

C言語講座第 3 回～関数～

担当 田中 駿

まず関数ってなんだ.....？

$$y = ax + b ?$$

難しそう.....

数学かよ死んだ.....



実は！

今まで使っていたmainも関数なんです！

printfも関数、scanfも関数！！

今回は自分で関数を作っちゃいましょう！！

どうやれば関数と呼べるものが
作れるのか？

int ,void

```
型名 名前(代入されてくる変数の型 )  
{  
    処理....  
}
```

これだけで関数として成立する！！！！

こんなんでも自作関数！ (変数xを2にして返す関数)

- #include <stdio.h>

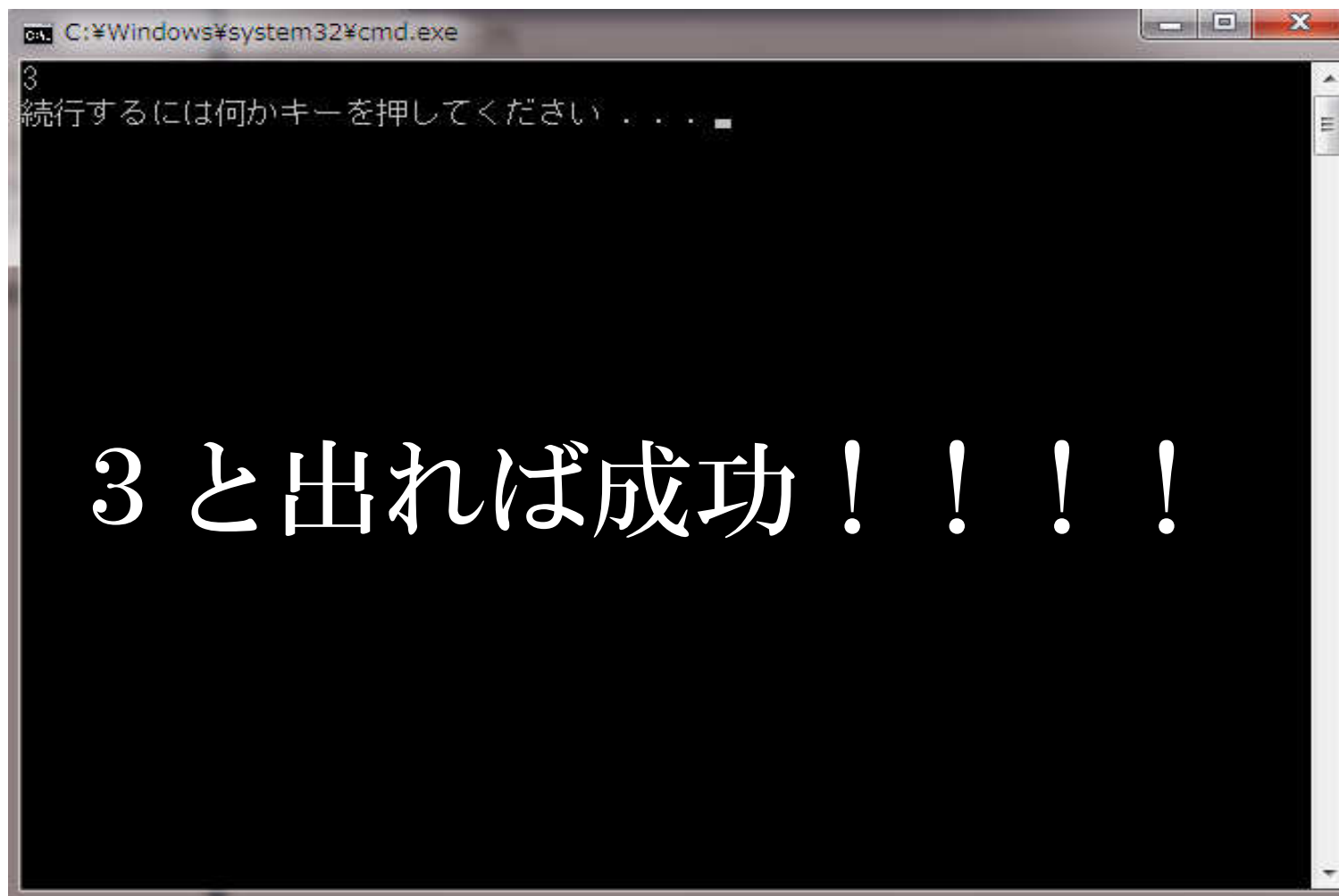
```
int Kansuu(int x){  
    x=2;  
    return x;  
}
```

```
int main(){  
    int x ;  
    x=1;  
    x=x+Kansuu(x);  
    printf("%d¥n",x);  
    return 0;  
}
```

今まで適当にmainに **return 0;**と書いていたと思うけど、この**"return"**とは"自作関数に変数を代入してくれた場所に処理結果を返すためのもの"だった！！

ここでxを自作関数に代入している

実行結果



```
C:\Windows\system32\cmd.exe
3
続行するには何かキーを押してください . . .
```

3 と出れば成功!!!!

問題！！

代入されてきた変数に数字を足して、
その結果を表示するプログラムを作っ
てみよう！！！！！！

さっき作ったプログラムを参考にしよ
う！！

答え

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

```
void Kansuu(int x)
```

```
{
```

```
    int y=2;
```

```
    x=x+y;
```

```
    printf("%d¥n",x);
```

```
}
```

```
int main()
```

```
{
```

```
    int x=2;
```

```
    Kansuu(x);
```

```
    return 0;
```

```
}
```

値を返す必要のない関数の型名はvoid
にするとreturn 0;
がいらない！

自作関数の中に
printfを書く！！

関数を先に書くのは面倒臭い....?

そんなあなたに「**プロトタイプ宣言**」！
名前は難しそうだけどただ関数名と引数の型を
先に書くだけ！

void Kansuu(int);

```
void main()
{
    ...処理
    Kansuu( x );
}
```

```
void Kansuu(int x)
{
    ...処理
}
```



引数というのは関数の
カッコ内に書かれて
いる変数のこと！



関数の理解を深めよう！！

次は関数の仕組みを深く理解出来る

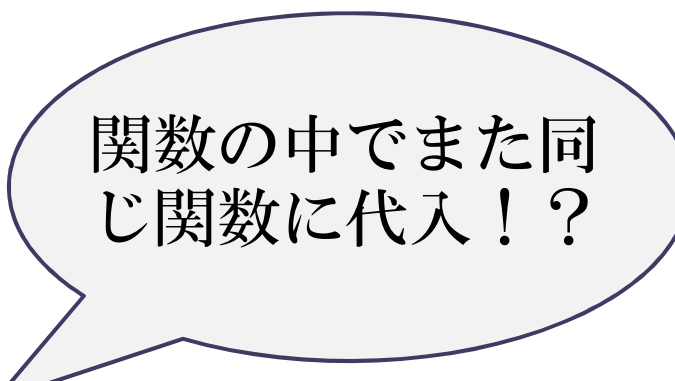
再帰関数

を説明していきます！

まずはプログラムを見よう！！

```
#include <stdio.h>
int SaikiKansuu(int);

void main(){
    int x=5;
    printf("%d¥n",SaikiKansuu(x));
}
int SaikiKansuu(int x){
    if(x==0)
        return 1;
    else
        return(x*SaikiKansuu(x-1));
}
```



関数の中でまた同じ関数に代入！？

この関数が再帰関数！！！！

```
int SaikiKansuu(int x){  
    if(x==0)  
        return 1;  
    else  
        return(x*SaikiKansuu(x-1));  
}
```

- ・なにやってんのこれ？

```
int SaikiKansuu(int x){  
    if(x==0)  
        return 1;  
    else  
        return(x*SaikiKansuu(x-1));  
}
```

まずこの関数に5が代入されてきます

5==0ではないので、下のreturn文に
飛びます

return(5 × SaikiKansuu(5-1))
の処理を行います



つまり・・・

5 × SaikiKansuu(4)
という状態になります！！

次は4を代入し
て同じ処理！



5 × 4 × 3 × 2 × 1 が出来た！

でもこれじゃ最後に $\times 0$ になって 0 になるんじゃない . . .

そのための `if(x==0)`
`return 1;`

これがあることによって最後 0 が代入されても
 $5 \times 4 \times 3 \times 2 \times 1 \times 1$ となってちゃんと答えが
出てくるぞ！！

もう一度ソースコードを見てみよう

```
#include <stdio.h>
int SaikiKansuu(int);

void main(){
    int x=5;
    printf("%d¥n",SaikiKansuu(x));
}
int SaikiKansuu(int x){
    if(x==0)
        return 1;
    else
        return(x*SaikiKansuu(x-1));
}
```

そう、このプログラムは
main関数から代入された
数字の階乗を印字するプログラム
だった！！



最後に！！！！

さきほどの

main関数で入力された数字の階乗を求めて印字するプログラム

を自分で1から打って作ってみよう！！！！

これで今回の講座は終わりです！！！！！！



お疲れ様でした！！！！