プログラミング入門
#1 入出力 変数 繰り返し 配列

高校 3 年 赤澤侑

2021/05/12

1　こんにちは、世界！

みなさんこんにちは[1]、高校 3 年の赤澤です。本シリーズ「プログラミング入門」では中学 1 年生に向けて C++ によるプログラムの書き方を説明していきます。厳密さよりも寧ろ、わかりやすさと、幅広い話題に触れ、皆さんの興味のある分野を発掘することの 2 点に重きを置きました。かなり発展的な内容を含みますが、全てを理解する必要はありません。楽しいプログラミングライフを送りましょう！

さて、人間にとって挨拶は重要です。本稿のはじめで私も挨拶をしたとおり、初対面の方には必ず挨拶をする必要があります[2]。それではまずはコンピュータに挨拶をしてみましょう！以下のソースコードを実行してください[3]。

Source Code (1): Hello, World!

```cpp
#include <iostream>
using namespace std;

int main(){
    cout << "Hello, World！" << endl;
    return 0;
}
```

このコードを実行してみると次のような出力が得られます。

Input/OutPut (1): Hello, World! - **実行結果**

```
Hello, World！
```

これは `Hello, World!` という文字列を出力するプログラムです[4]。

では、プログラムの内容を簡単に説明しましょう。早速、1 行目から解説……といきたいところですが、以下に記す部分は「おまじない」[5]だと思ってあまり意味を考えずに記述するようにしてください。最初のプログラムの解説としては少し難しいからです。

---

[1] ここでは世間一般に通じている挨拶を用いましたが、マイコン部では挨拶として「グッドモルﾆｯﾋ！ﾆﾋﾋﾋﾋ！」と言うことが慣習となっています。

[2] プログラムの書き方なんかよりよっぽど重要なことです。挨拶、ダイジ。

[3] 私の tex 環境上、半角の exclamation mark をソースコード上に出力できませんでした。実際にコードを書く際には半角で入力するほうが好ましいです。

[4] Hello, World というのは初学者が最初に書くコードとして非常に有名なものです。C 言語を解説した伝説的名著である、「プログラミング言語 C」という書籍が初出とされています。

[5] よくわからないけどこれを書いておかないとプログラムが動かないもの、という意味です。

Source Code (2): **プログラムの雛形**

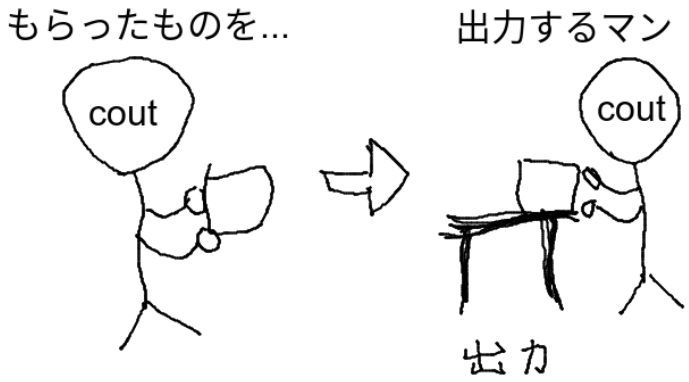
```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     //処理内容
6     return 0;
7 }
```

`//処理内容`と書いてある部分にプログラムの内容を記述します。この雛形についてはコラムや後の章で詳しく説明しますが、今はそういうものだと思っておいてください。しかし、次のことは極めて重要なので覚えておいてください。すなわち、プログラムは`int main()`と書いてある次の行からスタートし、`int main()`の後にある`{}`[6]の中に記述された内容が実行されます。

では処理内容を詳しく見ていきましょう。処理内容は次のようになっています。

```
1 cout << "Hello, World!" << endl;
```

まず、`cout`という部分に注目します。これはよく「シーアウト」と発音されるものでなにかを出力するときに使います[7]。ではこの cout は何者か、といえば、「もらったものを順番に出力するマン」ということができるでしょう。



この出力するマンに出力してほしい物を渡すための演算子[8]こそが`<<`です。このソースコードでは cout に"Hello, World!"と endl を順番に渡しているのです。先程説明したとおり、cout は渡されたものを順番に出力してくれるのでソースコードのように<<をいくつもつなげることができます。たとえば、処理内容を次のように書き換えてもソースコード 1 と同じ出力が得られます。

```
1 cout << "Hello" << ", " << "World" << "!" << endl;
```

ここで、出力したい文字は`""`[9]によって囲まれていることに気づくと思います。このように、出力したい文

[6] 開きカッコ`{`は 4 行目の最後、閉じカッコ`}`は 7 行目にあります。

[7] cout は console out の略とされます。console とはコンピュータとやり取りするための入出力装置 (キーボードとかディスプレイとか) のことを指しますが、ここではコマンドライン (文字がいっぱい出ているかっこいいウィンドウのやつ) を意味します。環境によっては入力、出力などと書かれていることもあります。

[8] 挿入子といいます。

[9] "はダブルクオーテーションと読みます。似た文字に’がありますが、こちらはシングルクオーテーションあるいはシングルクオー

字列は`""`でくくってやらねばなりません。試しに""を外してみると、エラーが出てしまうでしょう。

では一番最後の`endl`とは何でしょうか？これは改行を意味するものです。これを cout に渡すことによって改行することができます。プログラム中での改行は C++ においては動作に影響を与えない[10]ので注意しましょう。また、endl の部分を`"\n"`とすることもできます。すなわち、"\n"を改行文字といい、\n 自体が改行を意味する特殊な文字[11]です。以下のソースコードが最初のものと同じ出力を与えます。

```
1 cout << "Hello, Wolrd!\n";
```

最後に、5 行目の一番最後の文字である、`;`[12]についてです。これは文の終わりを意味する文字です。ちょうど、日本語において文の終わりに「。」を、英語において文の終わりに “.”を置くのと同じように C++ においては; を置く必要があります。

以上で第 1 章第 1 節は終了です。出力はプログラムの基本なのでしっかりとおさえましょう。しかし、慣れていけば自然と書けるようになるので、無理に「覚えよう！」と頑張る必要はありません。気楽にやりましょう。

2 変数

次のような場合を考えてみましょう。

> **Problem 1 : 配送手数料**
>
> マイコン通販社では商品を配送する際に商品代金に追加で配送手数料 300 円をお客さんに払ってもらっている。商品代金が X のとき、お客さんが払う金額を求めるプログラムを作成せよ。

計算自体は簡単です。ある料金 X に 300 を足してあげればいいのです。これを $X + 300$ と表現します[13]。ではこれをどのようにプログラム上で表現すればよいのでしょうか？実は C++ には変数という様々な値を入れておく領域と、プログラムの実行時に入力を受け取るための機能が備わっています。

2.1 変数と変数型

まずは変数について説明します。変数とは**何かしらの情報を保持しておくメモリ上の領域**のことです。では一つずつ詳しく説明します。はじめに、何かしらの情報とはなにを指すか。それは数字であったり文字列であったり画像であったり音声であったり様々です[14]。ここでは変数が保持する情報を数字である、と仮定して話を進めていきます。

トと読みます。一般には引用を意味する記号です。また英語におけるアポストロフィも同じ記号を用いて表現しますが、意味は違うので注意しましょう。

[10] これは、意味のある語 (int、main、cout、endl など) の途中を除いてプログラム中では任意の場所で改行をして良い、ということを意味します。見やすくなるように適宜改行しましょう。一方で、改行がプログラムの動作に影響を与える言語もあります。Python などがその一例です。厳密には Python においてはインデントが動作に影響を与えますが、こういった言語では改行の仕方も言語仕様に従いましょう。

[11] 制御文字という。

[12] セミコロンと読みます。コロン:と間違えないようにしてください。

[13] このような文字を含んだ式のことを文字式といいます。詳しくは代数の教科書を参照してください。

[14] コンピュータ上ではこれらは全て数字に置き換えて扱うので究極的には「変数は数字を保持している」ということができるでしょう。ただし、人間にとってこれらのデータを数字に置き換えることは極めて煩雑な処理なので人間がそれを意識しなくて済むように様々な型の変数があるのです。