

Programming I 0x0c



関数の活用(2010.06.21)

塩澤秀和 <http://vilab.org>

“関数”とは何か（復習）

関数を使った
プログラム

- 関数＝プログラムの部品
 - よく使う手順を部品にする
 - 部品（関数）をmainで利用

```
#include <stdio.h>

int main(void)
{
    printf("痛恨の一撃！¥n");
    printf("痛恨の一撃！¥n");
    printf("痛恨の一撃！¥n");
    return 0;
}
```

関数を使う前の
プログラム

部品化

本体

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

```
void tsukon(void)
{
    printf("痛恨の一撃！¥n");
}
```

関数

```
int main(void)
{
    tsukon();
    tsukon();
    tsukon();
    return 0;
}
```

回数を変
えてみよう

局所変数(復習)

```
#include <stdio.h>

int main(void)
{
    int dp; /* 変数定義 */

    printf("ダメージ ->");
    scanf("%d", &dp);

    printf("会心の一撃! ¥n");
    printf("%dのダメージ¥n", dp);

    return 0;
}
```

この部分を関数にしても...

```
#include <stdio.h>

void kaishin(void)
{
    printf("会心の一撃! ¥n");
    printf("%dのダメージ¥n", dp);
}

int main(void)
{
    int dp; /* 変数定義 */

    printf("ダメージ ->");
    scanf("%d", &dp);

    kaishin();
    return 0;
}
```

コンパイル
エラー!!

□ 局所変数(ローカル変数)

- 通常の変数は、定義されている {} の外には通用しない

関数の引数(復習)

□ 関数にデータを渡す方法

- 関数呼び出しの()のなかに、データ(式)を並べる
関数名(式); ← 2つ以上なら「関数名(式, 式, ...);」
- 関数に受け渡しするデータを“引数(ひきすう)”という

□ 引数をとる関数の定義

- 関数定義の()のなかに、データを受け取る変数を定義

```
void 関数名(データ型 変数, ...)  
{  
    変数を使ったプログラム  
}
```

double x, int a
などいくつでも

※ voidは空(なし)という意味

- 引数の変数は関数内でのみ通用する(ローカル変数)

関数にデータを渡す(復習)

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

仮引数
(値をもらう変数)

```
/* 関数定義 */
```

```
void buy(int price)  
{
```

```
/* データを使った処理 */
```

```
printf("まいどあり~¥n");  
printf("値段は%d円です¥n",  
    price);
```

```
}
```

関数呼び出し
(仮引数=実引数)

```
int main(void)
```

```
{
```

```
int yen;
```

```
/* 関数にデータを渡す */
```

```
buy(1480);
```

実引数
(関数に渡す値)

```
/* 変数で渡す */
```

```
yen = 4980;
```

```
buy(yen);
```

```
return 0;
```

```
}
```

受け渡し
price=1480

受け渡し
price=yen

今日のプログラム

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

```
int plus(int x, int y)
{
    int z;

    z = x + y;

    return z;
}
```

戻り値の
データ型

戻り値
(関数から返す値)

```
int main(void)
```

```
{
    int a, b, c;

    printf("a b -> ");
    scanf("%d %d", &a, &b);

    c = plus(a, b);

    printf("c = %d\n", c);

    return 0;
}
```

引数
(関数に渡す値)

関数の戻り値 (p.74)

□ 関数からデータを渡す方法

- 計算結果などの値を、関数から呼び出し元(main)に渡す
- “引数”とは逆方向 = “戻り値”、“返り値”、“返却値”

戻り値の型 関数名(引数並び)

```
{
  計算などの手順
  ...
  return 戻り値;
}
```

戻り値
の式



関数の概念

□ return文 (p.76)

- 関数を終了して、呼び出し元にすぐ戻る
- 戻り値がある関数の場合、値を“戻す”ために絶対に必要

引数と戻り値

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

```
int main(void)
```

```
{
```

```
    int a, b, c;
```

```
    printf("a b -> ");
```

```
    scanf("%d %d", &a, &b);
```

```
    c = plus(a, b);
```

```
    printf("c = %d\n", c);
```

```
    return 0;
```

```
}
```

```
int plus(int x, int y)
```

```
{
```

```
    int z;
```

```
    z = x + y;
```

```
    return z;
```

```
}
```

戻り値の
データ型

引数
(関数に渡す値)

戻り値
(関数から返す値)

計算を関数に分ける

```
#include <stdio.h>

int main(void)
{
    double pi = 3.14;
    double r, s;

    r = 5.0;
    s = pi * r * r;
    printf("%.4f¥n", s);
    return 0;
}
```

本体

部品化

```
#include <stdio.h>
double menseki(double r);
```

プロトタイプ宣言

```
int main(void)
{
    double r, s;

    r = 5.0;
    s = menseki(r);
    printf("%.4f¥n", s);
    return 0;
}
```

```
double menseki(double r)
{
    double pi = 3.1416;
    double s;

    s = pi * r * r;
    return s;
}
```

ここでは
printf
しない!!

□ 数学のような“関数”

- 「値を返す」「値を戻す」

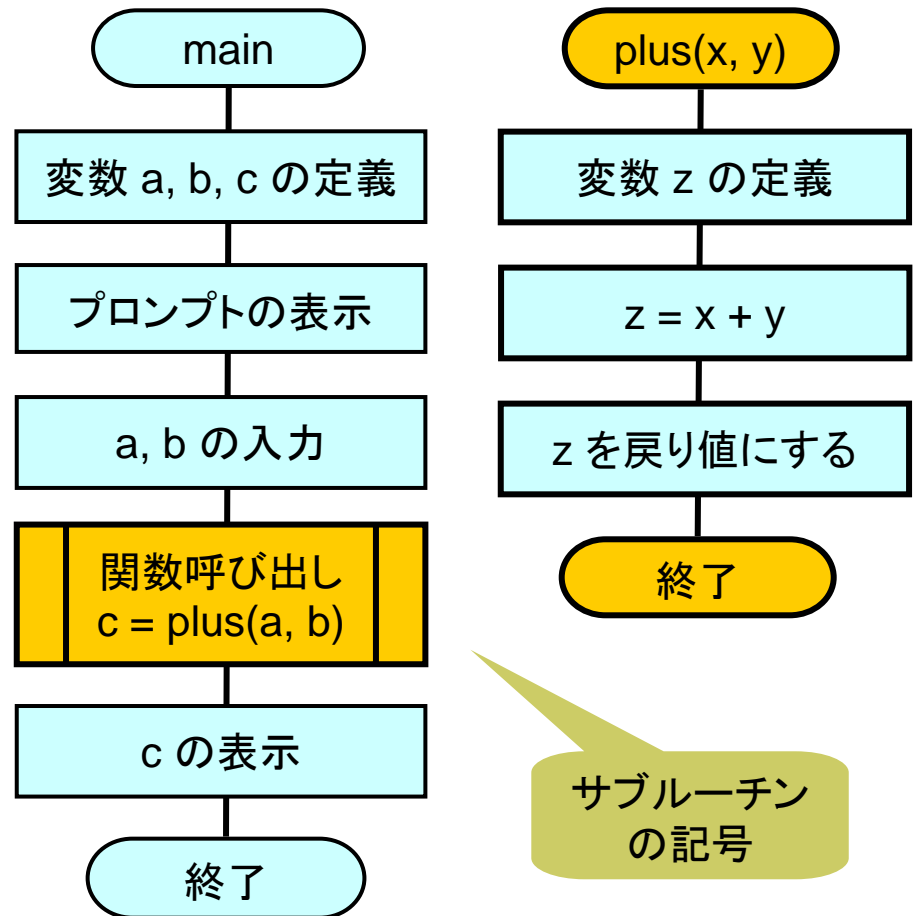
⇒ 結果は**表示 (printf) せず**に
“戻り値”として返す (return)

関数の実行をたどる

関数は部品
プログラム

- ステップ実行
 - F10キー: 1行ずつ実行
 - F11キー: 関数に入る
 - Shift+F5: デバッグ終了

- 関数の実行をたどる
 - [F10] で実行開始
 - [F10] で1行ずつ進む
 - 関数にきたら [F11]
 - [F10] で関数内を実行
 - 最後に [Shift]+[F5] で実行終了



関数(サブルーチン)のある
プログラムのフローチャート

関数の作成例

- プログラムの部品として
 - ひとことで何をする部品かズバリ言える関数がよい

```
int sum(int n, int m)
{
    int i, s;
    s = 0;
    for (i = n; i <= m; i++) {
        s += i;
    }
    return s;
}
```

```
double larger(double x,
               double y)
{
    double ans;
    if (x > y) ans = x;
    else      ans = y;
    return ans;
}
```

```
int smallest(
    int a, int b, int c)
{
    int ans;
    ans = a;
    if (b < ans) ans = b;
    if (c < ans) ans = c;
    return ans;
}
```

- C言語ではmainも関数
 - 戻り値は、プログラムの終了状態(成功かエラーかなど)
 - 正常終了なら0 ⇒ ふつうは「return 0;」しか使わない

関数を使ったプログラムの例

```
#include <stdio.h>
int read_age(void);
int entrance_fee(int age);
void write_yen(int yen);

int main(void)
{
    int a, y;

    /* キーボードから年齢を入力 */
    a = read_age();

    /* 年齢から入場料を求める */
    y = entrance_fee(a);

    /* 入場料を表示して終了する */
    write_yen(y);
    return 0;
}
```

```
int read_age(void)
{
    int age;
    printf("年齢? ");
    scanf("%d", &age);
    return age;
}

int entrance_fee(int age)
{
    if (age < 6) return 0;
    if (age < 12) return 1500;
    return 3000;
}

void write_yen(int yen)
{
    printf("価格%d円¥n", yen);
}
```

elseがなくてもいい
理由を考えよ

数学関数 (p.73)

□ 数学関数ライブラリ

- あらかじめ用意されている関数
例: $y = \sin(x) + \cos(x)$;
- `#include <math.h>` が必要

□ 主な数学関数 (p.191)

- 数値はdoubleが基本

<code>pow(x, y)</code>	x^y	<code>sqrt(x)</code>	\sqrt{x}
<code>exp(x)</code>	e^x	<code>log(x)</code>	$\log_e x$

- 三角関数

`sin`, `cos`, `tan`, `asin`, `acos`, `atan`, `atan2`

角度の単位は [ラジアン] = [度] $\times \pi / 180^\circ$

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

int main(void)
{
    double x, y;

    printf("x = ");
    scanf("%lf", &x);

    y = sqrt(x);
    printf("y = %f\n", y);
    return 0;
}
```

演習問題

- 12a. 「計算を関数に分ける」のプログラムを三角形の面積を計算するように改造しなさい。menseki は底辺と高さという2つの引数をとる関数に改造し、mainも修正する必要がある。
- 12b. 引数として整数nをとり、戻り値として、nが正ならば+1を、nが負ならば-1を、nがゼロならば0を返す sign という名前の関数を作成し、適当なmain関数をつけて動作確認をしなさい。
- 関数プロトタイプ宣言は `int sign(int n);` となる。
- 12c. 倍精度型の引数を3つとり、最大値を返す関数 max3 を作成しなさい。適当なmain関数をつけて動作確認をしなさい。
- 12d. 数学の表現で $f(x) = 2x + 1$ となる関数をC言語で作成し、mainで呼び出して $f(0)$, $f(1)$, ..., $f(10)$ の値を表示させなさい。
- 次回までの課題: **リスト5-7~5-12** を入力して教科書を予習