

LA CLASE MATH

Se echan de menos operadores matemáticos más potentes en Java. Por ello se ha incluido una clase especial llamada **Math** dentro del paquete **java.lang**. Para poder utilizar esta clase, se debe incluir esta instrucción (aunque normalmente no es necesario porque todo el paquete **java.lang** ya estará incluido en nuestro código):

```
import java.lang.Math;
```

Esta clase posee métodos muy interesantes para realizar cálculos matemáticos complejos. Por ejemplo:

```
double x= Math.pow(3,3); //x es 33, es decir 27
```

Math posee dos constantes, que son:

constante	Significado
double E	El número e (2, 7182818245...)
double PI	El número Π (3,14159265...)

Por otro lado posee numerosos métodos que son:

operador	significado
double ceil(double x)	Redondea x al entero mayor siguiente: ♦ Math.ceil (2.8) vale 3 ♦ Math.ceil (2.4) vale 3 ♦ Math.ceil (-2.8) vale -2
double floor(double x)	Redondea x al entero menor siguiente: ♦ Math.floor (2.8) vale 2 ♦ Math. floor (2.4) vale 2 ♦ Math. floor (-2.8) vale -3
int round(double x)	Redondea x de forma clásica: ♦ Math.round (2.8) vale 3 ♦ Math. round (2.4) vale 2 ♦ Math. round (-2.8) vale -3
double rint(double x)	Idéntico al anterior, sólo que éste método da como resultado un número double mientras que round da como resultado un entero tipo int
double random()	Número aleatorio decimal situado entre el 0 y el 1
tiponúmero abs(tiponúmero x)	Devuelve el valor absoluto de x.
tiponúmero min(tiponúmero x, tiponúmero y)	Devuelve el menor valor de x o y

<code>tiponúmero max(tiponúmero x, tiponúmero y)</code>	Devuelve el mayor valor de x o y
<code>double sqrt(double x)</code>	Calcula la raíz cuadrada de x
<code>double pow(double x, double y)</code>	Calcula x^y
<code>double exp(double x)</code>	Calcula e^x
<code>double log(double x)</code>	Calcula el logaritmo neperiano de x
<code>double acos(double x)</code>	Calcula el arco coseno de x
<code>double asin(double x)</code>	Calcula el arco seno de x
<code>double atan(double x)</code>	Calcula el arco tangente de x
<code>double sin(double x)</code>	Calcula el seno de x
<code>double cos(double x)</code>	Calcula el coseno de x
<code>double tan(double x)</code>	Calcula la tangente de x
<code>double toDegrees(double anguloEnRadianes)</code>	Convierte de radianes a grados
<code>double toRadians(double anguloEnGrados)</code>	Convierte de grados a radianes
<code>double signum(double n)</code>	Devuelve el valor del signo del número n. Si n vale cero, la función devuelve cero; si es positivo devuelve 1.0 y si es negativo -1.0 Esta función apareció en la versión 1.5 de Java.
<code>double hypot(double x, double y)</code>	Suponiendo que x e y son los dos catetos de un triángulo rectángulo, la función devuelve la hipotenusa correspondiente según el teorema de Pitágoras. Disponible desde la versión 1.5
<code>double nextAfter(double valor, double dir)</code>	Devuelve el siguiente número representable desde el valor indicado hacia la dirección que indique el valor del parámetro dir. Por ejemplo <code>Math.nextAfter(34.7, 90)</code>

devolvería 34.7000000001 Función añadida en la versión Java 1.6
--

NUMEROS ALEATORIOS

Una de las aplicaciones más interesantes de **Math** es la posibilidad de crear números aleatorios. Para ello se utiliza el método **random** que devuelve un número **double** entre cero y uno.

Para conseguir un número decimal por ejemplo entre cero y diez bastaría utilizar la expresión:

Math.random()*10

Si el número queremos que se encuentre entre uno y diez, sería:

Math.random()*9+1

Y si queremos que sea un número entero entre 1 y 10, la expresión correcta es:

(int) Math.floor(Math.random()*10+1)

Entre 10 y 30 sería:

(int) Math.floor(Math.random()*21+10)