

# C言語6

関数

例題

# 関数 (function) とは?

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

```
int main(void){
```

```
    文;
```

```
    return 0;
```

```
}
```

赤い部分をmain関数(main function)と言う。

# 関数の定義

戻り値の型 関数名(引数リスト)

{

文;

…;

return 式;

}

# 例1 戻り値が無い場合 引数の数だけ \* を表示する

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

```
// 星 * を表示する関数
```

```
int stars(int n){  
    int i;  
    for(i=0; i<n; i++)  
        printf("*");  
    printf("¥n");  
    return 0;  
}
```

```
// main関数
```

```
int main(void){  
    int num = 7;  
    stars(num);  
    return 0;  
}
```

【実行結果】

\*\*\*\*\*

## 例1-2 mコの空白の後にnコの\*を表示するプログラム

```
int stars(int m, int n){
    int i;
    for(i=0; i<m; i++){
        printf(" ");
    }
    for(i=0; i<n; i++){
        printf("*");
    }
    printf("¥n");
    return 0;
}

int main(void){
    int a,i;
    printf("正の整数を入力して下さい。");
    scanf("%d", &a);

    for(i=a-1; i >=0; i--){
        stars(i,a-i);
    }
    return 0;
}
```

### 【実行例】

正の整数を入力して下さい。7

```
 *
**
***
****
*****
*****
*****
```

# 例2 戻り値がある場合一足し算

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

```
// 足し算を行う関数
```

```
int plus(int x, int y){  
    int z;  
    z = x+y;  
    return z;  
}
```

【実行結果】

8

```
// main関数
```

```
int main(void){  
  
    int a=3;  
    int b=5;  
    printf("%d¥n", plus(a, b));  
  
    return 0;  
}
```

# 例3 最大値を返す

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

```
int max(int x, int y){  
    if(x>y)  
        return x;  
    else  
        return y;  
}
```

```
int main(void){  
  
    int n1, n2;  
  
    printf("2つの整数を入力してください。¥n");  
    printf("整数1:"); scanf("%d", &n1);  
    printf("整数2:"); scanf("%d", &n2);  
  
    printf("大きい方の値は%dです。¥n", max(n1, n2));  
  
    return 0;  
}
```

# 例4 3値の最大値を返す

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

```
int max3(int x, int y, int z){
```

```
    int max=x;
```

```
    if(y>x)
        max=y;
```

```
    if(z>max)
        max = z;
```

```
    return max;
```

```
}
```

```
int main(void){
```

```
    int n1, n2, n3;
    int mx;
```

```
    printf("3つの整数を入力してください。¥n");
    printf("整数1:"); scanf("%d", &n1);
    printf("整数2:"); scanf("%d", &n2);
    printf("整数3:"); scanf("%d", &n3);
```

```
    mx = max3(n1, n2, n3);
```

```
    printf("最大値は%dです。¥n", mx);
```

```
    return 0;
```

```
}
```

## 例5 3値の最大値(他の関数を呼び出す。)

```
#include<stdio.h>
// 2値の最大値を求める関数
int max(int x, int y){
    if(x>y)
        return x;
    else
        return y;
}
// 2値の最大値を求める関数を呼び出して、3値の最大値を求める関数
int max3(int x, int y, int z){
    int m;
    m=max(x, y);
    m=max(m, z);
    return m;
}

int main(void){
    int n1, n2, n3;
    int mx;

    printf("3つの整数を入力してください。¥n");
    printf("整数1:"); scanf("%d", &n1);
    printf("整数2:"); scanf("%d", &n2);
    printf("整数3:"); scanf("%d", &n3);

    mx = max3(n1, n2, n3);

    printf("最大値は%dです。¥n", mx);

    return 0;
}
```

# 関数式マクロ

- 定義

#define マクロ名(引数) (式)

- 例 最大値を求める

```
#define MAX(x, y) (x > y ? x : y)
```

( $x > y ? x : y$ ) の意味

もし  $x > y$  ならば  $x$  を、そうでなければ  $y$  を返さない。

# 例題1 最大値を求める

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

```
#define MAX(x, y) (x > y ? x : y)
```

```
int main(void){
```

```
    int n1, n2, ans;
```

```
    printf("1番目の整数を入力してください。¥n");  
    scanf("%d", &n1);
```

```
    printf("2番目の整数を入力してください。¥n");  
    scanf("%d", &n2);
```

```
    ans=MAX(n1, n2);
```

```
    printf("最大値は%dです。¥n", ans);
```

```
    return 0;
```

```
}
```

## 例題2 2値の和を求める

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

```
#define add(x, y) (x+y)
```

```
int main(void){
```

```
    int n1, n2, sum;
```

```
    printf("整数1を入力:"); scanf("%d", &n1);
```

```
    printf("整数2を入力:"); scanf("%d", &n2);
```

```
    sum = add(n1, n2);
```

```
    printf("%d+%d=%d¥n", n1, n2, sum);
```

```
    return 0;
```

```
}
```

### 実行例

整数1を入力:23

整数2を入力:73

23+73=96

# 再帰(自分自身を呼び出す)

- 例 階乗の計算

```
int factorial(int n){  
    if(n>0)  
        return n*factorial(n-1);  
    else  
        return 1;  
}
```

## $n!$ の再帰的定義

- (1)  $n=0$  ならば1を返す。
- (2)  $n>0$  ならば  $n*(n-1)!$ を返す。

# 例8

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

```
int factorial(int n){  
    if(n>0)  
        return n*factorial(n-1);  
    else  
        return 1;  
}
```

```
int main(void){  
    int num;  
    printf("整数を入力:");  
    scanf("%d", &num);  
    printf("%d!=%d¥n", num, factorial(num));  
    return 0;  
}
```

## 実行結果

整数を入力:5

5!=120

# 再帰関数の例2

## 1からnまでの和を求める再帰関数sum(n)

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

```
int sum(int n){  
    if(n>0)  
        return n+sum(n-1);  
    else  
        return 0;  
}
```

```
int main(void){  
    int num;  
    printf("正の整数を入力:");  
    scanf("%d", &num);  
    printf("1から%dまでの整数の和は%d¥n", num,  
sum(num));  
    return 0;  
}
```

### 【実行例】

正の整数を入力:20

1から20までの整数の和は210

# 関数Sum(n)を再帰を使わないで書いた場合

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

```
int sum(int n){
```

```
    int i, s=0;
```

```
    for(i=1; i<=n; i++)
```

```
        s+=i;
```

```
    return s;
```

```
}
```

```
int main(void){
```

```
    int num;
```

```
    printf("正の整数を入力:");
```

```
    scanf("%d", &num);
```

```
    printf("1から%dまでの整数の和は%d¥n", num, sum(num));
```

```
    return 0;
```

```
}
```

## 【実行例】

正の整数を入力:10

1から10までの整数の和は55

# 再帰関数の例3 rからnまでの和

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

```
int sum(int n, int r){  
    if(n>=r)  
        return n+sum(n-1, r);  
    else  
        return 0;  
}
```

```
int main(void){  
    int n1, n2;  
    printf("正の整数1を入力:"); scanf("%d", &n1);  
    printf("正の整数2を入力:"); scanf("%d", &n2);  
    printf("%dから%dまでの整数の和は%dです。¥n", n1, n2, sum(n2, n1));  
    return 0;  
}
```

## 【実行例】

正の整数1を入力:5

正の整数2を入力:12

5から12までの整数の和は68です。

# グローバル変数

- グローバル変数は、プログラムの開始から終了まで値を記憶している。

【例9】

```
#include<stdio.h>
int a=0; //グローバル変数です。
void func(void){
    a++;
}
int main(void){
    printf("a=%d\n", a);
    func(); //func()関数が呼び出されると、
    printf("a=%d\n", a); //aは1増える。
    return 0;
}
```

【実行結果】

```
a=0
a=1
```