

## プログラミング基礎演習A



プログラムの記述  
変数と代入  
printf関数による出力

## プログラム記述(1)

いちばんの基本 : 「メイン関数に書かれた命令が1つずつ順番に実行される」

上から順番に  
実行するよ



```
int main()
{
    実行される命令
}
```

メイン関数

2

## プログラム記述(2)

プログラムの構造



メイン関数の実行に必要な情報の記述

```
int main()
{
    実行される命令
}
```

メイン関数

まずはここを  
読まないで  
メイン関数が  
読めないんだ

3

## プログラム記述(3)

先週動かしたプログラムのメイン関数

```
int main()
{
    printf("Hello World!\n");
}
```

メイン関数

printf  
って何?



はじめに ヘッダファイル を読み込む!

↓  
命令に含まれる 関数の宣言 が記述されたファイル  
printf関数を使用するためには、ヘッダファイルstdio.hが必要

4

## プログラム記述(4)

先週動かしたプログラム

```
#include <stdio.h> ヘッダファイル読み込み

int main() メイン関数
{
    printf("Hello World!\n");
}
```



定義通りに  
実行しよう!

ふむふむ、  
printfって  
こういうもの  
なんだ

5

## ヘッダファイル

- ◆ C言語であらかじめ用意されている関数(命令)などの宣言が記述されているファイル
- ◆ 拡張子は.h
- ◆ 必要に応じて#include指令により読み込む  
例) #include <stdio.h>

- ◆ よく使うヘッダファイル

standard  
stdio.h, stdlib.h, string.h, math.hなど  
library  
input, output

6

## printf関数

printf("表示する文字列")

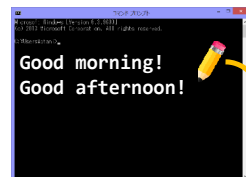
- ◆ 文字列を出力する関数
- ◆ ヘッダファイルstdio.hが必要

- ◆ C言語での決まり
- ◆ 改行は\nで表す
- ◆ 命令文の最後にはセミコロン;を付ける
- ◆ コメント文は/\*と\*/で囲まれた部分を書く

```
#include <stdio.h>
```

```
int main()
{
    printf("Good morning!\n"); /* (1) */
    printf("Good afternoon!\n"); /* (2) */
}
```

改行 文末のセミコロン



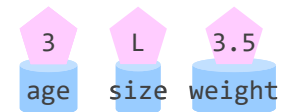
7

## 変数

- ◆ データを入れる箱のようなもの
- ◆ 中身がわかるような名前を付ける
- ◆ 入るデータの種類により型が異なる

【基本的な変数の型】

- ◆ int : 整数
- ◆ char : 文字(1バイト)
- ◆ float : 実数(単精度浮動小数点)
- ◆ double : 実数(倍精度浮動小数点)
- ◆ 1つの箱に1つの値しか入らない
- ◆ 値は上書きされる



型によって  
箱の大きさが  
違うね



整数 : integer  
浮動小数点 : floating point  
文字 : character

8

## 変数名

- ◆ 変数には何を表しているかがわかりやすい名前を付ける
  - ◆ できれば英単語 or 英単語の省略形
    - ◆ 日本語のローマ字表記はちょっとカッコ悪い
    - ◆ スペルミスに注意!
- ◆ 半角の英数字およびアンダースコア\_のみ使用できる
  - ◆ 大文字と小文字は区別される
  - ◆ 先頭に数字は使えない
- ◆ C言語で予約されている単語は使えない  
予約語の例: if, else, for, while など



9

## 記号の読み方(1)

| 記号 | 読み方                   | 記号 | 読み方             |
|----|-----------------------|----|-----------------|
| !  | エクスクラメーションマーク         | \  | バックスラッシュ        |
| "  | ダブルクォーテーション           | `  | バッククォーテーション     |
| #  | ナンバー                  | @  | アットマーク          |
| \$ | ダラー                   | +  | プラス             |
| %  | パーセント                 | *  | アスタリスク          |
| &  | アンパサンド                | ;  | セミコロン           |
| '  | アポストロフィ, シングルクォーテーション | :  | コロン             |
| =  | イコール                  | ,  | カンマ             |
| -  | ハイフン, マイナス            | .  | ピリオド, ドット       |
| ~  | チルダ                   | ?  | クエスチョンマーク       |
| ^  | caret, ハット            | /  | スラッシュ           |
|    | バー, パーティカルライン         | _  | アンダースコア, アンダーバー |
|    |                       | ¥  | 円マーク            |

10

## 記号の読み方(2)

| 始め<br>(開き) | 終わり<br>(閉じ) | 読み方                  |
|------------|-------------|----------------------|
| (          | )           | 丸括弧, パーレン            |
| [          | ]           | 角括弧, ブラケット           |
| {          | }           | 波括弧, カーリーブラケット, プレース |
| <          | >           | 山括弧, アングルブラケット       |

| 記号 | 読み方          |
|----|--------------|
| <  | レスザン, 小なり    |
| >  | グレイターザン, 大なり |



11

## 変数の宣言

- 型名 変数名
- ◆ 使用する変数の領域を確保する
- ◆ 関数の冒頭で宣言する
- ◆ 変数名をカンマ, で区切ることで, 複数の変数を同時に宣言できる

```
int main()
{
    char size;
    int age, year;
    ...
}
```

文字が入る箱を1個,  
整数が入る箱を2個  
用意するんだね!

size age year

12

## 変数への代入

変数名 = 値

- ◆ 実行される命令文として書く
- ◆ 数学の=と意味が異なる
  - ◆ 数学の= : 「左辺と右辺の値が等しい」
  - ◆ C言語の= : 「左辺の変数に右辺を代入する」

```
int main()
{
    int age, year;
    age = 25;
    year = 1998;
}
```

変数の宣言

処理の内容

空行があると読みやすい

ageに25, yearに1998を入れるんだね!

25 age

1998 year

13

## int型変数への実数の代入

int型の変数に実数を代入すると小数部分が切り捨てられる

```
#include <stdio.h>

int main()
{
    int a, b;
    double x;

    x = 3.14;
    a = x;
    b = 1.41;
    printf("aの値は%dです\n", a);
    printf("bの値は%dです\n", b);
}
```

小数部分  
切捨て

3 a

1 b

3.14 x

aの値は3です  
bの値は1です

14

## 変数の初期化

型名 変数名 = 初期値

- ◆ 変数の宣言と同時に初期値を設定する

```
int main()
{
    int year = 2045;
    ...
}
```

箱を用意すると同時に値が入るよ!

2045 year

以下のように書くのと同じ

```
int year;

year = 2045;
```

15

## printfによる変数の内容の表示

`printf("フォーマット文字列", 変数, 変数, ...)`

- ◆ フォーマット文字列中の変換指定子を後に続く引数が表す値に置き換えて文字列を出力する関数
- ◆ 第2引数以降の引数の個数は、フォーマット文字列中の変換指定子の個数に一致
- ◆ 主な変換指定子
  - ◆ **%d** : 整数(int型)を10進数で表示
  - ◆ **%f** : 実数(float型, double型)を表示
  - ◆ **%c** : 文字(char型)を表示
  - ◆ **%s** : 文字列を表示

10進数 : decimal number  
浮動小数点 : floating point  
文字 : character  
文字列 : string

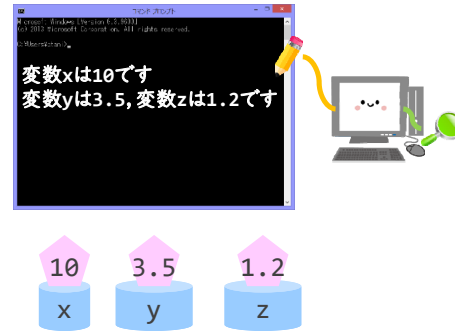
16

## プログラム例

```
#include <stdio.h>
```

```
int main()
{
    int x;
    double y, z;

    x = 10;
    y = 3.5;
    z = 1.2;
    printf("変数xは%dです\n", x);
    printf("変数yは%f, 変数zは%fです\n", y, z);
}
```



17

## 演習

自分の生まれた年, 月, 日をそれぞれ変数year, month, dayに代入し, 実行例のように出力するプログラムを作成せよ。

プログラム名はe2とすること。

### <実行例>

```
year = 2002
month = 5
day = 7
生年月日 : 2002/5/7
```

赤字部分は自分の生年月日とすること

18

## 演習 ～手順～

1. どんなプログラムにも絶対にあるのは?
  - メイン関数
  - まずは中身が空のメイン関数を書く

```
int main()
{
}
```

開き括弧を書いたら、  
すぐに閉じ括弧を書こう!



19

## 演習 ～手順～

2. 必要なヘッダファイルは?
  - 出力にはstdio.hが必要
  - #includeの行を書く

```
#include <stdio.h>

int main()
{
}
```

stdio.hは  
ほとんどのプログラムに  
必要になるよ!



20

## 演習 ～手順～

### 3. 必要な変数は？

- 変数year, month, dayが必要
- 適切な型で変数を宣言する

```
#include <stdio.h>

int main()
{
    int year, month, day;
}
```

年も月も日も  
整数だね！



21

## 演習 ～手順～

### 4. 処理の順番は？

- yearに生まれた年を代入する
- monthに生まれた月を代入する
- dayに生まれた日を代入する
- 「year = XXXX」を表示する (printf)
- 「month = YY」を表示する (printf)
- 「day = ZZ」を表示する (printf)
- 「生年月日 : XXXX/YY/ZZ」を表示する (printf)

これを1つずつ  
メイン関数に  
書けばいいね



22