**全局和局部变量**

1. **局部变量**
2. 只能在函数内进行调用
3. 只在本函数内有效
   * 1. 即 sub1 没办法用sub2 里面的 x、y 反之则反

void sub1(){

int a, b;

}

void sub2(){

int x, y;

}

1. 不同函数可以定义相同的变量名（由上面我们可以得出吧），因为局部变量只在本函数内有效，且地址也是不一样的
2. 形参也是局部变量
3. 特殊写法【程序块】用大括号包起来，也就是复合语句
   * 1. 代码，我们c语言允许这么干的，因为 a、b 都是在{ }之前定义的也就是在main()里面由效，而{}又包含在 main() 之间，所以在{}可以进行调用 ，而c就只能在{}里面进行调用

int main(){

int a = 3;

int b = 4;

{

int c = a + b;

printf("%d", c);

}

}

1. **全局变量**
2. 可以为本文件中的所有函数进行复用
3. 在变量定义之后都可以进行用（如果之前的较函数就没办法啦）
4. 优缺点
   * 1. 优点
        1. 函数与函数之前可以进行联系，在一个函数中改变，另一个函数也可以用
     2. 缺点
        1. 程序运行时，始终占据内存，知道程序结束，而局部变量在某个函数跳出时就会释放
        2. 降低函数的通用性（如果全局变量是一样的）
        3. 全局变量太多，降低程序可读性，所以要谨慎使用哦
5. extern 关键字，就是就是可以引用在函后面定义的全局变量，如果我们在函数之后定义全局变量，是不是意味着在该函数内不能引用该全局变量呀
   * 1. 代码

void printDig(){

extern int a;

printf("%d", a);

}

int a = 3;

1. **全局和局部**
2. 当局部变量与全局变量同名时，全局变量失效，但是值不会受影响
   * 1. 代码

int a = 3;

int main(){

{

int a = 400;

printf("%d\n", a);

}

printf("%d", a);

}

1. 那我们来进行一个小的测试，大家猜想一下输出的结果是什么
   * 1. 代码

int a = 3;

void changeA(int a);

int main(){

printf("%d\n", a); //3

changeA(a);

printf("%d\n", a); //3

}

void changeA(int a){

a = 10;

return;

}