

プログラミング概論

Week6

本日の講義

- 変数の型と型変換
- scanf()関数で数値を扱う方法
- 演習

復習：変数の型

- C言語の変数では”型”を指定する必要あり

```
int x;
```

整数型

(=整数を代入することの出来る変数)

```
float x;
```

(浮動)小数型

(=整数を代入することの出来る変数)

```
char x;
```

文字型

(=1文字を代入することの出来る変数)

復習：文字と数値

- C言語では文字、小数、整数を区別する
 - 文字はシングルクォーテーション(')でくくる
 - 数値はくくらない
 - 小数は小数点以下まで記述する
 - 整数は小数点以下を記述しない



```
int x;  
x = 'a';
```



```
char c;  
c = 10.0;
```



```
float x;  
x = 'a';
```

数値は文字型の変数に代入できない
文字型の変数に数値は代入できない

暗黙的型変換

- 小数を整数型の変数に代入するとどうなる？

```
int x;  
x = 10.5;  
printf("%d", x);
```

暗黙的型変換

- 小数を整数型の変数に代入するとどうなる？

```
int x;  
x = 10.5;  
printf("%d", x);
```

コンパイルできるが
この行で警告(Warining)が
出る



10と表示される。
切り捨てられてる
(小数から整数への暗黙的型変換という)

暗黙的型変換

- 整数同士の割り算を小数型の変数に代入するとどうなる？

```
float x;  
x = 3 / 2;  
printf("%f", x);
```



1.000と表示される。
3/2の計算の段階で小数点以下が切り捨てられてる

明示的型変換(キャスト)

- 整数同士の割り算を小数型の変数に代入するとどうなる？

```
float x;  
x = (float) 3 / 2;  
printf("%f", x);
```



1.5000と表示される。

3/2の前に(float)を付けると小数点以下が切り捨てられず
計算結果が小数に変換される

型変換-まとめ-

- 暗黙的型変換
 - 小数を整数型変数に代入した場合
→ 小数点以下が切り捨て
 - 整数を小数型変数に代入した場合
→ 同等の小数に変換
- 明示的型変換
 - 通常、整数/整数の演算を行うと小数点以下が切り捨てられる
 - (float) 整数/整数と記述すると計算結果が小数に変換(キャスト)される

本日の講義

- 変数の型と型変換
- scanf()関数で数値を扱う方法
- 演習

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

③ xで読み込んだ文字列を代入する
変数を指定

scanf()関数での整数読み込み

- %dや%fを用いると数値の読み込みが可能

```
int x;  
scanf("%d", &x);
```

② %dで整数を読み込みを指定

① まずは整数型変数を定義

③ xで読み込んだ文字列を代入する
変数を指定。数値の場合は必ず&を付ける

scanf()関数での数値読み込み

- %dや%fを用いると数値の読み込みが可能

```
float x;  
scanf("%f", &x);
```

② %fで小数を読み込みを指定

① まずは小数型変数を定義

③ xで読み込んだ文字列を代入する
変数を指定。数値の場合は必ず&を付ける

本日の講義

- 変数の型と型変換
- scanf()関数で数値を扱う方法
- 演習

演習0(作業ディレクトリ作成)

- 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

演習1(端末の自動補完機能)

- 端末にはTabキーを押すとファイル名を自動で補完してくれる機能があります

\$ cd we ここまで入力したらTabキーを2回押す

\$ cd week weekまで自動で入力される

\$ cd week6 ここまで入力してエンター

(week6ディレクトリに移動する)

- 自動補完機能を積極的に使うことで作業スピードがアップします。
- 自動で補完する機能の使い方が分ければ演習2に進む

演習2(過去のコマンド履歴)

- 端末には過去に入力したコマンドを再度入力する機能があります

\$ (キーボードの↑キーを押す)

\$ (もう一度押す)

\$ (もう一度押す)

- ↑キーを積極的に使うことでコンパイル/実行などが素早く行えます
- 過去に入力したコマンドが自動で入力されることを確認できたら演習3へ進む

演習3 (C言語プログラム作成)

- scanf_practice.cというファイル名で以下のC言語プログラムを作成しなさい

```
/* 年齢Bot  
   プログラミング概論第6回  
   作成者:(自分の名前)  
*/  
  
#include<stdio.h>  
  
int main(void) {  
    printf("年齢Bot\n");  
    printf("-----\n");  
}
```

ー ヒント: 以下のコマンドでエディタが立ち上がります。
\$ gedit scanf_practice.c &

コンパイル/実行

- (忘れてしまった人向け)
- scanf_practice.cのコンパイル/実行は以下のコマンドで行います

```
$ cc scanf_practice.c コンパイル  
$ ./a.out 実行
```

演習4 (printf関数)

- scanf_practice.cに「あなたの名前は何ですか？」と表示する記述を追加しなさい

動作例:

\$./a.out

あなたの名前は何ですか？

演習5 (scanf()関数)

- scanf_practice.cに以下の機能を追加しなさい
 - 「名前は何ですか？」と表示した後、scanf()関数を使ってキーボード名前を入力させる
 - 入力が完了したら「ようこそ、○○さん！」と表示する
- 動作例(下線部はキーボード入力を表す))
名前は何ですか？
高専太郎
ようこそ、高専太郎さん！

演習6 (scanfの数値入力)

- scanf_practice.cに以下の機能を追加しなさい
 - 「ようこそ、〇〇さん！」と表示した後に、「歳はいくつですか？」と表示する
 - scanf()関数で年齢を整数で入力させる
 - 入力が完了したら、10年後の年齢を計算し、「10年後は〇〇歳ですね！」と表示する
- 動作例(スライドでは前半部分は省略))
ようこそ、高専太郎さん！
歳はいくつですか？
15
10年後は25歳ですね！

演習7(早く終わった人向け)

- 10人の年齢をscanf()関数で受け取り、10人の平均年齢を実数で出力するプログラム(10average_age.c)を作成しなさい
- 動作例)
1人目の年齢を入力してください
20
2人目の年齢を入力してください
...(省略。10人繰り返す)
10人の平均身長は(計算結果)です。

次回

- 次回は演習回になります
 - 変数、printf()、scanf()を使った模擬試験を用意します
 - 今週までの演習が終わっていない人は来週進めてもokです。
 - 定期試験の決まりごとについて説明するので必ず出席してください