

# Introducción a Java

Módulo 2

# Librería *Math*

# Math

Muchas veces nos vemos en la necesidad de hacer operaciones matemáticas, cálculos, usar constantes, etc.

Ahora bien, en el módulo anterior aprendimos operadores aritméticos, pero se hace un poco complejo hacer, por ejemplo, la raíz cuadrada de un número, el exponencial de un número u cualquier otra operación matemática.

Java nos proporciona una librería llamada *Math* para realizar la mayoría de estas operaciones y dos constantes importantes.

Para poder incluir una biblioteca o librería deberíamos importarla, pero ***Math* pertenece al paquete *Lang* por lo que no es necesario hacerlo.**

# Constantes

Tipo	Método	Descripción
Static double	E	Devuelve el valor más aproximado del número $e$
Static double	PI	Devuelve el valor más aproximado del número $\pi$



# Métodos

Tipo	Método	Descripción
Static double	<b>abs(double a)</b>	Devuelve el valor absoluto de un valor double introducido como parámetro.
Static float	<b>abs(float a)</b>	Devuelve el valor absoluto de un valor float introducido como parámetro.
Static int	<b>abs(int a)</b>	Devuelve el valor absoluto de un valor Entero introducido como parámetro.
Static long	<b>abs(long a)</b>	Devuelve el valor absoluto de un valor long introducido como parámetro.
Static double	<b>acos(double a)</b>	Devuelve el arco coseno de un valor introducido como parámetro.
Static int	<b>addExact(int x, int y)</b>	Devuelve la suma de sus argumentos, lanzando una excepción si el resultado desborda un <i>int</i> .
Static long	<b>addExact(long x, long y)</b>	Devuelve la suma de sus argumentos, lanzando una excepción si el resultado se desborda a <i>long</i> .
Static double	<b>asin(double a)</b>	Devuelve el arco seno de un valor introducido.

Tipo	Método	Descripción
Static double	<code>atan(double a)</code>	Devuelve el arco tangente de un valor introducido.
Static double	<code>cbrt(double a)</code>	Devuelve la raíz cúbica de un <i>double</i> .
Static double	<code>cos(double a)</code>	Devuelve el coseno trigonométrico de un ángulo.
Static double	<code>exp(double a)</code>	Devuelve el número e de Euler elevado a la potencia de un <i>double valor</i> .
Static double	<code>log(double a)</code>	Devuelve el logaritmo natural (base e ) de un <i>double valor</i> .
Static double	<code>log10(double a)</code>	Devuelve el logaritmo de base 10 de un <i>double</i> .
Static int	<code>multiplyExact(int x, int y)</code>	Devuelve el producto de los argumentos, lanzando una excepción si el resultado desborda un <i>int</i> .
Static double	<code>max(double a, double b)</code>	Devuelve el mayor de dos valores <i>double</i>
Static float	<code>max(float a, float b)</code>	Devuelve el mayor de dos valores <i>float</i> .
Static int	<code>max(int a, int b)</code>	Devuelve el mayor de dos valores Enteros.
Static long	<code>max(long a, long b)</code>	Devuelve el mayor de dos valores <i>long</i> .
Static double	<code>min(double a, double b)</code>	Devuelve el menor de dos valores <i>double</i> .

Tipo	Método	Descripción
Static float	<code>min(float a, float b)</code>	Devuelve el menor de dos valores <i>float</i> .
Static int	<code>min(int a, int b)</code>	Devuelve el menor de dos valores enteros.
Static long	<code>min(long a, long b)</code>	Devuelve el menor de dos valores <i>long</i> .
Static double	<code>pow(double a, double b)</code>	Devuelve el valor del primer argumento elevado a la potencia del segundo argumento.
Static double	<code>random()</code>	Devuelve un doublevalor con un signo positivo, mayor o igual que 0.0 y menor que 1.0.
Static double	<code>round(double a)</code>	Devuelve el long redondeado más cercano al double introducido.
Static float	<code>round(float a)</code>	Devuelve el int mas cercano y redondeado al <i>float</i> introducido.
Static double	<code>sin(double a)</code>	Devuelve el seno trigonométrico de un ángulo.
Static double	<code>sqrt(double a)</code>	Devuelve la raíz cuadrada positiva correctamente redondeada de un <i>double valor</i> .
Static double	<code>tan(double a)</code>	Devuelve la tangente trigonométrica de un ángulo.

**¡Nos vemos en  
el siguiente módulo!**

