プログラミング概論
講義資料を各自DL
http://bit.ly/kosen01

プログラミング概論

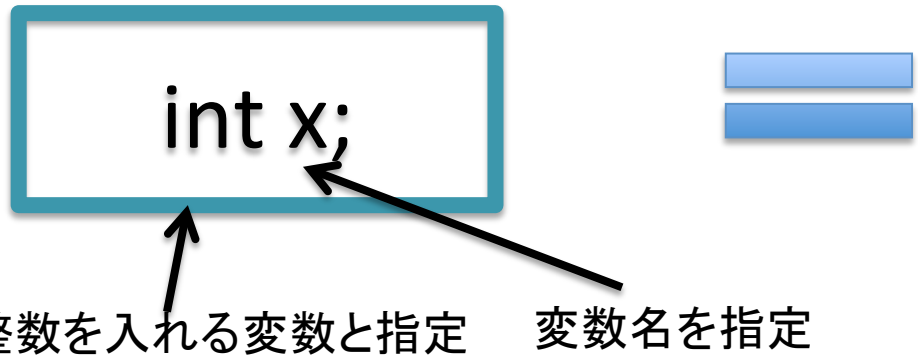
第5週

本日の講義

- 文字型の変数
  - 文字型の定義、代入、参照
- キーボードからの入力
- “コメント”の使い方

復習：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



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/floatなど「何を入れるか」指定する部分を「型」という

文字型(変数定義)

char x;

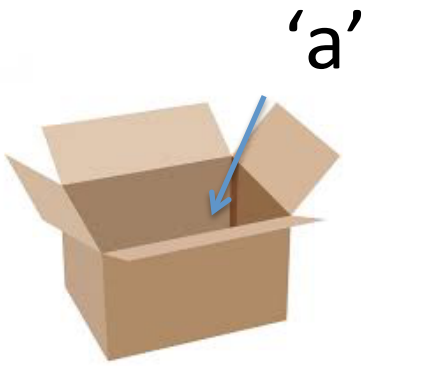
文字型charを
指定

変数名xを指定

文字を入れる
xという名前の変数を作る

文字型（代入）

x = ‘a’;



文字aを変数xに代入

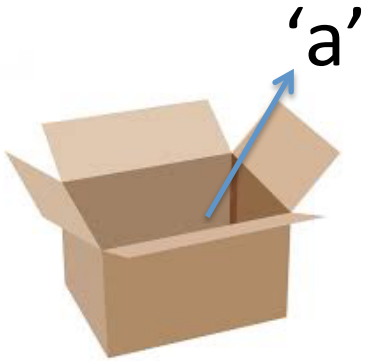
C言語では文字は必ずシングルクオーテーションでくくる決まりになっている。

文字型(参照)

printf(“%c\n”, x);

文字の表示には%cを使う
整数の表示には%d
小数の表示には%f

変数名xを指定



変数xから文字aを取り出して
printf関数で表示

サンプルプログラム（文字型）

```
#include <stdio.h>

int main(void) {
    char x;

    x = ‘a’;
    printf(“%c\n”, x);
    return(0)
}
```

char型変数で文字列を扱う

- char型の変数には1文字しか代入できない
- 複数の文字（文字列）を扱うには変数名の後に[最大の文字数]を追加する

```
char x[60];
```

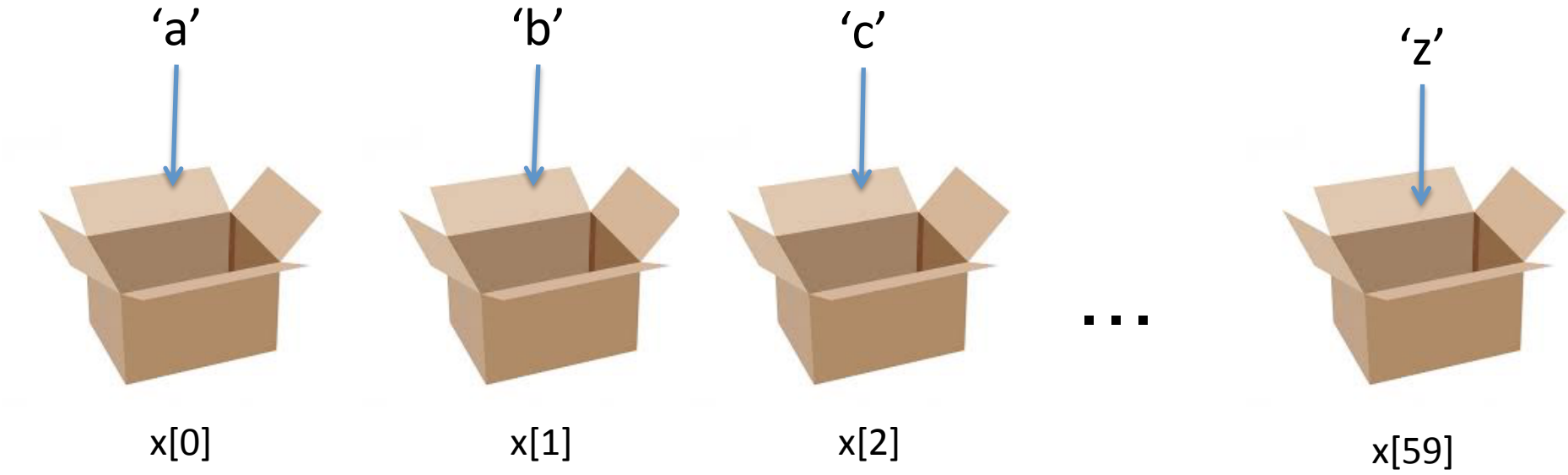
char型の変数が60個作られる
（＝配列という仕組みですが、後期に詳しく学びます）
60個の変数にはx[0]からx[59]のように番号が付く

本日の講義

- 文字型の変数
  - 文字型の定義、代入、参照
- キーボードからの入力
- 型変換
- “コメント”の使い方

scanf()関数

- キーボードからの入力にはscanf()関数を使う
- scanf()関数はキーボードから入力された文字を1文字ずつ変数に格納していく



scanf()関数の使い方

- scanf()関数
  - キーボードから変数に文字列を読み込む
- scanf()関数の使い方

```
char x[60];
scanf("%s", x);
```

②%sで文字列を読み込みを指定

①まずは変数（配列）を用意
char型の変数が60個生成

③xで読み込んだ文字列を代入する
変数を指定

printf()関数での文字列の表示

- printf()関数で文字列を表示するには%sを使う

```
printf(“%s”, x);
```

サンプルプログラム

```
#include <stdio.h>

int main(void) {
    char x[256];

    scanf(“%s”, x);
    printf(“%s\n”, x);
    return(0)
}
```

C言語のコメント

- C言語プログラムに“注釈”を入れる機能
  - 動作の説明などをコメントとして書いておく
- コンパイル時にはコメントは無視される

コメントの記述例

```
#include <stdio.h>
/*
コメントはこのように書きます
*/
int main(void) {
    printf(“サンプルプログラム”)
    return(0) // 1行の場合はこのように書きます
}
```

C言語でのコメント

```
#include <stdio.h>
/*
コメントはこのように書きます
*/
int main(void) {
    printf(“サンプルプログラム”)
    return(0) // 1行の場合はこのように書きます
}
```

コメントの書き方

1. /*と*/の間にコメントを書く
   - 複数行のコメントが書ける

2. //のあとにコメントを書く
   - コメントが1行の場合はこちらも使える

なぜコメントが必要か

理由：プログラムが読みやすくなる

- 自分で書いたプログラムであっても、数日経つとどのような動作か記憶が曖昧になる
  → コメントを入れると分かりやすくなる
- チームで開発する場合には他の開発者がプログラムを理解しやすいようにコメントを入れてあげる

コメントの例

```
/*
    年齢を表示するプログラム
    作成日：2016年5月
*/
#include <stdio.h>

int main(void) {
    int age; // 年齢を格納する変数
    age = 10; // 年齢を代入
    printf(“年齢は%d歳です。\n”, age); // printfで表示
    return(0) // プログラムを終了
}
```

演習0

- 端末を起動し、mkdirコマンドを用いて「week5」というディレクトリを作成しなさい
- lsコマンドを用いてweek5ディレクトリが正しく作成できたことを確認しなさい
- cdコマンドを用いてweek5ディレクトリに移動しなさい
  （演習1以降はweek5ディレクトリ内で行うこと）
- 前回、前々回の講義資料を見てCプログラムのコンパイル、実行の方法を確認しなさい

演習1(文字型)

- 以下のサンプルプログラムの動作を execise1.cというファイル名で作成し動作を確認しなさい

```
#include <stdio.h>

int main(void) {
    char x;

    x = ‘a’;
    printf(“%c\n”, x);
    return(0)
}
```

演習2(cpコマンド)

- ファイルを複製するcpというコマンドがあります。端末で以下を入力しexecise1.cをexecise2.cというファイルに複製しなさい

  $ cp execise1.c execise2.c

- lsコマンドでexecise2.cが作成されたことを確認しなさい。

演習4（キーボードからの文字列読み込み）

- キーボードから文字列を読み込み以下の例のように表示するプログラムexecise2.cを作成しなさい

- 動作例）
  文字を入力してください：
  xxxxxxxxxxx
  あなたの入力した文字列は：xxxxxxxxxxxです。

演習5（コメント）

- execise1.cとexecise2.cの1行目に名前、
  2行目に学籍番号をコメントとして記述しなさい

- コメントを記述したり、除去してもプログラムの
  動作に変化がないことを確認しなさい

演習6（時間が余った人向け）

- キーボードから名前と年齢を入力すると、「現在の年齢」を表示するプログラム(age.c)を作成しなさい

- 動作例：
  名前を入力してください：TARO
  年齢を入力してください：10
  TAROさんの年齢は10歳です。

次回

- 次回はキーボードから数字を入力し、
  数値計算をするプログラムを作成します
- 次々回は演習回になります。
  - 次回までの演習が終わっていない人は演習回に進めましょう
  - これまでの演習が早く終わってしまった人向けには少し難しい課題を用意します