

プログラミング概論

第4週

本日の内容

- C言語における「変数」
- 演習

数学における変数

- C言語の「変数」の前に数学での変数を考える
 - $x = 2$ と変数 x の値が決まれば
 - $2x + 3 = 11$ のように式の値が決まる

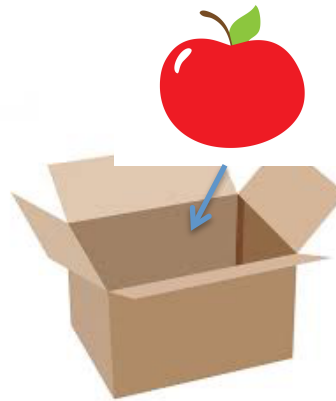
C言語における変数

- 変数 = 「何かを入れておく箱のようなもの」
– では「箱」とはなにか？

現実の箱：



リンゴを入れる
ための箱を**作る**



リンゴを**入れる**



リンゴを**取り出す**

C言語における変数

- 変数 = 「何かを入れておく箱のようなもの」

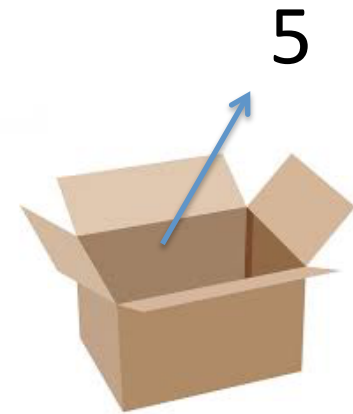
C言語の変数:



1. 整数を入れる
ための変数を**作る**
(= 変数定義)



2. 整数を**入れる**
(= 値の代入)



3. 整数を**取り出す**
(= 値の参照)

C言語での変数定義

- 変数を作ることをC言語では「変数を定義する」(変数定義)という

```
int x;
```

=



整数を入れる
xという名前の変数を**作る**

intをfloatに変えると「小数」を入れる変数
intをcharに変えると「文字」を入れる変数

int/float/charなど「何を入れるか」指定する部分を「**型**」という

C言語での値の代入

- 変数になにかを入れることを「変数に代入する」という

```
x = 5;
```

=



整数5を変数xに代入

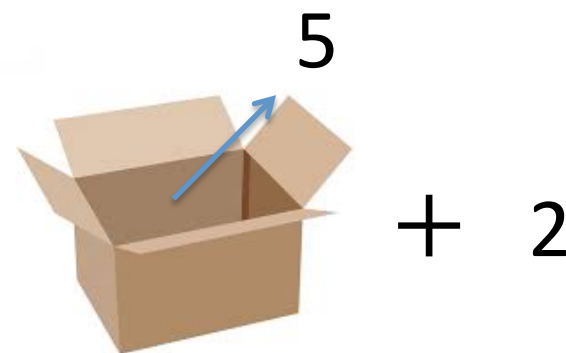
変数の「型」がintの場合は必ず整数を代入
変数の「型」がfloatの場合は必ず小数を代入
変数の「型」がcharの場合は必ず文字を代入

C言語での値の参照

- 変数に代入された値を取り出すことを「変数を参照する」という

```
printf("%d\n", x + 2);
```

=



整数5を取り出し、
2を加算して表示

2つ以上の変数を使う

1. 変数を2つ同時に定義(カンマで区切る)

```
int x, y;
```

2. 2つの変数に整数を代入

```
x = 1;  
y = x + 3;
```

3. 2つの変数の値をprintf()で表示

```
printf("xの値: %d\n", x);  
printf("yの値: %d\n", y);
```

演習0

- 端末を開き以下のコマンドで「week4」というディレクトリを作る
 - `$ mkdir week4`
- week4ディレクトリに移動する
 - `$ cd week4`

演習1

- 以下のコマンドを入力し空のC言語プログラムを作成
 - \$ gedit exercise1.c &
- テキストエディタが起動するので以下のプログラムを入力、コンパイル・実行し動作を確認

```
int main(void)
{
    int x, y;

    x = 15;
    y = 32;
    printf("%d\n", (x + y) / 2);

    return (0);
}
```

演習2

- 3人の身長の平均値を表示するプログラムを作成しなさい

– ヒント

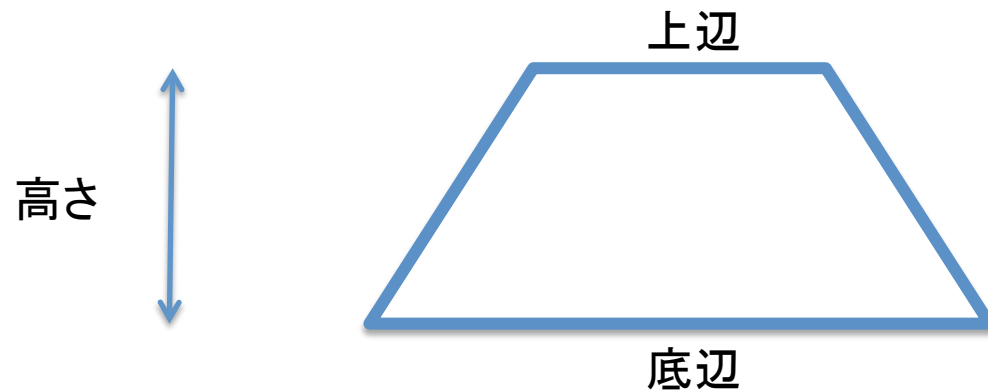
- average.cという空のファイルをgeditコマンドで作成
- 4つの変数(person1, person2, person3, average)を定義
- 変数の型はすべてfloatにする
- person1, person2, person3には身長を代入
- averageには3つの値の平均値を代入
- printf(“身長の平均:%f\n”, average)を用いて結果を表示

演習3

- 台形の面積を求めるプログラムを作成しなさい

ーヒント

- 台形の面積はの公式は
面積 = (上辺+底辺) × 高さ ÷ 2



補足：端末で使えるコマンド

- 新しい(空の)プログラムの作成
 - \$ gedit ファイル名 &
- コンパイル
 - \$ cc ファイル名
- 実行(コンパイルがうまく行った場合のみ)
 - \$./a.out
- ディレクトリの移動
 - \$ cd ディレクトリ名