默认参数

```
// 默认参数只能在参数列表最右边
void foo(int a, int b, double c=3.14, long d=123)
{
    ...
}
```

- 允许多个默认参数
- 从右向左，不间断
- 只能放在声明或定义处，最好在声明处

引用

指针的危害

1. 空指针，需检查代码
2. 野指针

左值引用

- 引用的本质上是一个常量指针
- 引用关系一旦建立，不能改变
- 必须初始化，不能使用`nullptr`或常量
- 常量引用可以使用常量初始化：`const int &a = 5`
- 不存在二级引用
- 数组的引用

```
int arr[5] = {1, 2, 3, 4, 5};
int (&arr_ref)[5] = arr;        // sizeof: 20
```

- 不能返回局部变量的引用

内联

宏的问题

1. 没有类型检查
2. 不方便调试

内联

- 允许函数像宏一样在调用处展开

- **Debug**版没有内联
- 编译器选项
  - **Ob0**禁用内联
  - **Ob1**只适用`__inline`项，如果没有`inline`，则不会内联。否则尝试内联
  - **Ob2**任何适用项，均进行尝试内联
- 内联的声明和实现不能分开，即放在头文件中
- `inline`关键字只是对编译器的建议，编译器只是尝试不保证必须内联。一般情况下会对简单函数进行内联