターネットやWebの仕組みを知る必要がありません。

そのため、短時間で習得でき、よく使う命令さえ覚えれば、すぐにプログラムを始められます。

②プログラマの負担を減らせる

フレームワークには、たくさんの便利な機能が備わっています。

プログラマは、それらの機能を利用することによって、自分で長いプログラムを書かなくても、その機能呼び出すだけの短いプログラムで実現できます。

そうすれば、プログラムを記述する時間が減り、効率が高まるのはもちろんですし、自分で書かない分だけ、プログラムに不具合が入り込む余地も少なくなります。

②は利点ですが、①は利点でありながら欠点でもあります。隠されることは、これからWebプログラムを始める人にとって、本質を理解しにくくします。

確かに、フレームワークで提供される命令だけを覚えれば、ひととおりWebプログラムができるかも知れません。

しかし、その環境に慣れしてしまうと、もう、フレームワークなしでプログラムを書くことができなくなり、さまざまなプログラミング言語を操るのが困難になります。

ですから、これからプログラムをはじめる人こそ、フレームワークを使わずに、仕組みから知ってもらいたいのです。

フレームワークは、プログラミング言語によって異なります。たとえば、PHPのフレームワークとしては、「CakePHP」「CodeIgniter」「FuelPHP」などがあります。Perlのフレームワークとしては「Catalyst」などがあります。Javaのフレームワークとしては「Struts」「Spring」「Seasar」などがあります。Rubyのフレームワークとしては「Ruby on Rails」が有名です。



MEMO



Webプログラムの周辺技術と省力化のためのライブラリ

さて、Webプログラムは、それ単体で完結することはありません。実用的なWebプログラムでは、いくつかの周辺技術と組み合わせられます。



データを保存するためのデータベース

なかでも、よく使われるのが、データベースです。データベースは、データの格納庫です。顧客情報や商品情報、購入履歴など、さまざまなデータを保存するときに使われます。

もし、あなたが、顧客、商品、購入履歴のような、永続的に保存したいデータを扱いたいと思うなら、データベースの知識が必要です。

これはWebプログラムとは関係ない、別の技術ですが、習得が必須です。



省力化できるライブラリ

Webプログラムは、年々進化しています。最近のトレンドやプログラムのコードを減らせる省力化の手法を身につけておくと、とても役立ちます。

いまでは、Webプログラムは、とても大きくなってしまったため、すべての機能を自分で記述することは、ありません。代わりに、「誰かが作った便利なプログラム」を組み合わせます。これらは「ライブラリ」と呼ばれます。無償で使えるフリーのライブラリも多く、それらを使えば、自分が記述するコードを減らせ、しかも、自分で書くよりも、ずっと安定性の高いプログラムになります。

ライブラリには、流行廃りがあります。皆がよく使っているライブラリのトレンドを知り、開発スタイルをそれに併せていくことが、短時間で良質なプログラムを作るのに欠かせません。



本書を読み進めるに当たって

本書は、ここまで述べた思想のもとに書かれています。

大きく、次の3つに分けられます。

➡ 1 インターネットやWebの仕組み -----

Chapter1からChapter3では、インターネットやWebの仕組みを説明します。Webプログラムをするには、そもそも、それが動作するWebサーバが必要です。実際にインターネット上にWebプログラムを置く場合、Webサーバは、どのような環境のものを用意し、どこに構築すればよいのか、その要件も説明します。

➡ 2 プログラミングの基本 -----

Chapter4とChapter5が、プログラミングの基本的な話になります。

この2つの章は、実際に、簡単に試せる構成にしました。

Chapter4では、Windowsに「XAMPP」というWebプログラムを実行できる環境をインストールし、PHPを使ってプログラムする方法を説明します。Chapter5では、PHPを題材に、実際に簡単なWebプログラムをしながら、プログラムの基本的な流れと構文を説明します。

言語はPHPですが、大きくPHPに依存する題材は、ありません。ここで扱う内容は、文法の差こそあれ、どのようなプログラミング言語にも通用するものです。

➡ 3 Webプログラムの応用 -----

Chapter6以降は、応用です。実際のWebプログラム開発の現場では、どのような技術が使われているのか、そのトレンドを紹介します。

Chapter6では、省力化できるライブラリや操作性を向上させるJavaScriptの扱い方を説明します。Chapter7では、ショッピングサイトで「カゴの中身」を実現するのに不可欠な、Cookieの仕組みを説明します。最後のChapter8では、データベースの使い方を説明します。

本書は、本当に基礎的な入門書です。手っ取り早く習得したい人にとっては、少し遠回りに思えるかも知れません。

読者の皆様は、本書を読み終えたとき、きっと別の、今度は、より実践的な入門書を手にとっていることでしょう。

そうしたときに、一見、遠回りに見える本書が、実践的な入門書を理解する手助けとなり、「このプログラムの意味は、実は、こういう意味だったのか」と思い起こすきっかけとなれば、筆者として幸いです。

2015年1月

大澤 文孝

Chapter 1	WebブラウザでWebページが表示される仕組み	17
	この章の内容	18
Section 01	Webブラウザの役割	20
	Webサイトのコンテンツのやりとり	20
	「静的ページ」と「動的ページ」	21
	Webブラウザの働き	23
Section 02	Webページを構成する「HTML」	25
	「HTML」という言語	25
	Webブラウザによる「レンダリング」	26
	リンクから別のページに辿る仕組み	27
	画像や動画は別に読み込まれてHTMLにはめ込まれる	28
Section 03	文字コードと文字化け	30
	文字化けとその解消方法	30
	文字コード	31
	UnicodeとUTF-8	33
	UTF-8が使われる理由	35
Section 04	レイアウトを指定するCSS	36
	要素に対してレイアウトを指定する	36
	CSSを変えるとデザインが変わる	37
Section 05	Webブラウザでプログラムを実行するJavaScript	39
	JavaScriptのプログラムを実行する	39
	JavaScriptの利用例	40
Chapter 2	WebサーバからWebブラウザにコンテンツが届くまで	41
	この章の内容	42
Section 01	インターネットを構成するネットワーク	44
	日本のインターネットを司る「IX（インターネット・エクスチェンジ）」	44
	ユーザーの回線とサーバの関係	45
Section 02	IPアドレスの割り当て	46
	相手を区別するIPアドレス	46
	IPアドレスの決め方	47

Section 03	ルータを使った環境でのIPアドレス	51
	ルータが内蔵するNAT機能	51
	同じIPアドレスでも、違うクライアントを指すことがある	54
Section 04	ドメイン名とIPアドレスとの関係	55
	ドメイン名をIPアドレスに変換する	55
	階層化されたドメイン名	56
Section 05	ドメイン名とIPアドレスを相互変換する「DNSサーバ」	58
	ドメイン名を検索するためのDNSサーバ	58
	DNSサーバでドメイン名がたどられる仕組み	60
	Webサーバをドメイン名で運用するために必要なこと	63
Section 06	通信を振り分ける「ポート」	64
	「保証された通信」と「保証されない通信」	64
	通信が混じらないようにする「ポート番号」	65
	ウェルノウンポートとエフェメラルポート	66
Section 07	WebサーバソフトとHTTP	69
	HTTPを使った通信	69
	Webサーバの応答	70
Section 08	暗号化するSSL	72
	データを暗号化するSSL	72
	公開鍵暗号方式	73
	SSLを使うには	79
Section 09	Webサーバを構築するには	83
	Webサーバを設置する	83
	DNSサーバを設定する	85
	Webサーバソフトをインストール・設定する	86
	Web以外に必要な設定	86

Chapter 3 Webプログラムがコンテンツを作る仕組み 89

	この章の内容	90
Section 01	クライアントサイドとサーバサイド	92
	クライアントサイドのプログラム	92
	サーバサイドのプログラム	95

Section 02	Webサーバで実行可能なプログラミング言語	98
	サーバサイドで実行できる主なプログラミング言語	98
	インタプリタとコンパイラの違い	100
	人気のプログラミング言語	101
Section 03	Webサーバでプログラムを実行する仕組み	102
	ファイルを見せるのではなくプログラムの実行結果を返す	102
	CGI	103
	モジュール	107
Section 04	入力された内容を送信する入力フォーム	109
	入力フォームとは	109
	フォームを構成する要素	110
	フォームの送信方式	111
	URL エンコード	113
Section 05	JavaScript と Ajax	114
	JavaScript は HTML と連携する	114
	イベントドリブンによるプログラミング	115
	要素を id で指定して操作するのが JavaScript 流のプログラミング	116
	サーバサイドのプログラムと連携する Ajax	117
Section 06	データベースの必要性	119
	データを保存して検索できるようにするデータベース	119
	主なデータベースソフト	119
Chapter 4	Web プログラムを動かしてみよう	121
	この章の内容	122
Section 01	Web プログラミングの開発環境を揃える	124
	プログラミング言語を決める	124
	オールインワンパッケージの実行環境「XAMPP」	124
Section 02	XAMPP の入手とインストール	127
	XAMPP のインストール	127
	XAMPP を起動する	129
	XAMPP のドキュメントルート場所	131
Section 03	テキストエディタで PHP のプログラムを記述する	133

	テキストエディタを用意する	133
	拡張子を表示するようにしておく	135
	PHPのプログラムを記述する	137
	実行してみる	141
Section 04	PHPプログラムの基本	143
	プログラムの部分を「<?php」と「?>」で囲む	143
	文末には「;」を記述する	144
	文字の並びを示すときは、「"」または「'」で囲む	144
	空白や改行は、見やすくするためのもの	146
	プログラムにはコメントを記述できる	146
	エラーがあるときは、エラーメッセージが表示される	147
	警告とエラーの違い	148
	バグはエラーとは限らない	149

Chapter 5 Webプログラミングをしてみよう 151

	この章の内容	152
Section 01	入力フォームのデータを読む - ①GETメソッドの場合	154
	入力フォームを作る	154
	入力された名前を、そのまま返すプログラム	156
	オウム返しにはHTMLエンコードが必須	159
Section 02	入力フォームのデータを読む - ②POSTメソッドの場合	162
	POSTメソッドを使った例	162
	開発者機能を使って送受信されるデータを見る	163
Section 03	関数について知る	167
	関数は、何かまとまった処理をするもの	167
	関数の例	168
	関数リファレンスを参照する	169
Section 04	変数、四則演算、データ型	173
	足し算する例	173
	変数と型	174
Section 05	条件判定する	181
	フォームを介さずに呼び出したときのエラー	181

	条件を判定する if 文	183
	POST の値が設定されていなかったときにエラーメッセージを表示する	184
	値の大小を比較する	187
	複数の条件は括弧で括って組み合わせるのが安全	190
Section 06	繰り返し処理	193
	繰り返し処理するための構文	193
	回数を数える	195
	繰り返しのための for 構文	199
	二項演算子で表記を簡単に記述する	201
	ループ処理を途中でやめたいとき	203
Section 07	データをまとめる配列とハッシュ	206
	一連のデータをまとめて扱う	206
	配列のメリット	208
	ハッシュ	209
Chapter 6	ライブラリや JavaScript を使ったプログラミング	215
	この章の内容	216
Section 01	ライブラリと API	218
	ライブラリと API	218
	ライブラリを使うとコードの記述量が減る	219
	ライブラリや API を使うには、インストールが必要	220
Section 02	正規表現で書式をチェックする	223
	正規表現を使ったバリデーション	223
	正規表現を使って文字列の並びをチェックする	223
	PHP で正規表現を使う	226
	文字列を分割したり置換したりする	229
Section 03	出力をテンプレート化する	233
	表形式でデータを出力する例	233
	テンプレートを作って流し込む	234
	Smarty を使ってテンプレート処理する	236
Section 04	CSS と JavaScript で装飾する	244
	操作性を良くする工夫	244

動きを付けるためにCSSとJavaScriptを組み合わせる	245
JavaScriptの互換性を吸収するライブラリ	250
各種ユーザーインターフェイスを提供するライブラリ	254

Chapter 7 ユーザーを追跡するCookieとセッション情報 259

この章の内容	260
Section 01 ユーザーの状態を管理する	262
複数のページから構成されるWebプログラム	262
ユーザーの操作情報を保存しておく	263
共通領域をクライアント側に保存するかサーバ側に保存するか	265
Section 02 クライアント側にデータを保持するCookieの仕組み	267
Cookieの仕組み	267
Cookieを使った実例	269
開発者ツールでCookieを見る	274
Cookieは確実ではない	277
Section 03 サーバ側にデータを保存するセッション情報	278
セッション情報ではユーザーを識別するIDを付ける	278
セッション情報を使った実例	280
セッションIDを見る	283
セッションのタイムアウト	284
セッション情報を使う場合の注意点	286

Chapter 8 データベースを使ったプログラミング 289

この章の内容	290
Section 01 データベースとは	292
データベースとプログラミング言語との関係	292
リレーショナルデータベース	293
SQLを使ったデータベース操作	297
Section 02 データベースやテーブルを作成する	300
MySQLを起動する	300

	MySQLの管理ツールを起動する	302
	データベースを作成する	303
	ユーザーを作成する	304
	テーブルを作成する	306
Section 03	SQLでデータ操作する	310
	INSERT文でレコードを追加する	310
	SELECT文でレコードを参照する	312
	UPDATE文でレコードを更新する	315
	DELETE文でレコードを削除する	317
Section 04	プログラムからテーブルを操作する	319
	データベース操作の基本	319
	PHPからMySQLデータベースにINSERT文を発行してレコードを追加する	320
	PHPからMySQLデータベースにSELECT発行してレコードを取得する	324
Section 05	フォームからレコードを操作する	327
	フォームに入力された情報をレコードとして登録する	327
	不正な文字が入力できないように対策する	330
	索引	333



Webブラウザで Webページが 表示される仕組み

CHAPTER

1

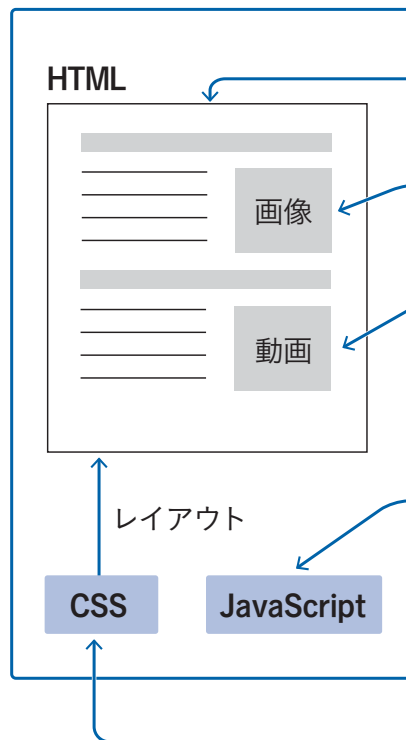
サイトを見るときに使うソフトが、Webブラウザです。サイトのコンテンツは、Webサーバに置かれています。この章では、Webブラウザが、Webサーバのデータを、どのように処理することで、Webページが見られるようになっているのか、その仕組みを説明します。



この章の内容



Web ページ

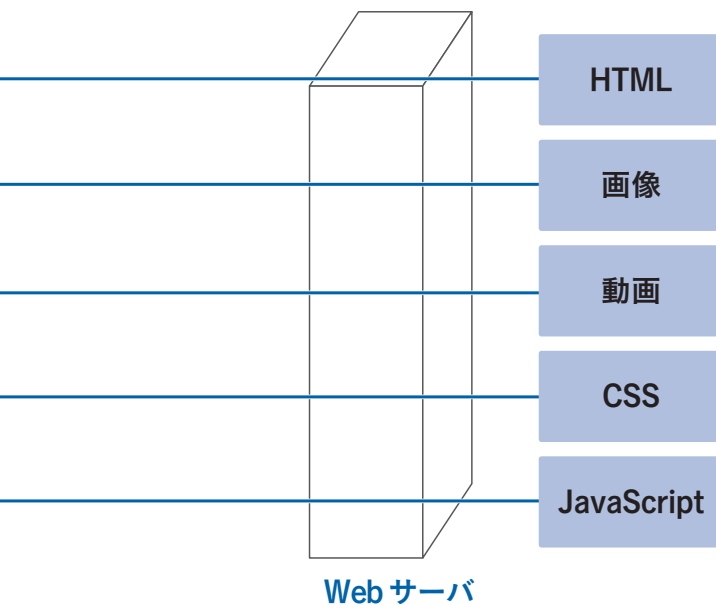


Webのコンテンツは、Webサーバに置かれています。置かれたコンテンツのデータをダウンロードして、ユーザーに表示する機能をもつのがWebブラウザです。

コンテンツは、HTMLという書式のデータで構成されています。これに画像をはめ込んだり、レイアウトを適用したりすることによって、最終的なWebページが表示されます。

Webブラウザには、ブラウザの内部でプログラムを実行する機能もあります。この機能を使うと、キーボード操作やマウス操作などによって動くWebページを構成できます。

1つのWebページは、主に、次の4つの要素で構成されています。必須なのは、①のHTMLのみです。それ以外はオプションです。



① HTML

Webのコンテンツが記述されています。

② 画像や動画など

①に差し込まれる画像や動画です。

③ CSS

①のHTMLの、タイトルや見出し、本文、画像などを、どのようにレイアウトするのかを指示します。

④ JavaScript

Webブラウザの内部で実行されるプログラムです。



Webブラウザの役割

SECTION

01



パソコンでもスマホでも、インターネットの閲覧には、Internet ExplorerやFirefox、Google Chromeなどの「Webブラウザ（Web browser）」を使います。Webブラウザは、サーバからコンテンツをダウンロードして、ユーザーに表示します。



Webサイトのコンテンツのやりとり

まずは、Webブラウザを使ってWebサイトを見ているとき、パソコンやWebブラウザは、どのような動きをしているのかを、見ていきましょう。

Webに関する規格は、「W3C（World Wide Web Consortium。http://www.w3c.org/）」という業界団体が策定しています。



クライアントとサーバ

たとえばユーザーが、Webブラウザのブックマーク（お気に入り機能）に登録している架空のWebサイト「マイナビ水族館（http://www.mynavi.aqua.co.jp/index.html）※1」にアクセスするとします。

Webサイトは、インターネットに接続されている、いずれかのコンピュータが提供しています。このコンピュータのことを、「**サーバ（Webサーバ）**」と言います。

ユーザーが、このブックマークをクリックすると、Webブラウザはマイナビ水族館のウェブサイトを提供しているサーバに接続します。すると、ページの内容が得られます。

このとき情報を取得する端末（Webブラウザが動作しているパソコンやスマートフォン）のことを、「**クライアント（Client。Webクライアント）**」と言います。

※1）マイナビ水族館は実在しない架空の水族館・Webサイトです。

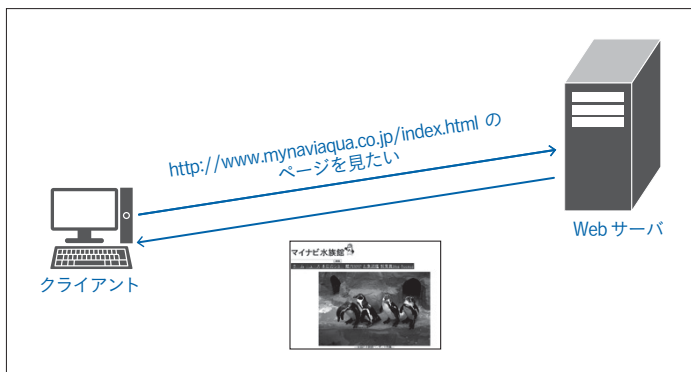


図 1-1-1 クライアントとサーバとのやりとり



URL

Webサイトの場所を特定するのが、**URL (Uniform Resource Locator)** です。URLには、接続方式を示す「プロトコル (protocol) の名前」、そのWebサイトの情報を提供する「サーバの名前」、そしてサーバ内の場所を特定する「パス (path) の名前」が含まれています。

ブックマークから辿ったり、Webブラウザのアドレス欄に「http://www.mynaviaqua.co.jp/index.html」と入力したりした場合、Webブラウザは、そのWebサーバ (www.mynaviaqua.co.jp) に接続し、URLのパス (/index.html) を要求します。するとWebサーバは、そのパスに該当するデータを返します。

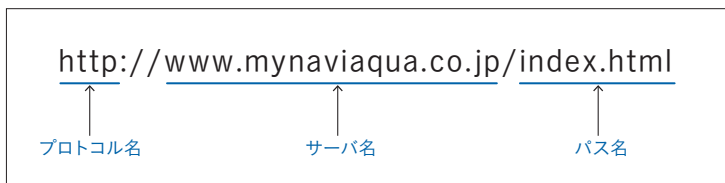


図 1-1-2 URLの構成



「静的ページ」と「動的ページ」

Webページには、アクセスしたときに、同じ内容が表示されるページ

(静的ページ) と、そうでないページ (検索結果などの動的ページ) があります。この2つは、サーバの構成が異なります。



静的ページ

サーバに、あらかじめファイルとしてページを置いておくものを「**静的ページ** (static page)」と言います。

Webサーバは、いつでも、そのファイルの内容を返す処理だけをします。つまり、誰かが、そのファイルを書き換えない限り、ずっと同じ内容が表示されます。



静的というのは、「コンテンツのデータが変化しない」という意味であり、「動きがない」という意味ではありません。たとえば、Flashや動画、アニメーションGIFなどで動くコンテンツであっても、誰かがサーバのファイルを書き換えるまで内容が変わらないなら、それは静的ページです。

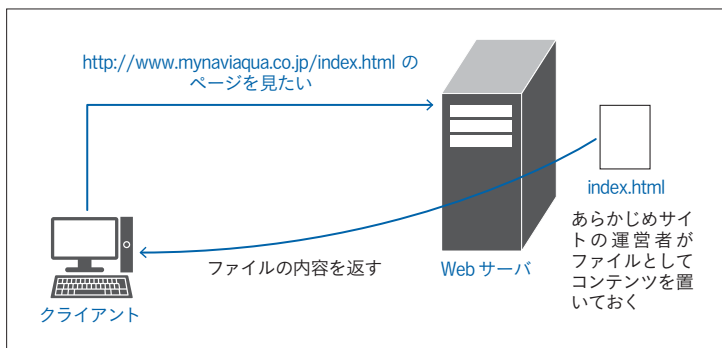


図 1-1-3 静的ページの構成



動的ページ

静的ページとは異なり、サーバ上のプログラムが毎回ページを生成しているのが、「**動的ページ** (dynamic page)」です。

プログラムでは、ユーザーの入力や現在の状態を見てコンテンツを生成することで、そのときに応じて、ページの内容に変化を与えられます。

検索サイトのように、入力した検索キーワードによって結果が変わるも

のや、掲示板のように書き込みのたびに更新されるページ、そして、メールフォームや通販サイトなども、動的ページとして構成されています。

動的ページを作るのに必要なのが、本書の主題でもあるWebプログラミングです。

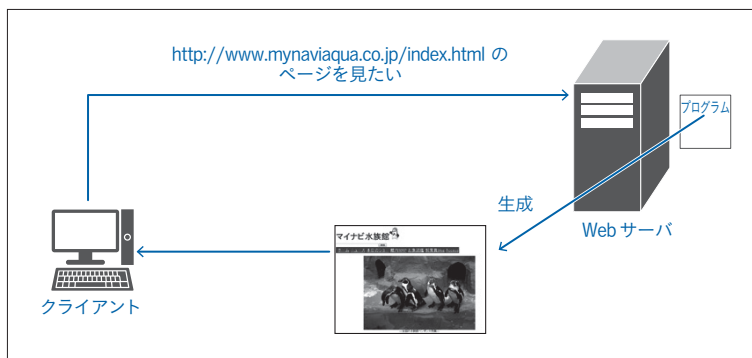


図 1-1-4 動的ページの構成



Webブラウザの働き

普段、意識することは少ないですが、Webブラウザとは、Webにアクセスする際に、人間と対話するソフトです。入力や選択など、人の要求に応じて働きます。

Webブラウザの主要な機能は、次の3つです。

⇒ 1 Webにアクセスしてページを表示する

URLを入力・選択したときに、対応するWebサーバにアクセスしてデータを取得し、その内容を表示する機能です。

Webサーバから取得したデータを整形して、ユーザーの目に見えるかたちで表示することを、「**レンダリング** (rendering)」と言います。

⇒ 2 入力フォームからのデータ入力

人が入力したデータを、相手のWebサーバに送信する機能です。検索サイトで検索の文言を打ち込んだり、掲示板サイトや通販サイトで名前や情報を書き込んだりしたデータを送信するのがこれにあたります。

Webページに設けられた入力欄のことを、「**入力フォーム**」と言います。

➡ 3 プログラムの実行機能

マウスやキーボードの動きなどによって、アクションを起こすプログラムを実行する機能です。たとえば、Google マップでは、マウスのドラッグで地図が動きますが、これは、「地図を動かすプログラム」が、Web ブラウザで実行されているからです。

動きを作るプログラムの手法は、いくつかあります。

Web ブラウザでは、JavaScript という仕組みを使って作られたプログラムを実行できます。

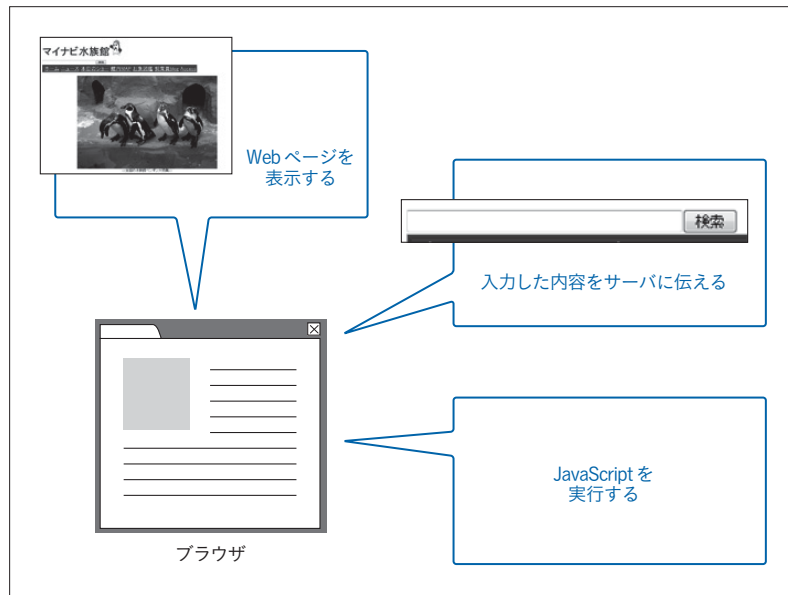


図 1-1-5 Web ブラウザの主要な機能



SECTION

02



Webページを構成する「HTML」

Webページは、「HTML (HyperText Markup Language)」という文法で記述されています。Webブラウザは、これを解釈して適切な書式に直して、ユーザーに表示します。



「HTML」という言語

HTMLは、Webページを表記するために考案された書式です。「言語」といっても複雑なことはなく、元となるテキストに、どの部分が「見出し」「本文」「画像」なのかなどの意味付けを示したものに過ぎません。



マークアップランゲージ

HTMLは、「**マークアップランゲージ** (markup language)」と呼ばれ、意味付けしたい箇所を「タグ (tag)」で囲んで（マークアップして）表現するのが特徴です。

タグは、たとえば、「<title>マイナビ水族館</title>」のように「<タグ>*****</タグ>」の形式で記述します。タグで示された部分を、「要素 (element)」と言います。

タグは、用途によって、次の3つに分類できます。



1 意味を決めるもの

「title (タイトル)」「h1 (大見出し)」「h2 (中見出し)」など。



2 別コンテンツへの参照を示すもの

「img (画像を埋め込むことを示す)」「video (動画を埋め込むことを示す)」など。



MEMO

「h1」「h2」などの「h」は、「header (見出し)」の略です。「h1」から「h6」まで定義されています。

➡ 3 装飾するもの

「b（ボールド＝太字）」「font（文字色や文字サイズ）」など。

ただし、装飾のタグに関しては、HTMLで指定することは現在では推奨されておらず、CSS（p.36 参照）で、装飾を指定することが求められています。

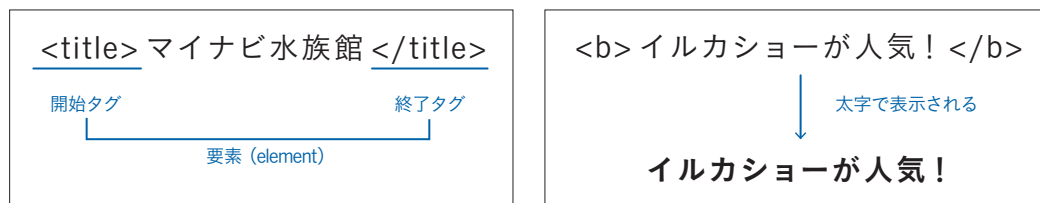


図 1-2-1 タグの例



Web ブラウザによる「レンダリング」

Web サーバからは、HTML の形式で、Web ページが送信されてきます。Web ブラウザは、この HTML を解釈・レイアウトして、ユーザーに表示します。たとえば、「見出しのタグで囲まれている部分を大きく」「太字のタグで囲まれているところは太く」というように整形します。

前の項目で、Web ブラウザが HTML を解釈して整形することを「レンダリング」と言うを書きましたが、Web ブラウザに内蔵されているレンダリング機能のことを、「**レンダリングエンジン** (rendering engine)」と呼びます。

基本的には、Web ブラウザが違えば、レンダリングエンジンも違います。

しかし最近では、独自のレンダリングエンジンではなく、Apple 社を中心に開発された「**WebKit** (ウェブキット)」と呼ばれる無償で使えるレンダリングエンジンを採用している Web ブラウザもあります。そうした Web ブラウザであれば、表示結果は、ほぼ同じです。

表 1-2-1 主な Web ブラウザと採用しているレンダリングエンジン

Web ブラウザ	レンダリングエンジン
Internet Explorer	Trident (3.0 以前は Spyglass など)
Mozilla Firefox	Gecko
Google Chrome	Blink (27 以前は WebKit)
Safari	WebKit
Opera	Blink (14 以前は WebKit、Presto など)



Webブラウザによって、見栄えや動きが違う

HTMLはレンダリングエンジンによって解釈・レイアウトされるので、レンダリングエンジンが違えば、Webページの表示が異なることがあります。

また、Webブラウザのなかでプログラムを実行できる機能である「JavaScript」の機能差もあるため、Webページの構成によっては、あるWebブラウザでは動かないということもあります。

そのため、Webサイトを作るときは、いくつかのWebブラウザで動作確認が必要です。

特に、企業ユーザーをはじめとする多くのユーザーは、OSに付属しているInternet Explorerを使っているため、このブラウザでの動作確認は欠かせません。

同じブラウザであってもバージョンが違えば、挙動が異なることもあります。



リンクから別のページに辿る仕組み

Webページに張られたリンクをクリックすると、別のページに飛ぶことができます。リンクは略称で、正式には「**ハイパーリンク** (hyperlink)」です。リンク (ハイパーリンク) は、リンクタグ「``」として構成されています。

たとえば、「マイナビ水族館お知らせのページ (<http://www.mynaviaqua.co.jp/oshirase.html>)」へのリンクであれば、

```
<a href="http://www.mynaviaqua.co.jp/oshirase.html">お知らせ</a>
```

のように記されています。

ユーザーがリンクをクリックすると、Webブラウザは「指定されたURLのデータを取得」して、それを表示します。

ページからページへと移動しているように感じますが、実際は、新しいページを次々とダウンロードして表示しているのに過ぎません。

Chapter
1

Chapter
2

Chapter
3

Chapter
4

Chapter
5

Chapter
6

Chapter
7

Chapter
8

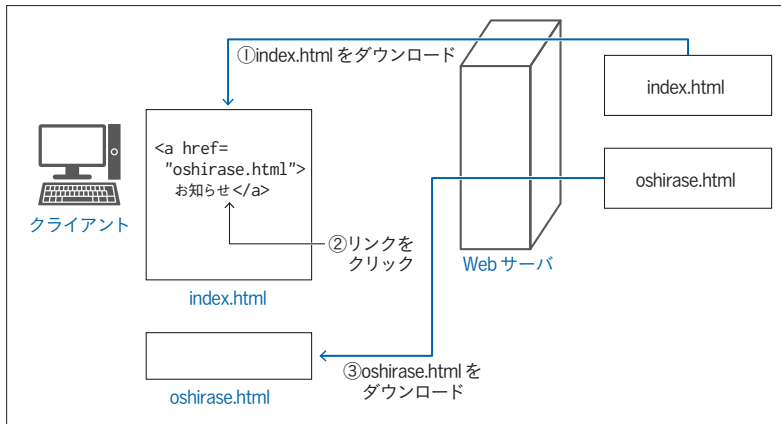


図1-2-2 リンクをクリックしてページを辿る



画像や動画は別に読み込まれてHTMLにはめ込まれる -

文章やタグは、HTMLに、すべて記述されています。

ただし、画像や動画ファイルは、そのHTMLとは別にWebサーバ上に置かれます。

HTMLには、「**画像ファイルはこの場所にある** ()」という情報しか記載されていません。

Webブラウザは、この画像データを、別途、Webサーバに要求してダウンロードします。そしてダウンロードした画像を、ページ内に、はめ込むように表示します。

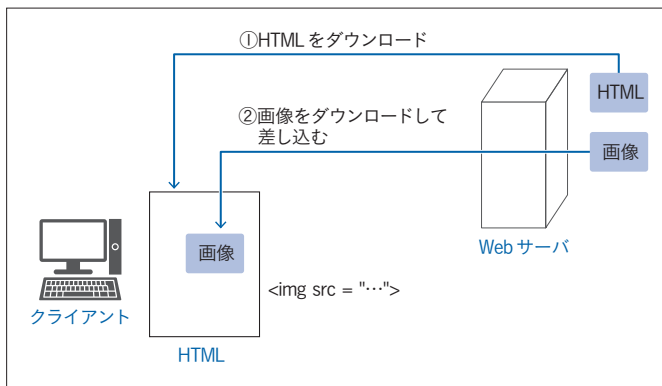


図1-2-3 画像は別途ダウンロードされ、ページにはめ込まれる

HTMLには、いくつかの規格があります。現在では、「XHTML 1.1」と「HTML5」がよく使われます。違いは、タグの種類と書き方です。

どの規格でHTMLを記述しているのかは、「DTD宣言」を、HTMLの冒頭に記述することで、Webブラウザに伝えます。たとえば、XHTML 1.1の場合は、冒頭に次のように記述します。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

DTD宣言は、レンダリングの結果にも影響を与えます。

Webブラウザはレンダリング機能として、「**互換モード**」と「**標準モード**」の2種類のモードを備えています。

「互換モード」は古いWebブラウザを模範にしてレイアウトするもので、最新のHTML規格とは、少し違った動きをします。HTMLの冒頭に「<!DOCTYPE>」が記述されていないときは、「互換モード」で動くので、レイアウトが少し異なることがあります。本書はデザインを目的とした書籍ではないので、提示するサンプルでは、DTD宣言を省略していますが、微細なレイアウトをするときは、DTD宣言の有無に注意してください。

表 1-2-A HTMLの主な規格

HTMLの規格	意味	DTD宣言
HTML 4.01 Transitional DTD	1990年12月にW3Cによって策定された規格。非推奨な要素を含む	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
HTML 4.01 Strict DTD	上記のうち、非推奨な要素を含まない	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
HTML 4.01 Frameset DTD	上記のうち、非推奨の一部であるフレーム機能（ページ内部を分割して、それぞれ別のURLのコンテンツを表示する機能）を含む	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
XHTML 1.1	要素を示すタグは、必ず閉じる（で始めたら、で終わる必要がある）ように構成したもの	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
HTML5	最新のHTML規格。動画機能や通信機能、データベース機能やファイルの読み書き機能などが含まれる総合的な規格	<!DOCTYPE html>



文字コードと文字化け

SECTION

03

Webページを見ているときに、意味不明な記号や漢字、カタカナなどが表示されて読めないことがあります。これが俗に言う「文字化け」です。「記述した文字コード」と「解釈した文字コード」が異なることによって起こります。



文字化けとその解消方法

文字化けが発生するのは、**文字コード**の解釈方法が違うのが原因です。

もし文字化けが発生したときは、Webブラウザの[エンコード]メニューから、適切な文字コードを選択することで直せます。



図 1-3-1 文字化けの例

文字コード

文字コードとは、文字に割り当てられた番号のことです。「A」「あ」「漢」など、パソコンで扱うすべての文字には、何らかの数値が割り当てられており、その数値によって、どの文字なのかが区別されます。

英語の文字コード

英語の文字コードは、ほとんどの場合「**ASCIIコード**（アスキーコード）」という番号が使われます。

アスキーコードでは、数値の32が「空白」、そのあと、いくつかの記号が続き、48～57が数字、またいくつかの記号が続き、65から「A」「B」「C」…と英語の大文字が割り当てられ、97から「a」「b」「c」と小文字が続くというように割り当てられています。

ASCIIコードでは、文字のコードは「0から255」の範囲の値です。この範囲は、2の8乗であり、「1バイト（byte）」で表現します。

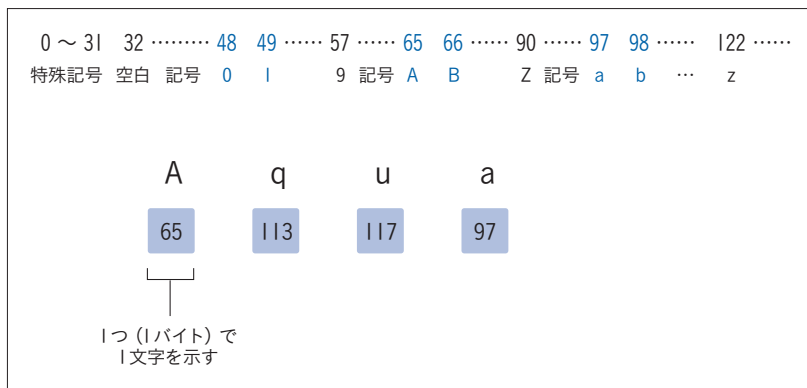


図1-3-2 ASCIIコード

日本語の文字コード

日本語は文字の数が多いので、ASCIIコードの「0から255」の範囲では表現できません。そこで日本語の文字コードは、「0から255」の範囲の場所を「2つ」使って、2バイトで表現します。



試し読みはお楽しみ
いただけましたか？

ここからはManatee
おすすめの商品を
ご紹介します。

Manatee Tech Book Zone 

3.6
2017

1

おすすめ
電子書籍

2

Manatee

体験型の本書でプログラミングの第1歩を踏みだそう!

『やさしくはじめるiPhoneアプリ作りの教科書 [Swift 3& Xcode 8.2対応]』は、iPhoneアプリを作ってみたく初心者のための入門書です。プログラミングが初めての人、苦手意識がある人でも楽しく学んでいけるよう、なるべくやさしく、イラストや図をたくさん使って解説しています。本書では実際にサンプルアプリを作りながら学んでいきますが、イラストによる解説で、一歩ずつ丁寧に、iPhoneアプリ作りの基本と楽しさを学べます。

豊富なイラストで「なぜ?」を解消! Javaの第一歩を踏み出そう

『スッキリわかるJava入門 第2版』は、Javaの基礎から初学者には難しいとされるオブジェクト指向まで、膨らむ疑問にしっかり対応しました。Javaプログラミングの「なぜ?」がわかる解説と約300点の豊富なイラストで、楽しく・詳しく・スッキリとマスターできる構成となっています。「なんとなくJavaを使っているけれど、オブジェクト指向の理解には自信がない」「学習の途中で挫折してしまった」という方にもおすすめです。

「プログラミング」

イラストによる解説で、プログラミングはじめての人でも学べる

300ものイラストで楽しく・詳しく・スッキリとマスター!

Chapter 1-3

アプリ開発の基本的な流れは?

アプリは5つの手順で作る

さて、iPhoneアプリはどのように作っていくのでしょうか? 最初はSwiftというプログラミング言語を使いますが、Swift言語だけで作るわけではなく、実際に動かすにはXcodeという開発環境が必要です。XcodeにはSwiftというプログラミング言語と、実際に動かすためのXcodeという開発環境がセットになっています。

1. プロジェクトを作る
2. アプリの画面を作る
3. 部品とプログラムをつなぐ
4. プログラムを書く
5. シミュレーターでテストする

テンプレートを選ぶ

アプリを作るには、あらかじめ用意されているテンプレートを選びます。テンプレートにはあらかじめ画面のレイアウトや、必要な部品が設定されています。これによって、アプリの作成が簡単になります。



テンプレートを選ぶ	説明
Single View Application	「Single View」を選択。これは単一の画面を持つアプリです。このテンプレートでは、画面のレイアウトや、必要な部品が設定されています。
Table View Application	「Table View」を選択。これはリスト表示のアプリです。このテンプレートでは、リストの表示方法や、必要な部品が設定されています。
Master-Details Application	「Master-Details」を選択。これはマスター/ディテール表示のアプリです。このテンプレートでは、マスター/ディテール表示の方法や、必要な部品が設定されています。
Page-Based Application	「Page-Based」を選択。これはページ表示のアプリです。このテンプレートでは、ページの表示方法や、必要な部品が設定されています。
Navigation Application	「Navigation」を選択。これはナビゲーション表示のアプリです。このテンプレートでは、ナビゲーションの方法や、必要な部品が設定されています。
Tab Bar Application	「Tab Bar」を選択。これはタブ表示のアプリです。このテンプレートでは、タブの表示方法や、必要な部品が設定されています。
Message Application	「Message」を選択。これはメッセージ表示のアプリです。このテンプレートでは、メッセージの表示方法や、必要な部品が設定されています。

作成するサンプルアプリはシンプルで、意味を理解しながら作っている

「書くのは、わかっていても面白くないから...」
「それは残念なことです!」「面白くないのは、まだ慣れないからです。Javaは簡単で、面白いですよ!」

書くのが楽しくなるように、この本では、Javaの基礎から、オブジェクト指向まで、やさしく解説しています。

実行結果

かんたん!かんたん!かんたん!

0.1 たくさんの文章を表示しよう

画面に2行以上の文章を表示させることもできます。次のように2行を書き足してコンパイル・実行しましょう。

```
リスト 0-4
public class Main {
    public static void main(String[] args) {
        System.out.println("こんにちは");
        System.out.println("ようこそ");
        System.out.println("ようこそ");
    }
}
```

実行結果

こんにちは
ようこそ
ようこそ

会話のやりとりの中にも、開発現場でのヒントが詰め込まれている

やさしくはじめる iPhone アプリ作りの教科書 [Swift 3& Xcode 8.2 対応]

マイナビ出版
森崎尚(著者)、
まつむらまきお(イラスト)
312 ページ 価格: 3,002 円 (PDF)



Swift
(iOS 開発)

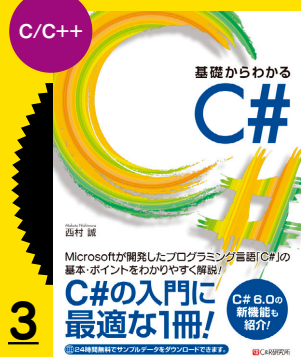
スッキリわかる Java 入門 第2版

インプレス
中山清義・国本大悟(著者)
658 ページ
価格: 2,376 円 (PDF)



Java

**幅広いジャンルで活躍！
C# がキホンから学べる本**



基礎からわかる C#

本書はプログラミングの経験がある人を対象とした、プログラミング言語「C#」の入門書です。C# の概要から基本的な文法、特徴的な機能まで、わかりやすく解説しています。C# 6.0 の新機能についても解説しています。

シーアンドオール研究所
西村誠(著者)
168 ページ 価格: 1,944 円(PDF)

**プログラミング未経験でも
Android アプリを開発！**



**イラストでよくわかる
Android アプリの作り方
Android Studio 対応版**

親しみやすいイラストや、ステップバイステップでの丁寧な解説が基本コンセプト。開発環境「Android Studio」に対応し、Android のプログラムを作りながら、自然に Java というプログラム言語の知識が身につきます。

インプレス
羽山博・めじろまち(著者)
価格: 2,138 円(PDF)

**JavaScript を網羅的に
取り上げた骨太の 1 冊**



JavaScript 逆引きハンドブック

JavaScript の逆引きリファレンスの決定版。JavaScript の機能を網羅的に取り上げていて、骨太の 1 冊になっています。JavaScript の基本的な処理や便利な Tips はもちろん、HTML5 の API についても数多く掲載しています。

シーアンドオール研究所
古旗一浩(著者)
993 ページ 価格: 3,694 円(PDF)

**初めてのウェブ開発も安心
Ruby の文法を基礎から解説**



**改訂 3 版
基礎 Ruby on Rails**

Ruby の文法やオブジェクト指向の考え方を初歩から解説。アプリケーションのモックアップ作り、データベースを導入し、ログイン・ログアウト機能を加え、最終的にはメンバーや記事の管理ページまでできあがります。

インプレス
黒田努・佐藤和人(著者)
536 ページ 価格: 3,240 円(PDF)

**初歩から順に理解できる
PHP とデータベース**

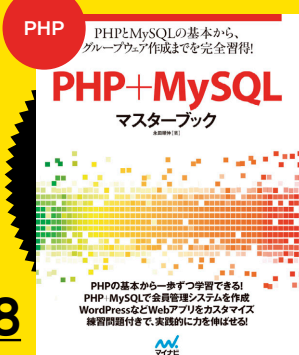


いちばんやさしい PHP の教本

PHP とデータベースの基本を順番に学んで、実践的なプログラムを完成させていく PHP の入門書です。大事なポイントや勘違いしやすいポイントは講師がフォロー。セミナーを受けている感覚で読み進められます。

インプレス
柏岡秀男・池田友子(著者)
240 ページ 価格: 1,836 円(PDF)

**PHP5.4 の基本から
MySQL との連携まで！**



PHP+MySQL マスターブック

この一冊で PHP と MySQL の基本と Web アプリケーションの構築法について学習できる実践的なプログラミング入門です。現場必須のプログラム構築法、API の活用法から、セキュリティ技術まで詳しく解説します。

マイナビ出版
永田順伸(著者)
384 ページ 価格: 2,916 円(PDF)

はじめてプログラミングに
触れる前に読んでおこう



9

プログラミングの基本から
手取り足取りじっくり解説



10

必ずアルゴリズムの意味が
わかるようになる入門書!



11

プログラミングの世界へようこそ

全くの初心者がプログラミングを勉強したいとき、さまざまな疑問が湧いてきます。「どの言語を覚えればいいのか?」「文系でも大丈夫?」本書はプログラミングに触れる前に知っておきたい基本をイラスト付きで解説します。

マイナビ出版
尾川一行・中川聡(著者)
192 ページ 価格: 1,933 円(PDF)

目指せプログラマー!
プログラミング超入門

本書は Windows 開発の標準ツールとも言える「Visual Studio」を使い、C# というプログラミング言語でプログラミングの基本を学びます。最終的には、ちょっとしたアクションゲームが作れるくらいになるのが目標です。

マイナビ出版
掌田津耶乃(著者)
312 ページ 価格: 2,074 円(PDF)

楽しく学ぶ
アルゴリズムとプログラミングの図鑑

図解とイラストを豊富に使ったアルゴリズムの入門書。アルゴリズムとは「問題を解決するための考え方」です。それが分かってくれば、8 種類のプログラミング言語を使ったサンプルプログラムを実際に試しましょう。

マイナビ出版 森巧尚(著者)、まつむらまきお(イラスト)
300 ページ 価格: 2,689 円(PDF)

プログラミングの初心者が
Python 3 を学ぶのに最適



12

Go 言語の基礎から応用まで
ポイントがよくわかる



13

R 言語の機能を
目的から見つけ出せる!



14

基礎 Python 基礎シリーズ

プログラミングの初心者を対象にした Python 3 の入門書です。変数の取り扱いから、リスト、タプルといった Python 固有のデータの操作、制御構造や関数などについて、初心者でも基礎から学習できるように説明しました。

インプレス
大津真(著者)
312 ページ 価格: 2,894 円(EPUB)

改訂 2 版
基礎からわかる Go 言語

Google が開発したプログラミング言語「Go」の基礎から応用までをわかりやすく解説した 1 冊です。最新の Go 1.4 のバージョンに対応して改訂しました。Linux、Mac OS X、Windows の各環境に対応しています。

シーアンドアール研究所 古川昇(著者)
240 ページ 価格: 2,138 円(EPUB)

改訂 3 版
R 言語逆引きハンドブック

本書では、最新バージョンの R 3.3.0 に対応し、R 言語の機能を目的から探すことができます。統計が注目を集めるなか、R を利用するユーザーも増えています。初心者でも使えるように、導入から丁寧に解説しています。

シーアンドアール研究所
石田基広(著者)
800 ページ 価格: 4,860 円(PDF)