

第1回 C言語勉強会

変数・標準入出力・条件分岐

ArkOak

生物機能科学課程 4年 丸山翔平

本日の勉強会の流れ

1. Hello, world!

2. 変数

3. 条件分岐

4. まとめと演習



1 . Hello, world!

1 . Hello, world!

```
#include <stdio.h>

int main() {
    printf( "Hello, world!¥n" );

    return 0;
}
```



2. 変数

2 - 1 . 変数とは

●変数

“扱われるデータを一定期間記憶し必要な時に利用できるようにするために、データに固有の名前を与えたもの”

by Wikipedia

つまり、データを格納しておく**入れ物**のようなもの
変数を使うための厳格な**ルール**がある

2-2. 変数使用のルール

- 変数を使用するには3つの手順が必要

宣言

名前付け

`int x`

代入

データの格納

`X = 10`

参照

変数を使う

`X + 5`
`printf(" %d¥n", x)`

```
int age = 20;
```

```
printf("彼は%d歳です.¥n", age);
```

宣言と同時に行う代入を初期化というよ



変数の型

```
int age = 20;  
printf(“彼は%d歳です.\n”, age);
```

- int の部分を変数の型という
変数の性質を表す
データが表す情報を定義づけるもの.

勉強会で使う主な変数の型

型名	データの性質
int	整数値
double	浮動小数値
char	文字

変数とprintf関数

●%dとはなんなの？

```
printf( “彼は%d歳です.\n” , age );
```

```
>>彼は20歳です
```

```
printf( “彼はage歳です.\n” );
```

だと...

```
>>彼はage歳です
```

変数を別を書くことで文字列と区別している
%dのようなものを**フォーマット指定子**という

```
printf( “彼は%d月%d日生まれの%d歳で、血液型は%c型です.\n” , month, day, age, bloodtype );
```

printf()関数で呼び出す変数はいくつも参照できる

フォーマット指定子

- 変数のフォーマット（＝書式）を決める
- printf()関数など，書式付の関数で使用する
 - ～～f()と関数にf（＝format）がついているもの
scanf, fprintf, fscanfなど

型名	データの性質	書式
int	整数値	%d
double	浮動小数値	%lf
char	文字	%c (%s)

2-2. 変数使用のルール

●変数名

- 変数に格納されているデータが分かるようにつける
- 変数の名前として使えないものもある

●変数名のルール

- 半角英数字・アンダースコア(_)のみ使用可
- 大文字小文字は区別される
- 先頭は数字で始めてはならない
- 予約語はつかえない

人にわかりにくい変数名

x, y, z
a1_1, v1234

機械が認めない変数名

a-45¥, !#\$%
2015, 5pts
int, void

予約語

- C言語がすでに意味を持たせているワード
 - `int`, `char`などの変数の型名
 - `if`(条件分岐文), `for`(繰り返し文) など, C言語の文法に関する語
 - `return`など, プログラムの制御に関する語
- 予約語をどうしても使いたいときは…
 - `FOR`, `IF`, `Return`など, 大文字と組み合わせて使う
 - `returnscore`など, 文字をくっつける

2 - 3 . 定数

- 定数は、基本的に値が変わることのないデータを格納しておく入れ物

```
#define score 100
```

こんな風に定義する
文末にセミコロン ; は不要

```
#include <stdio.h>
#define score 100

int main() {
    printf( “スコアは%dです. ¥n”, score);

    return 0;
}
```

参照の仕方は変数と同じ



2 - 4 . 実践的な使い方

- 入力を受け取って変数に格納
→ 表示がprintf()なら, 入力はscanf()

```
scanf( “入力する書式” , &格納する変数);
```

数字の入力を受け付ける

```
int x = 0;  
scanf(“%d”, &x);  
  
printf(“xは%dです¥n”, x);
```

文字 (文字列) の入力を受け付ける

```
char s[256];  
scanf(“%s”, &s);  
  
printf(“sは%sです¥n”, s);
```

演習

●演習 2－1：変数を表示する

- 1) 21787, 3.4, -24, Aを適当な型の変数に代入し,
適切に表示せよ
- 2) 3.141592653を定数PIに格納し,
少数第4位まで表示させよ

●演習 2－2：変数を使った計算

- 1) 半径5の円の面積を求めよ

●演習 2－3：入力した数値による計算

- 1) 自分のBMIを求めてみよう
- 2) 為替換算プログラム
円をドル, ユーロなどへ変換

為替レートは	
米ドル/円	: 118.60円
ユーロ/円	: 131.49円
英ポンド/円	: 182.75円

出力例

●演習 2－1：変数を表示する

- 1) 21787, 3.4, -24, Aを適当な型の変数に代入し,
適切に表示せよ
- 2) 3.141592653を定数PIに格納し,
少数第4位まで表示させよ

●演習 2－2：変数を使った計算

- 1) 半径5の円の面積を求めよ

```
21787
3.4
-24
A
```

```
3.1415
```

ヒント

変数の型は, 整数・少数・文字

2) は, 表示するときの書式で桁数を決める
2－2 (1) は, 前に定義した定数PIを使う

出力例

●演習 2 - 3 : 入力した数値による計算

- 1) 自分のBMIを求めてみよう
- 2) 為替換算プログラム

円をドル, ユーロなどへ変換

為替レートは
米ドル/円 : 118.60円
ユーロ/円 : 131.49円
英ポンド/円 : 182.75円

```
>>あなたの体重は？
```

```
65
```

```
>>身長は？
```

```
1.71
```

```
>>あなたのBMIは22.22です
```

```
>>変換したい値段を入力
```

```
100
```

```
>>100円は, 1.186ドルです
```

```
>>$1.186, €1.3149, £1.8275
```

ヒント

入力を受け付けるデータは変数で,
プログラム中で変化しない値は定数で
定義するのがよい.



3. 条件分岐

3. 条件分岐

- さらにプログラムらしい処理をさせたい
- 場合分けをして、出力結果を変える

たとえば、

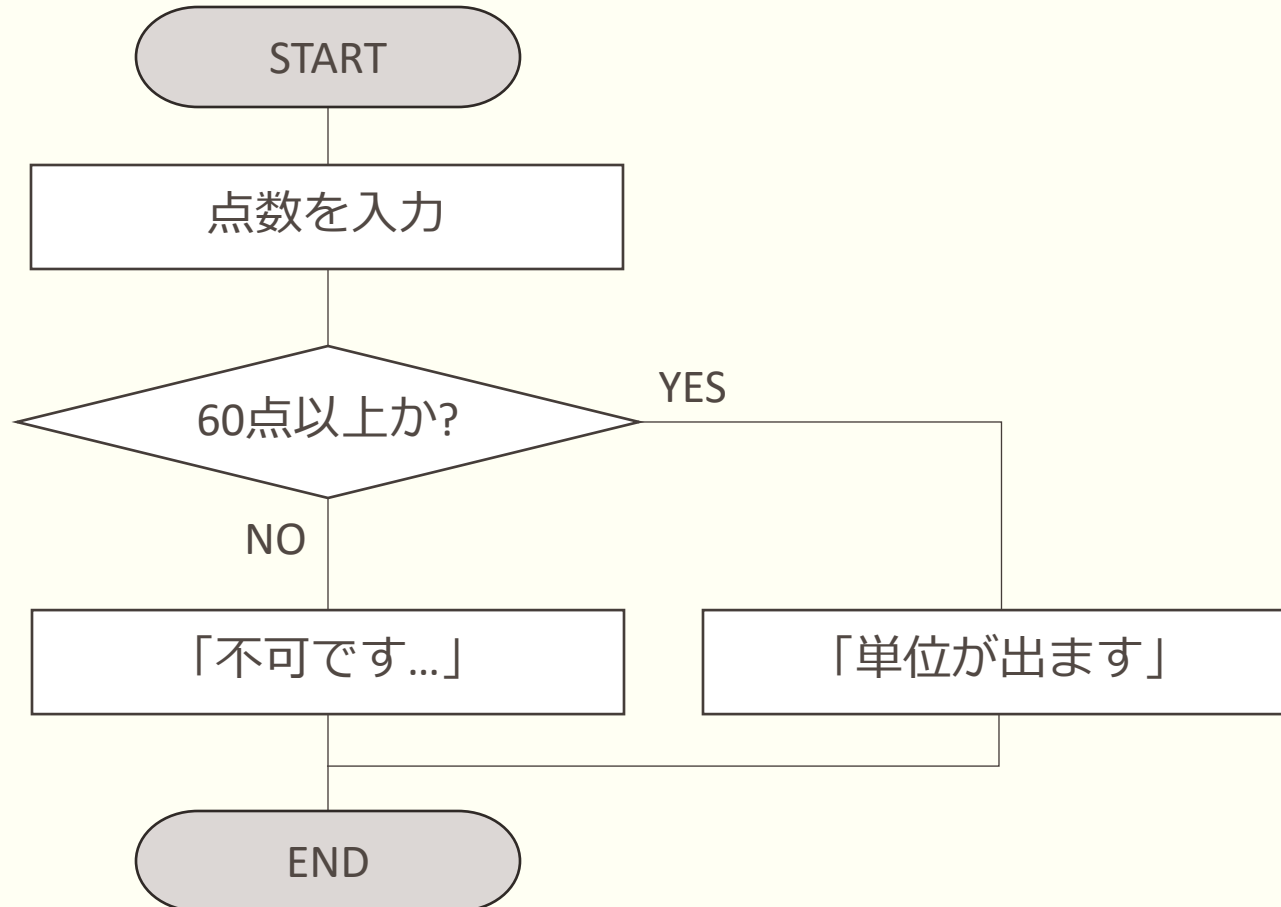
```
>>あなたの誕生日は？  
5月22日だよ  
>>ではあなたはふたご座ですね！
```

→入力された値をもとに計算して、異なる結果を出力させる。

3 - 1 . 条件分岐を表す

●C言語の文法の前に…

「フローチャート」



3 - 1 . 条件分岐を表す

- これをC言語の文法で表すと…

```
#include <stdio.h>

int main() {
    int score = 0;

    printf("点数を入力\n");
    scanf("%d", &score);

    if(score >= 60)
        printf("単位が出ます\n");
    else
        printf("不可です...\n");

    return 0;
}
```

3 - 2 . 条件分岐 —— if文

```
if(条件式) {  
    真のときの処理  
} else {  
    偽のときの処理  
}
```

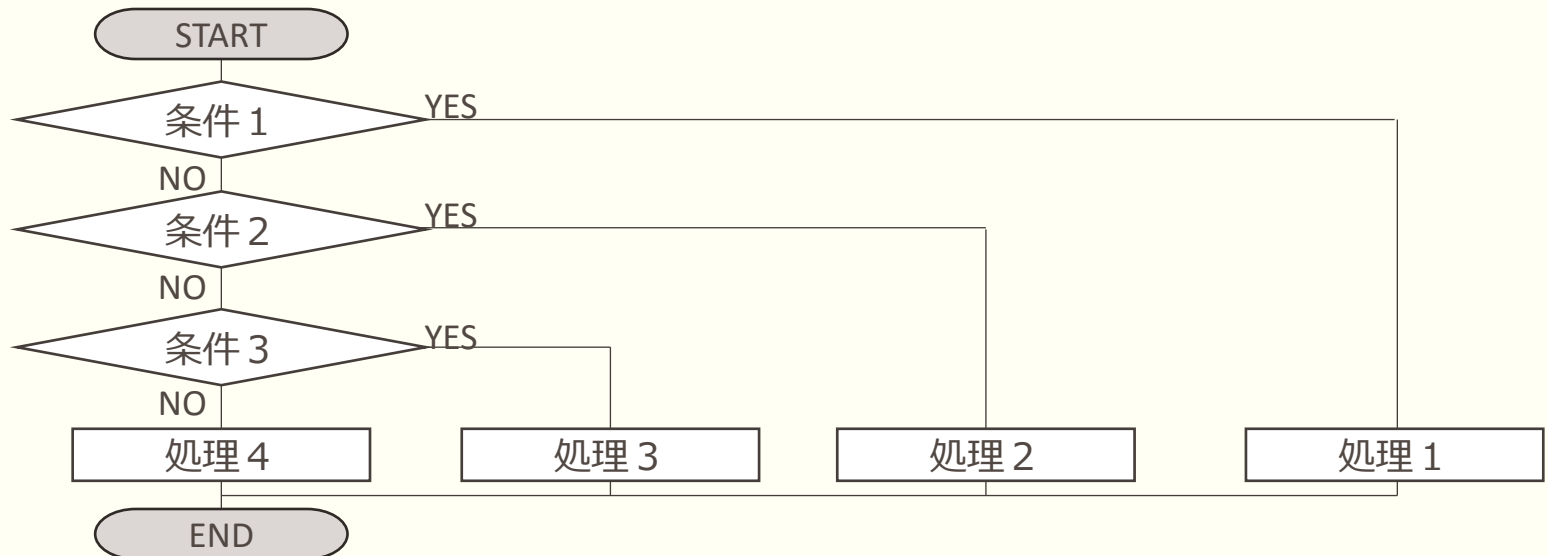
一行だけなら大括弧は省略できますよ



3 - 2 . 条件分岐 —— if文

● 選択肢を複数用意したいときは…

```
if( 条件 1 ){  
    条件1が真のときの処理  
}  
else if( 条件 2 ){  
    条件 1 は偽だが, 条件 2 は真のときの処理  
}  
else if( 条件 3 ){  
    条件 1 も 2 も偽だが, 条件 3 は真のときの処理  
}  
else{  
    すべての条件が偽の時の処理  
}
```



3 - 2 . 条件分岐 —— if文

●入れ子構造も可能

```
if( 条件A ){  
    if( 条件 1 ){  
        条件Aと条件 1 が真のときの処理  
    }else if( 条件 2 ){  
        条件Aが真で, 条件 1 は偽だが条件 2 が真のときの処理  
    }else{  
        条件Aが真で, 条件 1 も条件 2 も偽のときの処理  
    }else if( 条件B ){  
        if( 条件 3 ){  
            条件Aは偽だが条件Bは真で, 条件 3 が真のときの処理  
        }else if( 条件 4 ){  
            条件Aは偽だが条件Bは真で, 条件 3 は偽だが条件 4 は真のときの処理  
        }else{  
            条件Aは偽だが条件Bは真で, 条件 3 も条件 4 も偽のときの処理  
        }else{  
            条件Aも条件Bも偽のときの処理  
        }  
    }  
}
```

3 - 3 . 条件式

- 条件式は，真か偽かで判断できる形式で記述
- 関係演算子，論理演算子を用いる

- 関係演算子

$>$, $<$, $==$, $!=$

→演算子の前後の関係性を調べる

- 論理演算子

$\&\&$, $||$

→論理積，論理和，つまり「かつ」「または」
組み合わせ条件を記述する

3 - 4 . 条件分岐 —— switch文

- 真か偽かではなく，変数の中身で分岐させる

```
switch (A) {  
    case a:  
        Aがaだった時の処理  
        break;  
    case b:  
        Aがbだった時の処理  
        break;  
    ...    //caseは好きだけ増やせる  
    default:  
        Aがどの場合でもなかったときの処理  
        break;  
}
```

switch文はブロックの区切りに大括弧はいらないよ



演習

- 3-1 : あいさつプログラム

時間を入力させ、以下のような場合分けで挨拶を返すプログラムをかけ

6~10:おはようございます

11~17:こんにちは

18~21:こんばんは

0~5,22~24:Zzz...

それ以外は, "正しい時間を入力してください"を返す

- 3-2 : 図形面積算出プログラム

三角形, 長方形, 台形, 円などの面積を求めるプログラムをかけ

例

```
> 図形を選んでください  
> 1 : 三角形  2 : 長方形  3 : 台形  4 : 円  
> 底辺を入力してください (など)  
> 面積は〇〇cm2 です.
```

- 3-3 : if - switch変換

演習3-1をif文ではなく switch文で記述して同じ結果を得るには？



4. まとめと課題演習

変数と標準入出力

- 変数とは、データをしまっておく**入れ物**
- 変数は **宣言**・**代入**・**参照**の手順で使う
- printf()関数で変数を用いるときは、**書式**と引数を指定する形で変数の参照を行う
- scanf()関数はキーボードからの入力を受取る**書式**と引数を指定する形で変数に格納する
- **定数**もある

変数と入出力を組み合わせることで、
少しプログラムっぽいことが出来る！

条件分岐

- 条件分岐は、行う処理の**場合分け**をする
- if文は、**真か偽かで判断**できる条件に強い
- switch文は、**複数の場合分け**につよい
- 条件分岐を記述するときは、**フローチャート**をあらかじめ書いて整理しておくとうい

条件分岐を使いこなすことで、
よりプログラムっぽいことが出来る！

課題

●Ex- 1 : BMI算出プログラムの応用

A氏…身長192cm, 体重115kg

B氏…身長154cm, 体重38kg

C氏…身長178cm, 体重96 k g

三人のうち最も太っているのはだれか

※「肥満体型」

BMIが25以上は肥満の恐れがあるとされる

>>もっとも太っているのは, ○氏

ヒント

まずはそれぞれのBMIを算出
BMIどうしの比較をして,
一番太っている人を表示

簡単に解けたら, 何人かの身長体重
を入力して, 一番太っている人を表
示させるプログラムも作ってみよう

課題

●Ex- 2 : intの限界 階乗を計算させてみる

- i) 7!
- ii) 13!

13!でエラーが出てしまうのはなぜか？

ヒント

電卓アプリなどで, 13の階乗を計算してみよう

→桁数は？

int のデータ型について調べてみよう

課題

●Ex- 3 : 計算順序

半径125の円の面積を求める

ただし円周率は $355/113$ の近似値を用いること

※円周率を先に掛けた時と、最後に掛けた時での結果の違いを考える

ヒント

355/113単体で表示してみると... ?
出力書式が整数だとどうなる ?