# static

# 分类

## 限定在当前文件可见

* 修饰**全局变量**时，表明一个全局变量只对定义在同一文件中的函数可见。
* 修饰**函数**时，表明该函数只在同一文件中调用。

## 唯一的，程序结束生命才结束

* 修饰**局部变量**时，表明该变量的值不会因为函数终止而丢失。
* 修饰**类的数据成员**，表明对该类所有对象这个数据成员都只有一个实例。即该实例归所有对象共有。

## 约束类的成员函数，防止改变非静态数据成员

* 修饰**类的成员函数**，以表明此成员函数逻辑上属于该类，而且该函数只对静态数据、全局数据或者参数进行操作，而不对非静态数据成员进行操作。

# 为什么

## 为什么静态成员函数不能是虚函数，即static与virtual不能共存？

因为static成员函数中，没有this指针。

如果有virtual修饰，就应该是运行时的动态绑定。

但是没有this指针，所以不知道绑定到哪个类。

假如static成员函数可以定义为virtual：

class Base

{

protected:

static virtual void fun(){}

}

class Child : public Base

{

protected:

static virtual void fun(){}

}

main()

{

Child c;

Base \*bp = &c;

bp->fun();//编译报错，static和virtual不能并存，这里是假设编译通过

//bp->fun()调用的应该是Base::fun()；因为对于静态函数，哪个对象调用它，

//它就执行哪个类（对象属于的那个类）的方法

//如果是动态绑定调用的应该是Child::fun()

}

## 为什么静态成员函数不能访问该类的非static成员变量？

因为static成员函数中，没有this指针。

如果修改非static成员变量，应该是修改this->variable。

但是没有this指针，所以不能访问非static成员变量。

## 为什么const成员函数中的static成员变量可以改变？

因为const成员函数约定为不能修改该对象的属性，而static成员变量是对类的属性，不属于某个对象。