**Java 21 – 2019 JSE EIT – Math**



Ya nos habíamos referido a la biblioteca “**java.lang**”.

El compilador importa automáticamente a este paquete, por lo que no es obligatorio especificarlo.

Es una biblioteca que no es necesaria especificarla, porque se la asume.

En otras palabras, podemos especificarla de la siguiente manera…

**import java.lang.\*;**

…o ignorar esta especificación.

Son clases e interfaces básicas que usa **Java**.

Su nombre proviene de los cuatro primeros caracteres de “lenguaje”, que en inglés es “**lang**uage”

**La clase “Math”**

Pertenece a la biblioteca “**java.lang**” y dispone de los siguientes métodos:

|  |  |
| --- | --- |
| **Método** | **Descripción** |
| **Math.abs(x)** | Valor absoluto |
| **Math.acos(x)** | Arco coseno |
| **Math.asin(x)** | Arco seno |
| **Math.atan(x)** | Arco tangente |
| **Math.atan2(x,y)** | Componente angular de la representación polar de x, y |
| **Math.ceil(x)** | Menor entero mayor que x |
| **Math.cos(x)** | Coseno |
| **Math.E** | Constante 2.718281828459045 |
| **Math.exp(x)** | ex |
| **Math.floor(x)** | Mayor entero menor que x |
| **Math.IEEEremainder(x,y)** | Resto de la división x/y según el estándar IEEE 754 |
| **Math.log(x)** | Logaritmo natural: ln(x) |
| **Math.max(x,y)** | El mayor de x e y |
| **Math.min(x,y)** | El menor de x e y |
| **Math.PI** | Constante 3.141592653589793 |
| **Math.pow(x,y)** | xy |
| **Math.random()** | Número aleatorio entre 0 y 1 |
| **Math.rint(x)** | Entero más cercano a x (devuelve un doble) |
| **Math.round(x)** | Entero más cercano a x (devuelve un entero o un long) |
| **Math.sin(x)** | Seno |
| **Math.sqrt(x)** | Raíz cuadrada |
| **Math.tan(x)** | Tangente |

A continuación tenemos un programa que nos hace practicar algunos métodos de la clase **Math**.

**package hm;**

**public class HolaMundo {**

**public static void main(String args[]) {**

**double x = 2.51;**

**System.out.println( "Constante Pi: " + Math.PI );**

**System.out.println( "Constante E: " + Math.E );**

**System.out.println( "cos(pi): " + Math.cos( Math.PI ) );**

**System.out.println( "round(x): " + Math.round( x ) );**

**}**

**}**

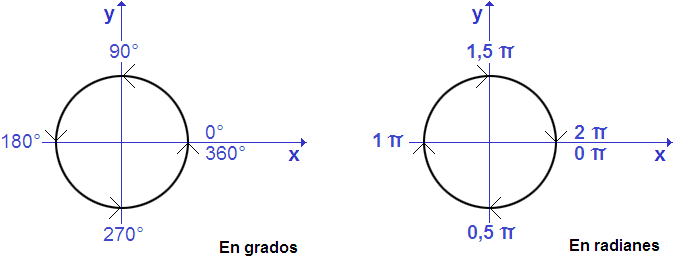
|  |  |  |
| --- | --- | --- |
| La salida es: |  |  |
| **Constante Pi:** | **3.141592653589793** |
| **Constante E:** | **2.718281828459045** |
| **cos(pi):** | **-1.0** |
| **round(x):** | **3** |

Las funciones trigonométricas usan **radianes** y no **ángulos**.

Por eso, en el ejemplo anterior, el coseno de π radianes vale -1.

Para convertir **grados** en **radianes**, podemos usar la siguiente relación:

**180°** son **π** (**pi**) **radianes**.



Para convertir **grados** en **radianes** podemos escribir la siguiente instrucción **Java**, basado en la regla de tres simple:

**float radianes = ( Math.PI / 180 ) \* grados ;**