

CLASE MATH JAVA

Fuente traducción de:

<https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html>

- **Campo Resumen**

Campos

Modificador y Tipo	Campo y Descripción
static double	<u>E</u> El <code>double</code> valor que está más cerca que cualquier otra para <i>ae</i> , la base de los logaritmos naturales.
static double	<u>PI</u> El <code>double</code> valor que está más cerca que cualquier otra a <i>pi</i> , la relación de la circunferencia de un círculo y su diámetro.

método Resumen

Todos los métodos Los métodos estáticos métodos concretos

Modificador y Tipo	Método y Descripción
static double	<u>abs</u> (double a) Devuelve el valor absoluto de un <code>double</code> .
static float	<u>abs</u> (float a) Devuelve el valor absoluto de un <code>float</code> .
static int	<u>abs</u> (int a) Devuelve el valor absoluto de un <code>int</code> .
static long	<u>abs</u> (long a) Devuelve el valor absoluto de un <code>long</code> .
static double	<u>acos</u> (double a) Devuelve el arco coseno de un valor; el ángulo devuelto es en el rango de 0.0 a través de <i>pi</i> .
static int	<u>addExact</u> (int x, int y)

	Devuelve la suma de sus argumentos, lanza una excepción si el resultado se desborda .
static long	<u>addExact</u> (long x, long y) Devuelve la suma de sus argumentos, lanza una excepción si el resultado se desborda en long.
static double	<u>asin</u> (double a) Devuelve el arco seno de un valor; el ángulo devuelto es en el rango de $-pi / 2$ a través de $pi / 2$.
static double	<u>atan</u> (double a) Devuelve el arco tangente de un valor; el ángulo devuelto es en el rango de $-pi / 2$ a través de $pi / 2$.
static double	<u>atan2</u> (double y, double x) Devuelve el ángulo <i>theta</i> de la conversión de coordenadas rectangulares (x, y) a coordenadas polares (r, <i>theta</i>).
static double	<u>cbrt</u> (double a) Devuelve la raíz cúbica de un double.
static double	<u>ceil</u> (double a) Devuelve el más pequeño (el más cercano al infinito negativo) doublevalor que es mayor que o igual al argumento y es igual a un entero matemático.
static double	<u>copySign</u> (double magnitude, double sign) Devuelve el primer argumento de punto flotante con el signo del segundo argumento de coma flotante.
static float	<u>copySign</u> (float magnitude, float sign) Devuelve el primer argumento de punto flotante con el signo del segundo argumento de coma flotante.
static double	<u>cos</u> (double a) Devuelve el coseno de un ángulo trigonométrico.
static double	<u>cosh</u> (double x) Devuelve el coseno hiperbólico de un double.

static int	<u>decrementExact</u> (int a) Devuelve el argumento disminuido en uno, lanza una excepción si el resultado se desborda
static long	<u>decrementExact</u> (long a) Devuelve el argumento disminuido en uno, lanza una excepción si el resultado se desborda en long.
static double	<u>exp</u> (double a) Devuelve el número de Euler e elevado a la potencia de un double.
static double	<u>expm1</u> (double x) Devuelve $e^x - 1$.
static double	<u>floor</u> (double a) Devuelve el valor más grande (el más cercano al infinito positivo) que es menor o igual al argumento y es igual a un entero matemático.
static int	<u>floorDiv</u> (int x, int y) Devuelve el más grande (el más cercano al infinito positivo) valor que es menor o igual al cociente algebraico.
static long	<u>floorDiv</u> (long x, long y) Devuelve el más grande (el más cercano al infinito positivo) que es menor o igual al cociente algebraico.
static int	<u>floorMod</u> (int x, int y) Devuelve el módulo de piso de los argumentos.
static long	<u>floorMod</u> (long x, long y) Devuelve el módulo de piso de los argumentos.
static int	<u>getExponent</u> (double d) Devuelve el exponente imparcial utilizado en la representación de una double.
static int	<u>getExponent</u> (float f) Devuelve el exponente imparcial utilizado en la representación de una float.

static double	<p><u>hypot</u>(double x, double y)</p> <p>Devoluciones $\sqrt{x^2 + Y^2}$ sin rebosadero intermedio o subdesbordamiento.</p>
static double	<p><u>IEEEremainder</u>(double f1, double f2)</p> <p>Calcula la operación resto en dos argumentos según lo prescrito por el estándar IEEE 754.</p>
static int	<p><u>incrementExact</u>(int a)</p> <p>Devuelve el argumento incrementa en uno, lanza una excepción si el resultado se desborda una int.</p>
static long	<p><u>incrementExact</u>(long a)</p> <p>Devuelve el argumento incrementa en uno, lanza una excepción si el resultado se desborda una long.</p>
static double	<p><u>log</u>(double a)</p> <p>Devuelve el logaritmo natural (base e) de un double.</p>
static double	<p><u>log10</u>(double a)</p> <p>Devuelve el logaritmo en base 10 de un double.</p>
static double	<p><u>log1p</u>(double x)</p> <p>Devuelve el logaritmo natural de la suma de los argumentos y 1.</p>
static double	<p><u>max</u>(double a, double b)</p> <p>Devuelve el mayor de dos double.</p>
static float	<p><u>max</u>(float a, float b)</p> <p>Devuelve el mayor de dos float.</p>
static int	<p><u>max</u>(int a, int b)</p> <p>Devuelve el mayor de dos int.</p>
static long	<p><u>max</u>(long a, long b)</p> <p>Devuelve el mayor de dos long.</p>
static double	<p><u>min</u>(double a, double b)</p>

	Devuelve el menor de dos double.
static float	<u>min</u> (float a, float b) Devuelve el menor de dos float.
static int	<u>min</u> (int a, int b) Devuelve el menor de dos int.
static long	<u>min</u> (long a, long b) Devuelve el menor de dos long.
static int	<u>multiplyExact</u> (int x, int y) Devuelve el producto de los argumentos, lanza una excepción si el resultado se desborda una int.
static long	<u>multiplyExact</u> (long x, long y) Devuelve el producto de los argumentos, lanza una excepción si el resultado se desborda una long.
static int	<u>negateExact</u> (int a) Devuelve el opuesto de un argumento, una excepción si el resultado se desborda una int.
static long	<u>negateExact</u> (long a) Devuelve el opuesto de un argumento, una excepción si el resultado se desborda una long.
static double	<u>nextAfter</u> (double start, double direction) Devuelve el número de coma flotante adyacente al primer argumento en la dirección del segundo argumento.
static float	<u>nextAfter</u> (float start, double direction) Devuelve el número de coma flotante adyacente al primer argumento en la dirección del segundo argumento.
static double	<u>nextDown</u> (double d) Devuelve el valor de punto flotante adyacente a den la dirección de infinito negativo.

static float	<p><u>nextDown</u>(float f)</p> <p>Devuelve el valor de punto flotante adyacente a f en la dirección de infinito negativo.</p>
static double	<p><u>nextUp</u>(double d)</p> <p>Devuelve el valor de punto flotante adyacente a den la dirección de infinito positivo.</p>
static float	<p><u>nextUp</u>(float f)</p> <p>Devuelve el valor de punto flotante adyacente a f en la dirección de infinito positivo.</p>
static double	<p><u>pow</u>(double a, double b)</p> <p>Devuelve el valor del primer argumento elevado a la potencia del segundo argumento.</p>
static double	<p><u>random</u>()</p> <p>Devuelve un double con signo positivo, mayor o igual que 0.0 y menor que 1.0.</p>
static double	<p><u>rint</u>(double a)</p> <p>Devuelve el double que más se aproxime al valor del argumento y es igual a un entero matemático.</p>
static long	<p><u>round</u>(double a)</p> <p>Devuelve el más cercano longn al argumento, con lazos de redondeo hasta el infinito positivo.</p>
static int	<p><u>round</u>(float a)</p> <p>Devuelve el más cercano int al argumento, con lazos de redondeo hasta el infinito positivo.</p>
static double	<p><u>scalb</u>(double d, int scaleFactor)</p> <p>Devuelve $d \times 2^{\text{scaleFactor}}$ redondeadas como si se realiza por un solo redondeadas correctamente de punto flotante se multiplican a un miembro del conjunto de valor doble.</p>
static float	<p><u>scalb</u>(float f, int scaleFactor)</p>

Devuelve $f \times 2^{\text{scaleFactor}}$ redondeadas como si se realiza por un solo redondeadas correctamente de punto flotante se multiplican a un miembro del conjunto de valor flotante.

static double	<p><u>signum</u>(double d)</p> <p>Devuelve la función signo del argumento; cero si el argumento es cero, 1,0 si el argumento es mayor que cero, -1,0 si el argumento es menor que cero.</p>
static float	<p><u>signum</u>(float f)</p> <p>Devuelve la función signo del argumento; cero si el argumento es cero, 1.0f si el argumento es mayor que cero, -1.0f si el argumento es menor que cero.</p>
static double	<p><u>sin</u>(double a)</p> <p>Devuelve el seno trigonométrico de un ángulo.</p>
static double	<p><u>sinh</u>(double x)</p> <p>Devuelve el seno hiperbólico de un double.</p>
static double	<p><u>sqrt</u>(double a)</p> <p>Devuelve la raíz cuadrada positiva correctamente redondeado de un double.</p>
static int	<p><u>subtractExact</u>(int x, int y)</p> <p>Devuelve la diferencia de los argumentos, lanza una excepción si el resultado se desborda una int.</p>
static long	<p><u>subtractExact</u>(long x, long y)</p> <p>Devuelve la diferencia de los argumentos, lanza una excepción si el resultado se desborda una long.</p>
static double	<p><u>tan</u>(double a)</p> <p>Devuelve la tangente trigonométrica de un ángulo.</p>
static double	<p><u>tanh</u>(double x)</p> <p>Devuelve la tangente hiperbólica de un double.</p>
static double	<p><u>toDegrees</u>(double angrad)</p>

Convierte un ángulo medido en radianes en un ángulo aproximadamente equivalente medido en grados.

static int

[toIntExact](#)(long value)

Devuelve el valor del argumento `long`; lanza una excepción si el valor desborda una `int`.

static double

[toRadians](#)(double angdeg)

Convierte un ángulo medido en grados a un ángulo de aproximadamente equivalente medido en radianes.

static double

[ulp](#)(double d)

Devuelve el tamaño de un ULP del argumento.

static float

[ulp](#)(float f)

Devuelve el tamaño de un ULP del argumento.

