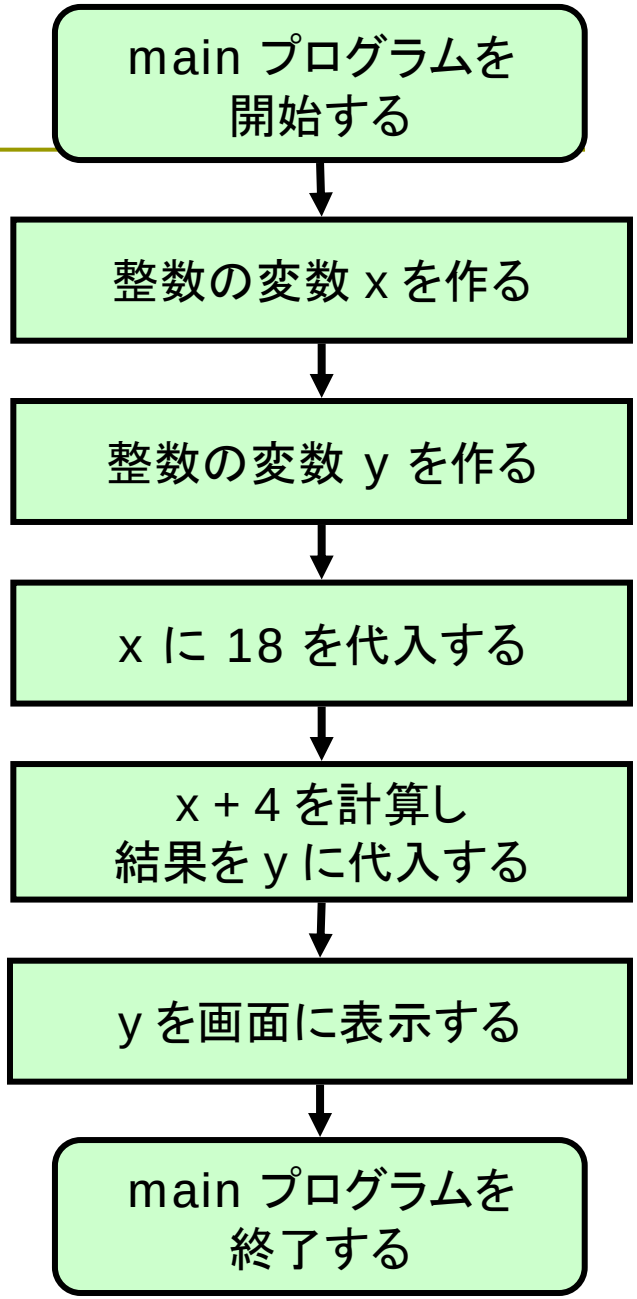
Programming I 0x04

標準入力と演算子（2010.04.26）

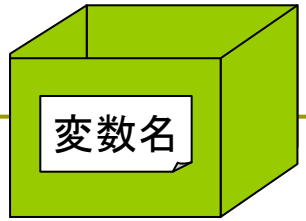
塩澤秀和 http://vilab.org

変数の使用例(復習)

```
#include <stdio.h>

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

    x = 18;
    y = x + 4;

    printf("答え:%d¥n", y);

    return 0;
}
```



変数の作りかた（復習）

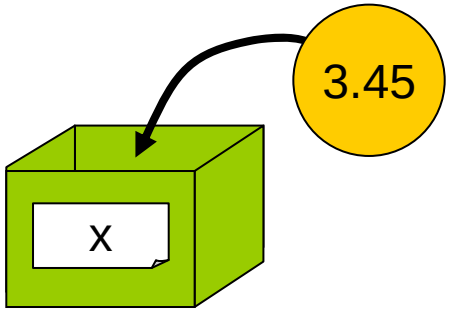


- 変数の“定義”（宣言）
  - 変数定義：「データ型名 変数名;」

    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

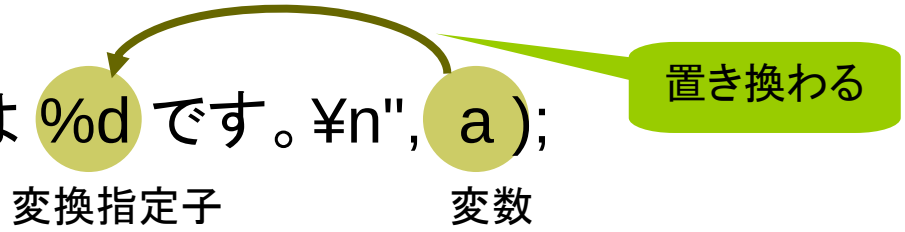


- 変数への代入
  - 変数に値を入れるには「=」を使う

    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！**

データの表示（復習）

- printf 関数の書式
  - printf("メッセージ文字列", **式**);
  - メッセージ文字列に“変換指定子”を埋め込むと、その部分が「式」の値に置き換わる
  - 例： printf("変数aの値は %d です。¥n", a);



- 変換指定子

int	**%d**	intを16進数で	**%x**
char	**%c**	floatとdouble	**%f**

  - 上記以外は、教科書の表3.4参照
  - なお、「%」文字自体を表示したいときは、「%%」とする

代入と初期化

- 変数への代入（p.40）
  - 変数に、「=」で新しい値を入れる（代入演算子）
    〇よい例　x = 3.45;　　×ダメな例　3.45 = x;
    〇よい例　x = a + b;　×ダメな例　a + b = x;
  - イコールよりも、「←」のようなイメージ（x←3.45）
  - 「a = b」と「b = a」は、まったく反対の意味になる

- 変数の初期化（p.38）
  - 変数を作るときに、「=」で値を設定できる（初期化子）
  - 例：int age = 18;

- プログラムの中で、変数の値はどんどん変わる
  - よく変化する変数には、特に分かりやすい名前をつける

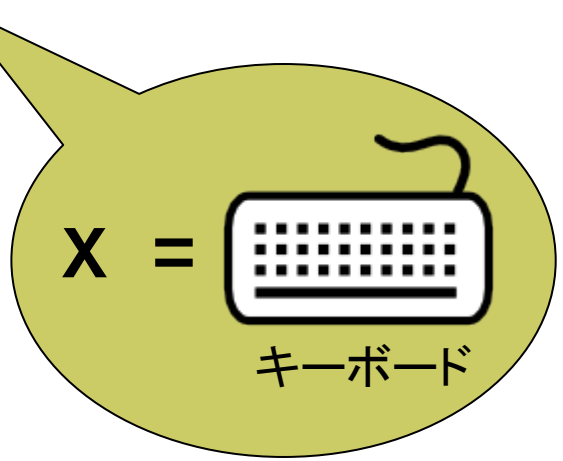
キーボードからの入力

```
#include <stdio.h>

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

    x = 18;
    y = x + 4;

    printf("答え:%d¥n", y);

    return 0;
}
```

実行するとプログラムが停止するので
キーボードから何か整数を打ち込んで
[Enter]キーを押す

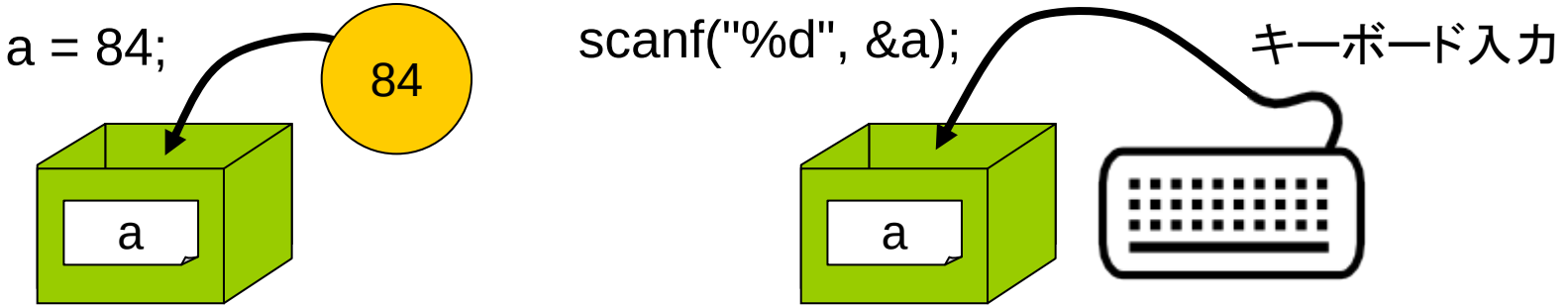
1行書き換えてみよう

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



scanf 関数（p.31）

- 標準入力からデータを読み込む
  - scanf("変換指定子", &変数);
  - 要注意!!「&」（変数の“アドレス”）が必要 ⇒ 意味はプロII



- データ型によって書式が違う
  - 整数（int）の入力: scanf("**%d**", &変数);
  - 実数（float）の入力: scanf("**%f**", &変数);
  - 実数（double）の入力: scanf("**%lf**", &変数); ←**l**は“エル”
  - 1文字（char）の入力: scanf("**%c**", &変数);

scanfの使いかた

- 1つのscanfで複数項目の入力（p.32）
  - scanf("%d %d %d", &kazu1, &kazu2, &kazu3);
  - 項目の間が空白なら、入力は改行または空白で区切る
  - カンマ等なら、入力もその文字で区切らないといけない

- “プロンプト”を表示しよう（p.35）
  - scanfの前にprintfでユーザに入力指示を表示する
  - 例： printf("時速を入力してください： ");
    scanf("**%lf**", &speed);

- 注意：scanfの文字列は表示されない
  - ダメな例： × scanf("x -> %d", &x);
  - キーボードから「x ->」という文字列も**読み込め**という意味

コンパイルエラーについて

- エラーと警告
  - エラー(error): 文法などの間違いで、コンパイル、実行できない
  - 警告(warning): 何とかコンパイルはできるが、多分間違っている
- エラーや警告が出たら…
  - 必ずよく読んで、「エラー」や「警告」がゼロになるようになおす
  - エラーメッセージをダブルクリックすると、発生した行に飛ぶ
  - その行だけでなく、前後の行もよーく見て間違いを探すこと
- scanf の警告対策
  - scanf を使うと警告が出るのが、以下のようにすると出なくなる
  1. Webブラウザで http://vilab.org/c1_2010/ にアクセス
  2. set_cl_options.wsf を右クリックして、「保存」
  3. 保存した set_cl_options.wsf をダブルクリックして実行する

算術演算子(p.37)

```
#include <stdio.h>

int main(void)
{
  int x, y;

  x = 7;
  y = 3;

  printf("加算 %d¥n", x + y);
  printf("減算 %d¥n", x - y);
  printf("乗算 %d¥n", x * y);
  printf("除算 %d¥n", x / y);
  printf("剰余 %d¥n", x % y);

  return 0;
}
```

その他の演算	書きかた
マイナス (符号反転)	−x
カッコ (優先順位)	( ) だけを使う 例: ((x + y) / 2 + c)
べき乗 x^y (xのy乗)	演算子はないので x*x*x などと書く (数学関数 pow を用いてもよい)

このプログラムを
キーボードからdouble型の
数値を読み込んで、
計算結果もdouble型で表示
するように改造してみよう
(ただし剰余は計算できない)

データ型の変換

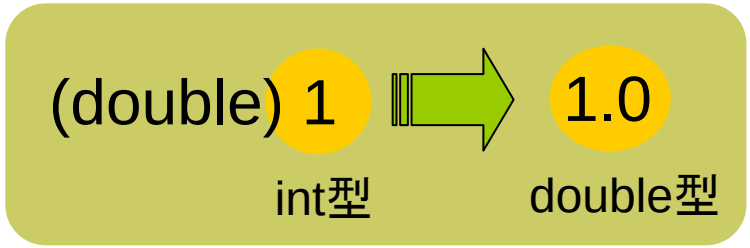
- 演算結果のデータ型
  - 整数型 と 整数型 ⇒ 整数型 （例: 10 / 4 → 2）
  - 実数型 と 実数型 ⇒ 実数型 （例: 10.0 / 4.0 → 2.5）
  - 整数型 と 実数型 ⇒ 実数型 （例: 10 / 4.0 → 2.5）
  - 精度が高いほうのデータ型に暗黙的に変換される

- キャスト演算子（p.41）
  - データ型を明示的に変換する
    文法: (データ型名) 式
  - 例: 整数→実数の変換（実数に直してから割り算）

```
heikin = (double)(a + b) / 2.0;    /* a, bはint型 */
```

  - 例: 実数→整数の変換（小数点以下を切り捨てる）

```
discount = (int)(price * 0.8);
```



キャストの使用例

```
#include <stdio.h>

int main(void)
{
  int a, b;
  int c;
  double d, e, f;

  printf("整数を2個入力:");
  scanf("%d %d", &a, &b);

  c = a / b;
  printf("整数計算 %d¥n", c);

  d = (double)a / (double)b;
  printf("実数計算 %f¥n", d);

  e = (double)(a / b);
  printf("ダメな例 %f¥n", e);

  f = (double)a / b;
  printf("これはOK %f¥n", f);

  return 0;
}
```

それぞれの理由を考えてみよう

ビット演算子(p.42)

記号	演算	意味
& (x & y)	ビット単位の論理積(AND)	対応するビットが両方とも1ならば、結果のビットも1にする
| (x | y)	ビット単位の論理和(OR)	対応するビットが片方でも1ならば、結果のビットを1にする
^ (x ^ y)	ビット単位の排他的論理和(XOR)	対応するビットが異なるならば、結果のビットを1にする
~ (~x)	ビット単位の否定(NOT)	全ビットを反転する(1の補数)
<< (x <<n)	左シフト	構成ビットを、nビットだけ左にずらす
>> (x >>n)	右シフト	構成ビットを、nビットだけ右にずらす

10進数	2進数	16進数
0	0000	0x00
1	0001	0x01
2	0010	0x02
3	0011	0x03
4	0100	0x04
5	0101	0x05
6	0110	0x06
7	0111	0x07
8	1000	0x08
9	1001	0x09
10	1010	0x0A
11	1011	0x0B
12	1100	0x0C
13	1101	0x0D
14	1110	0x0E
15	1111	0x0F

語源集

- main
  - 本体・メイン
- include
  - 取り込む
- stdio.h
  - standard input / output
    標準入出力
- int
  - integer 整数
- char
  - character 文字
- float
  - floating-point number
    浮動小数点数
- double
  - double precision 倍精度
- unsigned
  - 符号(sign)なし
- return
  - 戻る
- printf
  - print 印刷する + format 書式
- scanf
  - scan 読み取る + format 書式
- ¥n
  - newline 改行
- %d
  - decimal 10進数
- %x
  - hexadecimal 16進数
- %f
  - float
- %lf
  - long float = doubleの古い別名
- %c
  - char

演習問題

4a. int型の変数nを定義してキーボードから整数を読み込み、それを3で割ったあまりを表示するプログラムを作成しなさい。

- 0, 1, 2, 3, 4, 5, ... と順に整数を入力してみると、どうなるか。

4b.「算術演算子」のプログラムを、キーボードから倍精度型（double型）のデータを読み込んで、計算結果も倍精度型で表示するように改造しなさい。ただし、剰余は計算しない。

- まず、変数の宣言（定義）は、double x, y; になる。

4c. 単価がx円のおにぎりをy個買ったとき、z円札を出した場合のお釣りを表示するプログラムを作成しなさい。x, y, zの値はそれぞれプロンプトを表示してユーザに入力させること。

4d. キーボードから整数a, b, c, dを読み込み、それらの平均を実数xに代入してから表示するプログラムを作成しなさい。

- 例えば 3, 5, 6, 7 を入力したら、(3 + 5 + 6 + 7) / 4 → 5.25 が正解。

演習問題(宿題)

- 次回までの課題: リスト4-3～4-6 を入力して実行しておく

4e. 文字型の変数cにキーボードから1文字を入力し、cに1を足してから(c = c + 1)、cを文字として表示するプログラムを作成しなさい。(書式指定は"%c")

4f. double d = 1.0 / 3.0 と float f = 1.0F / 3.0F をともに小数点下20桁まで表示させて計算の精度を比較してみなさい。

- 小数点は実数(double)を示す。末尾に「F」をつけるとfloat型になる。

4g. 円の半径 r をキーボードから読み込み、その面積を計算して表示するプログラムを作成しなさい(π =3.1416とする)。

4h. キーボードから整数aを読み込み、aと0x01との“ビット単位の論理積”(&)を表示するプログラムを作成しなさい。

- 0, 1, 2, 3, 4, 5, ... と順に整数を入力してみると、どうなるか。