

Mathematical constants

Constants (since C++20)

Defined in header <code><numbers></code> Defined in namespace <code>std::numbers</code>	
e_v	the mathematical constant e (variable template)
log2e_v	$\log_2 e$ (variable template)
log10e_v	$\log_{10} e$ (variable template)
pi_v	the mathematical constant π (variable template)
inv_pi_v	$\frac{1}{\pi}$ (variable template)
inv_sqrtpi_v	$\frac{1}{\sqrt{\pi}}$ (variable template)
ln2_v	$\ln 2$ (variable template)
ln10_v	$\ln 10$ (variable template)
sqrt2_v	$\sqrt{2}$ (variable template)
sqrt3_v	$\sqrt{3}$ (variable template)
inv_sqrt3_v	$\frac{1}{\sqrt{3}}$ (variable template)
egamma_v	the Euler–Mascheroni constant γ (variable template)
phi_v	the golden ratio Φ ($\frac{1+\sqrt{5}}{2}$) (variable template)
inline constexpr double e	<code>e_v<double></code> (constant)
inline constexpr double log2e	<code>log2e_v<double></code> (constant)
inline constexpr double log10e	<code>log10e_v<double></code> (constant)
inline constexpr double pi	<code>pi_v<double></code> (constant)
inline constexpr double inv_pi	<code>inv_pi_v<double></code> (constant)
inline constexpr double inv_sqrtpi	<code>inv_sqrtpi_v<double></code> (constant)
inline constexpr double ln2	<code>ln2_v<double></code> (constant)
inline constexpr double ln10	<code>ln10_v<double></code> (constant)
inline constexpr double sqrt2	<code>sqrt2_v<double></code> (constant)
inline constexpr double sqrt3	<code>sqrt3_v<double></code> (constant)
inline constexpr double inv_sqrt3	<code>inv_sqrt3_v<double></code> (constant)
inline constexpr double egamma	<code>egamma_v<double></code> (constant)
inline constexpr double phi	<code>phi_v<double></code> (constant)

Notes

A program that instantiates a primary template of a mathematical constant variable template is ill-formed.

The standard library specializes mathematical constant variable templates for all floating-point types (i.e. `float`, `double``long double`, and fixed width floating-point types(since C++23)).

A program may partially or explicitly specialize a mathematical constant variable template provided that the specialization depends on a program-defined type.

Feature-test macro	Value	Std	Feature
<code>_cpp_lib_math_constants</code>	201907L	(C++20)	Mathematical constants

Example

Run this code

```
#include <cmath>
#include <iomanip>
#include <iostream>
#include <limits>
#include <numbers>
#include <string_view>

auto egamma_aprox(const unsigned iterations)
{
    long double s{};
    for (unsigned m{2}; m != iterations; ++m)
        if (const long double t{std::riemann_zetal(m) / m}; m % 2)
            s -= t;
        else
```

```

        s += t;
    return s;
};

int main()
{
    using namespace std::numbers;
    using namespace std::string_view_literals;

    const auto x = std::sqrt(inv_pi) / inv_sqrtpi +
        std::ceil(std::exp2(log2e)) + sqrt3 * inv_sqrt3 + std::exp(0);
    const auto v = (phi * phi - phi) + 1 / std::log2(sqrt2) +
        log10e * ln10 + std::pow(e, ln2) - std::cos(pi);
    std::cout << "The answer is " << x * v << '\n';

    constexpr auto y{"0.577215664901532860606512090082402"sv};
    std::cout
        << "γ as 106 sums of ±ζ(m)/m    = "
        << egamma_aprox(1'000'000) << '\n'
        << "γ as egamma_v<float>          = "
        << std::setprecision(std::numeric_limits<float>::digits10 + 1)
        << egamma_v<float> << '\n'
        << "γ as egamma_v<double>          = "
        << std::setprecision(std::numeric_limits<double>::digits10 + 1)
        << egamma_v<double> << '\n'
        << "γ as egamma_v<long double>     = "
        << std::setprecision(std::numeric_limits<long double>::digits10 + 1)
        << egamma_v<long double> << '\n'
        << "γ with " << y.length() - 1 << " digits precision = " << y << '\n';
}

```

Possible output:

```

The answer is 42
γ as 106 sums of ±ζ(m)/m    = 0.577215
γ as egamma_v<float>          = 0.5772157
γ as egamma_v<double>         = 0.5772156649015329
γ as egamma_v<long double>    = 0.5772156649015328606
γ with 34 digits precision = 0.577215664901532860606512090082402

```

See also

ratio (C++11) represents exact rational fraction
(class template)

Retrieved from "https://en.cppreference.com/mwiki/index.php?title=cpp/numeric/constants&oldid=178946"