

変数と算術式

情報処理演習

(テキスト: 第3章, 第5章, 第6章)

今日の内容

1. 変数
2. 代入
3. 関数(システム関数)
4. 標準出力関数 printf()
5. 標準入力関数 scanf()
6. Xcode統合環境の使い方(2)
7. 今日の課題

1. 変数

■ 変数

➤ データ(数値や文字)を入れるもの(箱)

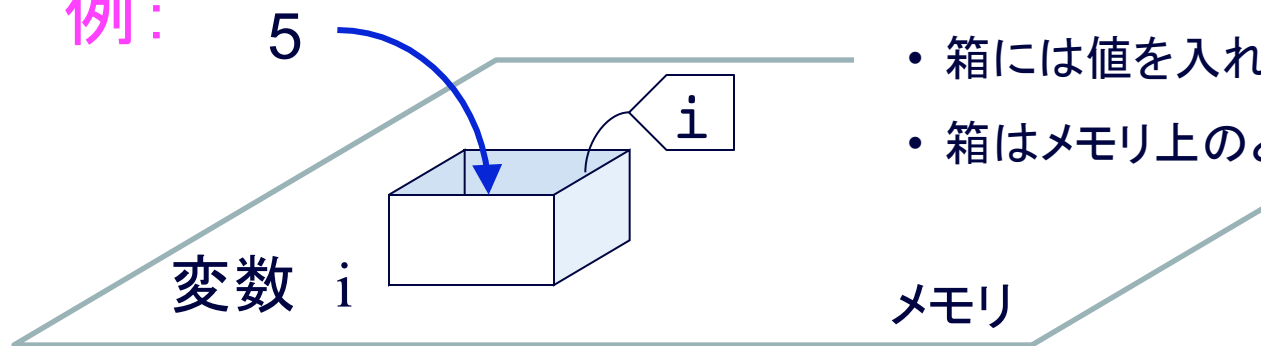
■ 変数名 (識別子: identifier)

➤ 英数字かアンダーバー(_)で作られる.

➤ 数字は最初の文字に使えない.

➤ 大文字と小文字を区別する.

例:



- 箱には名前が付いている.
- 箱には値を入れることができる.
- 箱はメモリ上のどこかに作られる.

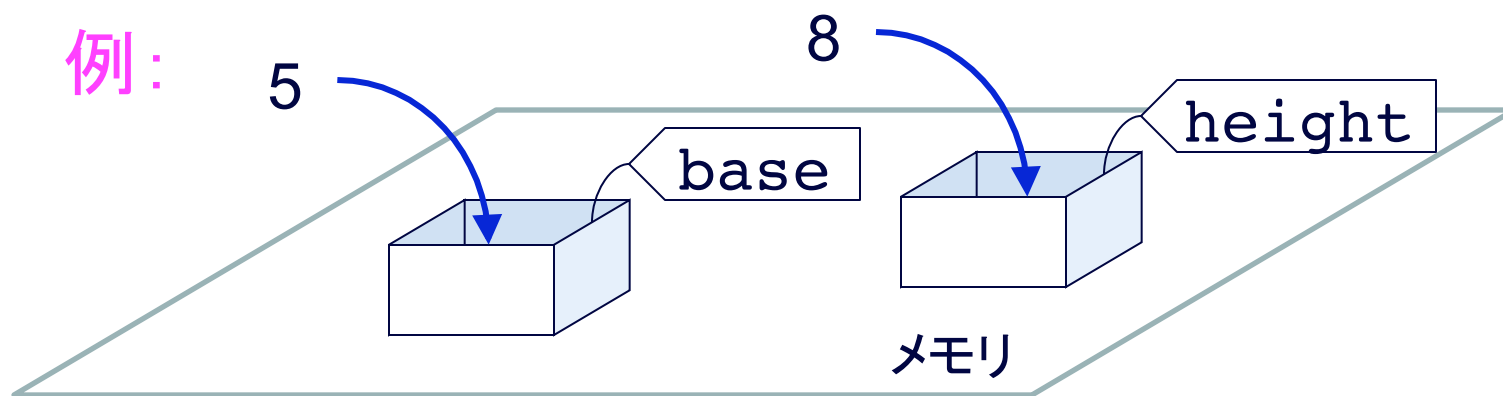
変数名をつける際の注意点

■ 変数名（識別子: identifier）

- 混同しにくい名前(識別子)を使う.
- 変数の意味を正確に伝える名前を使う.
- 短すぎる名前はよくない.
- 長すぎる名前もよくない.

前ページの例よりわかりやすい.

- baseは底辺、heightは高さ
- 変数の用途が名前からわかる.



数の表現

(テキスト19ページ)

1ビット(bit) = 0か1かという2つの状態を表す情報量
= 0か1かが入る箱

1バイト(byte) = 8ビット

(アルファベット1文字は1バイトで表現可能)

7	6	5	4	3	2	1	0	
0	0	0	0	0	1	0	1	= $2^2 + 2^0 = 5$
↑	↑	↑	↑	↑	↑	↑	↑	
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	

8ビットで表すことの出来る数字は?

$$0 \sim 2^8 - 1 = 255$$

もっと大きな数は？

- ビットをたくさん並べると大きな数を表せるようになる
- 負の数も表さなければならない。

1バイト(8ビット) → $-128 \sim 127$

2バイト(16ビット) → $-32,768 \sim 32,767$

4バイト(32ビット) → $-2,147,483,648 \sim 2,147,483,647$

機械で扱うので、あらかじめビット数を決めておく必要がある。

変数宣言

(テキスト21ページ)

■ 宣言

- コンパイラにユーザーが作った名前をどのように解釈するのかを伝えるもの
- 変数を使う前に必ず宣言しなければならない。
- 値の種類を教える必要がある(型)。

箱の大きさが異なる。

Xcodeは4バイト

型名	種類	バイト数
char	文字	1バイト
short	整数	2バイト
int	整数	2または4バイト
long	整数	4または8バイト
unsigned	符号なし整数(組合せ)	
float	単精度浮動小数点数	4バイト
double	倍精度浮動少数点数	8バイト

変数の上限下限はヘッダファイル `limit.h` に記述されている。

整数変数

変数宣言 `int base, height, area;`

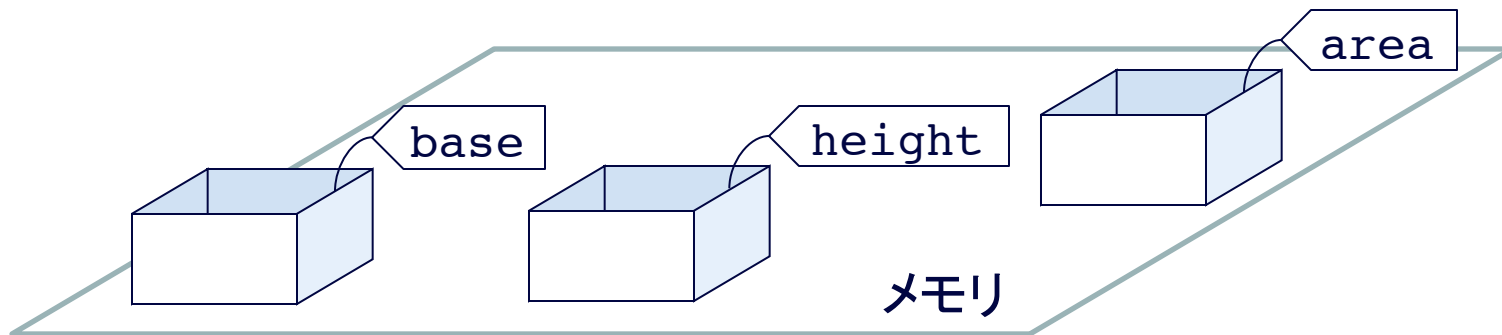
コンピュータの中で `int` 型の変数とは、
整数 (... , -2, -1, 0, 1, 2, ...) を蓄えておく32ビットの箱

下の範囲の整数を1つだけ箱の中に入れることができる。

$$-2^{31} \quad \sim \quad 2^{31} - 1$$

$$-2,147,483,648 \quad 2,147,483,647$$

上の変数宣言をすることで、下の図のような3つの箱が用意されると思ってよい。



2. 代入

(テキスト22ページ)

- 変数に数値を格納すること
- C言語ではイコール(=)を使用するが、左辺と右辺が等しいという意味ではない。
- 変数の型と同じ型のデータを格納する。

変数 = 式



- 左辺は変数を1つだけ書く。
- 値を格納する変数

代入式

(テキスト53ページ)

変数 = 式

右辺はリテラル, 演算式を記入できる.

式を計算した結果(値)を左辺の変数に代入する.

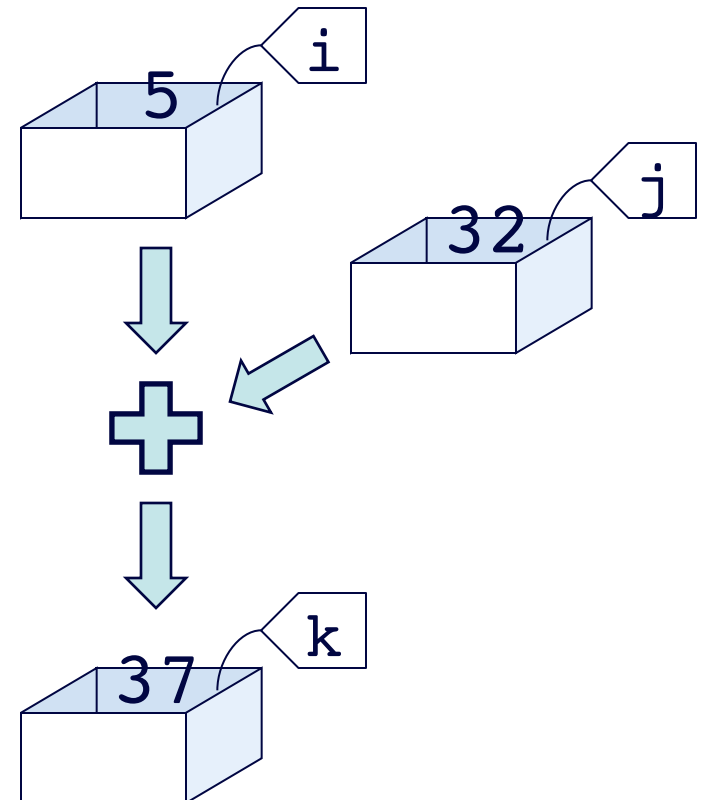
リテラルの代入

$i = 5$ ← 値5が代入される.

$j = 0 \times 20$ ← 値 $20_{(16)} = 32_{(10)}$ が代入される.

演算結果の代入

$k = i + j$ ← 足し算を計算した値が代入される.



リテラル

- “Hello, world” (文字列) や 5 (整数) などの具体的なデータをリテラルと言う。

文字リテラル	'A' 'a' ...	シングルクォート
文字列リテラル	"Hello, world" ...	ダブルクォート
整数リテラル	5 (10進数)	
	074 (8進数)	
	0x2f (16進数)...	
浮動小数点数リテラル	5.0 5. 0.2 .2 ...	

浮動小数点数リテラルは指定のない限りdouble型となる。

3.141592	(double型)
3.1415f	(float型)

代入式の文

- 代入式をプログラム中に書くときには、その後ろにセミコロンをつけて文にする。

代入式;

代入式の文

例

```
i = 5;  
j = 0x20;
```

リテラルの代入

```
k = i + j;
```

演算式を計算した値を変数に代入
この場合は $5 + 32$ で 37 を変数に代入する。

```
f0 = (float)(j / i);
```

```
f1 = (float) j / (float) i;
```

キャスト

直後（右側）にある式，変数，
リテラルの型を変更する。

3. 関数(ライブラリ関数)

(テキスト22ページ)

- 数学的な関数はライブラリ関数として用意されている.
 - ヘッダファイル math.h を include すると使用できる.
 - 関数 sin(), cos(), tan(), ln() など
 - 括弧の中に引数を書ける.
 - 使えるライブラリ関数はできるだけ使う.

使用例:

```
x = r * cos(theta);
```

関数名

引数:

- 関数に与えるパラメータ
- リテラル, 変数, 式などを書くことができる.
- 変数の値が関数に伝わる.
- 式を計算した結果が関数に伝わる.

関数 cos の求め方は
知らなくても使える

4. 標準出力関数 printf() (テキスト43ページ)

- 変数の値をコンソールに出力する際には 標準出力関数 (printf 関数) を使用する.
- プログラム中では下のように記述
`printf(第1引数, 第2引数, 第3引数, …);`
- printf では第1引数(出力書式文字列)を使って **引数の数**とその **型**が決められる.
- 2つ目以降の引数として変数(もしくは式)を書く.

書式付出力

(テキスト46ページ)

出力変換仕様

%文字幅, 精度, 変換指定子

```
printf("%5d + %5d = %5d\n", i, j, k);
```



- 変数が3つある.
- すべて整数である.
- 表示幅は5桁

```
printf("(float)(%5d/%5d) = %7.3f\n", i, j, f0);
```

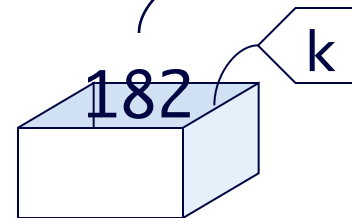
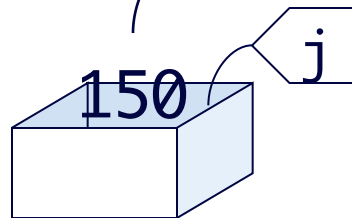
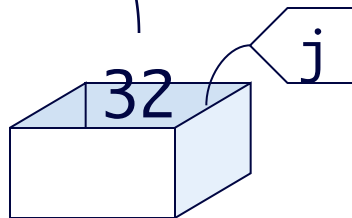
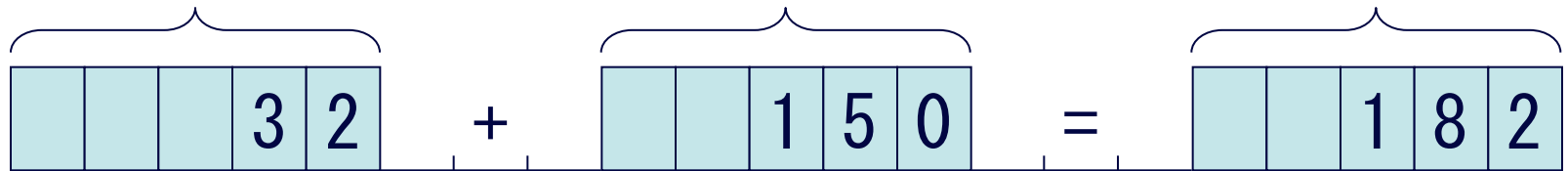


- 3つ目の変数は小数である.
- 3つ目は小数点以下3桁で表示
- 3つ目は小数点を含めて全体で7桁で表示

書式指定出力の例

```
printf("%5d + %5d = %5d\n", i, j, k);
```

表示の枠が決まる.



変数の値が表示される.



表示する値が枠に収まらない場合は、右にはみ出る (6桁で表示).

変換指定子, 文字幅, 精度

変換指定子	変換の文字列の形式	引数のデータ型
c	文字	char
s	文字列	先頭文字のアドレス
d	符号付10進整数	int
u	符号なし10進整数	int
o	符号なし8進整数	int
x	符号なし16進整数	int
f	浮動小数点数	float/double
e	浮動小数点数(指数表現)	float/double

%5d 符号付10進整数(少なくとも5文字幅)

%8f 浮動小数点数(少なくとも8文字幅)

%8.3f 浮動小数点数(少なくとも8文字幅, 小数点以下は3桁)

書式付出力での出力例

(テキスト50ページ)

出力する形式	出力変換仕様	表示例
整数	%d	321 12987
桁指定の整数(6桁)	%6d	321 12987 12345678
桁指定の整数(6桁) (頭に0を付ける)	%06d	000321 012987 12345678
小数	%f	123.456789
桁指定の小数 (全体8桁, 小数部3桁) (小数点は1桁になる)	%8.3f	1.235 123.457 98765.432

値の桁数の
ほうが大きい
場合は揃わ
ない。

整数部が5桁なので, 小数点が揃わない。

例題(1)

どのような出力になるか確認しなさい。

```
#include <stdio.h>

int main(int argc, const char * argv[]) {
    int i, j, k;    /* 変数宣言 */
    float f0, f1;
    i = 5;         /* 代入 */
    j = 3;
    k = i + j;     /* 算術演算 */
    printf("%5d + %5d = %5d\n", i, j, k);
    printf("%5d - %5d = %5d\n", i, j, i - j);
    return 0;
}
```

例題(2)

どのような出力になるか確認しなさい。

```
#include <stdio.h>

int main(int argc, const char * argv[]) {
    int i, j, k;    /* 変数宣言 */
    float f0, f1;
    i = 5;         /* 代入 */
    j = 3;
    f0 = (float)(i / j);    /* キャスト */
    f1 = (float)i / (float)j;
    printf("(float)(%d / %d) = %6.3f\n", i, j, f0);
    printf("(float)%d / (float)%d = %6.3f\n", i, j, f1);
    return 0;
}
```

5. 標準入力関数 scanf()

(テキスト39ページ)

- 前もって, stdio.hを読み込むこと
- プログラム中では下のように記述

```
scanf(第1引数, 第2引数, 第3引数, …);
```

```
char c;  
int i, j, k;  
float f;  
double df;
```

```
scanf("%c", &c); /* 変数cに文字を入力 */  
scanf("%d", &i); /* 変数iに整数を入力 */  
scanf("%d %d %f %lf", &j, &k, &f, &df);
```

float型の変数の場合

double型の変数の場合

↑
入力変換仕様
%変換指定子

↑
&は変数のメモリアドレス
を計算する演算子

scanf関数で使用可能な変換文字

変換指定子	入力文字列の形式	変数の型
c	文字	char
s	文字列	先頭文字のアドレス
d	10進整数	int
ld	10進整数	long int
o	8進整数	int
x	16進整数	int
f	浮動小数点数	float
lf	浮動小数点数	double

scanfでは、第2引数以降に与えた変数の型を考慮しないので、注意が必要である変換指定子で示された型の値だとして、変数に代入する。

`scanf("%d", &f);` ← 変数 f の型が何であっても10進整数の値を代入する。

例題(3)

```
#include <stdio.h>

int main(int argc, const char * argv[]) {
    int i, j, sum, dif;           // 変数宣言

    printf("Enter two integer:");
    scanf("%d %d", &i, &j);     // 入力
    sum = i + j;                 // 算術演算
    dif = i - j;
    printf("%5d + %5d = %5d\n", i, j, sum);
    printf("%5d - %5d = %5d\n", i, j, dif);
    return 0;
}
```

例題(4)

```
#include <stdio.h>

int main(int argc, const char * argv[]) {
    float x, y, f0, f1;           // 変数宣言

    printf("Enter two float:");
    scanf("%f %f", &x, &y);     // 入力
    f0 = x + y;                  // 算術演算
    f1 = x * y;

    printf("%8.3f + %8.3f = %8.3f\n", x, y, f0);
    printf("%8.3f * %8.3f = %8.3f\n", x, y, f1);
    return 0;
}
```


6. Xcode統合環境の使い方(2)

- 別の資料にまとめているので、そちらを参照してください。

7. 今日の練習問題(1)

- 下のプログラムを入力して, 出力を確かめなさい.
- 各変数の内容として(77, 3.14, -255, 10.0)が正しく出力されるようにプログラムを修正しなさい.

```
#include <stdio.h>
int main()
{
    int i, j;
    unsigned int u;
    float f;
    i = 077;
    j = 3.14;
```

```
    u = -255;
    f = 10;
    printf("i = %d\n", i);
    printf("j = %d\n", j);
    printf("u = %u\n", u);
    printf("f = %d\n", f);
    return 0;
```

```
}
```

7. 今日の練習問題(2)

Level	問題
C	2つの実数 a , b に対し, a/b の値を求めるプログラムを作成せよ.
B	上のプログラムの a/b の値を書式付き出力にして表示するプログラムを作成せよ.
B	上のプログラムの a/b の値を小数第1位で四捨五入して表示するプログラムを作成せよ.
A	6.3練習問題4(鶴亀算)を解くプログラムを作成せよ.