

Programming I 0x03



変数とデータ型 (2010.04.19)

塩澤秀和 <http://vilab.org>

最初のCプログラム(復習)

```
/* 最初のプログラム */
```

コメント(人間
向けのメモ)

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

#include 指令
プログラムで使う機能を指定
(この場合は画面への表示)

```
int main(void)
```

プログラム
本体(メイン)

```
{
```

```
printf("Hello, world!¥n");
```

画面表示を
する処理

```
return 0;
```

mainの処理
の終了

mainの
範囲

```
}
```

Cプログラムの基本形(復習)

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

```
int main(void)
```

```
{
```

データの初期設定



数値などの入力



計算



画面への表示

```
return 0;
```

```
}
```

とにかく
順番が大事!

基本形は
暗記!

今日理解するプログラム

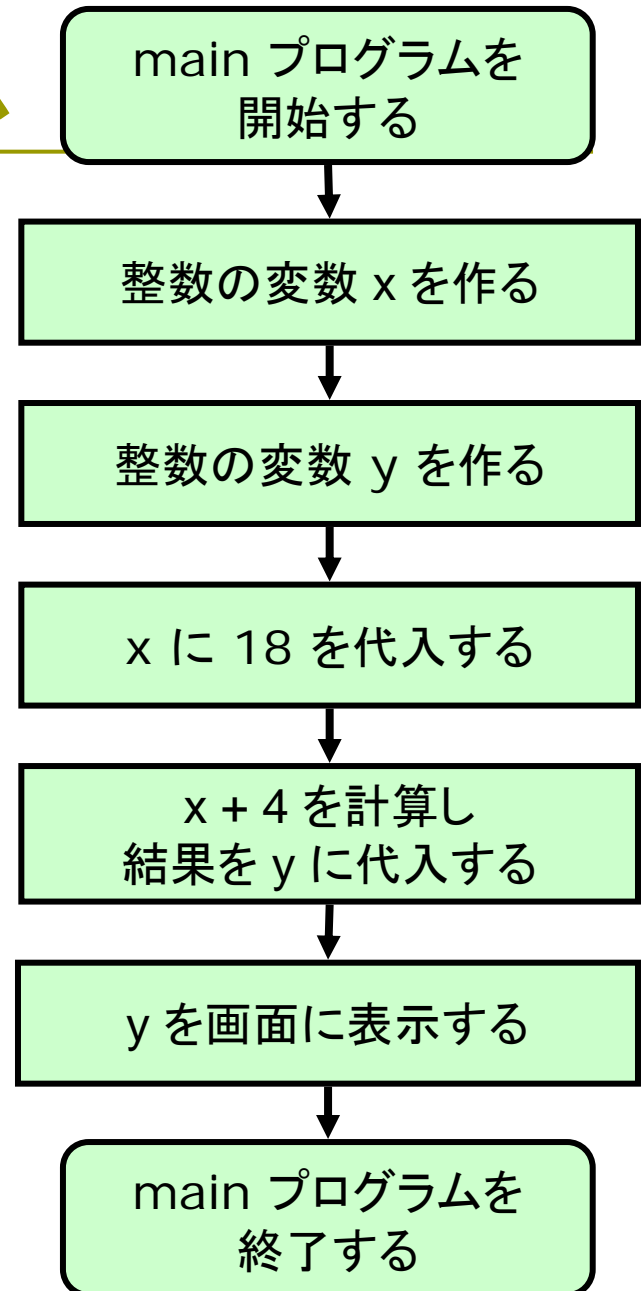
```
#include <stdio.h>

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

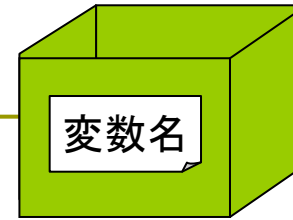
    x = 18;
    y = x + 4;

    printf("答え:%d\n", y);

    return 0;
}
```



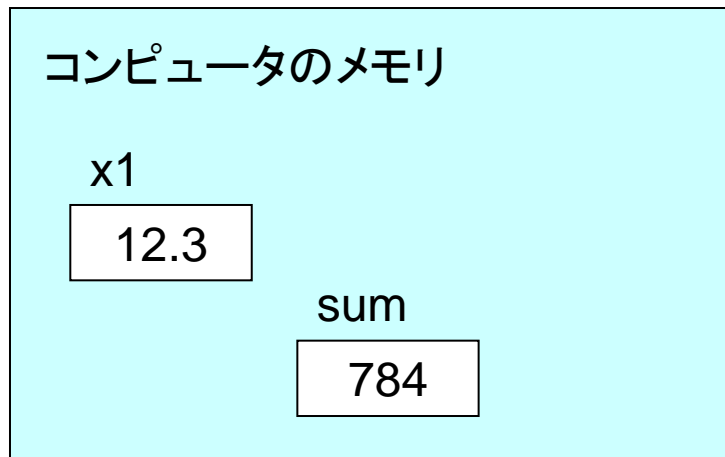
変数 (p.24)



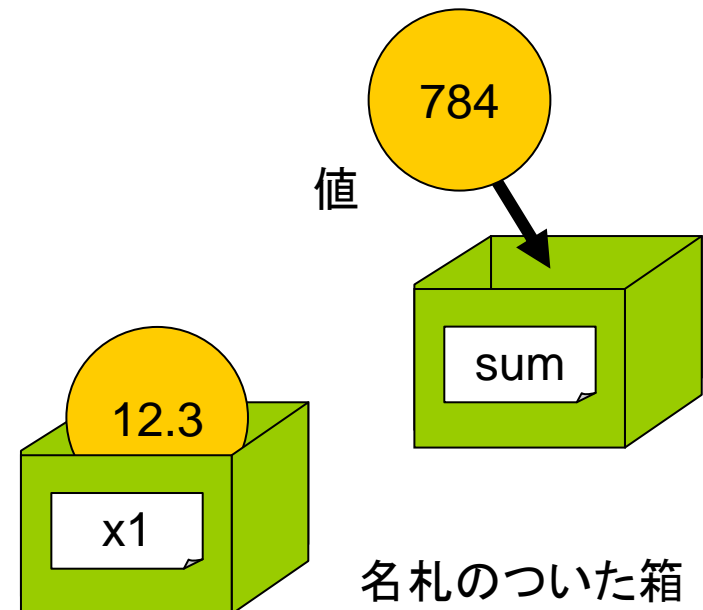
□ 変数とは

- 名前付きのメモリ(データを入れておく場所≡データの箱)
- 箱に入れるデータ(“値”)を自由に**変**えられる ⇒ 「**変**数」
- プログラムを、いろいろなデータに対応できるようにする

□ 変数のイメージ



メモリの記憶領域に名前をつける



変数の作りかた(p.24)

□ 変数の“定義”(宣言)

- 変数定義: 「データ型名 変数名;」

int size; ← **整数**のデータを入れる変数 size を作る

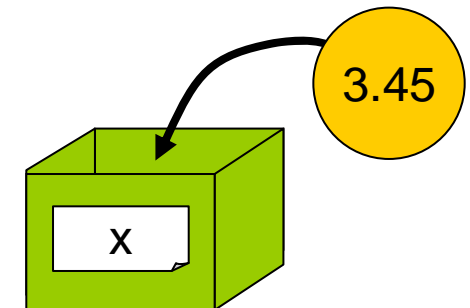
double x, y; ← **実数**のデータを入れる変数 x と y を作る

char ch; ← **1文字**のデータを入れる変数 ch を作る

- 変数は**ブロック({}で囲まれた範囲)のはじめ**で作る
- 分かりやすい変数名をつける
※ 命名のルールについては、教科書p.25を参照

□ 変数への代入

- 変数に値を入れるには「=」を使う
例: $x = 3.45$;
- 注意: 数学の「いつでも等しい」という意味**ではない!**



C言語のデータ型 (p.21)

□ データ型とは？

- プログラムのなかで使えるデータの種類(タイプ)
- 基本的に、プログラマーが適切に選択する必要がある
- 例: 高速な処理⇒整数型、高精度の計算⇒実数型

分類	データ型	名称	データの例
文字	char	文字型	'A', '8' (「」で囲う)
整数 (小数点 がない)	int	整数型	45, -1289
	unsigned (int)	符号なし整数型	3487, 0
実数 (小数点 がある)	float 単精度	浮動小数点型 (有効桁数の違い)	2.56, -0.345, 1.4E3
	double 倍精度		

変数の使用例

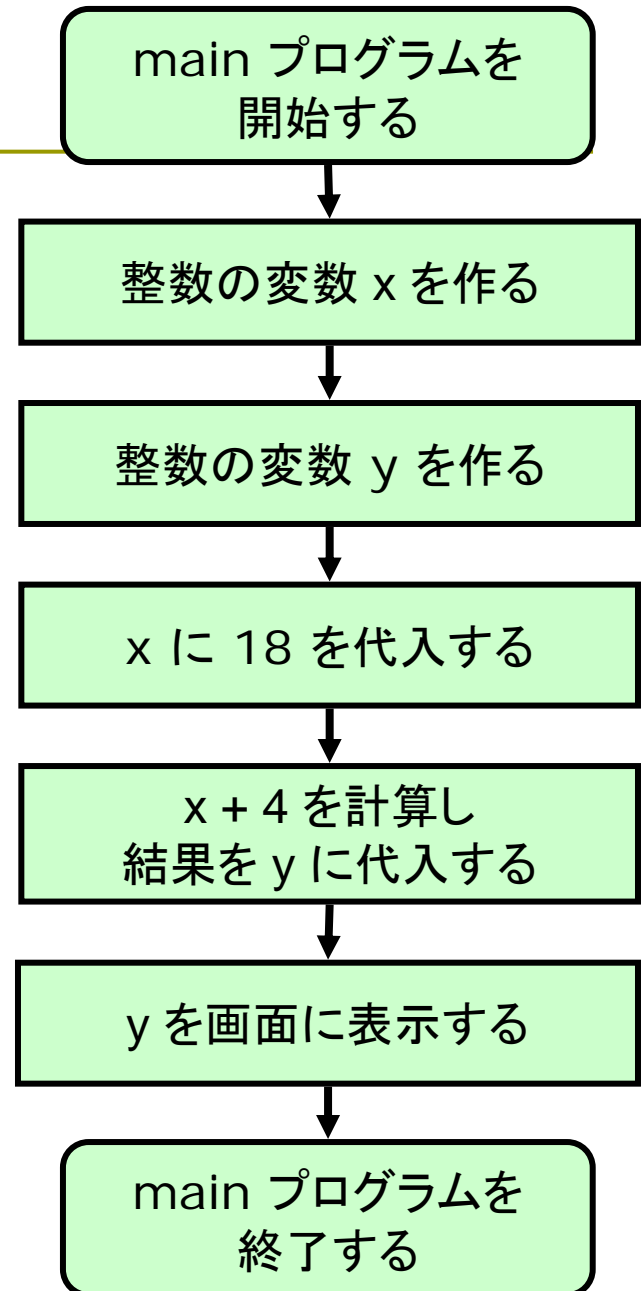
```
#include <stdio.h>

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

    x = 18;
    y = x + 4;

    printf("答え:%d\n", y);

    return 0;
}
```



文字型と実数型の例

```
/* char型の例 */  
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    char moji;
```

```
    /* 半角英数1文字 */
```

```
    moji = 'A';
```

```
    printf("文字 %c\n", moji);
```

```
    return 0;
```

```
}
```

```
/* double型の例 */
```

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

```
int main(void)
```

```
{
```

```
    double x;
```

```
    x = 4.3;
```

```
    printf("x = %-7f\n", x);
```

```
    x = 0.07;
```

```
    printf("x = %.3f\n", x);
```

```
    return 0;
```

```
}
```

printfで凝った表示

□ フィールド幅と桁数 (p.29)

- `%10d` 幅10桁の欄に右詰め `%-10d` 同じく左詰め
- `%.2f` 精度を小数第2位まで `%10.2f` 幅と精度の指定

□ 複数の変換指定子 (p.31)

- 1行に複数の変数や式の値を表示できる
- `printf("文字=%c 整数=%d¥n", moji, seisu);`
- `printf("%d に %d をかけると %d になる。¥n", x, y, z);`

□ 計算式を直接printfに書く (p.39)

- `printf("結果 = %d¥n", 10 + 15);`
- `printf("x / 2.0 の答えは %f です。¥n", x / 2.0);`

演習問題

- 3a. int型の変数a, bを作り、aに12を代入してから、bにaの値をコピー ($b = a$)して、bを表示するプログラムを作成しなさい。
- 3b. double型の変数x, yを定義し、xに10.0を代入してから、yに $x / 3.0$ の計算結果を代入し、最後にyの値を小数第5位まで表示するプログラムを作成しなさい。
- 3c. char型の変数iniを定義し、自分の名前のイニシャル1文字を代入してiniの値を表示し、続けてiniに苗字のイニシャルを代入して同様に表示するプログラムを作成しなさい。
- 文字型データは、'A' のようにシングルウォーテーションマークで囲う
- 3d. 変数a, b, cを定義し、aに123、bに456を代入してから、aとbの中身を交換し、結果を表示するプログラムを作成しなさい。
- 次回までの課題: リスト3-8~4-2 を入力して実行しておく