

### 【関数の使用】

関数とは「機能」のことで、英語では function と称します。

「printf 関数」はコマンドプロンプトに出力する機能、「scanf 関数」はコマンドプロンプトから入力する機能になります。

英語では関数を使用することを「calling function」と言い、日本語でも関数を使用することを「関数を呼び出す」と表現します。

C 言語にはさまざまな関数が用意されており「標準関数」と称します。

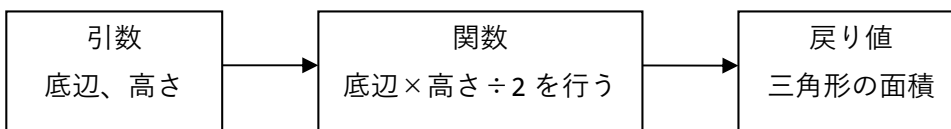
#### <引数と戻り値>

「引数（ひきすう）」は関数を使用するにあたって必要な値で、「戻り値」は関数を使用した結果です。

例えば、三角形の面積を取得する関数があったとします。

三角形の面積を求めるには「底辺」と「高さ」が必要です。これが引数になります。

演算した結果、三角形の面積が取得できます。これが戻り値になります。



引数を入れると戻り値が出てくる。

関数によって引数の数は異なりますが、戻り値はひとつです。

戻り値が無い関数もありますが、詳しくは「関数の作成」で学習します。

#### <関数の使用>

```
戻り値を入れる変数 = 関数名( 引数 );
```

- 引数が複数ある場合は、(カンマ) で区切って記述すること。
- 引数がない場合は()のみ記述すること。
- 引数の順番、データ型を揃えること。
- 関数の戻り値のデータ型と、戻り値を入れる変数のデータ型は同じ、もしくは変換可能であること。
- 戻り値が無い関数や、戻り値を取らない場合は「戻り値を入れる変数」を記述する必要はない。

<例文>

数値 x の y 乗を算出する pow 関数を使ってみます。

第 1 引数に数値、第 2 引数に乗数を指定すると、戻り値に「べき乗 (累乗)」が返ってきます。

第 1 引数、第 2 引数、戻り値のデータ型は double 型になっています。

```
double answer;          /* 戻り値を入れる変数 */
answer = pow(2, 3);     /* 2 の 3 乗を算出 */
printf("%f", answer);  /* 8.0 と出力 */
```

pow 関数を使用するのは「math.h」をインクルードする必要があります。

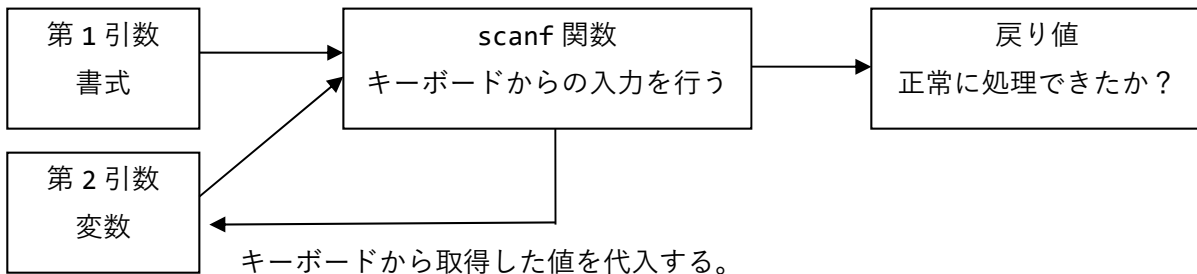
<参照渡し>

```
int var;
scanf("%d", &var);    /* キーボードからの入力値を変数 var に代入する */
```

scanf 関数はキーボードから取得した値を戻り値ではなく、第 2 引数に代入しています。

scanf 関数の戻り値は「キーボードからの入力が正常に処理されたかどうか？」を返すので、キーボードから取得した値は引数に代入しているのです。

このように関数の結果を取得する引数を「参照渡し」と称し、一部の標準関数で使用されています。



【インクルード】

```
#include <ヘッダーファイル名>
```

標準関数はヘッダーファイル（拡張子は h）に記述されており、役割ごとに分類されています。標準関数が記述されているヘッダーファイルを「標準ライブラリ」と称し、C コンパイラに含まれています。標準関数を使用するには、その関数が記述されているヘッダーファイルを組み込む＝インクルードする必要があります。

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

「stdio.h」は standard input output の略で、入出力に関する関数が記述されているヘッダーファイルです。

printf 関数、scanf 関数を使用する場合は stdio.h をインクルードする必要があります。

【メイン関数】

メイン関数は特殊な関数で、プログラムを実行した時に最初に実行される関数です。関数名が必ず「main」になります。

```
int main(void) {  
    処理を記述  
    return 0;  
}
```