

## C++20引入constexpr关键字用来标识函数返回常量

### 与constexpr类似

constexpr 函数在参数不是常量时会退化成普通函数

constexpr 函数在参数不是常量表达式时直接报错

### Demo

```
#include <iostream>
#include <array>

//constexpr 函数在参数不是常量时会退化成普通函数
//constexpr 函数在参数不是常量表达式时直接报错

template<typename...Args>
constexpr decltype(auto) foo1(Args...args)
{
    int sum{};
    ((sum += args), ...);
    return sum;
}

template<typename...Args>
constexpr decltype(auto) foo2(Args...args)
{
    int sum{};
    ((sum += args), ...);
    return sum;
}

int main()
{
    int x = 1, y = 1, z = 1;
    std::cout << foo1(x, y, z) << std::endl; //OK
    //std::cout << foo2(x, y, z) << std::endl; //error

    std::cout << foo1(1, 2, 3) << std::endl; //6 OK
    std::cout << foo2(1, 2, 3) << std::endl; //6 OK

    std::array<int, foo1(1, 2, 3)> a1{}; //OK
    std::array<int, foo2(1, 2, 3)> a2{}; //OK
    //std::array<int, foo2(x,y,z)> a2{}; //error
    std::array<int, [](int a, int b, int c)constexpr {
        return a + b + c;
    }(1, 2, 3)> a3{}; //OK
}
```

