

# C++ 标准库 <cmath>

C++ 标准库提供了丰富的功能，其中 <cmath> 是一个包含数学函数的头文件，它提供了许多基本的数学运算和常数。

<cmath> 是 C++ 标准库中的一个头文件，它定义了一组数学函数，这些函数可以执行基本的数学运算，如幂运算、三角函数、对数、绝对值等。

要使用 <cmath> 中的函数，你需要在你的 C++ 程序中包含这个头文件：

```
#include <cmath>
```

## 常用函数

<cmath> 提供了许多数学函数，以下是一些常用的函数。

### 1. 基本数学函数

函数	功能	示例
abs(x)	计算整数 x 的绝对值	abs(-5) // 5
fabs(x)	计算浮点数 x 的绝对值	fabs(-5.5) // 5.5
fmod(x, y)	计算 x 除以 y 的余数	fmod(5.3, 2) // 1.3
remainder(x, y)	计算 x 除以 y 的余数	remainder(5.5, 2) // 1.5
fmax(x, y)	返回 x 和 y 中的较大值	fmax(3.5, 4.2) // 4.2
fmin(x, y)	返回 x 和 y 中的较小值	fmin(3.5, 4.2) // 3.5
hypot(x, y)	计算 $\sqrt{x^2 + y^2}$	hypot(3, 4) // 5

### 2. 指数和对数函数

函数	功能	示例
exp(x)	计算 $e^x$ , e 为自然对数的底数	exp(1) // 2.71828...
log(x)	计算 x 的自然对数	log(2.71828) // 1
log10(x)	计算 x 的以 10 为底的对数	log10(100) // 2
pow(x, y)	计算 x 的 y 次方	pow(2, 3) // 8
sqrt(x)	计算 x 的平方根	sqrt(16) // 4

函数	功能	示例
<code>cbrt(x)</code>	计算 $x$ 的立方根	<code>cbrt(27) // 3</code>
<code>expm1(x)</code>	计算 $e^x - 1$	<code>expm1(1) // 1.71828...</code>
<code>log1p(x)</code>	计算 $\log(1 + x)$ ，适用于 $x$ 接近 0 的情况	<code>log1p(0.00001) // 0.00001</code>

### 3. 三角函数

函数	功能	示例
<code>sin(x)</code>	计算 $x$ 的正弦值， $x$ 以弧度为单位	<code>sin(3.14159 / 2) // 1</code>
<code>cos(x)</code>	计算 $x$ 的余弦值， $x$ 以弧度为单位	<code>cos(3.14159) // -1</code>
<code>tan(x)</code>	计算 $x$ 的正切值， $x$ 以弧度为单位	<code>tan(0) // 0</code>
<code>asin(x)</code>	计算 $x$ 的反正弦值，返回弧度	<code>asin(1) // 3.14159/2</code>
<code>acos(x)</code>	计算 $x$ 的反余弦值，返回弧度	<code>acos(-1) // 3.14159</code>
<code>atan(x)</code>	计算 $x$ 的反正切值，返回弧度	<code>atan(1) // 3.14159/4</code>
<code>atan2(y, x)</code>	计算 $y/x$ 的反正切值，返回弧度	<code>atan2(1, 1) // 3.14159/4</code>

### 4. 双曲函数

函数	功能	示例
<code>sinh(x)</code>	计算 $x$ 的双曲正弦	<code>sinh(0) // 0</code>
<code>cosh(x)</code>	计算 $x$ 的双曲余弦	<code>cosh(0) // 1</code>
<code>tanh(x)</code>	计算 $x$ 的双曲正切	<code>tanh(1) // 0.7616</code>
<code>asinh(x)</code>	计算 $x$ 的反双曲正弦	<code>asinh(1) // 0.8814</code>
<code>acosh(x)</code>	计算 $x$ 的反双曲余弦， $x \geq 1$	<code>acosh(1) // 0</code>
<code>atanh(x)</code>	计算 $x$ 的反双曲正切， $x$ 在 $(-1, 1)$	<code>atanh(0.5) // 0.5493</code>

### 5. 取整和浮点数操作

函数	功能	示例
<code>ceil(x)</code>	返回不小于 $x$ 的最小整数	<code>ceil(2.3) // 3</code>

函数	功能	示例
floor(x)	返回不大于 x 的最大整数	floor(2.3) // 2
trunc(x)	返回去除小数部分的整数值	trunc(2.8) // 2
round(x)	返回四舍五入到最接近的整数	round(2.5) // 3
lround(x)	返回四舍五入到 long 类型	lround(2.5) // 3
llround(x)	返回四舍五入到 long long 类型	llround(2.5) // 3
nearbyint(x)	返回舍入到最接近整数（但不引发浮点异常）	nearbyint(2.5) // 2
rint(x)	返回四舍五入到整数，符合当前舍入方式	rint(2.5) // 3
modf(x, &intpart)	将 x 的整数和小数部分分离	modf(2.3, &intpart)

6. 浮点数检查

函数	功能	示例
isfinite(x)	检查 x 是否为有限值（非无穷大或非 NaN）	isfinite(3.0) // true
isinf(x)	检查 x 是否为无穷大	isinf(1.0 / 0.0) // true
isnan(x)	检查 x 是否为 NaN	isnan(0.0 / 0.0) // true
isnormal(x)	检查 x 是否为正常的非零浮点数	isnormal(1.0) // true
signbit(x)	检查 x 的符号是否为负	signbit(-5.3) // true

实例

下面是一个使用 <cmath> 的简单示例，展示了如何计算一个数的平方根、正弦值和绝对值。

实例 1

```
#include <iostream>
#include <cmath> // 包含 <cmath> 头文件

int main() {
    double num = 9.0;
    double root = sqrt(num); // 计算平方根
    double sinValue = sin(M_PI / 2); // 计算正弦值，M_PI 是 π 的近似值
    double absValue = abs(-5.0); // 计算绝对值

    std::cout << "The square root of " << num << " is " << root << std::endl;
    std::cout << "The sine of " << M_PI / 2 << " is " << sinValue << std::endl;
    std::cout << "The absolute value of -5.0 is " << absValue << std::endl;
}
```

```
    return 0;
}
```

输出结果:

```
The square root of 9 is 3
The sine of 1.570796 is 1
The absolute value of -5 is 5
```

## 实例 2

```
#include <iostream>
#include <cmath>

int main() {
    // 基本数学运算
    std::cout << "abs(-5) = " << abs(-5) << std::endl;
    std::cout << "fmod(5.3, 2) = " << fmod(5.3, 2) << std::endl;

    // 指数和对数函数
    std::cout << "exp(1) = " << exp(1) << std::endl;
    std::cout << "log(2.71828) = " << log(2.71828) << std::endl;
    std::cout << "pow(2, 3) = " << pow(2, 3) << std::endl;

    // 三角函数
    std::cout << "sin(3.14159 / 2) = " << sin(3.14159 / 2) << std::endl;
    std::cout << "cos(3.14159) = " << cos(3.14159) << std::endl;

    // 取整函数
    std::cout << "ceil(2.3) = " << ceil(2.3) << std::endl;
    std::cout << "floor(2.3) = " << floor(2.3) << std::endl;

    // 浮点数检查
    double x = 1.0 / 0.0;
    if (isinf(x)) {
        std::cout << "x is infinite" << std::endl;
    }

    return 0;
}
```

## 注意事项

`<cmath>` 中的函数通常接受 `float` 或 `double` 类型的参数，并返回相应类型的结果。对于 `long double` 类型，你可以使用 `<cmath>` 中的函数，但需要在函数名后加上 `l` 后缀，例如 `sqrtl`。

某些函数，如 `pow` 和 `log`，可以接受整数作为参数，但结果仍然是浮点数。

`<cmath>` 中的函数可能会抛出异常，例如当 `sqrt` 函数的参数为负数时，会抛出 `std::domain_error` 异常。

`<cmath>` 是 C++ 标准库中一个非常有用的头文件，它提供了许多基本的数学函数，可以帮助开发者在编写程序时进行数学运算。