

## 一、auto 简介

在C语言中，使用 auto 修饰的变量，是具有自动存储器的局部变量，但很少有人去使用它，在C++11中，auto 有了新的含义，它不再是一个存储类型指示符，而是作为一个新的类型指示符来指示编译器，auto 声明的变量必须由编译器在编译时期推导而得。

我们通过一个简单的代码来了解一下

```
#include <iostream>
using namespace std;
```

```
double TestFun()
{
    return 3.14;
}
```

```
int main()
{
    int a = 10;
    auto b = a;
    auto c = 'c';
    auto d = TestFun();
```

```
    cout << typeid(a).name() << endl;
    cout << typeid(b).name() << endl;
    cout << typeid(c).name() << endl;
    cout << typeid(d).name() << endl;
    system("pause");
    return 0;
}
```



<https://blog.csdn.net/xinwenhuayu>

注意：\*\*auto就是一个占位符，编译器会根据初始化来推演出类型，填充位置，所以auto必须要初始化\*\*

## 二、使用细则

1.用auto声明指针类型时，用auto和auto\*没有任何区别

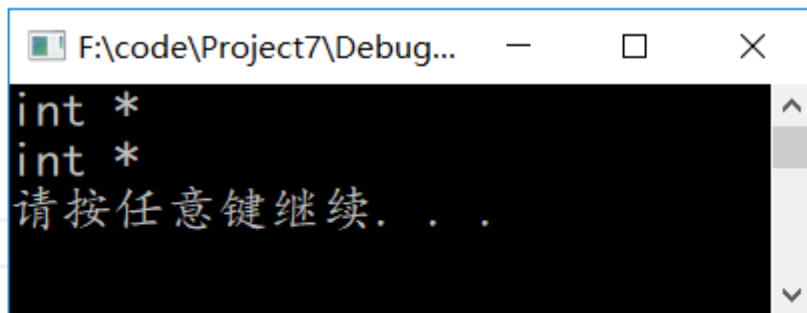
```

#include <iostream>
using namespace std;

int main()
{
    int a = 10;
    auto pa1 = &a;
    auto *pa2 = &a;

    cout << typeid(pa1).name() << endl;
    cout << typeid(pa2).name() << endl;
    system("pause");
    return 0;
}

```



```

F:\code\Project7\Debug...
int *
int *
请按任意键继续. . .

```

<https://blog.csdn.net/xinwenhuayu>

用auto声明引用类型时则必须加&

```

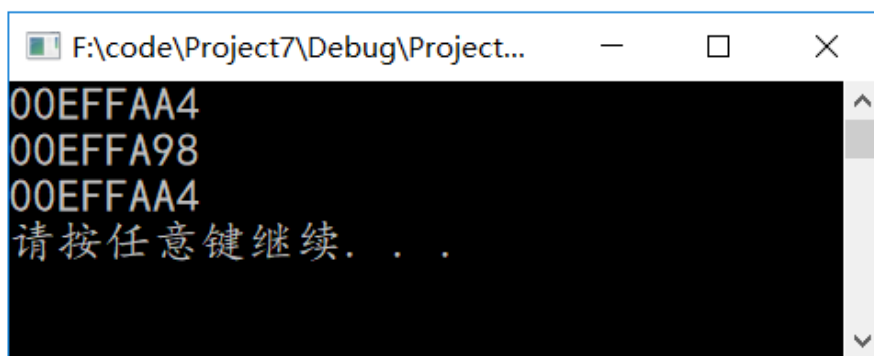
#include <iostream>
using namespace std;

int main()
{
    int a = 10;
    auto ra1 = a;
    auto& ra2 = a;

    cout << &a << endl;
    cout << &ra1 << endl;
    cout << &ra2 << endl;

    system("pause");
    return 0;
}

```



```

F:\code\Project7\Debug\Project...
00EFAA4
00EFA98
00EFAA4
请按任意键继续. . .

```

<https://blog.csdn.net/xinwenhuayu>

ra1就是与a相同类型的初始化变量

2.当在同一行声明多个变量时，这些变量必须是相同的类型，否则编译器将会报错，因为编译器实际只对 第一个类型进行推导，然后用推导出来的类型定义其他变量。

```
#include <iostream>
using namespace std;

int main()
{
    auto a = 1, b = 2;
    auto c = 3, d = 4.0;

    system("pause");
    return 0;
}
```

有红色的波浪线，代码编译失败，c和d的初始化表达式类型不同

### 三、在哪些情况下，auto 会推演失败

1. auto不能作为函数的参数
2. auto不能直接用来声明数组
3. 为了避免与C++98中的auto发生混淆，C++11只保留了auto作为类型指示符的用法，只是一个占位符
4. auto不能定义类的非静态成员变量
5. 实例化模板时不能使用auto作为模板参数

### 四、新式for循环

```
#include <iostream>
using namespace std;

int main()
{
    int array[] = { 1, 2, 3, 4, 5 };
    for (auto& e : array)
        e *= 2;

    for (auto e : array)
        cout << e << " ";

    cout << endl;
    system("pause");
    return 0;
}
```



使用这种方法，比原来的方法简单了很多，使用这种方法有什么要注意的呢？

- 1.对于数组而言，就是数组中第一个元素和最后一个元素的范围；对于类而言，应该提供begin和end的方法，begin和end就是for循环迭代的范围。
- 2.迭代的对象要实现++和==的操作。