

- explicit 修饰构造函数时，可以防止隐式转换和复制初始化
- explicit 修饰转换函数时，可以防止隐式转换，但按语境转换除外

## explicit使用

```
struct A
{
    A(int) { }
    operator bool() const { return true; }
};
struct B
{
    explicit B(int) {}
    explicit operator bool() const { return true; }
};
void doA(A a) {}
void doB(B b) {}
int main()
{
    A a1(1); // OK: 直接初始化
    A a2 = 1; // OK: 复制初始化
    A a3{ 1 }; // OK: 直接列表初始化
    A a4 = { 1 }; // OK: 复制列表初始化
    A a5 = (A)1; // OK: 允许 static_cast 的显式转换
    doA(1); // OK: 允许从 int 到 A 的隐式转换
    if (a1); // OK: 使用转换函数 A::operator bool() 的从 A 到 bool 的隐式转换
    bool a6 (a1); // OK: 使用转换函数 A::operator bool() 的从 A 到 bool 的隐式转换
    bool a7 = a1; // OK: 使用转换函数 A::operator bool() 的从 A 到 bool 的隐式转换
    bool a8 = static_cast<bool>(a1); // OK : static_cast 进行直接初始化
    B b1(1); // OK: 直接初始化
    B b2 = 1; // 错误: 被 explicit 修饰构造函数的对象不可以复制初始化
    B b3{ 1 }; // OK: 直接列表初始化
    B b4 = { 1 }; // 错误: 被 explicit 修饰构造函数的对象不可以复制列表初始化
    B b5 = (B)1; // OK: 允许 static_cast 的显式转换
    doB(1); // 错误: 被 explicit 修饰构造函数的对象不可以从 int 到 B 的隐式转换
    if (b1); // OK: 被 explicit 修饰转换函数 B::operator bool() 的对象可以从 B 到 bool 的按语境转换
    bool b6(b1); // OK: 被 explicit 修饰转换函数 B::operator bool() 的对象可以从 B 到 bool 的按语境转换
    bool b7 = b1; // 错误: 被 explicit 修饰转换函数 B::operator bool() 的对象不可以隐式转换
    bool b8 = static_cast<bool>(b1); // OK: static_cast 进行直接初始化
    return 0;
}
```