

JAVA.LANG.MATH CLASS

http://www.tutorialspoint.com/java/lang/java_lang_math.htm

Copyright © tutorialspoint.com

Introduction

The **java.lang.Math** class contains methods for performing basic numeric operations such as the elementary exponential, logarithm, square root, and trigonometric functions.

Class declaration

Following is the declaration for **java.lang.Math** class:

```
public final class Math
    extends Object
```

Field

Following are the fields for **java.lang.Math** class:

- **static double E** -- This is the double value that is closer than any other to e, the base of the natural logarithms.
- **static double PI** -- This is the double value that is closer than any other to pi, the ratio of the circumference of a circle to its diameter.

Class methods

S.N.	Method & Description
------	----------------------

- | | |
|---|---|
| 1 | <u>static double absdoublea</u>
This method returns the absolute value of a double value. |
| 2 | <u>static float absfloata</u>
This method returns the absolute value of a float value. |
| 3 | <u>static int absinta</u>
This method returns the absolute value of an int value. |
| 4 | <u>static long abslonga</u>
This method returns the absolute value of a long value. |
| 5 | <u>static double acosdoublea</u>
This method returns the arc cosine of a value; the returned angle is in the range 0.0 through pi. |
| 6 | <u>static double asindoublea</u>
This method returns the arc sine of a value; the returned angle is in the range -pi/2 |

through $\pi/2$.

7

[static double atan\(double\)](#)

This method returns the arc tangent of a value; the returned angle is in the range $-\pi/2$ through $\pi/2$.

8

[static double atan2\(double, double\)](#)

This method returns the angle θ from the conversion of rectangular coordinates x, y to polar coordinates r, θ .

9

[static double cbrt\(double\)](#)

This method returns the cube root of a double value.

10

[static double ceil\(double\)](#)

This method returns the smallest *closest to negative infinity* double value that is greater than or equal to the argument and is equal to a mathematical integer.

11

[static double copySign\(double magnitude, double sign\)](#)

This method returns the first floating-point argument with the sign of the second floating-point argument.

12

[static float copySign\(float magnitude, float sign\)](#)

This method returns the first floating-point argument with the sign of the second floating-point argument.

13

[static double cos\(double\)](#)

This method returns the trigonometric cosine of an angle.

14

[static double cosh\(double\)](#)

This method returns the hyperbolic cosine of a double value.

15

[static double exp\(double\)](#)

This method returns Euler's number e raised to the power of a double value.

16

[static double expm1\(double\)](#)

This method returns $e^x - 1$.

17

[static double floor\(double\)](#)

This method returns the largest *closesttopositiveinfinity* double value that is less than or equal to the argument and is equal to a mathematical integer.

18

[static int getExponentdoubled](#)

This method returns the unbiased exponent used in the representation of a double.

19

[static int getExponentfloatf](#)

This method returns the unbiased exponent used in the representation of a float.

20

[static double hypotdoublex, doubley](#)

This method returns $\sqrt{x^2 + y^2}$ without intermediate overflow or underflow.

21

[static double IEEEremainderdoublef1, doublef2](#)

This method computes the remainder operation on two arguments as prescribed by the IEEE 754 standard.

22

[static double logdoublea](#)

This method returns the natural logarithm *basee* of a double value.

23

[static double log10doublea](#)

This method returns the base 10 logarithm of a double value.

24

[static double log1pdoublex](#)

This method returns the natural logarithm of the sum of the argument and 1.

25

[static double maxdoublea, doubleb](#)

This method returns the greater of two double values.

26

[static float maxfloata, floatb](#)

This method returns the greater of two float values.

27

[static int maxinta, intb](#)

This method returns the greater of two int values.

28

[static long maxlonga, longb](#)

This method returns the greater of two long values.

29	<u>static double mindoublea, doubleb</u>	This method returns the smaller of two double values.
30	<u>static float minfloata, floatb</u>	This method returns the smaller of two float values.
31	<u>static int mininta, intb</u>	This method returns the smaller of two int values.
32	<u>static long minlonga, longb</u>	This method returns the smaller of two long values.
33	<u>static double nextAfterdoublestart, doubledirection</u>	This method returns the floating-point number adjacent to the first argument in the direction of the second argument.
34	<u>static float nextAfterfloatstart, doubledirection</u>	This method returns the floating-point number adjacent to the first argument in the direction of the second argument.
35	<u>static double nextUpdoubled</u>	This method returns the floating-point value adjacent to d in the direction of positive infinity.
36	<u>static float nextUpfloatf</u>	This method returns the floating-point value adjacent to f in the direction of positive infinity.
37	<u>static double powdoublea, doubleb</u>	This method returns the value of the first argument raised to the power of the second argument.
38	<u>static double random</u>	This method returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.
39	<u>static double rintdoublea</u>	This method returns the double value that is closest in value to the argument and is equal

to a mathematical integer.

- 40 [static long rounddoublea](#)
This method returns the closest long to the argument.
- 41 [static int roundfloata](#)
This method returns the closest int to the argument.
- 42 [static double scalbdoubled, intscaleFactor](#)
This method returns $d \times 2^{\text{scaleFactor}}$ rounded as if performed by a single correctly rounded floating-point multiply to a member of the double value set.
- 43 [static float scalbfloatf, intscaleFactor](#)
This method return $f \times 2^{\text{scaleFactor}}$ rounded as if performed by a single correctly rounded floating-point multiply to a member of the float value set.
- 44 [static double signumdoubled](#)
This method returns the signum function of the argument; zero if the argument is zero, 1.0 if the argument is greater than zero, -1.0 if the argument is less than zero.
- 45 [static float signumfloatf](#)
This method returns the signum function of the argument; zero if the argument is zero, 1.0f if the argument is greater than zero, -1.0f if the argument is less than zero.
- 46 [static double sindoublea](#)
This method returns the hyperbolic sine of a double value.
- 47 [static double sinhdoublex](#)
This method Returns the hyperbolic sine of a double value.
- 48 [static double sqrtdoublea](#)
This method returns the correctly rounded positive square root of a double value.
- 49 [static double tandoublea](#)
This method returns the trigonometric tangent of an angle.r
- 50 [static double tanhdoublex](#)

This method returns the hyperbolic tangent of a double value.

51

[static double toDegreesdoubleangrad](#)

This method converts an angle measured in radians to an approximately equivalent angle measured in degrees.

52

[static double toRadiansdoubleangdeg](#)

This method converts an angle measured in degrees to an approximately equivalent angle measured in radians.

53

[static double ulpdoubled](#)

This method returns the size of an ulp of the argument.

54

[static double ulpfloaf](#)

This method returns the size of an ulp of the argument.

Methods inherited

This class inherits methods from the following classes:

• java.lang.Object

Processing math: 100%