MDN Web Docs
moz://a

Math

`Math` is a built-in object that has properties and methods for mathematical constants and functions. It’s not a function object.

`Math` works with the [`Number`] type. It doesn't work with [`BigInt`].

Description

Unlike many other global objects, `Math` is not a constructor. All properties and methods of `Math` are static. You refer to the constant pi as `Math.PI` and you call the sine function as `Math.sin(x)`, where `x` is the method’s argument. Constants are defined with the full precision of real numbers in JavaScript.

> **Note:** Many `Math` functions have a precision that’s *implementation-dependent.*
>
> This means that different browsers can give a different result. Even the same JavaScript engine on a different OS or architecture can give different results!

Static properties

`Math.E`

Euler's constant and the base of natural logarithms; approximately `2.718`.

`Math.LN2`

Natural logarithm of `2`; approximately `0.693`.

`Math.LN10`

Natural logarithm of `10`; approximately `2.303`.

`Math.LOG2E`

Base-2 logarithm of `E`; approximately `1.443`.

`Math.LOG10E`

Base-10 logarithm of `E`; approximately `0.434`.

`Math.PI`

Ratio of a circle's circumference to its diameter; approximately `3.14159`.

`Math.SQRT1_2`

Square root of ½; approximately `0.707`.

`Math.SQRT2`

Square root of `2`; approximately `1.414`.

Static methods

`Math.abs(x)`

Returns the absolute value of `x`.

`Math.acos(x)`

Returns the arccosine of `x`.

`Math.acosh(x)`

Returns the hyperbolic arccosine of `x`.

`Math.asin(x)`

Returns the arcsine of `x`.

`Math.asinh(x)`

Returns the hyperbolic arcsine of a number.

`Math.atan(x)`

Returns the arctangent of `x`.

`Math.atanh(x)`

Returns the hyperbolic arctangent of `x`.

`Math.atan2(y, x)`

`Math.atan2(y, x)`

Returns the arctangent of the quotient of its arguments.

`Math.cbrt(x)`

Returns the cube root of `x`.

`Math.ceil(x)`

Returns the smallest integer greater than or equal to `x`.

`Math.clz32(x)`

Returns the number of leading zero bits of the 32-bit integer `x`.

`Math.cos(x)`

Returns the cosine of `x`.

`Math.cosh(x)`

Returns the hyperbolic cosine of `x`.

`Math.exp(x)`

Returns `e^x`, where `x` is the argument, and `e` is Euler's constant (`2.718`…, the base of the natural logarithm).

`Math.expm1(x)`

Returns subtracting `1` from `exp(x)`.

`Math.floor(x)`

Returns the largest integer less than or equal to `x`.

`Math.fround(x)`

Returns the nearest single precision float representation of `x`.

`Math.hypot([x[, y[, …]]])`

Returns the square root of the sum of squares of its arguments.

`Math.imul(x, y)`

Returns the result of the 32-bit integer multiplication of `x` and `y`.

`Math.log(x)`

Returns the natural logarithm ($\log_e$; also, `ln`) of `x`.

`Math.log1p(x)`

Returns the natural logarithm ($\log_e$; also `ln`) of `1 + x` for the number `x`.

`Math.log10(x)`

Returns the base-10 logarithm of `x`.

`Math.log2(x)`

Returns the base-2 logarithm of `x`.

`Math.max([x[, y[, …]]])`

Returns the largest of zero or more numbers.

`Math.min([x[, y[, …]]])`

Returns the smallest of zero or more numbers.

`Math.pow(x, y)`

Returns base `x` to the exponent power `y` (that is, `x^y`).

`Math.random()`

Returns a pseudo-random number between `0` and `1`.

`Math.round(x)`

Returns the value of the number `x` rounded to the nearest integer.

`Math.sign(x)`

Returns the sign of the `x`, indicating whether `x` is positive, negative, or zero.

`Math.sin(x)`

Returns the sine of `x`.

`Math.sinh(x)`

Returns the hyperbolic sine of `x`.

`Math.sqrt(x)`

Returns the positive square root of `x`.

`Math.tan(x)`

Returns the tangent of `x`.

`Math.tanh(x)`

Returns the hyperbolic tangent of `x`.

`Math.trunc(x)`

Returns the integer portion of `x`, removing any fractional digits.

Examples

Converting between degrees and radians

The trigonometric functions `sin()`, `cos()`, `tan()`, `asin()`, `acos()`, `atan()`, and `atan2()` expect (and return) angles in *radians*.

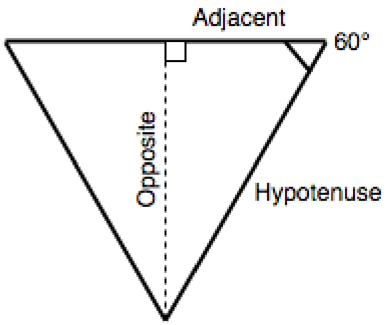
Since humans tend to think in degrees, and some functions (such as CSS transforms) can accept degrees, it is a good idea to keep functions handy that convert between the two:

```
function degToRad(degrees) {
  return degrees * (Math.PI / 180);
};

function radToDeg(rad) {
  return rad / (Math.PI / 180);
};
```

Calculating the height of an equilateral triangle

If we want to calculate the height of an equilateral triangle, and we know its side length is 100, we can use the formulae *length of the adjacent multiplied by the tangent of the angle is equal to the opposite.*



In JavaScript, we can do this with the following:

```
50 * Math.tan(degToRad(60)).
```

We use our `degToRad()` function to convert 60 degrees to radians, as `Math.tan()` expects an input value in radians.

Returning a random integer between two bounds

This can be achieved with a combination of `Math.random()` and `Math.floor()`:

```
function random(min, max) {
  const num = Math.floor(Math.random() * (max - min + 1)) + min;
  return num;
}

random(1, 10);
```

Specifications

No specification found

No specification data found for `javascript.builtins.Math`.

Check for problems with this page or contribute a missing `spec_url` to mdn/browser-compat-data. Also make sure the specification is included in w3c/browser-specs.

Browser compatibility