

計算機の歴史と仕組み

暮らしに生きるコンピュータ

アトリエ未来 [著]



算盤からiPhoneまで

アルゴリズム & プログラミング

計算機の歴史と仕組み

— 暮らしに生きるコンピュータ —

[著] アトリエ未来

「技術書典 11 新刊」
令和三年七月七日 ver 1.0.0

■免責

本書は情報の提供のみを目的としています。

本書の内容を実行・適用・運用したことで何が起きようとも、それは実行・適用・運用した人自身の責任であり、著者や関係者はいかなる責任も負いません。

■商標

本書に登場するシステム名や製品名は、関係各社の商標または登録商標です。

また本書では、TM、^(R)、^(C)などのマークは省略しています。

始めに

楽しいプログラミングの世界へようこそ。

情報技術「IT」に囲まれた生活を送るわたくしたち。多くの先人が築いた歴史の上に今日があります。

ページ数の関係で、二冊に分冊いたしました。算盤から iPhone まで、コンピュータの歴史と仕組みを学びます。また、アルゴリズムや学校教育におけるプログラミングについても簡単にご紹介しています。

姉妹巻となる「じゃんけんゲームを創ろう - 初めてのウェブプログラミング」では、HTML / CSS / JavaScript を学び、じゃんけんアプリを作り上げていきます。

巻末には、今後の成長へつなげて欲しいとの願いから、参考書籍を付けました。

現代の魔法、それがプログラミングです。自由自在にコンピュータを操つて、幸せな未来へと大きく羽ばたいてください。

対象読者

パソコンの操作ができる、初心者の方を想定しています。歴史や仕組みを知りたい方、プログラミングに興味がある方への最初の一歩になればと願っています。

また、誰かに教えようと思っている方にも、生徒の方にお渡しするテキストとして、お使い頂けるかと思います。

謝辞

Re:VIEW Starter^{*1}を用いて、綺麗に製本することができました。作者の kauplan 氏に厚く御礼申し上げます。

^{*1} <https://kauplan.org/reviewstarter/>

著者紹介

卓越した技能を有する者として認められる国家資格「応用情報技術者」を保持。平成 30 年より、プログラミングの個人指導やウェブ開発を行う「アトリエ未来^{*2}」を創業。HTML 講座や Ruby 講座、IT パスポート講座等を開催している。

趣味の将棋は、日本将棋連盟より三段の免状を允許。日本の美しい自然や豊かな精神性を宿す熊野古道を歩くことや、たくさんの花に囲まれた日々を愛している。

^{*2} <https://atelier-mirai.net/>

目次

始めに	i
対象読者	i
謝辞	i
著者紹介	ii
第1章 暮らしの中のコンピュータ	1
第2章 コンピュータの歴史	3
[コラム] ムーアの法則	6
第3章 コンピュータを創った人々	9
3.1 ジョン・フォン・ノイマン	9
3.2 クロード・シャノン	10
3.3 アラン・チューリング	10
3.4 ジョージ・ブール	11
3.5 ニクラウス・ヴィルト	11
第4章 コンピュータの仕組み	13
4.1 ハードウェア	14
4.2 ソフトウェア	17
♣ OS (Operating System / 基本ソフト)	17
♣ アプリケーションソフト	17
第5章 コンピュータ よもやま話	19
♣ インターネット	19
♣ クラウド	19
♣ IoT	20
♣ 量子コンピュータ	20

♣ 人工知能	21
♣ 電子署名（デジタル署名）	21
第6章 二進数の話	23
6.1 二進数と十進数	23
[コラム] 金の延棒クイズ	24
6.2 単位のお話	26
♣ おまけ	28
6.3 計算機理論入門	29
第7章 アルゴリズムとは	31
7.1 アルゴリズムとは	31
♣ ナップザック問題 ^{*3}	32
♣ ハノイの塔 ^{*4}	32
♣ 巡回セールスマン問題 ^{*5}	32
7.2 1から10までの合計を求める	33
7.3 より良いアルゴリズム	33
[コラム] カール・フリードリヒ・ガウス	34
第8章 プログラムとは	37
8.1 プログラムとは何でしょうか？	37
8.2 プログラミング言語の種類	38
♣ 機械語とアセンブリ言語	39
♣ C言語	39
♣ Ruby(ルビー)	40
♣ Scratch(スクラッチ)	40
8.3 学校教育でのプログラミング	40
♣ なぜプログラミング教育を導入するのか	40
♣ 小学校でのプログラミング教育のねらい	41
♣ 中学校でのプログラミング教育のねらい	42
♣ 高等学校でのプログラミング教育のねらい	42
♣ プログラミング的思考とは	42
♣ 「正多角形を描く」場合について考える	43

♣ 実践例の紹介	44
付録 A 珠玉の名著のご紹介	45
A.1 HTML / CSS / JavaScript を学ぶ	45
♣ CSS グリッドで作る HTML5 & CSS3 レッスンブック	45
♣ JavaScript Primer 迷わないための入門書	46
♣ 動く Web デザインアイディア帳	46
A.2 コンピュータサイエンスの基礎を学ぶ	46
♣ みんなのコンピュータサイエンス	47
♣ プログラマの数学	48
♣ 教養としてのコンピュータサイエンス講義	48
♣ キタミ式イラスト IT 塾 IT パスポート	49
A.3 プログラマとして上達したい方のために	49
♣ 新装版 達人プログラマー 職人から名匠への道	49
♣ コーディングを支える技術	50
A.4 アルゴリズムを学ぶ	50
♣ アルゴリズム図鑑 絵で見てわかる 26 のアルゴリズム	51
♣ C 言語による標準アルゴリズム事典	51
A.5 Ruby を学ぶ	51
♣ 初めてのプログラミング 第 2 版	52
♣ たのしい Ruby 第 6 版	52
♣ プロを目指す人のための Ruby 入門	53
♣ Ruby によるデザインパターン	53
終わりに	55

第1章

暮らしの中のコンピュータ

今日の情報社会では、暮らしの中の至る所にコンピュータが使われています。どのような箇所に使われているのか、それを見ていきます。

日々の暮らしの中の様々なところにコンピュータを使ってわたしたちは生きています。「コンピュータ」の代表は、パソコンやスマートフォンですね。



▲図1.1:暮らしの中に生きるコンピュータ

コンピュータ (Mac / iPad / iPhone)

ウェブサイトの閲覧、書類作成や経理など事務作業に、自動車や飛行機の設計に、電車のダイヤを組んだり、絵を描いたり、電話やビデオ通話、

電子メールやチャットで連絡を取り合ったり、ゲームで遊んだりなどなど、さまざまなことができます。そして、もちろん「プログラミング」も楽しめますね。

クーラー

暑い日々が続くとき、心地よく過ごせるよう冷風を送ってくれるクーラー。これにもコンピュータが使われています。室温を検知して、設定温度を保つよう、風量を調整します。「感じて」「考えて」「動く」。コンピュータの基本です。

炊飯器

毎朝美味しい御飯を炊き上げてくれる炊飯器。「初めちょろちょろ、中ぱっぱ、赤子泣いてもふた取るな」と火力を調整したり、指定した時刻に炊き上げるためのタイマー機能など、小さなコンピュータ（マイコン）が組み込まれています。

給湯器

お風呂の給湯器、「38度」のお湯が出るように設定すると、浴槽が満水になるまで自動で給湯してくれます。五右衛門風呂のように、水を汲み、お湯を沸かし、湯加減を確かめてと、手間をかけることなく、ボタン一つで入浴できるようになりました。

信号機

赤信号、青信号と、点灯させる色を変更するのはもちろん、周囲の道路状況を見て、近隣の信号機と連携、青信号の時間を調整するなど、ここでもコンピュータが活躍しています。

バス

行き先表示の電光掲示板を始め、車内では料金決済のためのICカードや、バス停では運行状況を表示したりしています。エンジンや電気自動車の出力制御などにも使われています。

他にも、身の回りのいろいろなところにコンピュータが使われています。どこに使われているでしょうか。家の中で、お店で、学校や会社でなど、探してみましょう。

第 2 章

コンピュータの歴史

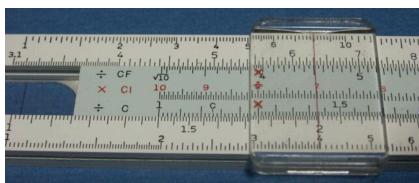
昔から人は暦や税、土木工事や建築など、様々な計算を行ってきました。この章では、太古から現代に至るまでの計算機の歴史を振り返っていきます。

「コンピュータ」とは何でしょうか？ とても高速に多くの計算をする機械、それがコンピュータです。昔から人は暦や税、土木工事や建築など、様々な計算を行ってきました。暗算や筆算をしたり、算盤や計算尺を発明しました。たくさんの歯車を組み合わせた機械式計算機が生まれ、電気に関する理解が深まり、「電子計算機」＝「コンピュータ」が現れました。



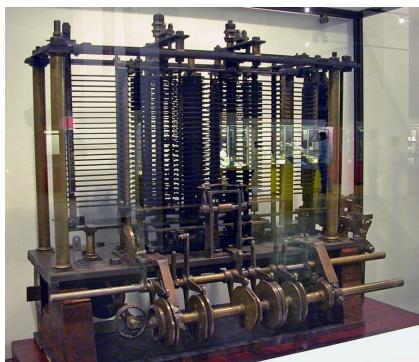
算盤

江戸時代には「読み書きそろばん」といわれ寺子屋や私塾などで実用的な算術が教えられており、昭和中期までは、事務職や経理職に就くには珠算が必須条件だった。



計算尺

対数の原理を利用したアナログ式計算用具で、乗除算および三角関数、対数、平方根、立方根などの計算用に用いる。



バベッジの解析機関 試作品 (1837年)

イギリス人數学者チャールズ・バベッジが設計した、蒸氣機関で動くはずだった機械式汎用コンピュータであり、対数や三角関数の数表を作ることに特化した計算機である。



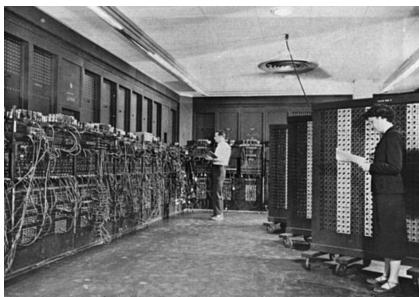
タイガー計算機 (大正 13 年)

歯車などの機械要素により計算を行う機械式計算器としての代表的存在であり、大正時代に発明されたタイガー計算機は昭和 45 年まで発売された。



東京帝国大学 九元連立方程式求解機 (昭和 19 年)

昭和 13 年頃、東京帝国大学航空研究所の佐々木達治郎を中心として、機械式計算機の開発が進んでいた。MIT の Wilbur の作った連立方程式の求解機の情報に基づき、志賀亮と三井田純一が九元連立方程式求解機の製作を担当した。昭和 19 年頃に完成、9 個の変数を持つ連立方程式を解く、我が国初のアナログ計算機となつた。



米国陸軍 ENIAC(エニアック)(昭和 21 年)

米国ペンシルベニア大学で開発された黎明期の電子計算機。パッチパネルによるプログラミングは煩雑ではあるが、米陸軍による砲撃射表の計算、マンハッタン計画、円周率計算など、汎用的な計算問題を求解できた。

17468 本の真空管で作られていて、幅 24m、高さ 2.5m、奥行き 0.9m、総重量 30 トンと大掛かりな装置でした。消費電力は 150kW で、開発費は 49 万ドルでした。



Z80 (昭和 51 年)

米国ザイログによって製造された 8 ビット・マイクロプロセッサ。昭和 60 年頃までは、パソコンの CPU としてなど、幅広い用途に使用された。現在でも組み込み用途など、目に見えないところで多用されている。



日本電気 PC-8001(昭和 54 年)

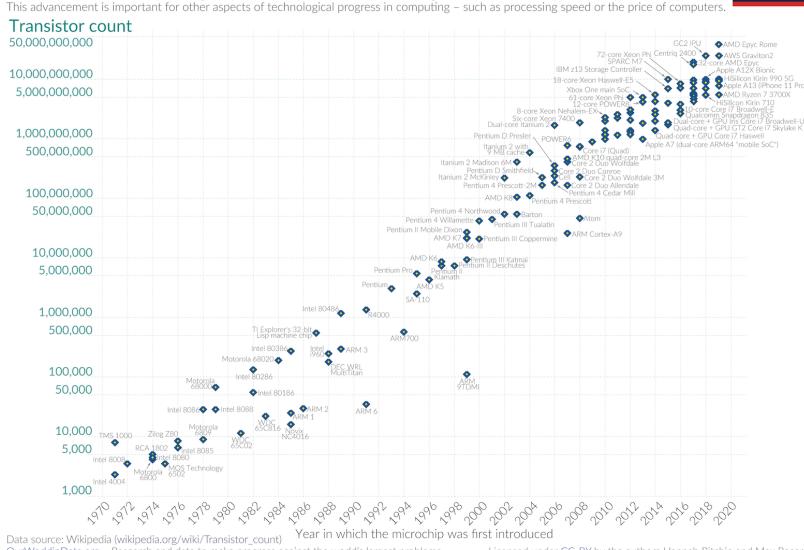
日本電気 (NEC) が発売したパソコン。キーボードと本体が一体化したデザインで、ハード・ソフトともに高い機能と完成度を有しており、数多くのソフトウェアや周辺機器が販売された日本のパソコンの代表機種。

【コラム】ムーアの法則

ムーアの法則^{*1}とは、大規模集積回路 (LSI) の製造・生産における長期傾向について論じた 1 つの指標であり、経験則に類する将来予測である。米インテル社の創業者のひとりとなるゴードン・ムーアが、集積回路あたりの部品数が毎年 2 倍になると予測した。

Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years.



▲図2.1: 集積回路のトランジスタ数の増大

毎年二倍とは

毎年 2 倍!! すごい増加率ですね。対数を知っていると簡単に計算できますが、一日 0.2% ずつ複利で成長すると一年で二倍になります。 $(1.002 \times 1.002 \times 1.002 \times \dots)$ と、365 回繰り返してみてください。)

どれくらいの増加率なのか、速さに置き換えて感じてみましょう。今年の新車は時速 100km で走るが、来年型は時速 200km。まああり得ません。ハイハイを始めた赤ちゃんは、時速 1km と言われています。マラソン選手は、時速 20km で走ります。ジェット機は、時速 900km で空を飛びます。ロケットは、時速 40,000km で、惑星探査に向かいます。数万倍に速くなりましたね。

黎明期のコンピュータとして有名な ENIAC(エニアック) は、

17468 本の真空管を使い、10 行の整数の足し算を毎秒 5000 回実行することができました。設置面積は 167 m² (約 100 噁)、消費電力は 150kW(電気ポット約 150 台分) でした。最新の iPhone は、118 億個のトランジスタを搭載し、一秒間に 11 兆回の計算ができます。驚異的な性能向上ですね。

多くの計算を速く行うために積み重ねられてきた人の営み。それが今日の豊かな社会を作りました。^{*2}

.....

^{*1} 出典: Wikipedia

^{*2} トランジスタ数のグラフであり、計算速度のグラフではありませんが、傾向を掴むためのものとして掲載しています。

第3章

コンピュータを創った人々

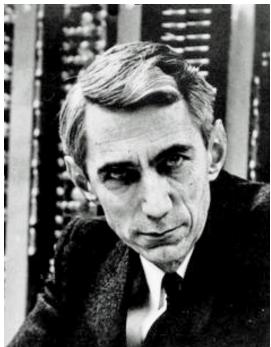
今日の情報社会が築かれるにあたっては、多くの方の貢献が欠かせませんでした。数多くの科学者や技術者が関わりましたが、ここではその中から著名な科学者をご紹介します。

3.1 ジョン・フォン・ノイマン



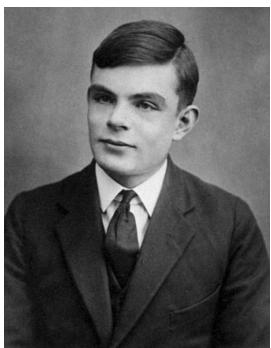
ジョン・フォン・ノイマンは、ハンガリー出身の数学者です。ほとんどのコンピュータの動作原理であるプログラム内蔵方式を考案しました。原子爆弾（マンハッタン計画）や黎明期の電子計算機ENIAC(エニアック)の開発でも有名です。ゲーム理論（複数人の意思決定を数学的に研究する学問）の成立に貢献しました。企業経営や軍事戦略理論や、将棋やチェスなどの零和ゲームの戦略など、社会に大きな影響を与えました。

3.2 クロード・シャノン



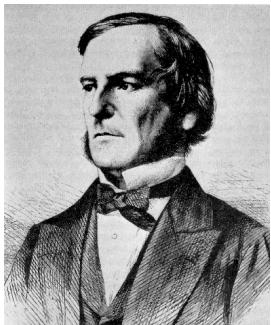
クロード・シャノンは米国の電気工学者、数学者です。情報、通信、暗号、データ圧縮、符号化など今日の情報社会に欠かせない分野を研究し、「情報理論の父」と呼ばれています。マサチューセッツ工科大学で論文を書き、電気回路・電子回路が論理演算に対応することを示しました。これにより、デジタル回路・論理回路の概念が確立され、コンピュータの実現に向け、とても大きな一歩となりました。情報量の単位「ビット」もシャノンの貢献です。

3.3 アラン・チューリング



アラン・チューリングはイギリスの数学者、論理学者、暗号解読者、コンピュータ科学者です。「チャーチ=チューリングのテーゼ(提唱)」と計算可能性理論への貢献で広く知られています。アルゴリズム(算法)を実行する機械を形式的に記述した「チューリングマシン」にその名を残しています。また「停止性問題の決定不能性」(無限の計算能力を持つコンピュータでも、解けない問題がある!)を示しました。また、エニグマの暗号解読への貢献や、チューリングテスト、コンピュータチエス、さらに実際面でもコンピュータの誕生に重要な役割を果たし、計算機科学、人工知能の父とも言われています。

3.4 ジョージ・ブール



ジョージ・ブールは、イギリスの数学者・哲学者。今日のコンピュータ科学の分野の基礎的な理論であるブール代数を確立した。組み合わせ回路（論理回路）はブール代数で表現できる。0と1を電圧の高低に対応させると、デジタル回路の入力と出力をブール論理の式で表現することができる。これにより、ANDゲート、ORゲート、NOTゲートのような基本論理回路や、NANDゲート、NORゲート、XORゲートなどを組み合わせてデジタル回路を構成することができる。

3.5 ニクラウス・ヴィルト



ニクラウス・ヴィルトはスイスの計算機科学者。プログラミング言語 Pascal、Modula-2などの開発や、ソフトウェア工学分野の開拓的研究で知られる。ヴィルトは、プログラミング言語 ALGOL W、Pascal、Modula、Modula-2 やオペレーティングシステム Oberon の開発などの功績により、ヴィルトは 1984 年にチューリング賞を受賞した。

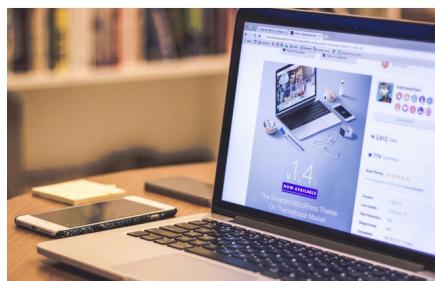
プログラミングの教育法について書いた記事 Program Development by Stepwise Refinement は、ソフトウェア工学の分野における古典である。1975 年の著作『アルゴリズム + データ構造 = プログラム』は広く知られ、今なお価値を失っていない。同書では、コンパイラ設計の説明のために、単純なプログラミング言語 PL/0 を設計。様々な大学のコンパイラ設計の授業で利用された。

第 4 章

コンピュータの仕組み

一台のコンピュータ。その中身はどのようになっているのでしょうか。
ハードウェアの仕組みを紹介します。

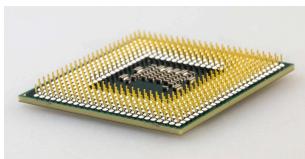
机の上に置かれたコンピュータとスマートフォン。
その中身を覗いてみると、演算・制御装置、記憶装置、入力装置、出力装置など、たくさんの部品からできています。詳しい説明は、IT 用語辞典 ^{*1} にあります、抜粋して見ていきましょう。



▲図 4.1: コンピュータとスマートフォン

^{*1} IT 用語辞典 <https://e-words.jp/>

4.1 ハードウェア



CPU

中央処理装置 (Central Processing Unit) とは、コンピュータの主要な構成要素の一つ。他の装置・回路の制御やデータの演算などを行う装置で、演算装置と制御装置が統合されている。

CPUはメインメモリ (RAM) に格納された機械語のプログラムを、バスを通じて一命令ずつ順番に読み出し (フェッチ)、その内容を解釈して行うべき動作を決定 (デコード) し、内部の回路を駆動して実際に処理を実行する。



メモリ

メモリとは、記憶、記憶力、回想、追憶、記念などの意味を持つ英単語。IT分野ではコンピュータに内蔵される半導体集積回路 (IC) を利用したデータの記憶装置を指す。コンピュータを構成する装置の一つで、CPUから直接読み書きができる記憶装置のことを「主記憶装置」という。

一般に主記憶装置は外部記憶装置よりはるかに高速に動作する装置が用いられるが、単価や装置構成上の制約から少ない搭載容量となっている。このため、コンピュータは起動すると外部記憶から主記憶に必要なプログラムやデータを読み込んで実行し、不要となったデータは主記憶から消去して新たに必要になったものに入れ替える。

永続的に保管する必要があるデータなどは外部記憶へ書き込まれて保存される。



ストレージ

ストレージとは、コンピュータの主要な構成要素の一つで、コンピュータが利用するプログラムやデータなどを永続的に記憶する装置。磁気ディスク（ハードディスク HDD）や光学ディスク（CD/DVD/Blu-ray）、フラッシュメモリ（USB メモリ/メモリカード/SSD）、磁気テープなどがある。

同じコンピュータに搭載される装置同士で比較すると、ストレージはメモリに比べて記憶容量が数桁（数十～数千倍）大きく、容量あたりのコストが数桁小さいが、読み書きに要する時間が数桁大きい。



マザーボード

マザーボードとは、コンピュータの主要な構成部品の一つで、マイクロプロセッサやメモリなど他の部品を装着し、通電したり相互に通信できるようにする基板のこと。

プラスチックなどでできた板状の装置で、表面や内部に各装置を結ぶ配線や制御用の半導体チップ（チップセットなど）、電子部品などが高密度に実装されている。

プロセッサやメモリモジュール、拡張カードなどを装着するためのスロットやソケットなどの接続部品、電源ユニットからのコードを差し込む電源コネクタ、ストレージ（外部記憶装置）など周辺機器を接続するためのケーブルを差し込むコネクタなども配置されている。



入力装置(キーボード)

正方形や横長の小さなボタンが縦横に整然と並び、文字や記号、コンピュータへの指示などを送信するための入力装置。

100 前後のキーが4~5段に並び、各キーの上部に入力される文字や機能が記されており、キーを押すと、そのキーが押されたという信号がコンピュータへ送信される。



出力装置(ディスプレイ)

ディスプレイとは、表示、展示、陳列、掲示、飾り付け、示すなどの意味を持つ英単語。IT分野では、コンピュータの出力装置の一つで、画面を発光させて像を映し出す表示装置(display device、ディスプレイ装置)のこと。「モニター」(monitor)とも呼ぶ。

コンピュータの操作画面を映像として電気的に映し出し、処理状況の変化や利用者の操作に即時に反応して表示内容を変化させることができる。データとして記録された動画像を再生・表示することもできる。ディスプレイ以前に主要な出力装置として利用されていたのは印字装置(プリンタ)であり、状況や操作を表示内容にリアルタイムに反映する特徴は画期的で便利な特性だった。

ディスプレイの画面は格子状に規則正しく並んだ微細な画素(ドット/ピクセル)から成り、その発光状態を電気的に制御してコンピュータから受信した映像信号を表示する。一つの画素を光の三原色に対応する微細な素子で構成し、カラー表示を行う。

4.2 ソフトウェア

コンピュータを構成する電子回路や装置などの物理的実体を「ハードウェア」と呼ぶのに対し、それ自体は形を持たないプログラムや付随するデータなどをソフトウェアという。

コンピュータを動作させる命令の集まりであるコンピュータプログラムを組み合わせ、何らかの機能や目的を果たすようまとめたもので、その役割により、ハードウェアの制御や他のソフトウェアへの基盤的な機能の提供、利用者への基本的な操作手段の提供などを行なう「オペレーティングシステム」(OS : Operating System / 基本ソフト)と、特定の個別的な機能や目的のために作られた「アプリケーションソフト」に大別される。

♣ OS (Operating System / 基本ソフト)

機器の基本的な管理や制御のための機能や、多くのソフトウェアが共通して利用する基本的な機能などを実装した、システム全体を管理するソフトウェア。

入出力装置や主記憶装置、外部記憶装置の管理、外部の別の装置やネットワークとのデータ通信の制御などが主な役割で、コンピュータに電源が投入されると最初に起動し、電源が落とされるまで動作し続ける。利用者からの指示に基いて記憶装置内に格納されたソフトウェアを起動したり終了させたりすることができる。

パソコン向けの OS としては、Microsoft 社の Windows や Apple 社の macOS や、Linux がある。スマートフォンやタブレットでは、Apple 社の iOS や、Google 社の Android OS が普及している。

♣ アプリケーションソフト

OS の機能を利用し、OS の上で動作するソフトウェアである。アプリケーションの開発者は、OS の提供する機能を利用することによって、開発の手間を省くことができ、操作性を統一することができる。また、ハードウェアの仕様の細かな違いは OS が吸収してくれるため、ある OS 向けに開発されたソフトウェアは、基本的にはその OS が動作するどんなコンピュータでも利用で

きる。

用途や目的に応じて多種多様なアプリケーションソフトがある。一例を挙げると、ワープロソフトや表計算ソフト、画像閲覧・編集ソフト、動画・音楽再生ソフト、ゲームソフト、Web ブラウザ、電子メールソフト、カレンダー・スケジュール管理ソフト、電卓ソフト、カメラ撮影ソフト、地図閲覧ソフト、プレゼンテーションソフトやデータベースソフト、財務会計ソフト、人事管理ソフト、在庫管理ソフト、プロジェクト管理ソフト、文書管理ソフトなどなど。

利用者が配布・販売パッケージ入手・購入して OS に組み込む作業「インストール」を行うことで使用可能となる。

第 5 章

コンピュータ よもやま話

さまざまなミニ知識のご紹介です。

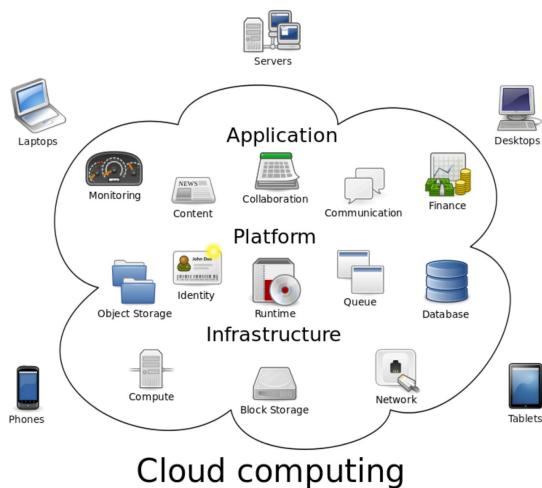
♣ インターネット

世界中のコンピュータ同士が繋がった巨大なネットワーク(網)。前身は ARPANET(アーパネット、高等研究計画局ネットワーク)。ジョゼフ・カルル・ロブネット・リックライダー等がアイディアを創り上げた。日本では、村井純氏等が普及に多大な貢献をしている。

♣ クラウド

英語で「雲」の意味。今まで自分のコンピュータで行った処理を、インターネット上のほかのコンピュータ上で行う。演算の他、情報伝達、データを保存など様々な役務が提供されている。^{*1}

^{*1} 画像出典: Wikipedia



▲図 5.1: クラウド

♣ IoT

モノのインターネットとも訳される。「インターネットに多様かつ多数の物が接続され、及びそれらの物から送信され、又はそれらの物に送信される大量の情報の円滑な流通が国民生活及び経済活動の基盤となる社会の実現」として定義されている。

♣ 量子コンピュータ

量子コンピュータは、量子力学的な重ね合わせを用いて並列性を実現するコンピュータ。従来のコンピュータ（以下「古典コンピュータ」）の基本素子は、情報量が0か1のいずれの値しか持ち得ない1ビットを扱うものであるのに対して、量子コンピュータでは量子ビット（qubit; quantum bit、キュービット）により、1キュービットにつき0と1の値を任意の割合で重ね合わせて保持する。n量子ビットあれば、2のn乗の状態を同時に計算できる。もし、数千qubitのハードウェアが実現した場合、この量子ビットを複数利用して、量子コンピュータは古典コンピュータでは実現し得ない規模の並列コン

ピューティングが実現する。理論上、現在の最速スーパーコンピュータで数千年かかっても解けないような計算でも、例えば数十秒といった短い時間でこなすことができる、とされている。

$$\Delta x \Delta p \geq \frac{\hbar}{2}$$

▲図 5.2: 不確定性原理

♣ 人工知能

人間にしかできなかつた知的な行為（認識、推論、言語運用、創造など）を、どのような手順（アルゴリズム）とどのようなデータ（事前情報や知識）を準備すれば、それを機械的に実行できるか」を研究する学問。（情報工学者・通信工学者 佐藤理史 氏）画像認識（顔を見て持ち主を判断する）、音声認識・音声合成（声を聞いて応答する）、外国の人と意思疎通が図れるように翻訳・会話する、絵を描いたり、歌を唄ったり、将棋を指すなど、多くの実用化がなされている。ニューラルネットワーク（脳機能の特性を計算機上のシミュレーションによって表現した数学モデル）でディープラーニング（深層学習）研究が進んでいる。

♣ 電子署名（デジタル署名）

電子署名とは、文書やメッセージなどのデータの真正性を証明するために付加される、短い暗号データ。作成者を証明し、改竄やすり替えが行われていないことを保証する。欧米で紙の文書に記されるサインに似た働きをするためこのように呼ばれる。^{*2}

*2 出典：IT 用語辞典

第6章

二進数の話

コンピュータは、「0」と「1」の二つの値を用いる二進数で動いています。黎明期には十進数を用いた計算機もありましたが、実行速度や作成費用などから二進法を用いるものが主流となりました。今日のコンピュータを動かす基礎となる二進数をご紹介します。

また、より深く計算機理論を学びたい方へ向けて、デジタル回路のご紹介と、富山大学幸山教授が書かれた「計算機理論入門」への参照を挙げました。

6.1 二進数と十進数

わたくしたちが日常使っている十進数では、0、1、2、…、8、9と大きくなっています、その次は、「十」の位へと繰り上がります。10、11、…、99 その次は、「百」の位へと繰り上がります。100、101、…、999 その次は、「千」の位へと繰り上がります。1000、1001…一つ上の位に上がるたびに、その桁の重みが十倍になっています。「7 6 8」という数でしたら、百が7つと、十が6つ、一が1つを併せた数のことですね。

二進数でも同様ですので、見てていきましょう。0、1と大きくなっています、その次は、「二」の位へと繰り上がります。10、11 その次は、「四」の位

へと繰り上がります。100、101、110、111 その次は、「八」の位へと繰り上がります。1000、1001・・・「1101」という数でしたら、八が1つと、四が1つと、一が1つを併せた数のことです。つまり、「13」です。

二進数では、すぐに桁数が増えていくので、四桁をひとまとめにした、十六進数もよく使われます。十六進数では、10～15までの数を一桁で表せるように記号を準備する必要があります。黎明期には様々な記号が提案されました。今日では「A」～「F」までのアルファベットの使用が定着しました。

以下に対応表を掲載いたします。

十進数と二進数、十六進数の表

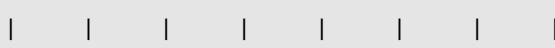
十進数	二進数	十六進数
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

【コラム】金の延棒クイズ

7日分の給料の支払いとして金の延べ棒が1本あります。仕事をしてく

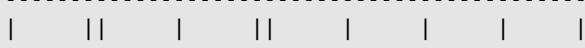
れる方へ、毎日の日当を支払いたい。6回はさみを入れて切り取れば、金の延べ棒は7等分されるので日払いできます。しかし、金の延べ棒を6回も切り取るのは大変なため、最小限の切り取り回数で済む2回にしたいと思います。どことどこを切り取ればよいでしょうか？

▼ 金の延棒



2回、はさみを入れて、1と2と4に分割してみては、いかがでしょう。

▼ 金の延棒



一日目のお支払いには、1の延棒を渡します。二日目のお支払いには、2の延棒を渡して、先に渡した1の延棒は返してもらいます。三日目のお支払いには、1の延棒も渡します。四日目のお支払いには、大きな4の延棒を渡して、2の延棒と1の延棒は返してもらいます。五日目のお支払いには、1の延棒も渡します。六日目のお支払いには、2の延棒を渡して、先に渡した1の延棒は返してもらいます。七日目のお支払いには、全ての延棒を渡します。

金の延棒と日当の支払い

金の延棒	日当
421	
001	1
010	2
011	3
100	4
101	5
110	6
111	7

二進数との対応が見られて、美しいですね。

他にもトーナメント表のシード選手の決定など、身近なところにも二進数は使われています。どのようなところに使われているか、調べてみるのも良いですね。====[/column]

6.2 単位のお話

パソコンでよく使われる単位のお話です。

- 1 ビット (bit)

コンピュータで扱うことができる最小の情報量です。スイッチが入っている (= 1) か、切れている (= 0) かなど、2通りの状態を表すことができます。二進数の1桁分の情報量です。

#@# * 10進法では、10個あつまると、次の位へ繰り上がりますが、2進法では、2つ集まると次の位へ繰り上がります。ですから、10進法の1桁では、0～9までの10通りの状態を表すことができましたが、2進法の1桁では、0～1までの2通りの状態を表すことができます。

2桁分（2ビット）では、00, 01, 10, 11 の4通りを表すことができます。

3桁分（3ビット）では、000, 001, 010, 011, 100, 101, 110, 111 の8通りを表すことができます。

- 1 バイト (Byte)

半角文字1文字分（8ビット = 2進法の8桁分）の情報量です。

つまり、00000000 ~ 11111111までの256通りの情報量です。256通りありますから、数字(0 - 9)や、アルファベット(A - Z, a - z)、記号(@, #, !など)を区別するには十分です。

大文字のアルファベット「A」には、二進数の「01000001」、十六進数の「41」を対応させます。次の表は、ASCIIコード表です。「A」に「41」が対応していますね。

文 字	10	16	文 字	10	16	文 字	10	16									
NUL	0	00	DLE	16	10	SP	32	20	0	48	30	@	64	40	P	80	50
SOH	1	01	DC1	17	11	!	33	21	1	49	31	A	65	41	Q	81	51
STX	2	02	DC2	18	12	"	34	22	2	50	32	B	66	42	R	82	52
ETX	3	03	DC3	19	13	#	35	23	3	51	33	C	67	43	S	83	53
EOT	4	04	DC4	20	14	\$	36	24	4	52	34	D	68	44	T	84	54
ENQ	5	05	NAK	21	15	%	37	25	5	53	35	E	69	45	U	85	55
ACK	6	06	SYN	22	16	&	38	26	6	54	36	F	70	46	V	86	56
BEL	7	07	ETB	23	17	'	39	27	7	55	37	G	71	47	W	87	57
BS	8	08	CAN	24	18	(40	28	8	56	38	H	72	48	X	88	58
HT	9	09	EM	25	19)	41	29	9	57	39	I	73	49	Y	89	59
LF*	10	0a	SUB	26	1a	*	42	2a	:	58	3a	J	74	4a	Z	90	5a
VT	11	0b	ESC	27	1b	+	43	2b	;	59	3b	K	75	4b	[91	5b
FF*	12	0c	FS	28	1c	,	44	2c	<	60	3c	L	76	4c	¥	92	5c
CR	13	0d	GS	29	1d	-	45	2d	=	61	3d	M	77	4d]	93	5d
SO	14	0e	RS	30	1e	.	46	2e	>	62	3e	N	78	4e	^	94	5e
SI	15	0f	US	31	1f	/	47	2f	?	63	3f	O	79	4f	_	95	5f
												o	111	6f	DEL	127	7f

▲図 6.1: ASCII コード表

- 2 バイト全角文字1文字分の情報量です。1バイトでは256通りの文字

を区別することができましたが、「ひらがな」や「カタカナ」、「漢字」などから成り立つ日本語を表記するには、不十分です。

- そこで 2 バイトにすることにより、 $256 \times 256 = 65536$ 通りの文字を区別することができるようになります。以前は、Shift-JIS(シフトjis)と呼ばれる文字体系で、日本語を表現していましたが、現在では、世界中の文字を 3 バイトで表現する UTF-8(ユーティーエフエイト) が主流です。

- 1 キロバイト (kB) = 1,024 バイト

- 2 進数 10 桁 ($= 2$ の 10 乗) で 1024 通りを表現できます。

10 進法で、1,000 倍のことを k (キロ) といいます。コンピュータの世界では、 $2 \times 2 \times 2$ を十回掛けると 1024 になるので、1024 でひとまとめにして、1024 バイトのことを 1kB と言います。

- おおよそ A4 用紙一枚程度の情報量です。

- 1 メガバイト (MB) = 1,024 キロバイト = 1,048,576 バイト 小さめの写真約 1 枚分、A4 約 1000 枚分の情報量です。昔懐かしいフロッピーディスク 1 枚は 1.44MB でした。

- 1 ギガバイト (GB) = 1,024 メガバイト = 1,048,576 キロバイト = 1,073,741,824 バイト

CD 1 枚 74 分で 640MB です。

DVD 1 枚 2 時間で、4.7GB ~ 8.5GB です。

高画質の映画などブルーレイディスクは、25GB ~ 100GB です。

- 1 テラバイト (TB) = 1,024 ギガバイト = 1,048,576 メガバイト = 1,073,741,824 キロバイト = 1,099,511,627,776 バイト

ハードディスクはとても大容量です。1TB のハードディスクで毎日 2 時間ずつ動画を見るなら、おおよそ 100 日分の容量になります。

♣ おまけ

$1,000 \text{ m} = 1 \text{ km}$, $100 \text{ a} = 1 \text{ ha}$, $1 \text{ dL} = 1 / 10 \text{ L}$, $1 \text{ cm} = 1 / 100 \text{ m}$, $1 \text{ mm} = 1 / 1,000 \text{ m}$ などがお馴染みですが、大きな数、小さな数を表すために

次の記号が規格されています。

国際単位系(SI 単位系)での接頭辞一覧

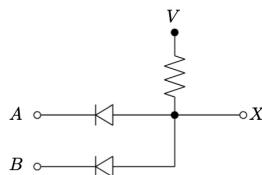
接頭辞	接頭辞	記号	乗数
ヨタ	yota	Y	10 の 24 乗=1,000,000,000,000,000,000,000,000 倍
ゼタ	zeta	Z	10 の 21 乗=1,000,000,000,000,000,000,000,000 倍
エクサ	exa	E	10 の 18 乗=1,000,000,000,000,000,000,000 倍
ペタ	peta	P	10 の 15 乗=1,000,000,000,000,000 倍
テラ	tera	T	10 の 12 乗=1,000,000,000,000 倍
ギガ	giga	G	10 の 9 乗=1,000,000,000 倍
メガ	mega	M	10 の 6 乗=1,000,000 倍
キロ	kilo	k	10 の 3 乗=1,000 倍
ヘクト	hecto	h	10 の 2 乗=100 倍
デカ	deca	da	10 の 1 乗=10 倍
			10 の 0 乗=1 倍
デシ	deci	d	10 の-1 乗=10 分の 1
センチ	centi	c	10 の-2 乗=100 分の 1
ミリ	milli	m	10 の-3 乗=1,000 分の 1
マイクロ	micro	μ	10 の-6 乗=1,000,000 分の 1
ナノ	nano	n	10 の-9 乗=1,000,000,000 分の 1
ピコ	pico	p	10 の-12 乗=1,000,000,000,000 分の 1
フェムト	femto	f	10 の-15 乗=1,000,000,000,000,000 分の 1
アト	atto	a	10 の-18 乗=1,000,000,000,000,000,000 分の 1
ゼプト	zepto	z	10 の-21 乗=1,000,000,000,000,000,000,000 分の 1
ヨクト	yocto	y	10 の-24 乗=1,000,000,000,000,000,000,000,000 分の 1

6.3 計算機理論入門

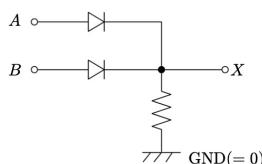
コンピュータは、0と1で動いていることはご紹介いたしました。黎明期には、真空管を用いて作られていたコンピュータですが、トランジスタ、IC、LSIと集積化が進み、小さなチップの中に、数百億ものトランジスタが実装されるようになりました。

多くのトランジスタが高速に動作することで今日のコンピュータの性能がありますが、デジタル回路を支える三つの論理素子 AND, OR, NOT 素子の回路図をご紹介いたします。 *1

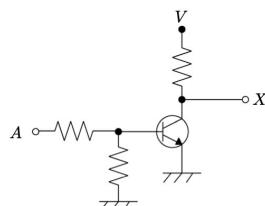
富山大学の幸山教授が書かれた「計算機理論入門」にとても良くまとまっていますので、是非ご一読なさってみてください。



▲図 6.2: AND 回路



▲図 6.3: OR 回路



▲図 6.4: NOT 回路

*1 引用：富山大学 幸山研究室 計算機理論入門 ([https://kouyama.sci.u-toyama.ac.jp
/main/etc/2003/ssh/sshi.pdf](https://kouyama.sci.u-toyama.ac.jp/main/etc/2003/ssh/sshi.pdf))

第 7 章

アルゴリズム とは

アルゴリズムとは、「計算や作業を遂行するための手順」のことです。良いプログラムを作る上で、アルゴリズムの理解は欠かせません。

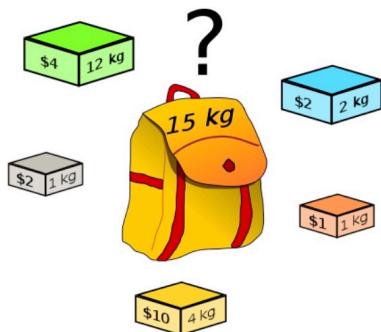
7.1 アルゴリズムとは

語源は、九世紀前半を生きたアラビアの数学者アル・フワーリズミーに因みます。日本語で「算法」と訳されることもあります。

アルゴリズムとは、「**計算や作業を遂行するための手順**」のことです。^{*1}並び替えや探索など基本的なものから、素数判定、最大公約数、円周率の計算など、数学的なもの、分類分け、データ圧縮などなど、さまざまなアルゴリズムが考案されてきました。

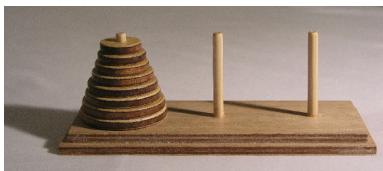
^{*1} 出典: アルゴリズム図鑑

♣ ナップザック問題 *2



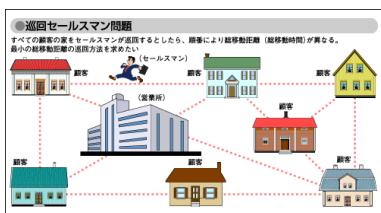
袋には 15kg まで入れられる。袋の価値を最大化するためには、どの品を詰めたら良いか？

♣ ハノイの塔 *3



パズルの一種。三本の杭と、大きさの異なる複数の円盤があり、左端の杭に小さいものが上になるように順に積み重ねられている。円盤を一回に一枚ずつどれかの杭に移動させることができるが、小さな円盤の上に大きな円盤を乗せることはできない。右端の杭に全ての円盤を移動させよ。

♣ 巡回セールスマン問題 *4



ある地域の営業担当のセールスマンが、その地域に住んでいる顧客全員の家を巡回して訪ねることになったとする。巡回の順番によって移動距離が大きく変わるので、移動距離が少なくなるような巡回の方法を求めたい。

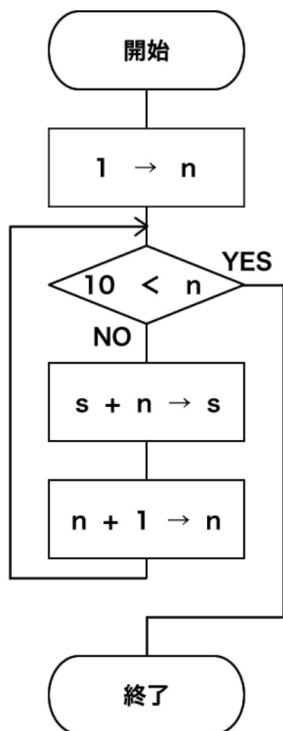
*2 出典: Wikipedia

*3 出典: Wikipedia

*4 出典: imidas

7.2

1から10までの合計を求める



ここでは、アルゴリズムの例として、1から10までの合計を求めます。左の図は「流れ図(フローチャート)」と呼ばれ、手順を示す際に良く使われます。

1. 「n」という名前の箱を用意し、1を入れます。
2. 10よりnが大きいか、条件判断します。
3. NOなら、「s」という名前の箱に、 $s + n$ を計算した答えを入れて、「n」という名前の箱に、 $n + 1$ を計算した答えを入れます。そして、2に戻ります。
4. YESなら、「終了」です。1から10までの合計が、「s」という名前の箱に入っています。

計算するための手順が「きちん」と書かれているので、確実に答えを求めることができます。

7.3

より良いアルゴリズム

先に紹介した、アルゴリズムの効率をみていきましょう。1から10までの合計を求める時の計算量(足し算の回数)は10回になります。1から100までの合計を求める時の計算量(足し算の回数)は100回になります。1から1000までの合計を求める時の計算量(足し算の回数)は1000回になります。このように、nに比例して計算量が増えることを、 $O(n)$ と書き、オーダーnと読みます。

もう少し良いアルゴリズムは無いでしょうか。

$$(1 + n) * (n / 2) \rightarrow s$$

として、求めてみるのはどうでしょう。計算量は $O(1)$ 」、 n によらず、一定の時間で計算することができ、最高に効率的です。

解きたい課題に対して、複数のアルゴリズムがあります。どれが効果的か思考することはとても大切です。

【コラム】カール・フリードリヒ・ガウス

後に数学王と呼ばれた、ガウス少年の逸話を紹介します。(出典: Wikipedia)

ガウスが7歳の時、数学の授業で教師が「1から100までの数字すべてを足せ」という問題を出した。教師は生徒たちが問題を解くには相当な時間がかかるだろうと考えていたが、ガウスはわずか数秒で「5050」という解答を出し、教師を驚かせた。

1から100までの数字を足していくと、 $1 + 100 = 101$ 、 $2 + 99 = 101$ 、…、 $50 + 51 = 101$ で、101の集まりが50個できるため、 $101 \times 50 = 5050$ になるとガウスは計算したのである。

その後もガウスは、素数定理や最小自乗法など赫々たる業績を挙げ、数学界に巨大な足跡を記しました。

- 1777年 - ドイツ、ブラウンシュヴァイクに生まれる
- 1792年 - 素数定理の成立を予想
- 1795年 - 最小二乗法発見
- 1796年 - 平方剰余の相互法則の証明。コンパスと定規のみで正十七角形を作図できることを証明
- 1799年 - 代数学の基本定理の証明
- 1801年 - 『整数論の研究』出版 複素数表記、現代整数の表記導入
- 1801年 - 円周等分多項式の研究

- 1807 年 - ゲッティンゲンの天文台長になり、以後 40 年同職につく
- 1809 年 - 『天体運行論』出版 最小二乗法を用いたデータ補正、正規分布
- 1811 年 - 複素積分、ガウス平面（複素数平面）ベッセルへの手紙
- 1827 年 - 『曲面の研究』出版、微分幾何学を創始
- 1855 年 - ゲッティンゲンで死去

第8章

プログラムとは

プログラムとは何か？ その語源を訪ね、そして今日使われているいくつかのプログラミング言語をご紹介します。そして学校教育でのいくつかの事例をご紹介します。

8.1 プログラムとは何でしょうか？

今日の「コンピュータ」＝「電子計算機」が登場する以前から「プログラム」という言葉は使られてきました。語源は、「前に書かれたもの」です。街頭で大勢に示す文書、「公文書」の意味です。

アルゴリズムとは、ある特定の課題への解法、解き方の手順を書き表したものでしたが、このアルゴリズムを、コンピュータが計算できるよう、コンピュータが理解できる言語で書き表したもの、それが「プログラム」です。「プログラム」＝「コンピュータへの指示書」です。

その昔のコンピュータは、特定の用途の計算のみを行うことができました。（専用計算機）黎明期のコンピュータは、回路と回路の間の配線を繋ぎ代える

ことで、異なる種類の計算も行うことができるようになりましたが、たくさんの配線を繋ぎ代えることはとても大変です。そのために生まれたアイディアが、「計算機への指令」 = 「プログラム」そのものを計算機に内蔵するというアイディアです。これにより、様々な手順を必要なときに読み込んで書かれた通りの計算を行うことができるようになりました。これが、現在ほとんどのコンピュータで広く採用されている「プログラム内蔵方式」 = 「ノイマン・アーキテクチャ」です。

また、シャノン等によりスイッチのオンとオフの回路（スイッチング回路）から、論理回路・デジタル回路への道が拓かれたことや、チューリングによるチューリングマシンの研究、19世紀を生きた数学者であるジョージ・ブールの仕事、ブール代数も、今日のコンピュータの存在に大きく貢献しています。

プログラムは、3つの要素で成り立ちます。

- 順次処理：上から下へ順番に進みます。
- 条件判断：YES か NO で答えられる質問です。
- 繰り返し：何回でも反復します。

8.2 プログラミング言語の種類

人間がコンピュータに指示するために作られた言語が、プログラミング言語です。用途に応じたさまざまなプログラミング言語が考案されてきました。ここではそのいくつかをご紹介いたします。

♣ 機械語とアセンブリ言語

18	CLC
F8	SED
A9 34 12	LDA #\$1234
69 21 43	ADC #\$4321
8F 03 7F 01	STA \$017F03
D8	CLD
E2 30	SEP #\$30
00	BRK

機械語は、「0」と「1」の二進数や「0」-「F」までの十六進数を書き連ねたものです。コンピュータはこれを読み取つて直接解釈し実行することができる命令データそのものの集まりです。

アセンブリ言語は、コンピュータが直接解釈実行可能な機械語を、人間にわかりやすいよう英略語を用いて書き表した言語です。

左側の機械語は分かりにくいが、右側のアセンブリ言語は、CLC:比較せよ, LDA:読み込め, ADC:足せ, STA:書き込め, BRK:中断 など、人に理解しやすくなっている。

♣ C 言語

```
int n;
int answer = 0;
for(n = 1; n <= 10; n++){
    answer += n;
}
printf("%d\n", answer);
```

1 から 10 までの合計を求める

略して C(シー)ともいいます。比較的人に読みやすい文法を持ちながら、機械に近いところまで書き表すことができる所以、小さなコンピュータ（マイコン）が組み込まれた電子機器のプログラミングや、UNIX（ユニックス）という OS（オペレーティングシステム）の開発などに広く用いられています。

♣ Ruby(ルビー)

```
answer = 0
(1..10).each do |n|
    answer += n
end
print answer
```

1から10までの合計を求める

まつもとゆきひろさん(通称 Matz)により開発されたオブジェクト指向スクリプト言語です。日本で開発されたプログラミング言語として初めて国際電気標準会議で国際規格に認証されました。開発者のまつもとゆきひろさんは、「Rubyの言語仕様策定において最も重視しているのはストレスなくプログラミングを楽しむことである(enjoy programming)」と述べています。

♣ Scratch(スクラッチ)



1から10までの合計を求める

MITメディアラボが開発した視覚的なプログラミング言語です。初めてプログラミングをする小学生が、遊び心のある実験やアニメーション、ゲームなどを作ったりすることができます。

8.3 学校教育でのプログラミング

昨今、小学校からのプログラミング教育が行われるようになりました。その狙いや概要を文部科学省より引用します。

♣ なぜプログラミング教育を導入するのか

今日、コンピュータは人々の生活の様々な場面で活用されています。家電や自動車をはじめ身近なもの多くにもコンピュータが内蔵され、人々の生活を便利で豊かなものにしています。誰にとっても、職業生活をはじめ、学校で

の学習や生涯学習、家庭生活や余暇生活など、あらゆる活動において、コンピュータなどの情報機器やサービスとそれによってもたらされる情報とを適切に選択・活用して問題を解決していくことが不可欠な社会が到来しつつあります。

コンピュータをより適切、効果的に活用していくためには、その仕組みを知ることが重要です。コンピュータは人が命令を与えることによって動作します。端的に言えば、この命令が「プログラム」であり、命令を与えることが「プログラミング」です。プログラミングによって、コンピュータ自分が求める動作をさせることができるとともに、コンピュータの仕組みの一端をうかがい知ることできるので、コンピュータが「魔法の箱ではなくなり、より主体的に活用することにつながります。

プログラミング教育は子供たちの可能性を広げることにもつながります。プログラミングの能力を開花させ、創造力を發揮して、起業する若者や特許を取得する子供も現れています。子供が秘めている可能性を発掘し、将来の社会で活躍できるきっかけとなることも期待できるのです。

このように、コンピュータを理解し上手に活用していく力を身に付けることは、あらゆる活動においてコンピュータ等を活用することが求められるこれからの中を生きていく子供たちにとって、将来どのような職業に就くとしても、極めて重要なこととなっています。^{*1}

♣ 小学校でのプログラミング教育のねらい

1. 「プログラミング的思考」を育むこと。
2. プログラムの働きや良さ、情報社会がコンピュータ等の情報技術によって支えられていることに気付き、コンピュータを上手に活用して身近な問題を解決したり、よりよい社会を築く態度を育むこと。
3. 各教科の内容を指導する中で実施する場合には、各教科で学びをより確実なものとすること。(プログラミングに取り組むことで、プログラミング技能を習得することは考え得るが、それ自体を目的とはしない。)

小学校においては、文字入力など基本的な操作を習得、新たにプログラミン

^{*1} 出典：文部科学省：小学校プログラミング教育の手引（第三版）

グ的思考を育成各教科等の特質に応じて、学習の基盤として必要となる情報手段の基本的な操作を習得するための学習活動や、プログラミングを体験しながらコンピュータに意図した処理を行わせるために必要な論理的思考力を身に付けるための学習活動を計画的に実施する。

♣ 中学校でのプログラミング教育のねらい

中学校においては、技術・家庭科（技術分野）においてプログラミング、情報セキュリティに関する内容を充実「計測・制御のプログラミング」に加え、「ネットワークを利用した双方向性のあるコンテンツのプログラミング」等について学ぶ。

♣ 高等学校でのプログラミング教育のねらい

高等学校においては、情報科において共通必履修科目「情報I」を新設し、全ての生徒がプログラミングのほか、ネットワーク（情報セキュリティを含む）やデータベースの基礎等について学習「情報I」に加え、選択科目「情報II」を開設。「情報I」において培った基礎の上に、情報システムや多様なデータを適切かつ効果的に活用し、あるいはコンテンツを創造する力を育成。【情報科】

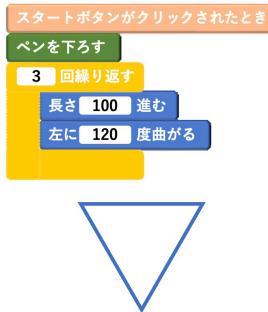
*2

♣ プログラミング的思考とは

「自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか」を論理的に考えていく力。

*2 出典：文部科学省 新学習指導要領のポイント（情報活用能力の育成・ＩＣＴ活用）

♣ 「正多角形を描く」場合について考える



▲ 図 8.1: 正三角形を描く

コンピュータで正三角形を描く場合を見てみます。「正三角形を描く」という命令は通常は用意されていませんので、そのままでは実行できません。そこで、コンピュータが理解できる(用意されている)命令を組み合わせて、コンピュータに命令することを考えます。

紙の上に作図する場合、正多角形がもっている「辺の長さが全て等しい」、「角の大きさが全て等しい」、「円に内接する」、「中心角の大きさが全て等しい」のような正多角形の意味や性質を使って作図します。

コンピュータで作図する場合にも同じことを考えます。ここでは、「辺の長さが全て等しく、角の大きさが全て等しい」という正多角形の意味を使って作図する場合を考えてみます。

この場合、「長さ 100 進む(線を引く)」、「左に 120 度曲がる」といったコンピュータが理解できる(用意されている)命令を組み合わせることで「正三角形を描く」ことができます。もっと大きな正三角形を描きたければ、「長さ 100 進む(線を引く)」を、「長さ 200 進む(線を引く)」というように修正します。曲がる角度を変えることで、正六角形や正八角形も描くことができます。

紙の上に鉛筆と定規、分度器やコンパス等を用いて正三角形を描くときも、用いる性質や手順そのものは異なるとしても、児童は同じように手順を考えた上で作図しているはずです。

♣ 実践例の紹介

未来の学びコンソーシアム*3



地域の魅力を発信しよう



スポーツとデータ分析。地域スポーツプログラミングで動くワークチームを応援しよう



自動車に搭載された技術と私たちの生活を便利にするプログラム



正多角形をプログラムを使ってかっこいいかえしをつかってリズムをつく



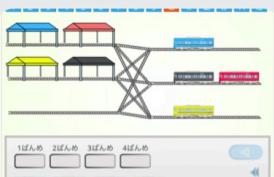
くりかえしをつかってリズムをつく



ブロックを組み合わせて47都道府県を見つけよう



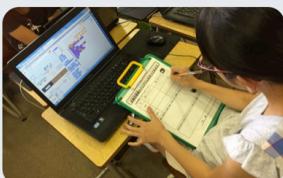
世界に発信！地域密着プログラミング学習による新潟市PRプロジェクト



プログラムを通して順番を学ぼう



主語と述語に気を付けながら場面に合ったことばを使おう



家族と食べる朝食を考えよう



ブロックを組み合わせて47都道府県を見つけよう

*3 <https://miraino-manabi.jp>

付録 A

珠玉の名著のご紹介

次の段階へ進むために読んで欲しい、名著をご紹介いたします。手にして読み、自らの血肉として頂ければ幸いです。^{*1}

A.1

HTML / CSS / JavaScript を学ぶ

♣ CSS グリッドで作る HTML5 & CSS3 レッスンブック

難易度 ★★☆



本書は CSS グリッドを基礎にした Web ページ制作を行うための解説書です。CSS グリッドを基礎になると、Web ページ制作がシンプルになります。サンプルを作りながら一歩一歩着実に学習することにより、モバイルファーストで本格的なレスポンシブに対応した実践的な Web 制作に関する知識がひと通り得られます。

- これから HTML5 & CSS3 を使った Web サイトの構築を学びたい人
- 最新の CSS グリッドに関する知識を得たいと考えている人に最適の一冊です。

^{*1} 紹介文は書籍紹介からの引用・改変です。

♣ JavaScript Primer 迷わないための入門書

難易度 ★★★



これから JavaScript を学びたい人が、ECMAScript 2015 以降をベースにして一から JavaScript を学べる書籍です。

この書籍は、JavaScript の仕様に対して真剣に向き合って書かれています。入門書であるからといって、極端に省略して不正確な内容を紹介することは避けています。そのため、JavaScript の熟練者であっても、この書籍を読むことで発見があるはずです。

♣ 動く Web デザインアイディア帳

難易度 ★★☆



「サイトに動きをつける時、同じ内容を何回も検索をしてソースコードを探している」「サイトで見つけたソースコードをそのままコピペしてみたけど、動かない」「JavaScript の本を購入してみたが、実際のサイトにどう組みこめばいいか具体的なイメージがわからない」「動きの原理を最低限理解して、とにかく早く実務にいかしたい」

本書は Web サイトを動かすことが苦手な右脳系ウェブデザイナーが、サイトを動かす第 1 歩を踏み出すための「動きの逆引き事典」です。近年のウェブサイトで使用されている基本的な動きの原理や仕組みをサンプルコードと共に紹介します。

A.2

コンピュータサイエンスの基礎を学ぶ

基礎編では軽く触れるだけでしたが、ぜひコンピュータサイエンスの世界をより深く学んでください。

♣ みんなのコンピュータサイエンス

難易度 ★☆☆



もはやコンピュータなしには生活が立ち行かなくなるレベルにまで到達しつつある現代社会。その圧倒的なパワーを問題解決に援用するためには小手先の知識だけでは追いつきません。かといって行き当たりばつたりで、全方位に知識を求めるには、その世界は広大にすぎますし、効率が悪すぎます。

本書はコンピュータサイエンスが扱う「基礎」「効率」「戦略」「データ」「アルゴリズム」「データベース」「コンピュータ」「プログラミング」という8つのジャンルにしづり、そのエッセンスと背景となる考え方を紹介します。

どのジャンルであれ、トップクラスのエンジニアを目指すにはコンピュータサイエンスが不可欠ですが、「どこから手を付ければいいのかわからない」「砂を噛むような分厚い理論書は敬遠したい」というステップアップしたいエンジニアやその予備軍、あるいは現役だけれどもう少しライトに全体像を俯瞰したい学生にも最適な1冊です。

♣ プログラマの数学

難易度 ★☆☆



プログラミングに役立つ「数学的な考え方」を身につけよう。

プログラミングや数学に関心のある読者を対象に、プログラミング上達に役立つ「数学の考え方」をわかりやすく解説しています。数学的な知識を前提とせず、たくさんの図とパズルを通して、平易な文章で解き明かしています。

2進数から人工知能に至るまで、ていねいに説明しています。

プログラミングや数学に関心のある読者はいうまでもなく、プログラミング初心者や数学の苦手な人にとっても最良の一冊です。

♣ 教養としてのコンピューターサイエンス講義

難易度 ★☆☆



デジタル時代で活躍するための「教養」をこの1冊で身につけよう!

プリンストン大学の一般人向け「コンピューターサイエンス」の講義が一冊に。デジタル社会をよりよく生きるために知識を伝説の計算機科学者がやさしく伝えます。(著者ブライアン・カーニハン氏は、プログラミング言語 C の発明者です)

本書は、わたくしたちの世界(デジタル社会)が、どのように動いているのか、なぜそのしくみになっているのかをもっとも明快かつ簡潔に説明しています。

♣ キタミ式イラスト IT塾 IT パスポート

難易度 ★☆☆

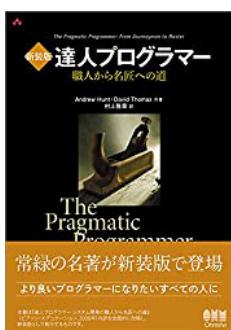


可愛いイラストでとてもわかりやすい解説を行っているため、IT パスポート試験にとって、まず大切な「解説書を一冊読み、用語や計算に慣れる」ことができる書籍です。

A.3 プログラマとして上達したい方のために

♣ 新装版 達人プログラマー 職人から名匠への道

難易度 ★★★



本書は、より生産的になりたいソフトウェア開発者に向け、アジャイルソフトウェア開発手法の先駆者として知られる二人により執筆されました。経験を積み、生産性を高め、ソフトウェア開発の全体をより良く理解するための、実践的なアプローチが解説されています。

先見性と普遍性に富んだ本書は、入門者には手引きとなり、ベテランでも読み直すたびに得るものがある、座右の一冊となるでしょう。

♣ コーディングを支える技術

難易度 ★★☆



本書は、プログラミング言語が持つ各種概念が「なぜ」存在するのかを解説する書籍です。世の中にはたくさんのプログラミング言語があります。そしてプログラミングに関する概念も、関数、型、スコープ、クラス、継承など、さまざまなものがあります。多くの言語で共通して使われる概念もあれば、一部の言語でしか使われない概念もあります。これらの概念は、なぜ生まれたのでしょうか。本書のテーマは、その「なぜ」を理解することです。

そのために本書では、言語設計者の視点に立ち、複数の言語を比較し、そして言語がどう変化してきたのかを解説します。いろいろな概念が「なぜ」生まれたのかを理解することで、なぜ使うべきか、いつ使うべきか、どう使うべきかを判断できるようになるでしょう。そして、今後生まれてくる新しい概念も、よりいつそう理解しやすくなることでしょう。

A.4 アルゴリズムを学ぶ

優れたプログラムを書くためには、アルゴリズムの理解も欠かせません。

♣ アルゴリズム図鑑 絵で見てわかる 26 のアルゴリズム

難易度 ★☆☆



基本的な 26 のアルゴリズム +7 つのデータ構造をすべてイラストで解説。アルゴリズムはどんな言語でプログラムを書くにしても不可欠ですが、現場で教わることはめったになく、かといって自分で学ぶには難しいものです。

本書は、アルゴリズムを独学する人のために作りました。はじめて学ぶときにはイメージしやすく、復習するときには思い出しやすくなるよう、基本的な 26 のアルゴリズム +7 つのデータ構造をすべてイラストで解説しています。

よいプログラムを書くために知っておかなきやいけないアルゴリズムの世界を、楽しく学びましょう。

♣ C 言語による標準アルゴリズム事典

難易度 ★★☆



コンピュータの算法に関するアルゴリズムの定石、レトリックを可能な限り収録した定番の書。手元に置いておきたい実用的な本が 30 年弱の時を経て新装改訂版として登場です。定評をいただいている基本的な内容はそのままに、時代にそぐわなくなっていた部分のみ改訂。これからも末長くご愛顧いただけるようにまとめ直しました。

A.5 Ruby を学ぶ

分かりやすい記法と優れた書き味で人気のプログラミング言語 Ruby のご

紹介です。

♣ 初めてのプログラミング 第2版

難易度 ★☆☆



初めてプログラミングを学ぶ入門者を対象に、プログラミングの基礎をていねいに解説した書籍。

教材には、誰でもどんな環境でも気軽に使える Ruby を使い、実際に簡単なコードを書きながら理解を深めます。プログラミングとは何かを無理なく理解してもらうために、要点をひとつひとつていねいに解説。簡単な概念から始めて、かなり高度なプログラミングの知識まで身に付けることができます。

プログラミングを学ぶなら、本書は最初の1冊に最適な入門書です。

♣ たのしい Ruby 第6版

難易度 ★☆☆



本書は、今までプログラミングをしたことがない、という方でも Ruby の使いこなし方の一端がつかめるように、ていねいな解説を行っています。プログラムに必要な変数・定数・メソッド・クラス・制御構造といった文法的な説明から、主なクラスの使い方と簡単な応用まで、できるだけわかりやすく説明することを心がけました。

チュートリアル、基礎、クラス、実践とテーマを切り分けて、平易に解説。Ruby の基礎から応用までがわかる一冊。

♣ プロを目指す人のための Ruby 入門

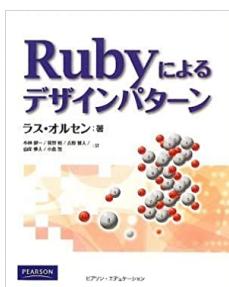
難易度 ★★☆



Ruby の文法をサンプルコードで学び、例題でプログラミングの流れを体験できる解説書です。ほかのプログラミング言語で開発経験のある人が、Ruby を学ぶ際に効率的に学べる内容を詰め込みました。プログラミング未経験者向けの「変数とは」「配列とは」といったプログラミング基礎知識の説明は最小限にし、そのぶん Ruby の特徴（他プログラミング言語との違い）、Ruby におけるリファクタリングの要点、テスト駆動開発やデバッグのやり方など開発現場で必要になる知識を解説しています。本書の内容を理解すれば、開発の現場で必要とされる Ruby 関連の知識を一通り習得できます。そして、「今まで呪文のようにしか見えなかつた不思議な構文」や「実はあまりよくわからないまま、見よう見まねで書いているコード」も自信をもって読み書きできるようになるはずです。

♣ Ruby によるデザインパターン

難易度 ★★★



スクリプト言語 Ruby の持つ力強さ・エレガントさ・シンプルさを、Gof をはじめ従来の代表的なデザインパターンと結合させ、少ないコードでより高度な効果的ソフトウェアを記述する方法を解説します。Metaprogramming や Rails-based Convention Over Configuration patterns などといった Ruby コミュニティから出てきた新しい革新的なパターンも紹介し、Ruby のプログラミング経験をより深く価値のあるものにしてくれる 1 冊です。

終わりに

基礎編では、コンピュータの歴史を学び、仕組みを理解しました。

実践編では、HTML / CSS / JavaScript を学び、ウェブアプリを作成いたしました。

全部で、108行のじやんけんプログラム、要所要所にコメントも付けていますので、今まで学んできた知識で読解できるはずです。ぜひ、遊んでみてください。自分で作ったプログラムの体験はいかがでしょうか？いろいろ創意工夫して、さまざまなアプリを作つていけそうですね。

「福祉」。「福」「祉^{*2}」どちらも「めぐみ、さいわい」という意味を持ちます。
「熱き心、^{たくま}逞^{かしいな}しき腕、冷静な頭脳」

学生時代に言われた言葉ですが、福祉を生きる者は、人としての熱い思い、暖かい心を持ち、その上で、冷静な判断力を以て、力強く行動するのだと。

「工学」の「工」は、「天の 理^{ことわり}を、地に下ろす」意味です。

技術の産物としての社会ではなく、世界を^{かがや}耀^{かがや}かせるために技術を用いてください。技術に使われるのではなく、技術を使いこなし、人の道に役立てる人となつてください。

令和の御世を生きる皆さんのが素晴らしい人生を生き、素晴らしい日本を創ることを願つて筆を置きます。

いやさか
彌榮

*2 「祉い」と書いて、「さいわい」と読みます。天からの恵みがその身に止まる意味です。

計算機の歴史と仕組み

暮らしに生きるコンピュータ

令和三年七月七日 ver 1.0.0

著 者 アトリエ未来

発行者 藤谷 真琴

連絡先 contact@atelier-mirai.net

<https://atelier-mirai.net>

© 令和三年 アトリエ未来