

# Theory: Math library

🕒 17 minutes   0 / 5 problems solved

Skip this topic

Start practicing

2493 users solved this topic. Latest completion was 1 day ago.

For efficient performing of various basic (and not so basic) numeric operations including exponential, modulus or finding the max/min of two numbers, Java provides the standard class `Math`. Let's consider some of the most common methods of this class.

## §1. Rounding methods

There's a number of popular methods for rounding the numbers up, down or the other way:

- `Math.min(..., ...)` returns the smaller value of two arguments;
- `Math.max(..., ...)` returns the greater value of two arguments;

```
1 int min = Math.min(11, 81); // min is 11
2 int max = Math.max(20, 30); // max is 30
```

- `Math.abs(...)` returns the absolute value of its argument;

```
1 int abs = Math.abs(-10); // abs is 10
2 double dabs = Math.abs(-10.33); // dabs is 10.33
```

- `Math.floor(...)` returns the largest double value that is less than or equal to its argument and is equal to an integer;
- `Math.ceil(...)` returns the smallest double value that is greater than or equal to its argument and is equal to an integer.

```
1 double floor = Math.floor(3.78); // floor is 3.0
2 double ceil = Math.ceil(4.15); // ceil is 5.0
```

## §2. Exponential functions

When we need to calculate a square or a cube root of the given number, we can apply the following methods:

- `Math.sqrt(...)` returns the square root of its argument;
- `Math.cbrt(...)` returns the cube root of its argument;

```
1 double sqrt = Math.sqrt(2); // sqrt is 1.4142...
2 double cbrt = Math.cbrt(27.0); // cbrt is 3.0
```

It is also possible to raise the number to any power we would like:

- `Math.pow(..., ...)` returns the value of the first argument raised to the power of the second argument.

```
1 double square = Math.pow(5, 2); // the square of 5 is 25.0
2 double cube = Math.pow(2, 3); // the cube of 2 is 8.0
```

## §3. Trigonometric functions

And here are some of the trigonometric functions, provided in `Math`:

- `Math.sin(...)` returns the trigonometric sin of the given angle in radians;
- `Math.cos(...)` returns the trigonometric cos of the given angle in radians;

```
1 double sin = Math.sin(pi / 2); // sin90° is 1.0
2 double cos = Math.cos(pi); // cos180° is -1.0
```

- `Math.toRadians(...)` converts an angle measured in degrees to an angle measured in radians (approximately).

Current topic:

`Math library` Stage 5 ...

Topic depends on:

- ✓ `Floating-point types` Stage 5 ...
- ✓ `Calling a method` Stage 2 ...

Table of contents:

- 1 [Math library](#)
- [§1. Rounding methods](#)
- [§2. Exponential functions](#)
- [§3. Trigonometric functions](#)
- [§4. And there's more...](#)
- [§5. The length of the hypotenuse](#)
- [Feedback & Comments](#)

```
1 | double grad = Math.toRadians(30); // grad is 0.5235...
```

## §4. And there's more...

There are also methods for hyperbolic, logarithmic, angular, and other functions. Check them out [here](#) and use when needed. Among them, you'll find such a useful method as:

- `Math.random()` returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.

```
1 | double random = Math.random(); // a random value >= 0.0 and < 1.0
```

Apart from functions, the `Math` class in Java contains two common constants:

- `Math.PI` is the ratio of the circumference of a circle to its diameter;
- `Math.E` is the base of the natural logarithm.

```
1 | double pi = Math.PI; // pi is 3.1415...
2 | double e = Math.E; // e is 2.71828...
```

## §5. The length of the hypotenuse

Now let's take a look at an example. Assume that we have a right triangle (one angle is 90 degrees). We know the lengths of both sides:  $a = 3$  and  $b = 4$ . Our task is to calculate the length of the hypotenuse. Now, it is time to go through the list of the `Math` class functions. After finding the one we need, the only thing left is to write the following code:

```
1 | double a = 3, b = 4;
2 | double c = Math.hypot(a, b); // c is 5.0
```

As you can see, the `Math` class has a lot of methods for performing numeric calculations and using them will make your life easier and your code nicer.

 Report a typo

315 users liked this theory. 1 didn't like it. What about you?



Start practicing

[Comments \(3\)](#)

[Hints \(0\)](#)

[Useful links \(0\)](#)

[Show discussion](#)