

◆書籍情報

タイトル：『中学生のためのプログラミング講座』

サブタイトル：なし

キャッチコピー：絶対に挫折しないはじめの一步

◆企画概要

幅広く使われているプログラミング言語である C 言語の解説と、その前提として必要となるコンピュータの仕組みについての解説書。小学校高学年から中学生の読者が理解できることを目的とする。コンピュータとプログラミングの基礎知識習得ののち、最終的には小規模プログラムが自力で書けるようになることを目標とし、最終課題として簡単な電卓を作成する。

◆企画者プロフィール

中学二年生の時から C 言語の学習をはじめ、同時に C 言語によるゲーム制作を開始。高校卒業までに二本の PC 向けシューティングゲームを文化祭で発表し、二本目は学校法人岩崎学園・神奈川新聞社主催の「第 28 回ソフトウェア・アイデア・コンテスト」にて岩崎学園賞を受賞。

大学は東京大学文科二類に進学後、進学振り分けで工学部電子情報工学科へ理転し本格的にコンピュータの学習を始める。趣味でのゲーム制作も継続し、2016 年 8 月現在までに PC 向けゲームを 7 本制作。

◆読者ターゲット

対象読者層としては、小学校高学年から中学生・高校生・大学生・社会人でプログラミングを初めて学ぶ人に幅広く利用されるような内容とする。ただし、タイトルの通り主なターゲットは中学生であり、特に今後始まるとされるプログラミング教育において参考書的に利用されることを期待する。

◆企画のねらい

プログラミングの基礎となる前提知識と概念を習得し、本書でプログラミングの簡単な体験をすることで、読者の更なる学習における強固な土台を作ることを目的とする。

具体的には、コンピュータの基礎となる考え方を理解したのち、プログラミングの根底となる概念を C 言語を通じて理解し、最終的に電卓という実用的なプログラムの作成を体

験することにより今までの学習に対する一区切りとしての達成感をもって別の書籍での学習に進むことができる。

◆企画の背景

情報化社会の発展によりプログラミングそれ自体への関心が高まっているとともに、義務教育におけるプログラミング教育の導入が決定し、学生およびそれを教える教師もプログラミング技術の学習が必要不可欠となる。そのためプログラミング知識の解説に対する需要は高まっており、その中でも特にプログラミングが持つ意味から理解することを目的とした書籍がその需要により良く応えられるものであると考えられる。

このように基礎からのしっかりとした学習を掲げる参考書籍は数多くある物の、基本的に従来のプログラミング教育が高校卒業程度の知識を持った学習者に対して行われていたため、現在存在する書籍もそれを前提としていることが多い。そのため中学・高校生向けのプログラミング学習書籍は需要に対して供給がほぼ空白となっており、本企画の書籍が発行されればその空白を補うことができると考えられる。

◆類書との差別化

ほぼ一切の前提知識を要求しない点と、徹底的に挫折しないことにこだわるという二点を類書との差別化の中核とする。以下、出版されている具体的な書籍との比較を行う。

『C の絵本』

大学のプログラミングの授業で参考書として挙げられていた書籍。イラストがふんだんに使われており、本企画の書籍と同様プログラミングの前提知識も解説される。

本企画書籍と異なるのは白黒印刷であることに加え、前提知識とプログラミング技術の解説が分けられていない点、および最終的に高度な内容まで扱われている点である。前者は知識獲得の流れを複雑化してしまう可能性、後者は初心者が初読で最後まで理解する難易度が上がり挫折感を感じてしまう可能性がそれぞれある。

『12 歳からはじめるゼロからの C 言語 ゲームプログラミング教室』

確認できる限り二十冊以上の十代前半を対象とした「ゲームプログラミング」シリーズが出版されており、市場の広さが伺える。その中で本企画と最も内容が似通っているものが本書である。

しかし、内容は読者にプログラミングに関する基礎知識があることが前提とされており、用語解説が少なく高度な内容まで扱うことが本企画の書籍とは異なっている。類似点としてはフルカラーでイラストが挿入されており手に取りやすくなっている点が挙げられるも

の、ソースコードは白黒でありこの点からも上級者向けであることが伺える。

『これからはじめるプログラミング基礎の基礎』

本企画書第一部から第二部冒頭部分までの類書に該当。導入→ハード→ソフト→C 言語の順に解説される。白黒ながら的確なイラストと高い文章力により内容は非常にわかりやすい。

扱う内容は本企画の書籍と似通っているが、プログラムを書きはじめるとは目的ではない点が大きく異なる。この書籍は「プログラミングに関する基礎知識の理解」が目的であり、本企画の書籍は「実際に簡単なプログラムが書けるようになること」が目的である。

◆企画者の要望

レイアウトとしては、図表およびイラストを多く挿入するものとする。

また現場においてはプログラムのソースコードは色付けレイアウトされていることがほとんどであるのに対し、一般的な解説書は白黒になっているために必要以上に難しそうに見える。そのためこの本ではフルカラーでソースコードの実践的な色付けを行う。同様に、実際のソースコードと同じように随所にコメントを挿入しソースコードで流れを読む際の補助とする。

読者がプログラミングを始めるまでの手間を省くために、環境構築を行うことのできるインストール用 CD を付属させられればなお良い。

◆構成案

はじめに

第 1 部 コンピュータとプログラムのしくみ

1-0 第 1 部の目標

1-1 コンピュータの本体

1-2 コンピュータの内側——CPU / メインメモリ / ハードディスク

1-3 コンピュータの外側——画面 / マウス・キーボード / CD ・ USB メモリ ・ SD カード

1-4 プログラムはどこで動いているのか

1-5 プログラミング言語とその種類——機械語 / アセンブラ / BASIC / C 言語

1-6 プログラミング言語から機械語へ——インタプリタ / コンパイラ

1-7 プログラミングの準備①：コンパイラのインストールと設定

1-8 プログラミングの準備②：エディタのインストールと設定

第 2 部 C 言語によるプログラミング

第 12 回出版甲子園

- 2-0 第 2 部の目標
- 2-1 一番簡単なプログラム
- 2-2 画面に文字を表示する——putchar / printf(1)
- 2-3 数字を保存する変数——int / float
- 2-4 計算のための記号
- 2-5 画面に数字を表示する——printf(2)
- 2-6 条件によって処理を変える——if
- 2-7 繰り返し処理を行う——for
- 2-8 キーボードからの入力を読み込む——scanf
- 2-9 最終課題：電卓プログラム

あとがき

◆見本原稿

「2-1 一番簡単なプログラム」より

では、第 1 部を経てプログラムを書き始めるための準備は整いました。ここから実際に C 言語のプログラムを書き始めてみることにしましょう。

まずは以下のソースコードを書き写してコンパイルして実行してみてください。

```
int main()           //ここから処理が始まる
{
    return 0;       //プログラム終了
}
```

実行すると黒いウィンドウ、すなわち**コマンドプロンプト**が表示されたとおもいます。それこそがこのプログラムの内容……ではなく、コマンドプロンプトの表示はコンパイラとエディタによってなされたものです。では、このプログラム自体は何の意味を持つかと言うと、実は何の意味も持っていません。ただ実行されるだけのプログラムで、中身には実行する内容が何も入っていないと考えてください。このプログラムは「一番簡単なプログラム」「一番小さなプログラム」などと呼ばれます。しかし、最も小さく単純であるからこそ、このプログラムから C 言語の最も基本的かつ重要なルールを学ぶことができます。それでは、くわしく見ていくことにしましょう。

まず、一行目の `int main()` と、その後ろについている `{`、そして最後の行の `}` に注目してください。 `int main()` の後に書かれた `{` は一つの大きなまとまりを作ります。これを **main 関数** と呼びます。C 言語のプログラムは、この **main 関数** のまとまりの一番上の行から実行さ

れることとなります。横書きの文章は一番上の行から読み始めるんだからそりゃ当たり前じゃないかと思うかもしれませんが、事はそう単純ではありません。次のプログラムを見てください。今は特に意味は分からなくても大丈夫です。近いうちにまたちゃんと出てきます。

```
#include <stdio.h>

int main()
{
    printf("あーあー、ただいまマイクのテスト中");
    return 0;
}
```

このプログラムの一行目は見ての通り「`#include <stdio.h>`」ですが、プログラムの実行はこの部分を見捨てて `main` 関数、正確にはその中の一行目である `printf` と書かれているところから始まるということになります。では一行目の「`#include <stdio.h>`」は？ と思うかもしれませんが、ここでコンパイラと実行ファイルの関係を思い出してください。

人間が書いたプログラムのソースコードはコンパイラがコンパイル、すなわち変換用のプログラムがコンピュータで実行可能な `exe` ファイルに変換します。ここで、上のソースコードのうち、コンパイラは確かに一行目の `#` から読み始めるのですが、ソースコードを `exe` ファイルに変換するとき、`main` 関数内部の一番初めの行である `printf` のところから始まる形で `exe` ファイルを作ります。よって、コンピュータが実行するときは `main` 関数の `{` の中の一番上の行から実行されるということになります。

(以下略)

※ソースコード・コンパイル・コマンドプロンプトなどの用語は既にこれ以前の章で解説済みのものとする。

※ソースコードの色付けはあくまで一例であり、実際の書籍において解説するエディタに応じて決定する。