

C言語講座

第五回

関数

▶ 関数とは？

- ▶ C言語は関数で成り立っている
- ▶ 関数というのはmain関数やprintf関数、scanf関数などのこと
- ▶ printfやscanfなどは仕組みがわからなくても動作さえわかれば引数を与えるだけで欲しい結果を得ることができる
- ▶ 既存の関数だけでなく自分で関数を定義し、使うことができる
- ▶ 関数をうまく組み合わせて使うことで書く量を減らし、すっきりしたプログラムになる
- ▶ 大きなプログラムを作るときは使わないと大変

▶ 今回はその関数を自分で作る！



関数 2

▶ 関数の構造

```
[記憶クラス][型指定]関数名([引数])  
{  
    ～関数の中身～  
}
```

▶ [記憶クラス]

- ▶ static(静的関数)、extern(外部関数)の事で省略するとexternになる。

▶ [型指定]

- ▶ intやcharなど。配列は不可。

▶ [引数]

- ▶ どんな値をこの関数の引数にするのか表記しておく。
 - ▶ 何も引数がない場合はvoidと表記しておく。
-

記憶クラス

- ▶ 静的関数: static
 - ▶ 関数の有効範囲が定義されているファイル内のみ有効
- ▶ 外部関数: extern
 - ▶ 他のモジュールでも参照できるように定義する方法
- ▶ 知能情報は2年で使うことになるので頭の片隅に置いておいたらいいよ



関数 3

▶ main関数

- ▶ C言語の核となる関数
- ▶ プログラムが実行されるときはこれの頭から実行される
- ▶ C言語の動作はmain関数に始まり、main関数に終わる



関数化例

[普通]

```
#include <stdio.h>
//main関数
int main(void){
    int x,y,ans;
    scanf("%d",&x);
    scanf("%d",&y);
    ans = (x*x + y*y);
    printf("%d¥n",ans);
    return 0;
}
```

[関数化]

```
#include <stdio.h>
//関数hoge宣言
int hoge(int a, int b){
    return(a*a+b*b);
}
//main関数
{
    int x,y,ans;
    scanf("%d",&x);
    scanf("%d",&y);
    ans = hoge( x, y );
    printf("%d¥n",ans);
    return 0;
}
```



関数 4

▶ プロトタイプ宣言

- ▶ 自分で関数を作るとき、main関数の前に関数宣言をして置かなければならない
- ▶ プログラムはmain関数から実行されるのでmain関数より上に関数を宣言しなければその関数を見つけることができなくてエラーになる
- ▶ プロトタイプ宣言はとりあえずの宣言をmain関数より上に表記しておくことで、関数の中身をどこにでも書くことができるようにする方法



プロトタイプ宣言例

[普通の関数宣言]

```
#include <stdio.h>
//関数y宣言
void y( int x ){
    int ans,i;
    for( i=0 ; i<x ; i++ ){
        ans += x;
        printf(“%d¥n“,ans);
    }
}
//main関数
int main(void){
    int a;
    scanf(“%d”,&a);
    y ( a );
    return 0;
}
```

[プロトタイプ宣言]

```
#include <stdio.h>
//プロトタイプ宣言
void y( int );
//main関数
int main(void){
    int a;
    scanf(“%d”,&a);
    y ( a );
    return 0;
}
//関数y宣言
void y( int x ){
    int ans,i;
    for( i=0 ; i<x ; i++ ){
        ans += x;
        printf(“%d¥n“,ans);
    }
}
```



関数 5

▶ 再帰関数

- ▶ 関数の中で自身の関数を呼び出す方法
- ▶ 動作が遅くなる上、プログラムが解読しにくくなる
- ▶ 階乗や数列計算で使うとスマートにまとめられる



再帰関数例

[nの階乗のプログラム]

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

```
//プロトタイプ宣言
```

```
int hoge(int);
```

```
int main()
```

```
{
```

```
    int n;
```

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

```
    int i;
```

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

```
        //関数呼出
```

```
        printf("%2d! = %10d¥n",i,func(i));
```

```
    }
```

```
}
```

```
//hoge関数
```

```
int hoge(int n)
```

```
{
```

```
    //nがのときを返して関数終了
```

```
    if(n==1)
```

```
        return 1;
```

```
    //nが以外するときhogeにn-1を引数に与えて
```

```
    //関数呼び出してnと掛けて返す
```

```
    else
```

```
        return n*hoge(n-1);
```

```
}
```

練習

- ▶ ピラミッドを作るプログラムを関数で作る

