基礎からきっちり覚える機械語入門

渡辺徹　[著]

Machine Language for Beginners

[初心者のための機械語読本]
コンピュータが直接理解できる唯一のコトバ
“機械語”の基礎の基礎から解説。

マイナビ
Mynavi Advanced Library

基礎からきっちり覚える機械語入門

渡辺徹　[著]

マイナビ

Mynavi Advanced Library

●本書のサポートサイト

http://book.mynavi.jp/support/pc/5067/

●本書は

「機械のコトバ」（2005年5月刊行）

を元にした電子版/オンデマンド版です。

●電子版/オンデマンド版制作にあたり、2014年1月現在での補足情報を記事として追加していますが、基本的に本書中の情報は書籍執筆段階のものです。

●本書に登場するソフトウェアのバージョン、URL、製品のスペックなどの仕様や情報は、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。

・本書中に掲載している画面イメージは、特定の設定に基づいた環境にて再現される一例です。ハードウェアやソフトウェアの環境によっては、必ずしも本書通りの画面にならない場合があります。あらかじめご了承ください。

・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

いきなり私事で恐縮なのですが、筆者は地方の小ソフト会社でゲームの開発に携わっていたりします。パッケージソフトもいくつかあり、小さい会社なのでユーザーサポートなども自分で担当していました。まあ売れた本数が本数だけにそんなに大したことにはならなかったわけですが…というのはともかく、そのときに一番困ったのが、お客様に結構言葉が通じないことでした。

もちろんお客様が外国人だったわけではなく、サポートするためには例えば「ファイル」とか「ドライバ」といった基本的な用語を使わざるを得ないのですが、それがたいてい通じないためです。そこで初心者向けのコンピューター解説書の必要性を常々感じていました。

そのための初心者向けの入門書などはそれこそ星の数ほど出ていますが、そういったものはたいていがOSやアプリケーションの「使い方」の本で、それがなぜそう動くのかといったようなことについては書かれていません。これは作者の手抜きと言うよりは、そういった基本的なことを説明しようとすると実はソフトウェアやOSの仕組みなどについて1から説明することになってしまって、ちょっと簡単には手を出せないからです。

ソフトウェアに関する本もこれまたたくさんありますが、今度は数学の教科書のようになってきて、興味がある人でも基本的な素養がないとまず理解不能です。

本書はそういったコンピューターに興味のある一般の人を対象とした、ソフトウェアの基礎の基礎に関する解説書です。

そのアプローチとして本書では、データの数値化やアルゴリズムの話と共に、本書のタイトルにもなっている「機械のコトバ」、すなわちコンピューターの「機械語」の解説を行います。

ある程度コンピューターを知っている人だと機械語は難しいものだというイメージがあるかもしれません。実際に機械語でプログラムを作成しようとすれば、それは大変な作業になります。しかしコンピューターがどのように動作するかというイメージを掴むためには、実はこの機械語を見てみるのが一番の早道なのです。機械語の動作が理解できれば、その上で動く高級言語やOSなどの動作も非常に分かりやすくなるのです。

もちろん本書を読んだからといって、明日からコンピューターがバリバリ使えるというわけにはいかないでしょう。しかし、今まで「何だかよく分からない箱」だったものが少しでも違ったように見えてきたならば、筆者が本書を書いた甲斐があったというものです。

最後に、本書の原稿を見てくれた算法研究所一同に感謝します。

もくじ

03 アルゴリズム

04 機械のコトバその1：機械語

プロローグ　A君の悲劇

21世紀になってコンピューターは生活の中にますます浸透してきました。パソコン本体の普及率もかなりのものですし、家電はいわゆるマイコン制御のものが当たり前です。仕事をする場合には、少なくとも部署に一台ぐらいは間違いなくパソコンがありますし、買い物をするためにコンビニやスーパーに行けばレジはPOSシステムになっています。なによりも、ほとんどの人が持っている携帯電話は小型のコンピューターそのものです。

このように現在の生活はもうコンピューターと切っても切れない関係になっています。しかしそれにも拘わらず、必ずしもコンピューターのことが一般の人に十分に理解されているとは思えません。多くの人はいまだコンピューターをなんとなく怪しげな装置だと見ているのではないでしょうか？

恐るべき悲劇に見舞われたA君も、そんな中の一人でした…。

A君は善良な一市民という言葉が特に似合うような、ごく普通の青年でした。彼はどちらかというとアウトドア派で、得意な科目は英語と社会でした。学校でコンピューターを使った授業もあったのですが、彼はそういったことにはあまり興味がなかったので、友達のレポートを丸写しして切り抜けてしまいました。

ですから彼はコンピューターのことなど全然知りません。でもそれで特に不都合もなかったのですが…。

そんな彼も学校を卒業して社会に出るときがやってきました。昨今はなにかと不況です。とりたてて特技を持たない彼がなんとか就職できたのは、とある田舎のちょっと怪しそうな会社でした。それでも職がないよりは遙かにましです。彼はやる気満々でした。

ところがその会社で命じられた業務というのが、コンピューターの操作だったのです！

彼の経歴を見れば分かる通り、A君はコンピューターのことなど毛頭分かりません。ただ運がよかったことは、作業がちゃんとマニュアル化されていたことです。そのマニュアルはなかなか親切で、パソコンの起動の際にはこういう手順で行い、作業のためにはこういう風に操作して…と絵入りで解説されています。その通りに操作していればよかったので、しばらくはなんとかなりました。

ところが、トラブルは思いもかけないときにやってきます。

その日は休日でした。しかしA君はちょっと作業の遅れがあったのでその日も出勤していました。もちろん、来ているのは彼一人です。

そしてA君がいつも通りパソコンを起動しようとしたときのことです。どうい

うことでしょう？　普段ならピポッと音がしたあとは放っておけば入力画面になるのですが、今回はなにか違います。真っ黒な画面になにやら

Invalid System Disk（16ページ参照）

とか表示されているではありませんか！

A君は、システムディスクというのはコンピューターの中にあるなにか記憶する装置らしいということは知っていました。Invalid とはなんでしたっけ？

彼が慌てて辞書を引いてみると…。

invalid

1.【形】効力のない、無効な
2.【形】病弱な、病人用の、病身で
3.【名】傷病兵、病弱者、病人

ヤバいです。なんだかとってもヤバそうです。もしかして、あのウイルスって奴かもしれません！

A君は慌ててコンピューター関連担当の先輩の家に電話をかけました。

「せ、先輩！　マシンがウイルスにやられました！」

先輩は慌てました。

「なに！　本当か？　今行く。待機してろ！」

先輩の家は郊外の住宅地なので、やってくるには一時間半かかります。しかし本当にウイルスなら事は急を要します。

そして一時間半後…。

「おい！　マシンはどんな様子だ？」

「はい。見てください。ほら、システムディスクが病気だって…」

それを見た途端に先輩の顔がなにか赤黒く変色しました。

「貴様ぁぁぁ！　こ、こんなことで俺を呼び付けやがったのか？」

「は？」

先輩の目はなんだか焦点が合っていません。これは例の逝っちゃってる目と言うのでは…。

「お、落ち着いてください先輩！　でも…」

「たわけが！　フロッピーが入ってるだけじゃねえか！」

「??」

A君は先輩がなにを言っているのか理解できませんでした。先輩はA君ににじり寄ってきます。

「おい！　俺はな、デートをな、デートを断ってきたんだぞ！」

「え？　そ、そうなんですか」
「今日のデートの約束を取り付けるのにどれだけかかったと思ってる？」
先輩の目には涙がにじんでいるように見えます。
「えっと、じゃあ、ウイルスじゃなかったんですか？」
「当たり前だ！」
「でも、あの、だったら、未然に防げてよかったですね。あはは」
その一言に先輩はついにキレました。
「うるさい！　おまえなんか殺してやる！」
「うわあああああ！　やめてください！　やめて！」

それから10分後…。
「どうして俺がこんな目に…？」
A君は冷たい床の上に横たわったまま力なくつぶやきました。思いっきりどつき回されたせいで、痛くて体を動かすこともできません。
「俺はなにもしてないのに…」
フロッピーを入れたまま帰ったのは彼ではありませんでした。というか、そんなことをするのは先輩ぐらいしかいません。それなのに、どうしてここまでされなければならなかったのでしょうか？　確かに先輩は普通よりは短気な性格でしたが、これはちょっと常軌を逸しているのではないでしょうか？

そう思ったとき、A君の心にふっとある考えがよぎりました。
“もしかしたら先輩はコンピューターに操られているのでは？”
A君の脳裏に昔見た映画のシーンが浮かび上がりました。確か最初は猿が変な鉄板の周りで踊っていて…それから宇宙になってHALとかいうコンピューターが出てきて…変な怪物が乗組員を皆殺しにし始めて…最後は自爆装置が「光あれ！」って…うわあああ！
ふと上を見上げると、机上のPCが見えます。それはじっと黙ってA君を見返しているような気がします。
「もしかしたら…こいつらは俺を監視しているのか？」
そう思った途端にA君は矢も楯もたまらなくなりました。
彼はふらふらと立ち上がると、そのまま部屋の中のコンピューターを片っ端からたたき壊し始めました。
物音を聞いて慌ててやってきた先輩に向かってA君はさわやかに言いました。
「これでやっと人間に戻れますよ！　先輩！」

こうして人間には戻れたA君ですが、人生は終わってしまいました。

…と、もちろんこれは実話ではありません。しかし中には、これが決して他人事に感じられない人がいるのではないでしょうか。

SF小説や映画ではコンピューターやロボットの反乱というのは非常に古典的なテーマです。アイザック・アシモフがロボット三原則を考案したのは1940年代の話です。それにも拘わらず、このテーマは現在に至るまで古びていません。

アイデアが古びないというのは、人の心の琴線に触れるなにかがあるからです。それは人がロボットやコンピューターを便利だと思う反面、同時に常に漠然とした不安感を抱き続けてきたからではないでしょうか。だからこそ機械が反乱するというお話を単純には笑い飛ばせないのです。

これが宇宙人の侵略の話ならどうでしょう。本当に宇宙人がいたら、その脅威度はロボットやコンピューターの比ではありませんが、幸運なことに今のところ地球外に生物が存在するという直接的な証拠はありません。近い将来、宇宙人と出くわすこともまずないでしょう。

しかし、これがコンピューターとなると話が違います。前述の通り現在は日常生活の中にコンピューターは当たり前のように浸透しています。近い将来にはロボットもまた同様に生活に入ってきそうです。

それにも拘わらず、あなたはそれがどういった素性のものか分からないわけです。これは大変まずい状況とは言えないでしょうか？

そのことに気づいたA君は、それらすべてを排斥するという手段に打って出ました。しかしこれはあまりよいやり方とはいえません。よく分からない外国人が隣に越してきたので、怖くなって皆殺しにしてしまったというようなものです。全然洒落になってません。

こういった場合に普通の人はどうすべきかといえば、まずその相手とコミュニケーションをとってみることでしょう。外国人でもなんとか会話さえできれば、彼らが日本人とかけはなれた風俗習慣を持っていたとしても、必ずや理解し合える部分も生まれます。

もちろん相手をある程度理解できたからといって、それで問題が解決というわけにはいかないでしょう。しかし、少なくとも相手がまったく理解できなかったときに感じたような不安はなくなっていることでしょう。

もうお分かりですね。相手がコンピューターだったとしても同じ手が使えるわけです。すなわち、コンピューターとコミュニケーションがとれればいいのです。そうやって腹を割って話してみれば、相手が決してそんなに危険な奴ではないことが分かるかもしれません。

では、どうすればコンピューターと話し合うことができるのでしょうか？

そのためにはコンピューターに通じる言葉を知る必要があります。それが**プログラム言語**と呼ばれる言語です。

コンピューターの**プログラム**もしくは**ソフトウェア**という言葉は皆さんも聞いたことがあるかもしれませんが、これはプログラム言語で書かれた文章のことです。コンピューターはその文章を読んで、そこに書かれていることを実行する機能を持った装置なのです。

というわけで、これから本書はこのプログラム言語、プログラム、ソフトウェアというものについてもっとも基本的なところから解説していきたいと思います。

ただひとつ注意しておいてほしいのは、本書を読了したからといって明日からコンピューターを自由自在に扱えるようになるなんてことはあり得ないということです。この本はコンピューターというものを理解するための最初の一歩の手助けにしかすぎません。

でも、その最初の一歩こそが重要だということはどなたにも理解していただけると思います。

補足@2014…Invalid System Disk

本書が最初に出たのは2005年のことでしたが、新装版を出すにあたっていきなりジョークの解説が必要になってしまいました(笑)。

これなんですが、まずパソコンが起動するには起動用のプログラムが入ったシステムディスクが必要になります。そこで当時のパソコンは電源を入れると最初にFDD(フロッピーディスクドライブ)にシステムディスクがあるかどうかを調べて、ディスクがなければ内蔵のHDD(ハードディスクドライブ)から起動するという動作が普通でした。内蔵HDDは当然システムディスクも兼ねています。

ところがここでFDDに「システムディスクではないフロッピーディスク」が入っていたらどうなったかというと――パソコンはビーッと嫌な音をたてて、真っ黒な画面に「Invalid System Disk」と表示されていたのです。

そこで初心者がFDDにデータディスクを入れっぱなしで電源を切ってしまって、次回の起動時にA君のように大慌て、というのがわりと日常の風景でした。

でも現在のパソコンにはもうFDDなんて搭載されておらず、もし今そんな表示が出てきたなら本当に内蔵のHDDが壊れている可能性大です。すなわち真剣に洒落にならない事態で、先輩も本気で顔面蒼白になったことでしょうが……。

――そんなわけで、このように初版当時と状況が変わってしまったような点については補足@2014というコラムの形でコメントすることにしました。

01

コンピューターとは何か

プログラム言語とかソフトウェアといったものに対する解説を始める前に、まず、もっとも根元的な疑問に対して回答を与えておかなければなりません。
その疑問とは、「コンピューターとはいったいなんなのか？」という疑問です。

01-01 電卓とコンピューターの違い

コンピューターは日本語ではよく「電子計算機」と訳されます。すなわち「電子」の「計算機」という意味ですが、コンピューターができる以前にも世の中に「計算機」は存在しました。たとえば算盤とか計算尺とか手回し計算機（これを見たことのある人は相当ご年輩のはず）などです。
この従来の計算機の最新型がいわゆる電子卓上計算機、要するに「電卓」です。それでは、このコンピューターと電卓はいったいなにが違うのでしょうか？ 算盤とかが相手ならばずいぶん違うように見えますが、電卓とコンピューターはちょっと見には結構似ているように思えます。
でも、この両者の間には日本海溝や男と女の間並の深い溝があるのです。

ちょっと青春の気分に

それではちょっと昔に戻って、二次方程式のことを思い出してください。

え？ 思い出すのもいやだって？ 困りましたね。もしコンピューターのことをある程度理解しようと思ったら、中学から高校1年程度の数学の素養がどうしても必要になります。

といっても、どうしてもイヤだという人に無理強いする気はありません。今まで何千年も人はコンピューターなしで生きてきたのですから。はっきり言って、そういう生きかたの方が人は幸せかもしれません。

ただ、そう言ってしまったらこの本はここで終わってしまいますので、これ以降は少し我慢してでもコンピューターのことを知りたいという読者を前提として書いていきます。大した数式は出てきませんから安心してください。

さて、二次方程式とは以下のような式を満たすxを求めなさいという問題です。中学校で習ったはずです（旧制中学ではどうだったかは知りませんが）。

$$ax^2+bx+c=0$$

上の式でa,b,cが例えば3,4,−1であったとすれば以下のようになります。

$$3x^2+4x-1=0$$

さて、ここでもう忘れているかもしれませんが、二次方程式には有名な**解の公式**というのがあります。

$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

こういった公式があると便利なことは、この式さえ覚えておけば式の中のa,b,cを上の例ならば3,4,−1とそれぞれ置き換えて機械的に計算すれば**自動的に答えが出てしまう**点にあります。

$$x=\frac{-4+\sqrt{4^2-4\cdot3\cdot(-1)}}{2\cdot3}\qquad x=\frac{-4-\sqrt{4^2-4\cdot3\cdot(-1)}}{2\cdot3}$$

では上の式を電卓で計算してみましょう。ここで使う電卓は、その辺で売っていたりWindowsのオマケに付いているような四則演算（＋－×÷）と平方根（ルート）と簡単なメモリ機能があるだけとします。

これを使って計算するためには、だいたい以下のようにキーを押す必要があるでしょう。以下の例はWindowsにおまけで付いてくる電卓ソフトを元にしています。

解1

[4][*][3][*][+/−][1][MS]
[4][*][4][−][MR][sqrt][−][4][÷][2][÷][3][=]

解2

[4][*][3][*][+/−][1][MS]
[4][*][4][−][MR][sqrt][+/−][−][4][÷][2][÷][3][=]

> 安い電卓にはカッコの機能がないので、足し算引き算と掛け算割り算が混在していると結構大変です。そのため、計算の順序が式とは少し変わっています。

結構面倒ですね。二次方程式には解が2つあるので二回計算しなければいけません。でもこのぐらいならまだいいです。以下のような問題になったらどうしますか?

$$3.25674x^2+245.456x-0.295816=0$$

解1

[4][*][3][.][2][5][6][7][4][*][+/−][.][2][9][5][8][1][6][MS]
[2][4][5][.][4][5][6][*][2][4][5][.][4][5][6][−][MR]

[sqrt][−][2][4][5][.][4][5][6][÷][2][÷][3][.][2][5][6][7][4][=]

解2

[4][*][3][.][2][5][6][7][4][*][+/−][.][2][9][5][8][1][6][MS]
[2][4][5][.][4][5][6][*][2][4][5][.][4][5][6][−][MR]
[sqrt][+/−][−][2][4][5][.][4][5][6][÷][2][÷][3][.][2][5][6][7][4][=]

こうなると、数字の桁数が多いのでキーをたくさん押さなければなりません。その上1カ所でもキーを押し間違っていたら大変なので、せめてもう一度は検算してみないと答えに安心ができません。

なんだかやる気が失せてきますね。それどころか、

1. $7.0346x^2-61.4456x-2.98321=0$
2. $2.4678x^2+124.331x-1.46169=0$
3. $6.11492x^2+89.136x+0.881361=0$

...

99. $1.92647x^2-51.63x-3.56561=0$
100. $5.74231x^2-76.2935x-0.51203=0$

などという問題を出されることだってあるかもしれません。これを全部やれと言われてしまったらどうでしょう？ もうほとんど拷問です。

もっとも正しい選択は、そんな羽目に陥らないようにすることなのですが…。

逃げられなかった場合

人生、運が悪いことはよくあります。上のような無理難題を吹っかけられてしまったとしたら、いったいどうすればいいのでしょうか？

解決策その1：戦う

まず第一に考えるべきことは、やはり戦うことでしょう。

戦い方はいろいろあります。おもむろに立ち上がって、そんなことを言い出した上司をぶん殴るというのも考えられますが、これはあまりお勧めできません。

せめて、こんなことをしても無駄だと主張すべきでしょう。それで運良く上司を説得できれば問題そのものがなくなってしまうわけで、万々歳です。

しかし、たいていの場合は必要があるからそういう指示が出ているわけで、簡単にはうまくいきそうもありません。

解決策その2：逃げる

次の手は逃げることでしょう。

ただその場合、中途半端な逃げ方は逆効果です。単に放り出して帰ってしまうというのは一番まずいやり方です。せめて医者に診断書を書いてもらって、できれば入院してしまうぐらいの勢いが必要です。それでも相手がしつこいようなら辞表をたたきつけてしまいましょう。

そうやって最後まで逃げ切ってしまえれば、それはそれで勝利と言えます。

しかし、辞表を出した後の行き先がないこともよくあります。もしあなたに家族がいたりしたら、そういった境遇にはなりたくないはずです。

解決策その3：手下を使う

逃げることにも失敗したとすれば、今度はうまいこと他人に押し付けてしまうことを考えてみましょう。世の中にはもしかしたらこんな計算が好きな人がいるかもしれません。そうでなくともあなたより立場の弱い人がいたならチャンスです。今こそ、その「立場」を有効利用すべきときです。

ただこれも世の常ですが、そういった「立場」とは決して恒久的なものではありません。それを維持するためには、あなたは不断の努力をする必要があるでしょう。それに失敗したら相手に二倍返し、三倍返しされてしまうかもしれません。

と、いろいろ策を考えてみましたが、どれも今ひとつぱっとしません。

こういった方法が使えないのならば、仕方なく自力でなんとかする他はありません。ともかく生きていく以上は文句を言っても始まらない状況が多々あります。そんな場合はどうしたらいいのでしょう?

絶対にまずいのは、いきなり諦めて電卓を取り出し、せこせこと計算を始めることです。そうではなく、あなたはそこでちょっと立ち止まって、せめて**どうすればこの作業が楽になるだろうか**と考えなくてはなりません。

いかに楽をするかという課題は人類の進化の原動力です。獲物を解体するためには歯で食いちぎるよりも、石器を使った方が楽でしょう。苦労して歩き回って獲物を狩るよりも、動物を飼育して増やした方が楽でしょう。剣で一人一人刺殺するよりも、爆弾を使った方がとっても楽です。

人類はこうやって進歩してきたのです。すなわち、ここであきらめてしまってはご先祖様に申しわけが立ちません。

では、どういう方法があるのでしょうか?

ここで解決策その3を思い出してください。この方法はある意味優れた方法だと言えるでしょう。問題は「人に奴隷労働を強いたら反抗される」ということにあります。

それならば、反抗さえされなければいいのではないでしょうか？ といっても人間相手では相当に難しいことです。しかし、人間以外ならばどうでしょうか？

まず考えられるのは動物です。例えば猿を訓練してやらせてみるのはどうでしょう？ 想像するだけで大変そうです。猿以外の動物だともっと大変そうです。

だとすれば、あとは**そういった機械を作る**ことですが…そんなものが作れるのでしょうか？ 作れるとしたらどんな機械でしょうか？

仮にあなたが例の二次方程式100問を解き始めたとします。はっきり言って恐ろしく退屈な作業でしょう。なぜ退屈かというと、やってることがまったくワンパターンだからです。個々の問題は別な問題なので、あなたはまったく同じことを繰り返しているわけではありません。それなのに、どうしてこうもワンパターンでつまらなく感じるのでしょうか？

それはこの場合、各問の係数だけは変わっているものの、すべて二次方程式であるのは共通だからです。そのため、問題の解き方そのものはまったく同じです。あなたはただひたすらなにも考えずに、ある手順に従った計算をしていくしかすることがありません。だからこそ退屈なわけです。

実はこのある手順に従った計算、すなわち**計算のやり方**というのが大変重要です。

そこである人が考えたわけです。

計算機に、この計算のやり方を覚えさせることができないだろうか？

機械がやり方を覚えていてくれれば係数を入力するだけで、あとは機械が勝手に計算してくれるじゃないか！

そう考えたのは**フォン・ノイマン**という人でした。

このフォン・ノイマンは天才だったので、本当にそういう機械を作ってしまいました（ただし、彼の暗算能力は作った機械を越えていたという話ですが）。

時は二次大戦のさなか、できた機械の名前は“ENIAC”と言いました。この機械は部屋いっぱいの真空管でできていて、一秒間に3,000回の足し算ができたそうです。彼らはその機械を使って大砲の弾の弾道計算をしました。

すなわち、コンピューターという機械の始まりです。

> と、一昔前までは言われていました。なぜ一昔前かというと、「いや実はABCマシンこそが真の元祖だ」とか「実はバベッジの解析機関までさかのぼる」とか主張する人が現れて、現在おおもめにもめている最中だからです。まあどうでもいいことですけどね。

ソフトウェアがなければ…

以上、なにが言いたかったかというと要するに

コンピューターとは計算の手順を覚えておくことができる計算機

なのです。もう少し具体的に言うと、コンピューターとは以下のような動作をする計算機です。

1. **計算の手順**を組み込んでおく。
2. 必要な**データ**を入力する。
3. 手順に従って**計算**させる。

これだけなのです。前の例で言えば次のような動作になります。

1. 二次方程式の解き方をコンピューターに教える。
2. 問題のデータ（この場合は*a,b,c*の係数）を入力する。
3. 問題を解け！ と命令する。

ここで重要な点は、1の計算の手順で**異なった手順を入れておけばコンピューターはまったく違った動作をする**ということです。

コンピューターに二次方程式を解く手順を入れておけば二次方程式を解きますが、代わりに連立方程式を解く手順を入れておけば連立方程式が解けます。その他、統計計算でもなんでも計算手順さえ交換すればできるようになるのです。

言い換えれば、コンピューターには計算の手順を教えることができるわけです。

この計算の手順のことを**プログラム**と言います。プロローグでコンピューターとコミュニケーションをとるにはプログラム言語を使うと言いましたが、プログラム言語とはこのプログラムを記述するための、言い換えれば計算の手順を記述するための言語に他なりません。

そして、このプログラムとデータをひっくるめたものをコンピューターの**ソフトウェア**と言います。このソフトウェアこそが、コンピューターというものを理解する上でもっとも重要かつ基本的な概念と言えます。

ソフトウェア
　プログラム：計算のやり方をプログラム言語で記述したもの。
　データ　　：計算される値。

さて、ここでソフトウェアのないコンピューターがあったら、それはなにができるでしょうか？

もうお分かりですね。ソフトウェアがないということは、そのコンピューターはなんの計算のやり方も知らないわけです。言い換えればなにもできません。

皆さんも「コンピューター、ソフトがなければただの箱」という表現をご存じだと思います。ソフトウェアを入れ替えることでコンピューターは様々な動作をしますが、逆に言えば**ソフトウェアがなければコンピューターはなにもできない**のです。

コンピューターがワープロになれるわけ

ここまでコンピューターは計算機だと表現してきました。しかし、普通の人はコンピューターをあまり計算には使っていないのではないでしょうか。たぶんほとんどの人はワープロで文書を作ったり、ゲームをしたり、インターネットでムフフな動画を集めたりするのに使っているだけのはずです。

実際、二次方程式のような「計算」しかできないのであればコンピューターの用途は非常に限られたものでしかありません。それでは元が算盤の成り上がりのようなものが、どうしてワープロに化けることができるのでしょうか？

ここでコンピューターには計算の手順を覚えさせることができる、と説明したことを思い出してください。ものを**覚えている**以上、コンピューター内部になんらかの**記憶装置**があることを意味します。

次節でもう少し詳しく解説しますが、ご存じの通りコンピューターには「メモリ」と呼ばれる記憶装置が付いています。コンピューターは、その記憶装置に書かれたデータを**読んだり書いたり**できるのです。

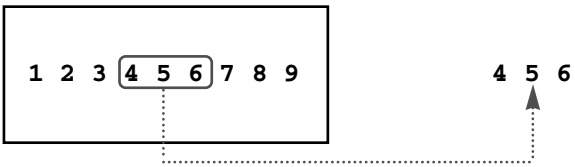
読んだり書いたりできるというのはどういうことでしょう？　これは、記憶装置の好きな場所にデータを書き込むことができるし、好きな場所のデータを読むことができるということです。まあ当然です。この程度のことができなければ困るのは分かるでしょう。

さて、それでは記憶装置のある場所に以下のような数字列が書いてあったとします。

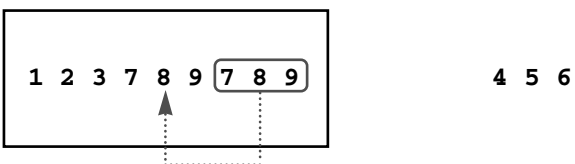
```
1 2 3 4 5 6 7 8 9
```

この数字列に次のような操作（読んで書くだけです）をします。

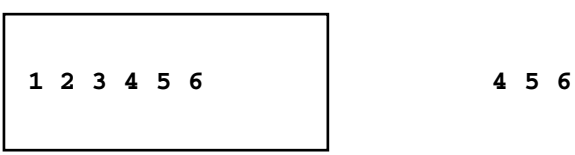
1. 真ん中の"456"を読み出して、どこか遠くの見えない場所にその値を書き込む。



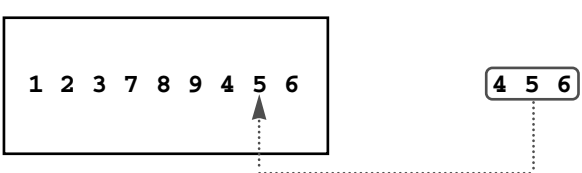
2. 右の"789"を読み出して、"456"が元あった場所に書き込む。



3. 読み出した"789"の場所を空にする。



4. 遠くに書いておいた"456"を読み出して、元の"789"のあった場所に書き込む。



これはなんでしょう？ 操作1～3をワンセットで考えれば、ワープロなどでよくある「カット」とか「切り取り」と言われる操作とそっくりです。同様に操作4の操作は「ペースト」とか「貼り付け」と言われる操作そっくりです。これが可能ならば、同じように挿入・削除や検索・置換なんかもできそうだと思うでしょう。

コンピューターがワープロになれる最大の理由は、このようにコンピューターが記

憶装置を持ち、そこからの**データを読み書きできる機能を持っている**からに他なりません。

コンピューターが電卓のような今までの計算機以上の機能を持てるのは、この点に尽きると言っていいでしょう。この操作ができるために、コンピューターはほとんど無限に近い可能性を持っているのです。

果てしなき野望

かくしてうまくソフトウェアを作ってやれば、コンピューターは非常に多彩な動作をすることができるのです。そこから極論、

コンピューターはなんでもできる！

という考え方が生まれてきます。

昔のSFにこんなのがありました。ある科学者が完璧なコンピューターを作ります。それが完成すると、科学者はコンピューターに「神はこの世に存在するか？」と質問しました。コンピューターはこう答えました。「然り。今こそ神は存在する」と。この話は、そのもっとも極端な例だと言えるでしょう。

でもこれは**嘘**です。心配する必要はまったくありません。なぜ嘘なのか、それが今後の重要なテーマになってきます。

01-02 ハードウェアの一番基本的な話

前節でソフトウェアという概念が出てきました。本書は今後それに関する話をしていきたいわけですが、そのためにはコンピューターが具体的にどういう構造を持った装置なのかということをある程度は知っておいてもらわないと話が進みません。
そこでこの節ではもう１つの重要な概念である「ハードウェア」、すなわちコンピューターの本体に関する簡単な解説をします。

もっとも手抜きで描いたコンピューターの構成

前節でコンピューターは次のような動作をすると書きました。

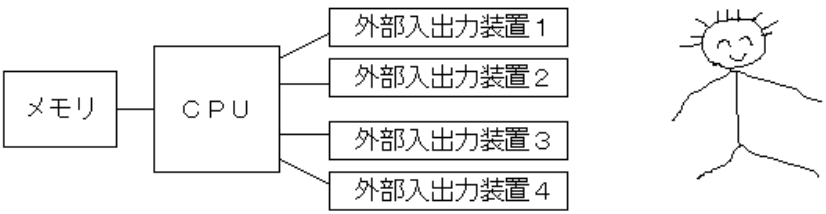
1. 計算の手順を組み込んでおく。
2. 必要なデータを入力する。
3. 手順に従って計算させる。

このような動作をさせるためには、計算機にはどういった装置が必要でしょうか？少なくとも以下のような部品が必要だということはお分かりになると思います。そして上記のような動作をさせるのであれば、これだけあればあとは特に必要なさそうなことも分かると思います。

CPU（中央演算装置）	計算機の本体、すなわち問題を解くという本命の動作をするための装置です。
メモリ	プログラムやデータを記録しておく装置です。当然のことながらコンピューターは教えたことを覚えておかなければなりません。したがって、このような「記憶装置」が必須になってきます。
外部入出力装置（IO機器）	プログラムやデータを入出力するための装置。人がコンピューターになにか教えたり答えを受け取ったりするためには例えばキーボードとかモニタといった装置が必要なのは明らかでしょう。通常のパソコンには他にもいろいろ装置がつながっていますが、それらは結局すべて外部入出力装置の一種だと言えます。

これらの装置のことをひっくるめて**コンピューターのハードウェア**と言います。

では、それがどのように組み合わされているのでしょうか？ 次の図は世界で一番いい加減に描いたコンピューターの図です。その横に同じぐらいいい加減に描いた人の絵が載っています。



ハードウェアの基本はこの図に集約されます。これならすぐ覚えられますね。それではそのおのおのについて説明してみましょう。

CPUとは

どんなにしぶとい人間でも、頭を潰せば生き返ってはきません。**コンピューターの頭に相当する装置**、それがCPUです。それではいったいCPUとはどのような動作をするのでしょう？ それはおおむね以下のようなものです。

1. メモリ上にある計算の手順を1行読み込む。
2. 書かれている内容に従って様々な動作（計算、データの読み書き、外部機器とのデータのやりとり）を行う。
3. 次の命令を読む。
4. それに書かれている命令に従って…以下略。

基本的にはこれだけです。具体的にどんな命令があるかとかは今後の話になりますので、とりあえず置いておいてください。

CPUはこの動作を周期的な電気信号のタイミングに合わせて行います。この周期的な電気信号が「クロック周波数」と呼ばれるもので、周期が短いほど動作は速くなります。例えば「Pentium4 3ギガヘルツ」といった表記がよくありますね。この3ギガヘルツというのがクロック周波数の速さです。この例だと一秒間に30億回の周期になります。すなわち、このCPUは一秒間に最大30億回の命令を実行できるという意味になります。

> ただし命令の中には１つの命令で数周期かかったり、データがやってくるの待っていたりしなければならないこともあり、実際にはもっと回数は減ってしまいますが。

細かいことはともかく、一般的にはクロック周波数が高ければCPUの性能も高いと考えてそれほど大きな間違いではありません。

この他に、CPUの性能を左右するものにデータのビット幅があります。

よくCPUの説明に16ビットCPUとか32ビットCPUとかいう表現があるのを見たことがあるでしょう。これは大ざっぱに言えば、一度（＝1命令で）に処理できるデータの量を示しています。当然ながら、一度に処理できるデータ量が増えれば性能もよくなることが想像できるでしょう。

初めてできたCPUは4ビットでした。パソコンが出たての頃は8ビットが普通でした。そして現在では32ビットが主流になっています（316ページ参照）。

また同じクロック数、同じビット幅のCPUでも、そのアーキテクチャ（設計）によって能力は異なります。さらに今後は1台のマシン上にたくさんのCPUが乗っているなどということも普通になっていくでしょう（316ページ参照）。

ただどれほど速度が上がったとしても、CPUの動作そのものが大きく変わることはないでしょう。

メモリとは

コンピューターをコンピューターたらしめるもっとも大きな要素は、実はこのメモリの存在です。コンピューターは計算のやり方を「覚える」ことができる以上、どこかに記憶装置が必要なのは自明です。

それでは、このメモリにどのように記憶が保存されるのでしょうか？　概念的に書けば下のようなものです。

番地	0	1	2	3	4	5	6	7	8	9	10	11	…
中身	130	203	130	177	145	229	141	68	130	171	13	10	…

メモリとはこのように

数値を書くことのできる領域がずらっと一列に並んでいるもの

です。各領域には0から始まる通し番号の**番地**とか**アドレス**と呼ばれる値が付いています。上記の例であれば、メモリの3番地には177という値が入っています。

CPUはこの番地を利用してメモリの中身を読み書きできます。「345番地の中身を読め」とか「647番地に値23を書け」といった感じです。

ちなみにこの領域に書ける数ですが、実際のコンピューターが使用しているメモリでは**0～255の範囲の整数**しか書けません。

コンピューターのカタログを見ると「メインメモリ512メガバイト」とかいった記述があることに気づくかもしれません。これは、このコンピューターのメモリは0～255まで書ける領域が536,870,912個ぐらいあることを示しています。

と、これでメモリがどういうものであるかの説明は終わりなのですが…。こんな仕組みで複雑な計算のやり方などというもの記憶することができるのでしょうか？ だいたい数値を書くにしても、0から255って、ちょっとなんだかしょぼすぎないでしょうか？

でも全然そんなことはないのです。ただ、ここにその説明を書くにはスペースがなさすぎます。それを説明するために本書の2～4章があるぐらいです。なので、ここではそれで問題ないのだと信じておいてください。

外部入出力装置とは

コンピューターのハードウェアの第三の要素として、外部入出力装置があります。
理論的にはこれはなくてもかまわないものですが、実際上、これがないとなにもできません。

データを入力したり表示したりする装置（31ページ参照）

モニタ	数値や文字、絵といった結果を表示して人の目で見えるようにする装置。
プリンタ	数値や文字、絵といった結果を紙に印刷する装置。
キーボード	コンピューターに手で文字データを入力するための装置。
マウス	コンピューターに手で位置データを入力するための装置。

データを保存しておくための装置（31ページ参照）

ハードディスク	様々なデータを一時的に記憶しておく装置。そこそこの容量がある。
フロッピーディスク	様々なデータを一時的に記憶する上に持ち運びもできるが、容量があまりにも少ないのでそろそろ時代遅れ。
MO	様々なデータを一時的に記憶する上に持ち運びもできる。フロッピーディスク数百枚分が入るが、結局あまり普及していない。
CD-ROM	CDに入っているデータを読み込む装置。読み込みしかできないが、そこそこの容量がある。たいていのパソコンに付いている。
DVD-ROM	DVDに入っているデータを読み込む装置。CDより容量が大きい。
紙テープ	様々なデータを一時的に記憶する上に持ち運びもできる。しかも、なにが書かれているのか目で読むことさえできる!! しかし、フロッピーディスク1枚分ぐらいのデータのために5,000メートルぐらい必要なのが欠点。

面倒くさいのでほんの一部しか書いていませんが、以上を見れば分かる通り、ないと困りますね。

というわけで、ハードウェアに関する一番基本の基本はこんなものです。簡単でしょう？
しかし、総論は簡単でも各論になるとややこしくなるのは世の常です。CPUといっても山のように種類があるし、外部入出力機器に至っては星の数ほどあります。
その上ここでは概念しか説明していないので、それが具体的にどのような構造を持っていてどういう原理で動作するかという点は結局不明のままです。

でもコンピューターの基本を理解するためだけなら、この程度を知っておけば十分なのです。

補足@2014…外部入出力装置の変化

時代と共に使われる入出力装置にも流行り廃りがあります。
まずデータを入力する装置には今ではタブレットを加えなければならないでしょう。またノートパソコンなどではマウスカーソルを操作するためにタッチパッドが付いているのが普通になりました。
データを保存する装置については、表にあった中では紙テープは冗談として、フロッピーディスクとMOも現在ではほぼ滅びてしまいました。
CD-RやDVD-Rは大きな容量のデータの受け渡し用として今でも使われていて、さらに容量の大きなブルーレイディスクもあるのはご存じだと思います。
しかし何といっても現在データを持ち運ぶ際にはUSBメモリや、SDカードなどのメモリカードを使っているのではないでしょうか。
この両者は見かけは違いますが、共に半導体ベースで、電源を切っても内容が消えないフラッシュメモリというもので作られています。
このフラッシュメモリそのものはかなり昔からありましたが、価格が高かったこと、データの書き込みが遅かったこと、書き込み可能な回数が限られていたことなどから、当初は各種機器の設定保存用などに使われていただけでした。
しかし技術の進歩によりそういった欠点が徐々に克服され、価格も安くなってきたため、現在では持ち運び用のメディアとしては最も利用されているものになったといえるでしょう。
それだけでなく長年パソコンのメインの記憶装置はHDD（ハードディスクドライブ）の牙城でしたが、それに代わるフラッシュメモリベースのSSD（Solid State Drive）も普及しつつあります。

01-03 コンピューターは数しか分からない

前にも書いた通り、コンピューターはソフトウェアがなければ粗大ゴミです。ソフトウェアとはデータとプログラムのことだと書きました。もちろん、それだけで分かってしまったという人がいるとは思えません。
ここでは、そのソフトウェアに関してもう少し詳しく説明をしましょう。

基本中の基本

まず最初に知っておかなければならないことは、コンピューターとは所詮のところ「計算機」なのだということです。したがって以下の大原則があります。

1. そもそもコンピューターは数値しか分からない。だからソフトウェアはすべて数値で記述される。
2. しかもコンピューターが直接理解できるのは2進法で表記された数だけである。

まず、これだけはしっかり覚えておいてください。
特に1は重要です。このあたりで本書を読むのにすっかり疲れてしまった人でも、これだけは覚えておいてください。

と、そう言われてもすぐにはピンとこないでしょうね。なぜなら実際のコンピューターでは明らかに数以外の対象も扱えているからです。

コンピューターがワープロになれる話を書いたように、コンピューターでは文字や絵の編集をすることができます。これはコンピューターが文字や絵を理解している証拠ではないでしょうか？ なにか話が矛盾していないでしょうか？

矛盾しているわけではありません。コンピューターは確かに数しか分からないのです。コンピューターで数以外の対象を扱えているように見えるのは、

その「対象」はあらかじめなんらかの方法で「数」に変換されていて、コンピューターは変換後の数を扱っているだけ

だからなのです。見かけ上はコンピューターが文字や絵を操作しているようには見えますが、実はコンピューター本体は数値の操作を行っているだけなのです。

　このためコンピューターを利用する際には、**扱いたい対象が数でなければまず最初にそれを数で表す**という作業が必要になります。これはコンピューターを利用する際に最初に直面する、もっとも重要な問題の1つになります。

　2章で様々な対象が具体的にどうやって数で表されているかを解説していますので、詳しくはそちらを読んでみてください。

10進法とか2進法とか

　というわけで、1はそういうものだと信じてもらえればいいのですが、2はちょっと問題です。**2進法**ってなんでしたっけ？　昔に習ったような微かな記憶があるかもしれませんが、いきなり説明しろと言われてもなかなか難しいですね。
　しかしこれはコンピューターを語る上では非常に重要なポイントなので、ちょっと詳しく説明しましょう。

　さて、下にたくさんの●があります。これはいくつあるでしょう？

●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
●●●●●●●●●

　なんだかものすごく人を馬鹿にした問題です。もちろんこれができない人はいないでしょう。少し我慢して数えてみれば●は237個です。
　ではこの237という数ですが、各桁に2とか3とか7という数字がなぜ出てくるのでしょう？　と言われたら少し頭を抱えてしまうかもしれません。先に答えを言ってしまえば

上の●の数を10進法で表したから2,3,7という数字が出てきた

のですが…。だからどうしてと言われそうですね。でも、10進法という言葉の意味がよく分かっていれば実は当然の話です。

　そこで、上記の●を次のように並べ替えてみましょう。

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。

Tech Book Zone Manatee

3.6
2017

おすすめ電子書籍

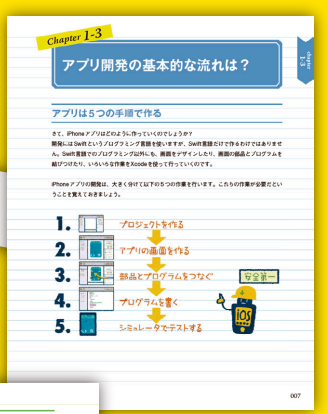
Tech Book Zone Manatee

1

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い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。

イラストによる解説で、プログラミングがはじめての人でも学べる



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ける

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】

Swift (iOS開発)

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

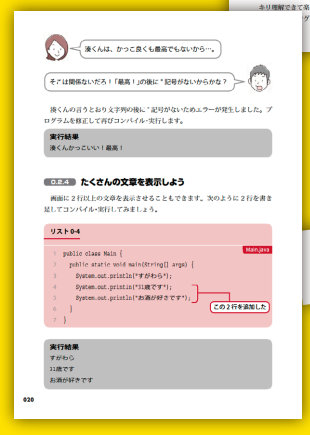
2

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」という方にもおススメです。

300ものイラストで楽しく・詳しく・スッキリとマスター!

「プログラミング」



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

スッキリわかる Java 入門 第2版

Java

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

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。

シーアンドアール研究所
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 アプリのつくり方 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。

シーアンドアール研究所
古籏一浩（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

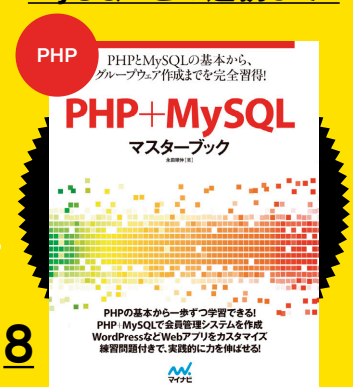
いちばんやさしい PHP の教本

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？」「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。それが分かってきたら、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)