

楽しく学ぶ



アルゴリズムと プログラミングの 図鑑

見て、読んで、試してわかる、実感できる入門書です！

「アルゴリズムの意味」を
イラストや図でやさしく解説



「プログラミング言語」の
サンプルを体験して納得

取り上げる言語

JavaScript・PHP・C・Java・Swift・Python・BASIC・Scratch



第 1 章 アルゴリズムってなに？

アルゴリズムってなんでしょう。プログラムとアルゴリズムの違いってなんでしょう。アルゴリズムと人間との関係を知りましょう。

第 2 章 いろいろなプログラミング言語

世の中にはいろいろなプログラミング言語があります。それぞれの言語の特徴、用途、動作環境、誕生時期、開発者、名前の由来などについて解説します。

第 3 章 データ構造とアルゴリズムの基本

アルゴリズムの基本について知りましょう。アルゴリズムの組み立て方や、書き方、データの使い方について解説します。

第 4 章 簡単なアルゴリズム

基本となるアルゴリズムを知りましょう。いろいろなプログラミング言語で、どのように記述するかも紹介します。

第 5 章 サーチアルゴリズム

サーチとは、「大量のデータの中から、目的のデータを見つけること」です。いろいろなサーチアルゴリズムを知りましょう。プログラミング言語での具体的な書き方も紹介します。

第 6 章 ソートアルゴリズム

ソートとは、大量のデータを「ある順番で整列させて、わかりやすくすること」です。いろいろなソートアルゴリズムを知りましょう。

楽しく学ぶ

アルゴリズムと プログラミングの 図鑑

森巧尚 [作]
まつむらまきお [絵]

本書のサポートサイト

本書のサンプルファイル、補足情報、訂正情報などを掲載します。適宜ご参照ください。

<http://book.mynavi.jp/supportsite/detail/9784839960216.html>

- ▼ 本書は2016年10月段階での情報に基づいて執筆されています。本書に登場する製品やソフトウェア、サービスのバージョン、画面、機能、URL、製品のスペックなどの情報は、すべてその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が、著者や出版社のいずれも、本書の内容に関してなん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いっさいの責任を負いません。あらかじめご了承ください。
- ▼ 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®マークは省略させていただいております。

はじめに

この本は、「必ずアルゴリズムの意味がわかるようになる、アルゴリズムの入門書」です。



みなさんの中で「アルゴリズム」と聞くと「なんだか難しくて、めんどくさい」と感じる人は多いのではないのでしょうか。もしも、そんな風に苦手に思っているとしたら、それはもったいないことです。



「アルゴリズム」とはズバリ、『問題を解決するための考え方』です。それは、コンピュータの中だけの特別なことではありません。みなさんの日常生活の中でも、よく行われていることです。問題をちゃんと解決するときは、「問題を整理して、何をどんな風にすれば、欲しい結果が得られるのか」に注目することが重要です。これが「アルゴリズム」なのです。



この本では、コンピュータのアルゴリズムを通して「問題を解決する考え方」を体験していきます。

- ・このアルゴリズムは、どんな考え方で問題を解こうとしているのか？

- ・この手順は、何をしようとしているのか？

など、「アルゴリズムの意味」に注目して、イラストや図を使って解説していきます。



難しそうに思えるアルゴリズムでも「意味」がわかれば、「なあんだ、こういうことだったのか」と気持ちよく納得できることでしょう。

苦手だと思っていた「アルゴリズム」を、ぜひ「問題解決をするときの道具」として、あなたの武器に加えましょう。



とはいえ、頭の中で「意味が理解できただけ」では、手ごたえがないので、すぐ忘れてしまいそうな気がします。そこでこの本では、実際に8種類のプログラミング言語を使ってプログラムを用意しました。あなたの使っているプログラミ

ング言語を使って、実行してみましょう。「理解した意味の通りにアルゴリズムが動くこと」を試して、実感してください。



この「8種類」というのは「よく目にするプログラミング言語だろう」と思って選んだ言語です。この言語こそが重要だとか、知っていると偉い言語だなどというわけではありません。もしもあなたが使っている言語がなかったらごめんなさいね。



Webサイト開発で使われる言語として、JavaScriptと、PHP。ハードウェアの開発で使われる言語として、C。スマホアプリの開発で使われる言語として、Swiftと、Java。機械学習やディープラーニングで使われる言語として、Python。そして、最近流行ってきている子供向けプログラミング教育で使われる言語として、Scratchと、BASIC、の8種類のプログラミング言語です。



こうやって「同じアルゴリズムのプログラム」を、8種類のプログラミング言語で並べていると面白いことがわかってきます。

「多くのプログラミング言語には共通点があるのだなあ」とか、「この言語には、こんな特長があるんだ」と気づく部分もあります。

もしみなさんが他のプログラミング言語を知りたいと思っているなら、この言語間の「共通点」や「違い」が見えてくると、言語を理解するハードルがぐっと下がりますよ。



さあ、「アルゴリズムの意味」と「プログラミング言語」について学んでいきましょう。



最後に、この本を書くにあたってご協力くださった福島 伸之さん、吉田研一さん、寺園聖文さん、そしてイラストを描いてくださったまつむらまきおさん、ありがとうございました。

2016年10月

森 巧尚

CONTENTS

第1章

アルゴリズムってなに? ————— 001

1.1 アルゴリズムってなに? ————— 002

アルゴリズムとは? ————— 002

1.2 プログラムとアルゴリズムの違い ————— 004

コンピュータを使う目的とは? ————— 004

プログラムとは? ————— 005

アルゴリズムとは? ————— 006

1.3 アルゴリズムを考えよう ————— 008

問題は、整理して考える ————— 008

ライブラリは「先人の知恵」 ————— 010

アルゴリズムを自分で作ろう! ————— 011

1.4 いろいろなアルゴリズム ————— 013

第2章

いろいろなプログラミング言語 — 015

2.1 いろいろなプログラミング言語 ————— 016

いろいろなプログラミング言語 ————— 016

プログラミング言語実行環境マップ — 018

2.2 JavaScript ————— 020

JavaScript 言語とは? ————— 020

実行環境 ————— 021

開発環境例 ————— 022

文法の特徴 ————— 022

2.3 PHP ————— 024

PHP 言語とは? ————— 024

実行環境 ————— 025

開発環境例 ————— 026

文法の特徴 ————— 026

2.4 C ————— 028

C 言語とは? ————— 028

実行環境 ————— 029

開発環境例 ————— 030

文法の特徴 ————— 030

2.5 Java ————— 032

Java 言語とは? ————— 032

実行環境 ————— 033

開発環境例 ————— 034

文法の特徴 ————— 034

2.6 Swift ————— 036

Swift 言語とは? ————— 036

実行環境 ————— 037

開発環境例 ————— 038

文法の特徴 ————— 038

2.7 Python ————— 040

Python 言語とは? ————— 040

実行環境 ————— 041

開発環境例 ————— 042

文法の特徴 ————— 042

2.8 BASIC ————— 044

BASIC 言語とは? ————— 044

実行環境 ————— 045

開発環境例 ————— 046

文法の特徴 ————— 047

2.9 Scratch ————— 048

Scratch 言語とは? ————— 048

実行環境 ————— 049

開発環境例 ————— 050

文法の特徴 ————— 050

第3章

データ構造と アルゴリズムの基本 ————— 053

3.1 データ構造 ————— 054

コンピュータが扱えるデータ	054
文字	054
画像	055
動画	055
音声	055
家電やロボット	055
データ構造	056
変数	056
配列	057
リスト	057
キュー：First In First Out	058
スタック：Last In First Out	059
木（ツリー）構造	059

3.2 フローチャート ————— 060

アルゴリズムを図式で書く方法	060
フローチャートで使う記号	060
端子	060
処理	061
選択（分岐）	061
反復	061
定義済み処理	061
流れ線	061

3.3 アルゴリズムの基本 ————— 062

3つの基本構造	062
順次構造：	
上から順番に、実行する	062
選択構造（条件分岐）：	
もしも～なら、実行する	063
反復構造（ループ）：	
繰り返し、実行する	063

3.4 アルゴリズムからプログラムへ ————— 064

イメージ	064
フローチャート	065
プログラム	065
JavaScript	066
PHP	066
C	067
Java	068
Swift	069
Python	070
BASIC	071
Scratch	071

第4章 簡単なアルゴリズム ————— 073

4.1 簡単なアルゴリズム ————— 074

4.2 合計値 ————— 075

アルゴリズムのイメージと手順 ——— 075

フローチャート ————— 077

プログラム ————— 078

JavaScript ————— 079

PHP ————— 080

C ————— 081

Java ————— 082

Swift ————— 083

Python ————— 084

BASIC ————— 085

Scratch ————— 086

4.3 平均値 ————— 088

アルゴリズムのイメージと手順 ——— 088

フローチャート ————— 089

プログラム ————— 089

JavaScript ————— 090

PHP ————— 091

C ————— 092

Java ————— 093

Swift ————— 094

Python ————— 095

BASIC ————— 096

Scratch ————— 097

4.4 最大値、最小値 ————— 099

アルゴリズムのイメージと手順 ——— 099

フローチャート ————— 101

プログラム ————— 101

JavaScript ————— 102

PHP ————— 103

C ————— 104

Java ————— 105

Swift ————— 106

Python ————— 107

BASIC ————— 108

Scratch ————— 108

4.5 データの交換 ————— 110

アルゴリズムのイメージと手順 ——— 110

フローチャート ————— 112

プログラム ————— 112

JavaScript ————— 113

PHP ————— 114

C ————— 115

Java ————— 116

Swift ————— 117

Python ————— 118

BASIC ————— 119

Scratch ————— 120

第5章

サーチアルゴリズム ————— 121

5.1 サーチ（探索）アルゴリズムとは ——— 122

サーチアルゴリズムとは ————— 122

5.2 リニアサーチ（線形探索法） ————— 123

ひとつひとつ調べていく

探索法 ————— 123

アルゴリズムのイメージ ————— 123

サーチアルゴリズムの工夫 ——— 124

具体的な手順 ————— 125

フローチャート ————— 126

このアルゴリズムの特徴（まとめ） — 127

プログラム ————— 127

JavaScript ————— 128

PHP ————— 129

C ————— 130

Java ————— 131

Swift ————— 132

Python ————— 133

BASIC ————— 134

Scratch ————— 135

5.3 バイナリサーチ（二分探索法） ——— 136

範囲を半分に絞りながら

探していく探索法 ————— 136

アルゴリズムのイメージ ——— 137

具体的な手順 ————— 139

フローチャート ————— 142

このアルゴリズムの特徴（まとめ） — 143

プログラム ————— 143

JavaScript ————— 144

PHP ————— 146

C ————— 148

Java ————— 150

Swift ————— 152

Python ————— 154

BASIC ————— 156

Scratch ————— 157

第6章

ソートアルゴリズム ————— 159

6.1 ソート（整列）アルゴリズムとは ————— 160

6.2 バブルソート（単純交換法） ————— 161

泡のように

値を浮かび上がらせるソート ————— 161

アルゴリズムのイメージ ————— 162

具体的な手順 ————— 163

フローチャート ————— 165

このアルゴリズムの特徴（まとめ） ————— 167

プログラム ————— 167

JavaScript ————— 168

PHP ————— 169

C ————— 170

Java ————— 172

Swift ————— 174

Python ————— 175

BASIC ————— 176

Scratch ————— 177

6.3 選択ソート（単純選択法） ————— 179

最小値を探して、

先頭から順番に並べていくソート ————— 179

アルゴリズムのイメージ ————— 179

具体的な手順 ————— 180

フローチャート ————— 183

このアルゴリズムの特徴（まとめ） ————— 185

プログラム ————— 185

JavaScript ————— 186

PHP ————— 187

C ————— 189

Java ————— 191

Swift ————— 193

Python ————— 194

BASIC ————— 196

Scratch ————— 197

6.4 挿入ソート（単純挿入法） ————— 199

データを抜き出して、

正しい位置に挿入していくソート ————— 199

アルゴリズムのイメージ ————— 200

具体的な手順 ————— 201

フローチャート ————— 203

このアルゴリズムの特徴（まとめ） ————— 205

プログラム ————— 205

JavaScript ————— 206

PHP ————— 207

C ————— 209

Java ————— 211

Swift ————— 213

Python ————— 215

BASIC ————— 217

Scratch ————— 218

6.5 シェルソート ————— 220

大雑把にソートを行ってから、

だんだん精度を高めていくソート ————— 220

アルゴリズムのイメージ ————— 220

具体的な手順 ————— 223

フローチャート ————— 226

このアルゴリズムの特徴（まとめ） ————— 228

プログラム ————— 228

JavaScript ————— 229

PHP ————— 231

C ————— 233

Java ————— 235

Swift ————— 237

Python ————— 239

BASIC ————— 241

Scratch ————— 242

6.6 クイックソート ————— 244

大小2つのグループに分割し、

分割をくり返していくソート ————— 244

アルゴリズムのイメージ ————— 244

具体的な手順 ————— 249

フローチャート ————— 253

このアルゴリズムの特徴（まとめ） ——— 255

プログラム ————— 255

 JavaScript ————— 256

 PHP ————— 258

 C ————— 260

 Java ————— 263

 Swift ————— 266

 Python ————— 268

付録

1 オブジェクトをソートする ————— 272

2 シャッフルするアルゴリズム ————— 274

3 O記法 ————— 278

4 アルゴリズムの組み合わせ一覧 ————— 280

言語別サンプル一覧 ————— 282

INDEX ————— 284

第 1 章

アルゴリズムって なに？

アルゴリズムってなんでしょう。

プログラムとアルゴリズムの違いってなんでしょう。

アルゴリズムと人間との関係を知りましょう。

- 1.1 アルゴリズムってなに？
- 1.2 プログラムとアルゴリズムの違い
- 1.3 アルゴリズムを考えよう
- 1.4 いろいろなアルゴリズム

1.1

アルゴリズムってなに？

アルゴリズムとは？

「プログラミング言語」はだいたい理解した、「開発環境の使い方」もだいたいわかった、なのにいざ自分でオリジナルのプログラムを作ろうとするとつまづいてしまった、という経験のある人は多いのではないのでしょうか。

その原因は「アルゴリズム」が見えていないことかもしれません。



「アルゴリズム」というと、抵抗を感じる人もいるのではないのでしょうか。

「複雑なコンピュータのしくみ」とか「数式を組み合わせで作った、難解なもの」といったイメージがあるかもしれませんね。

でも、実はアルゴリズムはもっとみなさんの身近なものなのです。



「アルゴリズム」とは、ズバリ「問題の解き方」です。



みなさんが、何かの問題を解決したいとき、「どんな考え方で解決するか?」「どんな手順で具体的に実行するか?」などと考えますよね。

これです。この、問題を解決するための「考え方と手順」が「アルゴリズム」なのです。

アルゴリズム = 問題の解き方

ですから、「アルゴリズム（問題の解き方）」がわからなければ、いざ自分でプログラムを作ろうと思っても、プログラムはうまく作れないのです。



技術書ではよく、アルゴリズムは「数式」で書かれています。

しかし、アルゴリズムは「日本語」でも書けます。「書いた通りに実行して問題が解決できる」のであれば、それはれっきとしたアルゴリズムです。

また書き方は、数式や文章だけではなくありません。「フローチャート」という図式で描く書き方もあります。



アルゴリズムは、自分に合った方法で形にすればいいのです。



では「アルゴリズム」とはどんなものなのか、見ていくことにしましょう。

JavaScript

PHP

C

Java

Swift

Python

BASIC

Scratch

1.2

プログラムとアルゴリズムの違い

コンピュータを使う目的とは？

みなさんの周りには、いろいろなコンピュータがありますね。パソコンやスマホやゲーム機、家電の中に入っていたりもします。

何のための使うのかは、人それぞれさまざまです。「データを検索したい」とか「データを集計したい」といったデータ処理だったり、「好きな曲を聴きたい」とか「ゲームで遊びたい」とか「3Dプリンタで立体物を作りたい」といったコンピュータのコントロールだったり。



ですが、もう少し大きな視点で見ると、共通点が見えてきます。それは、すべて「問題を解決するために」動かしているということです。



「問題」というと大げさなので「人の欲求」と言い換えた方がわかりやすいかも知れません。

「データを検索したい」「集計したい」「曲を聴きたい」「ゲームで遊びたい」「3Dプリンタで立体物を作りたい」

これらはみんな人の欲求です。いろいろなコンピュータは、すべて「人の欲求を満たすために動いている」のです。



コンピュータを使う目的 = 問題を解決すること (人の欲求を満たすこと)

プログラムとは？

問題を解決するために、コンピュータを動かします。このときコンピュータは「プログラム」で動きます。「いろんな種類のプログラム」があるから、コンピュータは「いろいろなこと」ができるのです。



この「プログラム」とは、いったいなんでしょう？

それは、「プログラム (program)」の語源を知ると少しわかります。

「program」とは、「pro (前もって)」という言葉と、「gram (書いたもの)」という2つの言葉でできた言葉です。つまり、プログラムとは「前もって、書いたもの」という意味なのです。

プログラム = 前もって、書いたもの

日常生活でも「プログラム」が使われることがあります。「コンサートのプログラム」とか「運動会のプログラム」とか「ダイエットのプログラム」などいろいろありますね。

これらの「プログラム」にも同じ意味があります。それは「これからする予定が、前もって書かれている」ということです。

コンピュータのプログラムも同じです。「これからコンピュータが行うことが、前もって書かれたもの」です。決して、理解不能な呪文が並んだものなどではないのです。

コンピュータのプログラム = コンピュータが行うことを、前もって書いたもの



Java
Script

PHP

C

Java

Swift

Python

BASIC

Scratch

アルゴリズムとは？

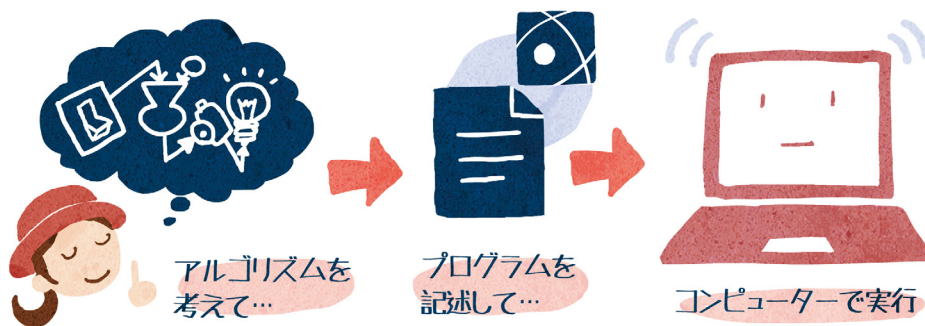
コンピュータは、プログラムに書かれた通りの命令を実行して、問題を解決します。では「書かれた通りに命令を実行するだけ」で、なぜ問題を解決できるのでしょうか？

もしも「ただ適当に命令が並んでいるだけのプログラム」だったら、問題を解決することはできません。ちゃんと問題が解けるように書かれているから、問題が解けるのです。



つまり、プログラムで重要なのは「問題の解き方」です。
そして、この「問題の解き方」こそが「アルゴリズム」なのです。

アルゴリズム = 問題の解き方



しかし、初心者の中にはアルゴリズムというものを意識されることは少ないようです。ときには、「プログラムはわかるけど、アルゴリズムって必要なの？」などと思われることもあるようです。

例えば、初心者向けのプログラミングの本などを見ると、アルゴリズムなんて考えなくても、プログラムが作れてしまいます。ネット上でプログラムを見つけたら、コピー＆ペーストしても動くプログラムは作れます。ですがそれは、用意されたアルゴリズムを書き写している状態であって、「問題の解き方」を理解しているわけではありません。



コピー＆ペーストしたプログラムを改造しようとして一部分を書き換えたら、思ったように動かなくなってしまうなんて経験はありませんか？「アルゴリズム（問題の解き方）」がわかっていない状態では、プログラムを作ろうとしても、うまくいかないことがあるのは当然です。



アルゴリズムとは問題を解く「考え」、プログラムとはそれを前もって書き表す「言葉」です。「考え」と「言葉」の両方が正しく揃わないと、コンピュータは思った通りに動いてくれないのです。

プログラムを作ること = アルゴリズムを考える + プログラミング言語で記述する

プログラミングに関する用語

プログラムに関するいろいろな言葉があります。ここで確認しておきましょう。

プログラム	=	コンピュータが行うことを前もって書いたもの
アルゴリズム	=	問題の解き方
プログラミング	=	プログラムを作ること（アルゴリズムを考えて、プログラムを記述すること）
プログラマー	=	プログラムを作る人（アルゴリズムを考えて、プログラムを記述する人）
プログラミング言語	=	プログラムを作るのに使う言語
ソースコード	=	人間が記述した状態のプログラム
バイナリコード	=	人間が記述したプログラムをコンピュータが実行できる状態にしたもの
コーディング	=	プログラムを記述すること
コーダー	=	プログラムを記述する人
コンパイラ	=	ソースコードをバイナリコードに変換するしくみ
インタプリタ	=	ソースコードをバイナリコードに変換しながら実行するしくみ

JavaScript

PHP

C

Java

Swift

Python

BASIC

Scratch

1.3

アルゴリズムを考えよう

それでは、「アルゴリズム」は、どのように考えればいいのでしょうか？

コンピュータの設計図なので、コンピュータみたいな頭脳を持った人間でないと作れないのでしょうか。



そうではありません。「アルゴリズムを考える」ということは、「問題の解き方を考える」ということです。

普通の人が日常生活の中で「問題解決をするのと同じこと」をすればいいのです。

問題は、整理して考える

「問題を解決する」にも、ぼんやりと考えていては考えがまとまりません。まずは、わかっている部分をハッキリさせて、「整理していくこと」が重要です。

アルゴリズムを考えるとときも同じです。いきなりアルゴリズムを考えるのではなく、まずわかっている部分をハッキリさせて整理していきましょう。



アルゴリズムを考えるとときに重要なのは、

1) 目的

2) 現状

3) 結果

の3つをハッキリさせることです。

1) 目的

まずは「何が目的なのか」をハッキリさせることが重要です。

当たり前のことのように思えますが、なんとなくハッキリさせないままプログラミングを始めてしまうことはありがちです。

ここをハッキリさせずに作ってしまい、目的がブレてしまうと、「使えないプログラム」になってしまいます。

2) 現状

「現状がどうなっているか」を把握することも大事です。

どういう状態かわからないままでは、解決のしようがありません。

どんなデータがあるのか、どういう前提条件があって解決しようとするものなのか、をハッキリさせます。

コンピュータではこれを、「インプット」といいます。

3) 結果

「どんな結果が欲しいのか」をハッキリさせます。

値が一つ出力されればいいのか、たくさんのデータとして出力するのか、絵やアニメーションとして表現されるべきなのか、結果といってもいろいろな形態があります。

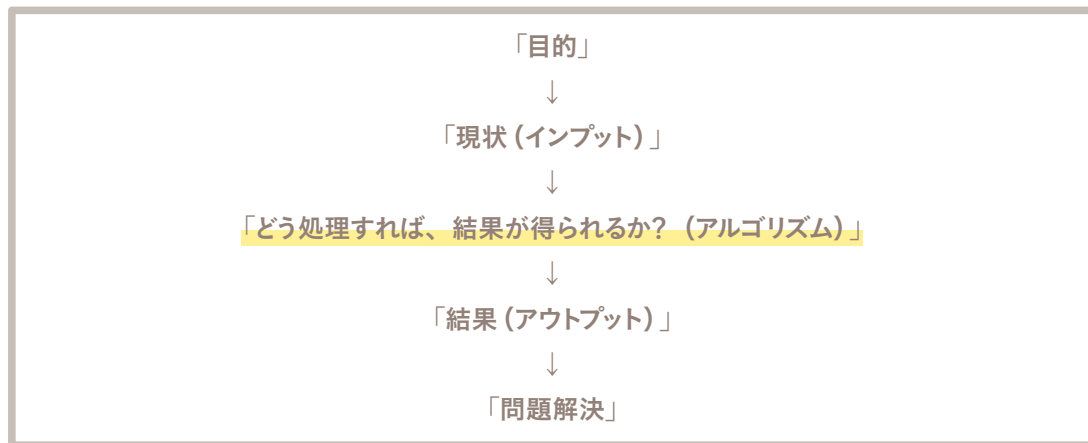
コンピュータではこれを、「アウトプット」といいます。



これら「目的」「現状」「結果」をハッキリさせることで、考えなければならない部分がしぼられます。



「現状（インプット）」から、何をどうすれば、「求める結果（アウトプット）」になるのか、という部分に注目して考えることができますようになります。



ライブラリは「先人の知恵」

問題を整理できたら、いよいよ「アルゴリズム（問題の解き方）」を考えます。

ですが、最初から自分で「アルゴリズム」を考えるのは、簡単なことではありません。難しいときは、他の人が作った「アルゴリズム」を利用しましょう。



人間が生きているうちに会う問題には、似たようなものがたくさんあります。そして、すでに昔の人たちも同じような問題を解決して乗り越えてきました。

その「昔の人たちが作った問題を解決する方法」のことを「先人の知恵」と言います。



囲碁や将棋で定番の問題を解決する方法を「定石」といったり、おいしい料理を作りたいという問題を解決してくれる方法を「レシピ」といったりしますが、これらもみな「先人の知恵」です。



コンピュータのアルゴリズムでも、「先人の知恵」を借りる方法があります。それが「ライブラリ」です。これまで多くの先輩たちが、プログラムを作る上で同じ問題に出会い、いろいろな方法で問題解決が行われました。その中で、特に効率がよくて使いやすい「アルゴリズム」をまとめたものが「ライブラリ」なのです。

ライブラリを使うときは、ぜひ作ってくれた先人たちに感謝しましょう。



アルゴリズムを自分で作ろう!

しかし、ライブラリだけで、すべての問題を解決できるわけではありません。今までになかった「新しい問題」や、個別案件などの「細かい問題」はライブラリだけでは対応できません。自分でアルゴリズムを考えましょう。

このようなときは、「いろいろなアルゴリズムを理解していること」が重要になります。

複雑なアルゴリズムも、実は多くの場合「いろいろなアルゴリズムの組み合わせ」でできています。初めて見ると、どうしてこんな難しいことが考えられるのだらうと感じますが、整理して見ていくと、アルゴリズムはいくつかのアルゴリズムを組み合わせでできていることがわかってきます。

アルゴリズムは、アルゴリズムの組み合わせで作ることができる

例えば、ゲームの世界では「アイテムとアイテムを合成する」と新しいアイテムを作ることができます。「強化されたアイテム」ができたり「新機能のアイテム」ができたりします。



同じように、アルゴリズムも組み合わせると、「強化されたアルゴリズム」ができたり「新機能のアルゴリズム」を作ることができるのです。

「アルゴリズムを組み合わせ、新しいアルゴリズムを作る」という知恵はうまく利用しましょう。

どのように組み合わせアルゴリズムができるかについては、この本の後半で説明していきます。

また、必ずしも組み合わせる必要はありません。ある「アルゴリズムのしくみ」をヒントにして、別の新しいアルゴリズムを考え出せる可能性もあります。



ぜひ、「いろいろなアルゴリズムのしくみ」を理解して、「アルゴリズムの本質」を身につけましょう。

1.4

いろいろなアルゴリズム

それでは、どんなアルゴリズムがあるのか、よく使われるアルゴリズムのダイジェストを見ていきましょう。

ソート（整列）アルゴリズム



「たくさんのデータを順番に並べたい」という問題を解決するアルゴリズムです。

たくさんのデータを扱うときに、まず必要になる基本的なアルゴリズムです。例えば、データベースのデータを指定した順番で表示させたり、Excelでデータを整列させたり、ファイルがたくさんあるときに日付順やサイズ順で並べ直したり、いろいろなところで使われています。

サーチ（探索）アルゴリズム



「たくさんのデータの中から、目的のデータを見つけたい」という問題を解決するアルゴリズムです。

これもたくさんのデータを扱うときに、必要になるアルゴリズムです。データを検索するといえばGoogleやYahoo!などの検索サイトが有名ですが、ここでもサーチアルゴリズムが使われています。

Java
Script

PHP

C

Java

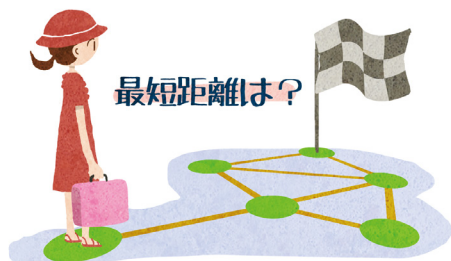
Swift

Python

BASIC

Scratch

経路探索アルゴリズム



「出発地点から到着地点へ、どの道が最短なのか知りたい」という問題を解決するアルゴリズムです。これは「カーナビ」や「乗換案内」で利用されています。所要時間が短い順、乗換回数が少ない順、料金が安い順、など重み付けによって求める最短路を変えることができます。

データ圧縮アルゴリズム



「大きなデータは通信や保存が大変なので、一時的に小さくしておきたい」という問題を解決するアルゴリズムです。画像を圧縮させるJPEG圧縮や、ファイルサイズを小さくするためのZIP圧縮などがあります。圧縮したデータはそのままでは使えないので、解凍してから使います。

暗号化アルゴリズム

秘密にしておきたい...



「データを他人に盗まれたり、変更されては困る」という問題を解決するアルゴリズムです。他人に読まれたり変更されたりできないような形式に変換します。個人情報や企業情報のセキュリティに重要なアルゴリズムです。

レイトレーシングアルゴリズム



「写真のようにリアルな3Dコンピュータグラフィックスを描きたい」という問題を解決するアルゴリズムです。光が物体に当たってどのように反射したり、通過したり、屈折したりするかなどを考慮して描画を行うので、リアルで美しい画像を作れます。

第 2 章

いろいろな プログラミング言語

世の中にはいろいろなプログラミング言語があります。
それぞれの言語の特徴、用途、動作環境、誕生時期、開発者、
名前の由来などについて解説します。

2.1 いろいろなプログラミング言語

2.2 JavaScript

2.3 PHP

2.4 C

2.5 Java

2.6 Swift

2.7 Python

2.8 BASIC

2.9 Scratch

2.1

いろいろなプログラミング言語

いろいろなプログラミング言語

それでは少し「プログラミング言語の種類」について目を向けてみましょう。

(プログラミング言語の種類のことは知っているよ、という人は第3章へ進んでもいいですよ。)

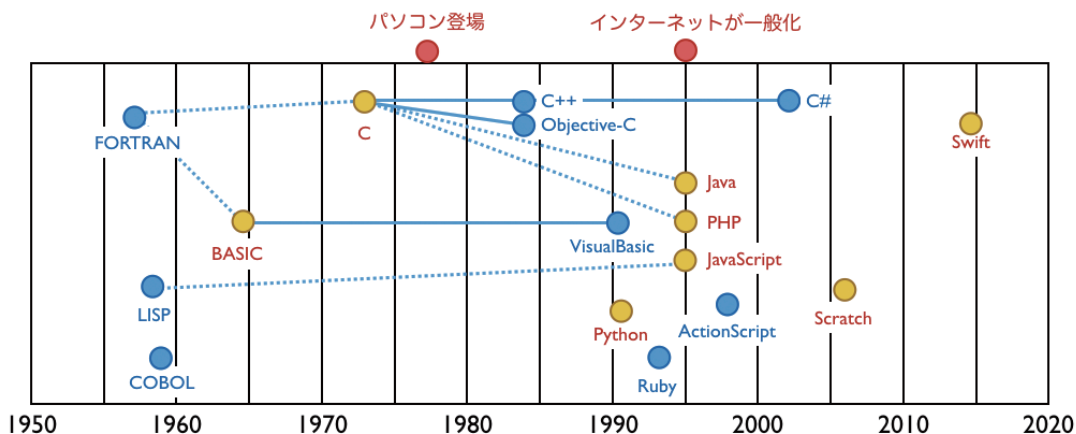


世の中にはプログラミング言語が、いろいろ存在しています。



いろいろあると勉強するのが大変なのでひとつに統一して欲しいように思えますが、どうしていろいろな言語があるのでしょうか。

それは、「使う目的」が違うからです。そして言語によって「得意な分野」に違いがあるからです。



事務処理の得意なプログラミング言語で面白いゲームを作るのは難しいですし、初心者向けのプログラミング言語で高度な科学計算を行うのは無理があります。

いろいろなプログラミング言語があるのは、それぞれの目的のために専門化していったからなのです。

プログラミング言語年表

いろいろなプログラミング言語の誕生を、時代を追って見ていきましょう。

新しい目的ができることでプログラミング言語が生まれたり、逆にプログラミング言語が生まれることで新たな世界が生まれてきました。



初期のコンピュータは大型コンピュータで、研究所や企業で使われていました。ですので、科学計算向けにFORTRAN、人工知能向けにLISP、事務処理向けにCOBOLという言語がありました。その後、初心者向けにBASIC、ハードウェアやOS向けにCができたことで、個人向けのパーソナルコンピュータが登場しました。



パソコンが普及して個人が使い始めるようになると、さらに使いやすい言語が求められるようになりました。Mac向けにObjective-Cが生まれ、BASICはWindows向けにVisualBasicへ変わりました。この他にも、数多くのプログラミング言語が生まれ、だんだん専門的で複雑化していきました。そこで、シンプルな言語でプログラミングができるようにとPythonができました。Pythonは最近注目を集めていますが、こんな昔からあったんですね。



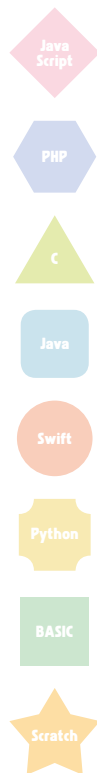
インターネットが一般化し始めると、インターネットに適した言語が登場してきました。サーバ向けにJavaやPHP、Ruby、ブラウザ向けにJavaScriptやActionScriptなどができました。その後、CはC++へと進化した後さらにC#へと進化しました。C#は、現在スマートフォン開発で多く使われているUnityで使用されている言語です。



子供向けのプログラミング教育にも目が向けられるようになり、Scratchができました。スマートフォンが普及しはじめると、iOSアプリを開発しやすくするためにSwiftができました。



このようにプログラミング言語は、新しい目的ができるたびに、それに適した言語が生まれてきたのです。



プログラミング言語実行環境マップ

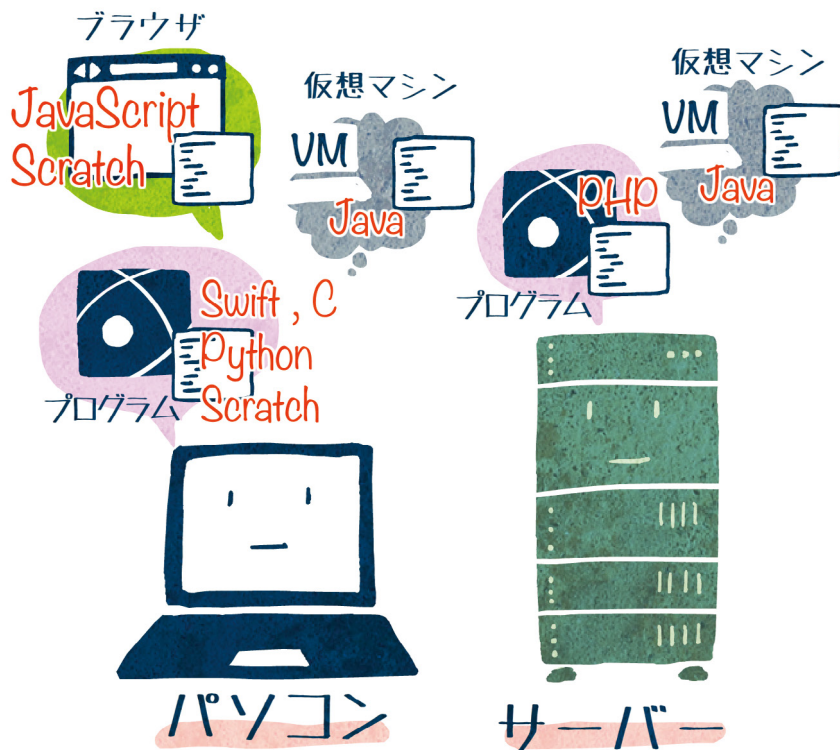
「使う目的」によってプログラミング言語は変わりますが、それは「実行する場所の違い」にも影響しています。プログラムはそれぞれ「最も適した場所」で実行されるのです。

JavaScript はブラウザ向けの言語なので、パソコンのブラウザ上で動きます。

PHP はサーバ向けの言語なので、サーバ上で動きます。

C はハードウェアやOS向けの言語なので、組み込み機器やパソコン上で動きます。

Java は仮想マシンのあるいろいろなコンピュータで動くようにできた言語なので、組み込み機器やパソコンや、サーバや、Androidの上で動きます。





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。

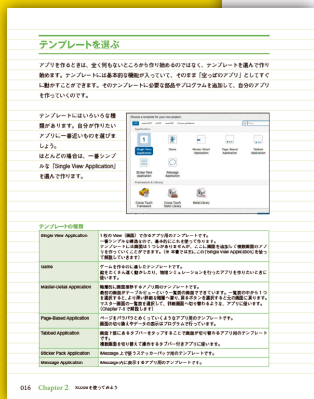
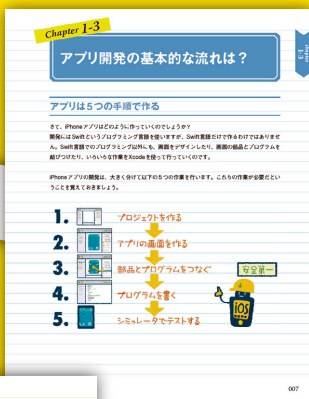
豊富なイラストで「なぜ?」を解消! Javaの第一歩を踏み出そう

『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。

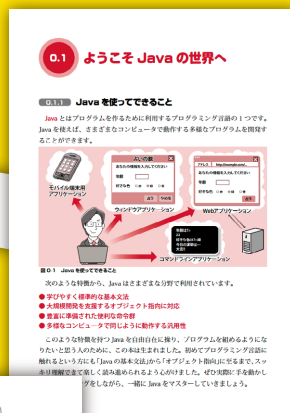
「プログラミング」

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

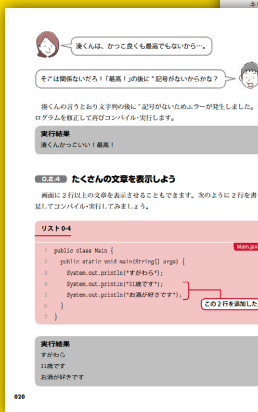
300ものイラストで楽しく・詳しく・スッキリとマスター!



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



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)

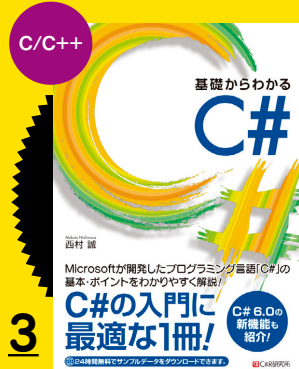
スッキリわかる Java 入門 第2版

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**



3

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！**



4

**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**



5

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**



6

**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**



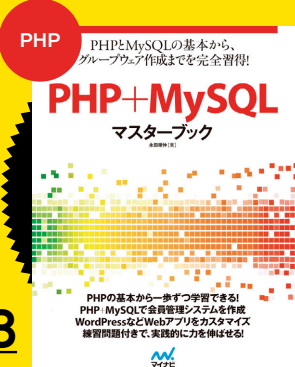
7

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！**



8

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 西川昇(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)