2018.5.22 小林 勝

7. 関数の基本構造と役割(2)

■ 関数の基本書式

(1) 戻り値なし、引数なし

```
void 関数名(void){
    処理
    return ;   ← 省略可
}
```

(2) 戻り値あり、引数なし

```
戻り値の型 関数名(void){
    処理
    return 戻り値 ;
}
```

(3) 戻り値なし、引数あり

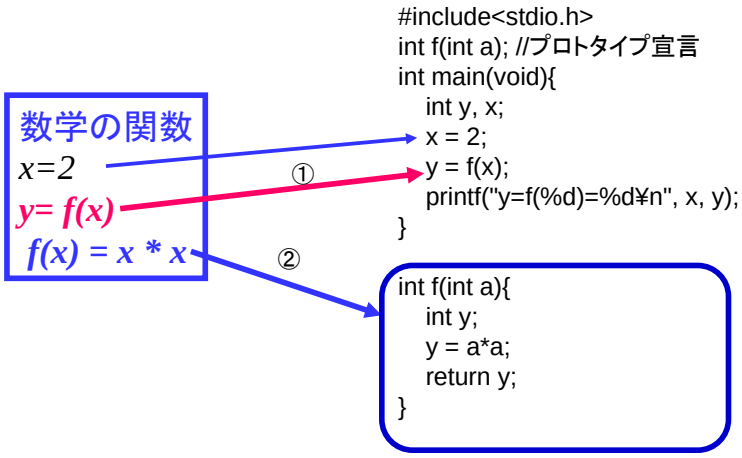
```
void 関数名(引数){
    処理
    return ;   ← 省略可
}
```

(4) 戻り値あり、引数あり

```
戻り値の型 関数名(引数){
    処理
    return 戻り値 ;
}
```

■ 「戻り値がある関数」は、「数学の関数」と同じように使える

戻り値を利用すると、数学の関数のように利用することができる。下の左側に「数学の関数」、右側に「戻り値のある関数」のプログラムを記す。①の左側に示した数学の y=f(x)は、右側のプログラム内では y=f(x)と記し、②に示す f(x)の具体的な計算式 x*x は、関数 f(int a)内に記述することができる。



＜図 1 関数の利用(戻り値のある関数、数学的な利用方法)＞

■ 関数の使用例

図 2 に関数の利用例を示す。関数冒頭の int の戻り値とは、下の例ならば「return x」「return y」の x と y の値のことである。従って、戻り値の型は、変数 x と変数 y の型になる。引数は、関数を呼び出す際に、関数に値を渡す役割を持つ。図 1 では、変数 x の値を関数 f(x)に渡している。

```
// 2 つの整数値の大きさ比較(小さな値を見つける)
#include<stdio.h>

// 小さな値を見つける関数(min)
int min( x, y){
        int x,y;                        //変数 x,y 宣言

        if( x<y )
                return x;               //戻り値(小さな値)
        else
                return y;               //戻り値(大きな値)
}

// メイン関数
int main(void){
        int a,b,c;                      //変数 a,b,c 宣言

        scanf( "%d",&a );                       //a 値入力
        scanf( "%d",&b );                       //b 値入力

        c=min( a,b );                           //関数呼び出し

        printf( "小さい値は%d です¥n",c );          //小さい値表示

        return;
}
```

＜図 2 関数の使用例＞