

void 这个关键字我老搞不明白不知道怎么用。不知道什么时候需要用它，很是郁闷，void 声明的方法可以有参数吗？比如 public void A(string names)这样可以不？？？ public void int A(int age)呵呵，请指正一下啊。。我就是不知道怎么用啊，什么时候需要参数，什么时候不需要参数？反正我脑子是乱的，如果说不清楚请发个参考资料，有网址也好啊，要个人介绍的那种 MSDN 的我没搞明白。。谢谢了。。。强烈的谢谢了

-----  
public void A(string names)是可以的,public void int A(int age)不行  
public int A(int age)又可以，跟括号里的参数无关，方法名前面的 void, int 是方法返回的类型，

每个方法都有一个返回类型，void 是一个比较特殊的东西，你可以叫它空类型，它表示你的方法不需要返回值，

而其他的比如 public int A(int age)这样的方法，表示在 A 方法里面必须返回一个 int 类型的值，

必须有 return 语句，而且 return 对象类型必须是 int 类型

举例：

```
public int A(int age)
{
    age += 1;
    return age;
}
```

这个方法返回类型是 int，它最后返回了一个 age，age 是 int 类型的

而对于 void 类型的方法

```
public void A(int age)
{
    age += 1;
}
```

或

```
public void A(int age)
{
    age += 1;
    return;
}
```

都是可以的

后一个虽然有 return 语句，但是他没有 return 任何对象，所以是可以的，

而 return 任何其他类型都不对。

应该很容易理解吧

-----  
这么说吧，你要写一个方法 A 执行某些代码。

首先你要一对花括号 “{ }” 告诉电脑那些代码在哪。就出现如下写法：

A

```
{  
    代码...  
}
```

你方法 A 想公开，在整个程序的任意一个角落都可以使用，那就需要加修饰符来说明：

```
public A  
{  
    代码...  
}
```

当然你也可以不加修饰符，那程序会默认你加的是 `private`，也就是说以下两种写法效果相同：

```
A  
{  
    代码...  
}  
private A  
{  
    代码...  
}
```

然后你的方法可能需要传入一个或多个数（参数），或者什么数也不需要传入，那就需要一对圆括号：

```
A(int a)  
{  
    代码...  
}  
A()  
{  
    代码...  
}
```

注意：你可以不写参数，就像上面那么空着，但你必须要加上圆括号告诉电脑你的参数是空着的。

最后方法可能会返回一个数或一个字母（返回值），也可能什么也不返回，但返回的话，个数肯定是一个，电脑唯一不知道的就是这一个是什么，数？还是字母？所以你要写上返回值的类型，注意，对应返回值只有一个的特点，返回值类型也只能写一个。

```
int A(int a)  
{  
    代码...  
}  
char A(int a)  
{  
    代码...  
}
```

然后就出现你说的这种问题，如果方法仅仅执行了一些操作，例如在屏幕上显示一行字就结束了，没有需要返回的值。但从编译原理上来说，方法肯定会返回一个数字，哪怕是任意一个数字，所以你必须给这个返回的数字规定一个类型。这也就是 void 出现的原因。它作为一种特殊的类型修饰符存在，代表什么也不是的量。其地位应该是与 int char... 相等的。所以你这样写是错的，你要明确的告诉电脑，你方法 A 返回的值是 int 量，还是 void 量，不能同时写两个。当然，如果是 void，就代表返回的值无意义，不可以被任何代码接收并使用。

```
public void A(int age)
```

---

void 标明的方法是指这个方法不“返回”任何“东西”(包括值啊, 引用这些)它和传入参数不相干;

例子:

```
int M(int a, int b) {...} // 有两个传入参数的方法, 并且它返回一个 int 类型的值
```

```
void M(int a, bool b) {...} // 有两入传入参数的方法, 并且它没有任何返回 (这种方法也称为一个“过程”)
```

---

不返回参数或值的时候都要用 void，要返回则带返回的类型

---

lz 的解释得都很清楚了。。

返回的类型就是

```
public int a()
```

```
{return int}; 这个就是带返回的类型
```

```
public void a()
```

```
{}; 这个就是不带返回的类型的
```

---

简单的解释是：如果有回传：就不用 VOID

如果没有回传就必须用 VOID

---

不返回参数的用 VOID 返回参数的用数据类型

---

谢谢各位的帮忙，我现在明白了，尤其是 anliuty，说的很好，但我还有点要问，这两个方法都实现了 age+1 的功能，在实际应用中我在什么情况下，怎么区分使用那个，比如在实际编程中

```
public int A(int age)
{
    age += 1;
    return age;
}
```

```
public void A(int age)
{
    age += 1;
}
```

---

什么情况下需要做一个返回值的方法，什么情况下做一个不需要返回值的方法

---

这就看你的具体需求了，

举个例子，你在页面上写一个绑定 GridView 的方法，就可以写成不要返回值的方法

```
private void BindGridView()
{
    string strSQL = "xxxxxxxxxxx";
    DataTable dt = GetDataTable();
    this.GridView1.DataSource = dt;
    this.GridView1.DataBind();
}
```

这个方法不需要返回值，它调用了方法 GetDataTable，为什么要调用这个方法，因为 GridView 绑定的是数据，它需要数据源，所以 GetDataTable 方法给它提供数据源，就需要返回值，

这个方法返回的是一个 DataTable，一个在内存中的数据表，

GetDataTable 方法就是去数据库拿数据的，拿到数据后在内存中呈现为一个表，

至于怎么实现，随便哪本 ASP.NET 入门的书上都会有，网上更是一搜一堆

总之方法要不要返回值，不是人为规定的，而是看设计者要实现的这个方法的具体目的

---

打个比方说

//做 2 个数的加法 需要用到相加以后的结果 所以需要返回值

```
private int add(int a,int b)
{
    return a+b;
}
```

//让一个数字只减 1 不需要用到结果 减了以后就不用管了

```
private void sub(int a)
{
    a--;
}
```