Programming I 0x0d

関数の応用と記憶域クラス
（2010.06.28）

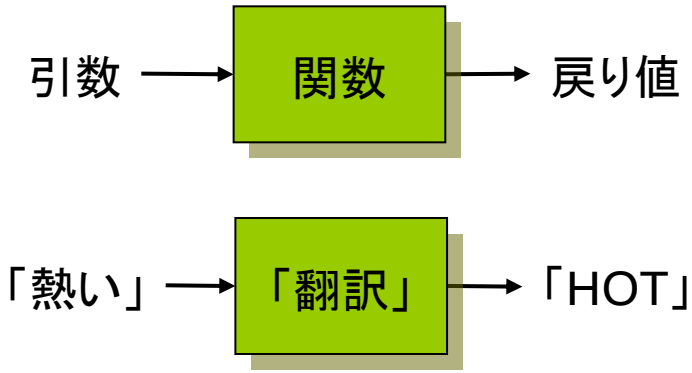
塩澤秀和 http://vilab.org

関数の戻り値（復習）

- 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（復習）

```
#include <stdio.h>

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

   z = x + y;

   return z;
}

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;
}
```

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）

数学関数（復習）

- 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
    pow(x, y)　x^y　　sqrt(x)　$\sqrt{x}$
    exp(x)　　e^x　　log(x)　$\log_e x$
  - 三角関数
    sin, cos, tan, asin, acos, atan, atan2
    角度の単位は［ラジアン］＝［度］×π／180°

```
#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;
}
```

C言語の関数

- <math.h>
  - pow(x, y)　x^y
  - sqrt(x)　$\sqrt{x}$
  - log(x)　log(x)
  - sin, cos, tan　三角関数
- <stdio.h>
  - printf, scanf　書式つき入出力
  - putchar(c)　1文字表示
  - getchar()　1文字入力
- <ctype.h>
  - isdigit(c)　文字が数字か判定
  - isalpha(c)　英文字か判定
  - isupper(c)　大文字か判定
  - tolower(c)　大文字→小文字
- <stdlib.h>
  - rand()　乱数の発生
  - srand(seed)　乱数の初期設定
  - exit(code)　プログラム終了
- <time.h>
  - time(NULL)　現在時刻の取得
- その他
  - main　メインルーチン
  - 自作の関数　サブルーチン
  - ライブラリ関数一覧（p.191）
- 関数でないもの
  - 制御文　if, while, for, return, ...
  - プリプロセッサ指令　#include, ...

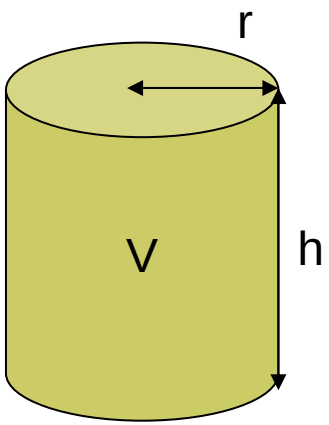
ライブラリ関数の利用例

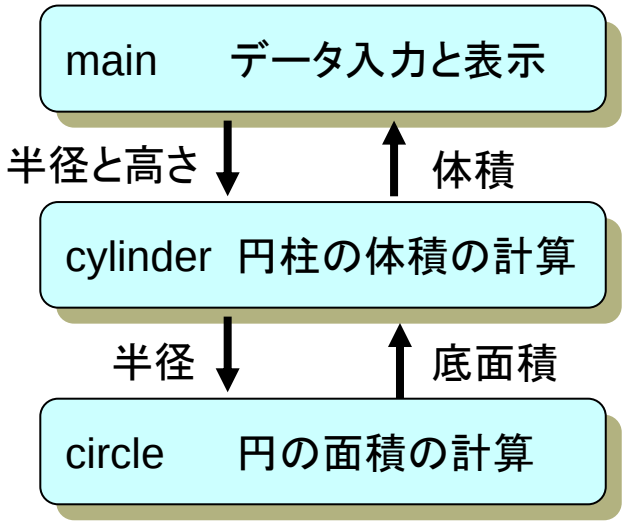
```
/* rand関数による乱数の発生 */
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void)
{
  int d1, d2;

  /* 現在時刻を使って乱数を初期化 */
  srand(time(NULL));
  printf("サイコロ2個振ります¥n");

  /* 発生した乱数を1～6に変換 */
  d1 = rand() % 6 + 1;
  d2 = rand() % 6 + 1;
  printf("%d %d¥n", d1, d2);
  return 0;
}
```

```
/* 文字種の判定と変換 */
#include <stdio.h>
#include <ctype.h>

int main(void)
{
  char c;

  printf("文字 -> ");
  c = (char) getchar();

  if (islower(c)) {
    c = toupper(c);
    printf("%cの小文字¥n", c);
  } else {
    printf("小文字でない¥n");
  }
  return 0;
}
```

関数の中で関数を使う

```
#include <stdio.h>

/* 円の面積 */
double circle(double r)
{
  double pi = 3.1415926536;
  return pi * r * r;
}

/* 円柱の体積＝円の面積×高さ */
double cylinder(double r,
   double h)
{
  return circle(r) * h;
}

int main(void)
{
  double r, h, V;

  printf("r = ");
  scanf("%lf", &r);
  printf("h = ");
  scanf("%lf", &h);

  V = cylinder(r, h);
  printf("V = %g¥n", V);
  return 0;
}
```





関数の呼び出し関係図

再帰（recursion）（p.86）

- 自分自身を呼び出す関数
  - たとえば、毎月100円ずつ貯金するなら…
    現在の全貯金 ＝ 先月までの全貯金 ＋ 今月の貯金

```
int chokin(int n)
{
  if (n == 0) return 0;
  return chokin(n-1) + 100;
}
```

$$chokin(n) = \begin{cases} 0 & (n = 0 \text{のとき}) \\ chokin(n-1) + 100 \end{cases}$$

nヶ月目の貯金の式

- C言語ではあまり使わない
  - 階層構造のデータ（フォルダなど）を処理するときに有効
  - この例や階乗（p.86）は説明用 ⇒ 普通はループを使う
  - 一般的にループは再帰でも書けるし、逆も可能

変数のスコープと寿命

- スコープ（通用範囲）
  - ソースコードの中で、変数や関数が “見える”範囲
  - 局所変数（ローカル） ⇔ 大域変数（グローバル）
- 記憶寿命
  - プログラム実行中に、変数が“存在”している期間
  - 自動変数 ⇔ 静的変数（p.87）
- 「宣言」と「定義」の区別
  - 宣言： 変数や関数の名前と型を明示（登録）すること
    - 関数プロトタイプ宣言は、定義も実行もせずに宣言だけをする
  - 定義： 新しい変数や関数を作ること（定義は宣言でもある）
    - 「代入」は新しい変数を作らないので、定義でも宣言でもない

変数の種類（記憶域クラス）

	定義場所	スコープ（通用範囲）	記憶寿命
ローカル変数 （局所変数）	関数の内部 関数の仮引数	関数の内部 ブロックで定義された場合はその内部	自動変数（auto） 関数の実行中のみ （p.87）
			静的変数（static） プログラムの実行中 （p.90）
グローバル変数 （大域変数） ※	全関数の外部 （外部変数）	プログラム全体 † 定義場所より前や別ファイルで使うには、別途、extern による宣言が必要（p.89）	プログラムの実行中 （静的記憶領域）

※ グローバル変数は、むやみに使うとプログラムが分かりにくくなる
† C言語には、このほかにグローバルな static 変数（ファイルスコープ）がある

グローバル変数の例

```
#include <stdio.h>

int points = 100;

void tokuten(int p)
{
  points += p;
  printf("得点%dポイント¥n", p);
}

void genten(int p)
{
  points -= p;
  printf("減点%dポイント¥n", p);
}
```

関数の外部で定義

```
int main(void)
{
  int no;

  while (points > 0) {
    printf("1.得点 2.減点:");
    scanf("%d", &no);
    if (no == 1)
      tokuten(10);
    else
      genten(10);
    printf("現在%dポイント¥n",
           points);
  }
  return 0;
}
```

演習問題

13a. 点P(x, y)の座標（double型）をキーボードから読み込み、Pから原点までの距離を表示するプログラムを作成しなさい。

- 距離は $\sqrt{(x^2 + y^2)}$ で計算する。数学ライブラリ関数の sqrt を用いる。

13b. 引数として3つの整数（int）を入力すると、それらの平均を実数（double）で返す関数を作成しなさい。適当なmain関数をつけて動作確認をしなさい。（出題範囲としては前回）

13c. 点P(x, y)の座標を引数にとり、Pから原点までの距離をreturnで戻す関数を作成しなさい。それを利用するように、13aのプログラムを書き換えなさい。

13d. グローバル変数langの値が0なら「こんにちは」と表示し、0以外なら「Hello」と表示する関数を作成しなさい。これを使って言語を切り替えて表示するmain関数も作成しなさい。

- 関数のプロトタイプ宣言： void print_greeting(void);

今後の日程

- 今後の日程
  - 7 / 5（月） プリプロセッサ
  - 7 /12（月） 演習
  - 7 /19（祝） 休講
  - 7/ 21（水） 統一テスト 17:00～19:00 教室未定

- 統一テスト
  - 日時：7月21日（水） 17:00～19:00
  - 場所：教室未定
  - 形式：中間試験と同じような形式
  - 範囲：教科書最初～プリプロセッサ（p.100）
    主に、ループと関数（自作） 6割以上は関数がらみ？