

2.1. 関数

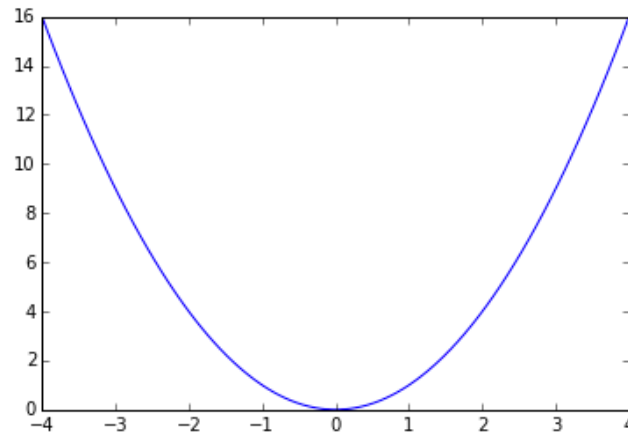
関数の基礎（引数、戻り値）
値渡し

関数の中で他の関数を呼ぶ

関数の基礎：数学の例

$$f(x) = x^2$$

f という関数は、x という値を入力されると、その値を2乗して返す



- ⇒ 関数とは、入力された何かに対して、決められた処理を行い、その結果を返す働きを持つもの
- ⇒ プログラミングにおいても同様

関数の基礎：f(x)=x^2のプログラム例

List1.1

```
#include <stdio.h>
#pragma warning( disable : 4996 )
int square(int x) {
    return x * x;
}

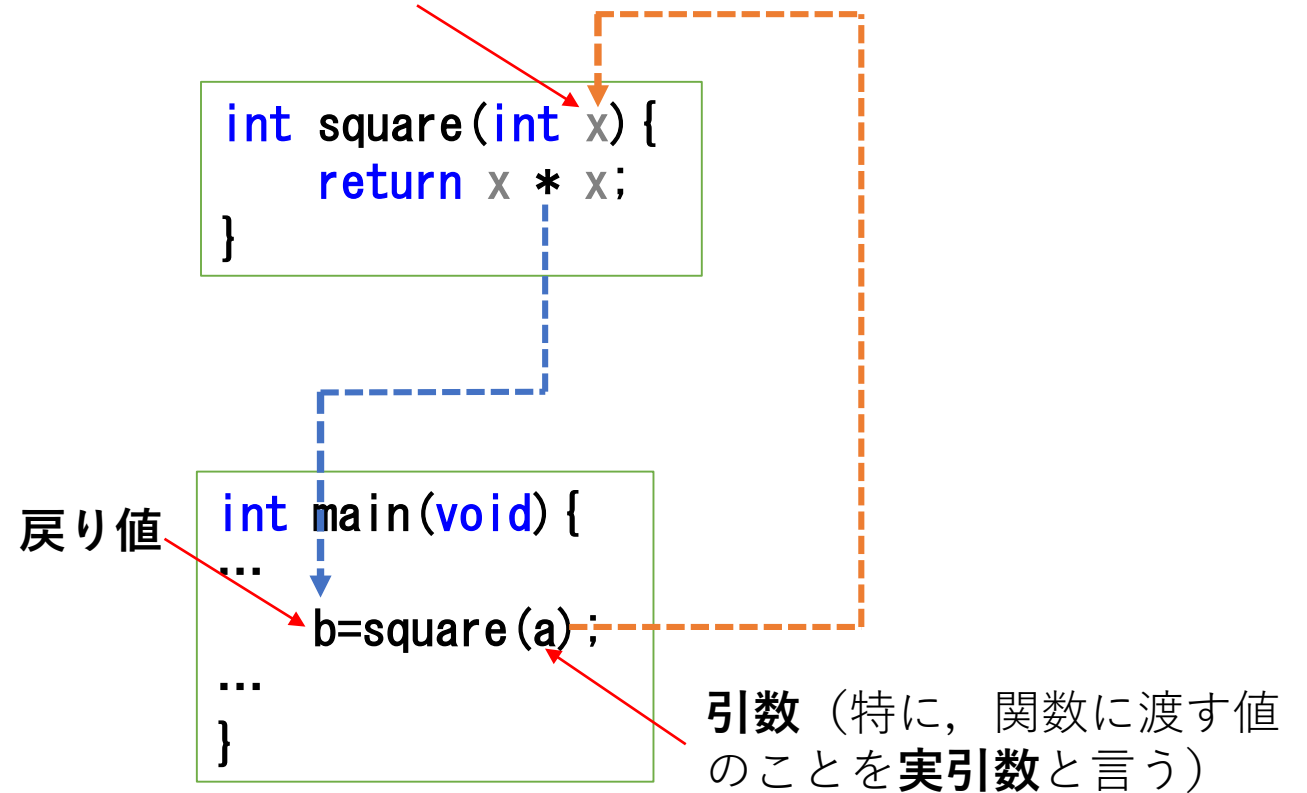
int main(void) {
    int a, b;

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

    b=square(a);
    printf("%d %d\n", a, b);

    return 0;
}
```

引数（特に、関数に渡された側では
仮引数と言う。ここではxが仮引数）



※ポイント： `int square(int x)` について
`square`の手前の`int`は戻り値の型、`x`の手前の`int`は引数の型

値渡し

List1.2

```
#include <stdio.h>
#pragma warning( disable : 4996 )
int power(int x, int n) {
    int y = 1;
    while (n-- > 0) y *= x;

    return y;
}

int main(void) {
    int x, n;
    printf("Input x n>");
    scanf("%d %d", &x, &n);
    printf("x=%d n=%d x^n=%d", x, n, power(x, n));

    return 0;
}
```

power()の中でnが減算されても、main()内のnに変化はない

実行結果

```
Input x n>3 4
x=3 n=4 x^n=81
```

```
int power(int x, int n) {
    int y = 1;
    while (n-- > 0) y *= x;

    return y;
}
```

```
int main(void) {
    ...
    power(x, n);
    ...
}
```

値がコピー
される

※ポイント：関数へ値を送るとき、その値はコピーされる（つまり、関数内でその値が変更されても、元の値には影響がない）

※ポイント：二つ以上の値を引数とするときは、その順番通りにコピーされる

【例題】 整数の大小を比較する関数

List1.3

```
#include <stdio.h>
#pragma warning( disable : 4996 )
int max2(int a, int b) {
    if (a > b)
        return a;
    else
        return b;
}
int main(void) {
    int a, b;
    printf("Input a b>");
    scanf("%d %d", &a, &b);
    printf("max: %d ¥n", max2(a, b));

    return 0;
}
```

関数の中で他の関数を呼ぶ

List1.4

```
#include <stdio.h>
#pragma warning( disable : 4996 )
```

```
void print_number( int n ) {
    printf( "%-3d", n );
}
```

print_number関数はprint_star関数の前で宣言しないといけない

```
void print_star( int n ) {
    print_number( n );
    while ( n-- > 0 )
        printf( "*" );
    printf( "\n" );
}
```

print_star関数内でprint_number関数を呼び出す

```
int main( void ) {
    int i;
    for ( i = 0; i < 5; i++ )
        print_star( i );

    return 0;
}
```

実行結果

```
0
1  *
2  **
3  ***
4  ****
```

※ポイント：呼び出される関数は、常に呼び出す側の関数より手前に書かれないといけない。従ってmain関数は常に一番下。

演習問題

A. 整数の絶対値を求める関数を設計せよ

B. 1から整数 n までの和を求める関数を設計せよ

発展問題

- A. 3つの整数のうち最大の整数を求める関数を設計せよ
- B. 下のような実行結果が得られる関数を設計せよ

実行結果



```
0
 * 1
** 2
*** 3
**** 4
```