

# Les fonctions mathématiques en Java

## La classe Math

### Table des matières:

<b>1</b>	<b>PRESENTATION DE LA CLASSE MATH</b> .....	<b>1</b>
<b>2</b>	<b>LES CONSTANTES DE LA CLASSE MATH</b> .....	<b>2</b>
<b>3</b>	<b>LES FONCTIONS TRIGONOMETRIQUES</b> .....	<b>3</b>
<b>4</b>	<b>LES FONCTIONS DE COMPARAISONS MIN ET MAX</b> .....	<b>3</b>
<b>5</b>	<b>LES ARRONDIS</b> .....	<b>4</b>
5.1	LA METHODE ROUND(N) .....	4
5.2	LA METHODE RINT(DOUBLE) .....	5
5.3	LA METHODE CEIL(DOUBLE) .....	5
5.4	LA METHODE ABS(X) .....	6
<b>6</b>	<b>LA METHODE IEEEEREMAINDER(DOUBLE, DOUBLE)</b> .....	<b>6</b>
<b>7</b>	<b>LES EXPONENTIELLES ET PUISSANCES</b> .....	<b>7</b>
7.1	LA METHODE POW( DOUBLE, DOUBLE) .....	7
7.2	LA METHODE SQRT(DOUBLE) .....	7
7.3	LA METHODE EXP(DOUBLE) .....	7
7.4	LA METHODE LOG(DOUBLE) .....	7
<b>8</b>	<b>LA GENERATION DE NOMBRES ALEATOIRES</b> .....	<b>8</b>
8.1	LA METHODE RANDOM() .....	8
<b>9</b>	<b>EXEMPLES DE PROGRAMME JAVA</b> .....	<b>9</b>
9.1	VALEURSMATHEMATIQUES.JAVA .....	9
9.2	CALCULEREMPRUNT.JAVA .....	9
9.3	CALCULERINTERETTEST.JAVA .....	10
9.4	CALCULERMENSUALITETEST.JAVA.JAVA .....	11

## 1 Présentation de la classe Math

La classe `java.lang.Math` contient une série de méthodes et variables mathématiques. Comme la classe `Math` fait partie du package `java.lang`, elle est automatiquement importée. De plus, il n'est pas nécessaire de déclarer un objet de type `Math` car les méthodes sont toutes statiques.

Exemple: Calculer et afficher la racine carrée de 3

```
public class Math1 {
    public static void main(java.lang.String[] args) {
        System.out.println(" = " + Math.sqrt(3.0));
    }
}
```

Le diagramme de classes suivant présente la classe Math (J2SE 1.3):

Math (from lang)
+ E : double = 2.718281828 + PI : double = 3.141592654
+ sin(arg0 : double) : double + cos(arg0 : double) : double + tan(arg0 : double) : double + asin(arg0 : double) : double + acos(arg0 : double) : double + atan(arg0 : double) : double + toRadians(arg0 : double) : double + toDegrees(arg0 : double) : double + exp(arg0 : double) : double + log(arg0 : double) : double + sqrt(arg0 : double) : double + IEEEremainder(arg0 : double, arg1 : double) : double + ceil(arg0 : double) : double + floor(arg0 : double) : double + rint(arg0 : double) : double + atan2(arg0 : double, arg1 : double) : double + pow(arg0 : double, arg1 : double) : double + round(arg0 : float) : int + round(arg0 : double) : long + random() : double + abs(arg0 : int) : int + abs(arg0 : long) : long + abs(arg0 : float) : float + abs(arg0 : double) : double + max(arg0 : int, arg1 : int) : int + max(arg0 : long, arg1 : long) : long + max(arg0 : float, arg1 : float) : float + max(arg0 : double, arg1 : double) : double + min(arg0 : int, arg1 : int) : int + min(arg0 : long, arg1 : long) : long + min(arg0 : float, arg1 : float) : float + min(arg0 : double, arg1 : double) : double

## 2 Les constantes de la classe Math

PI représente pi dans le type double ( 3,14159265358979323846 )

E représente e dans le type double ( 2,7182818284590452354 )

Extrait de la javadoc de la classe Math (JAVA SE 5.0):

Field Summary	
static double	<a href="#">E</a> The double value that is closer than any other to <i>e</i> , the base of the natural logarithms.
static double	<a href="#">PI</a> The double value that is closer than any other to <i>pi</i> , the ratio of the circumference of a circle to its diameter.

Exemple :

```
public class Math2 {
```

```
public static void main(java.lang.String[] args) {
    System.out.println(" PI = " + Math.PI);
    System.out.println(" E = " + Math.E);
}}
```

### 3 Les fonctions trigonométriques

Les méthodes `sin()`, `cos()`, `tan()`, `asin()`, `acos()`, `atan()` ont des prototypes de type suivant:

```
public static double <fonctiontrigo>(double <angle>)
```

Les angles doivent être exprimés en radians. Pour convertir des degrés en radian, il suffit de les multiplier par  $\text{PI}/180$ .

Extrait de la javadoc de la classe **Math** (JAVA SE 5.0):

#### Method Summary

static double	<a href="#"><code>cos</code></a> (double a) Returns the trigonometric cosine of an angle.
static double	<a href="#"><code>sin</code></a> (double a) Returns the trigonometric sine of an angle.
static double	<a href="#"><code>tan</code></a> (double a) Returns the trigonometric tangent of an angle.

static double	<a href="#"><code>acos</code></a> (double a) Returns the arc cosine of an angle, in the range of 0.0 through $\pi$ .
static double	<a href="#"><code>asin</code></a> (double a) Returns the arc sine of an angle, in the range of $-\pi/2$ through $\pi/2$ .
static double	<a href="#"><code>atan</code></a> (double a) Returns the arc tangent of an angle, in the range of $-\pi/2$ through $\pi/2$ .

### 4 Les fonctions de comparaisons min et max

Ces méthodes existent pour les types `int`, `long`, `float` et `double` : elles retournent respectivement les valeurs maximales et minimales des deux paramètres.

Extrait de la javadoc de la classe **Math** (JAVA SE 5.0):

#### Method Summary

static double	<a href="#"><code>max</code></a> (double a, double b) Returns the greater of two double values.
static float	<a href="#"><code>max</code></a> (float a, float b) Returns the greater of two float values.
static int	<a href="#"><code>max</code></a> (int a, int b) Returns the greater of two int values.
static long	<a href="#"><code>max</code></a> (long a, long b) Returns the greater of two long values.
static double	<a href="#"><code>min</code></a> (double a, double b) Returns the smaller of two double values.

static float	<a href="#"><u>min</u></a> (float a, float b) Returns the smaller of two float values.
static int	<a href="#"><u>min</u></a> (int a, int b) Returns the smaller of two int values.
static long	<a href="#"><u>min</u></a> (long a, long b) Returns the smaller of two long values.

Exemple:

```
public class Math1 {
    public static void main(String[] args) {
        System.out.println(" le plus grand = " + Math.max(5, 10));
        System.out.println(" le plus petit = " + Math.min(7, 14));
    }
}
```

Résultat :  
le plus grand = 10  
le plus petit = 7

## 5 Les arrondis

La classe Math propose plusieurs méthodes pour réaliser différents arrondis sur les nombres à virgule flottante.

### 5.1 La méthode round(n)

Cette méthode ajoute 0,5 à l'argument et restitue la plus grande valeur entière (int) inférieure ou égale au résultat. La méthode est définie pour les types float et double.

Extrait de la javadoc de la classe **Math** (JAVA SE 5.0):

Method Summary	
static long	<a href="#"><u>round</u></a> (double a) Returns the closest long to the argument.
static int	<a href="#"><u>round</u></a> (float a) Returns the closest int to the argument.

Exemple :

```
public class Arrondis1 {
    static double[] valeur = {-5.7, -5.5, -5.2, -5.0, 5.0, 5.2, 5.5, 5.7 };
    public static void main(String[] args) {
        for (int i = 0; i < valeur.length; i++) {
            System.out.println("round("+valeur[i]+") = "+Math.round(valeur[i]));
        }
    }
}
```

Résultat :  
round(-5.7) = -6  
round(-5.5) = -5  
round(-5.2) = -5  
round(-5.0) = -5  
round(5.0) = 5  
round(5.2) = 5  
round(5.5) = 6  
round(5.7) = 6

## 5.2 La méthode rint(double)

Cette méthode effectue la même opération mais renvoie un type double.

Extrait de la javadoc de la classe **Math** (JAVA SE 5.0):

static double	<a href="#">rint</a> (double a) Returns the double value that is closest in value to the argument and is equal to a mathematical integer.
---------------	--

Exemple :

```
public class Arrondis2 {
    static double[] valeur = {-5.7, -5.5, -5.2, -5.0, 5.0, 5.2, 5.5, 5.7 };
    public static void main(String[] args) {
        for (int i = 0; i <valeur.length; i++) {
            System.out.println("rint("+valeur[i]+") = "+Math.rint(valeur[i]));
        }
    }
}
```

Résultat :

```
rint(-5.7) = -6.0
rint(-5.5) = -6.0
rint(-5.2) = -5.0
rint(-5.0) = -5.0
rint(5.0) = 5.0
rint(5.2) = 5.0
rint(5.5) = 6.0
rint(5.7) = 6.0
```

### 6.4.3. La méthode floor(double)

Cette méthode renvoie l'entier le plus proche inférieur ou égal à l'argument.

Extrait de la javadoc de la classe **Math** (JAVA SE 5.0):

static double	<a href="#">floor</a> (double a) Returns the largest (closest to positive infinity) double value that is less than or equal to the argument and is equal to a mathematical integer.
---------------	--

Exemple :

```
public class Arrondis3 {
    static double[] valeur = {-5.7, -5.5, -5.2, -5.0, 5.0, 5.2, 5.5, 5.7 };
    public static void main(String[] args) {
        for (int i = 0; i <valeur.length; i++) {
            System.out.println("floor("+valeur[i]+") = "+Math.floor(valeur[i]));
        }
    }
}
```

Résultat :

```
floor(-5.7) = -6.0
floor(-5.5) = -6.0
floor(-5.2) = -6.0
floor(-5.0) = -5.0
floor(5.0) = 5.0
floor(5.2) = 5.0
floor(5.5) = 5.0
floor(5.7) = 5.0
```

## 5.3 La méthode ceil(double)

Cette méthode renvoie l'entier le plus proche supérieur ou égal à l'argument.

Extrait de la javadoc de la classe **Math** (JAVA SE 5.0):

static double	<a href="#">ceil</a> (double a) 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.
---------------	---

Exemple :

```
public class Arrondis4 {
    static double[] valeur = {-5.7, -5.5, -5.2, -5.0, 5.0, 5.2, 5.5, 5.7};
};
    public static void main(String[] args) {
        for (int i = 0; i < valeur.length; i++) {
            System.out.println("ceil("+valeur[i]+") = "+Math.ceil(valeur[i]));
        }
    }
}
```

résultat :

```
ceil(-5.7) = -5.0
ceil(-5.5) = -5.0
ceil(-5.2) = -5.0
ceil(-5.0) = -5.0
ceil(5.0) = 5.0
ceil(5.2) = 6.0
ceil(5.5) = 6.0
ceil(5.7) = 6.0
```

## 5.4 La méthode *abs(x)*

Cette méthode donne la valeur absolue de x (les nombre négatifs sont convertis en leur opposé). La méthode est définie pour les types int, long, float et double.

Extrait de la javadoc de la classe Math (JAVA SE 5.0):

### Method Summary

static double	<a href="#">abs</a> (double a) Returns the absolute value of a double value.
static float	<a href="#">abs</a> (float a) Returns the absolute value of a float value.
static int	<a href="#">abs</a> (int a) Returns the absolute value of an int value.
static long	<a href="#">abs</a> (long a) Returns the absolute value of a long value.

Exemple :

```
public class Math1 {
    public static void main(String[] args) {
        System.out.println(" abs(-5.7) = "+abs(-5.7));
    }
}
```

Résultat :

```
abs(-5.7) = 5.7
```

## 6 La méthode *IEEERemainder(double, double)*

Cette méthode renvoie le reste de la division du premier argument par le deuxième.

Extrait de la javadoc de la classe **Math** (JAVA SE 5.0):

static double	<a href="#">IEEERemainder</a> (double f1, double f2) Computes the remainder operation on two arguments as prescribed by the IEEE 754 standard.
---------------	---

Exemple :

```
public class Math1 {
    public static void main(String[] args) {
```

```
System.out.println(" reste de la division de 3 par 10 = "
+Math.IEEEremainder(10.0, 3.0) );
}}
```

Résultat :  
reste de la division de 3 par 10 = 1.0

## 7 Les Exponentielles et puissances

### 7.1 La méthode *pow(double, double)*

Cette méthode élève le premier argument à la puissance indiquée par le second.

Extrait de la javadoc de la classe **Math** (JAVA SE 5.0):

static double	<a href="#"><code>pow</code></a> (double a, double b) Returns the value of the first argument raised to the power of the second argument.
---------------	--

Exemple :

```
public static void main(java.lang.String[] args) {
System.out.println(" 5 au cube = "+Math.pow(5.0, 3.0) );
}
```

Résultat :  
5 au cube = 125.0

### 7.2 La méthode *sqrt(double)*

Cette méthode calcule la racine carrée de son paramètre.

Extrait de la javadoc de la classe **Math** (JAVA SE 5.0):

static double	<a href="#"><code>sqrt</code></a> (double a) Returns the correctly rounded positive square root of a double value.
---------------	---

Exemple:

```
public static void main(java.lang.String[] args) {
System.out.println(" racine carree de 25 = "+Math.sqrt(25.0) );
}
```

Résultat :  
racine carree de 25 = 5.0

### 7.3 La méthode *exp(double)*

Cette méthode calcule l'exponentielle de l'argument

Exemple :

```
public static void main(java.lang.String[] args) {
System.out.println(" exponentiel de 5 = "+Math.exp(5.0) );
}
```

Résultat :  
exponentiel de 5 = 148.4131591025766

### 7.4 La méthode *log(double)*

Cette méthode calcule le logarithme naturel de l'argument.

Exemple :

```
public static void main(java.lang.String[] args) {
```

```
System.out.println(" logarithme de 5 = "+Math.log(5.0) );  
}
```

Résultat :

logarithme de 5 = 1.6094379124341003

## 8 La génération de nombres aléatoires

### 8.1 La méthode *random()*

Cette méthode renvoie un nombre aléatoire compris entre 0.0 et 1.0.

Exemple:

```
public static void main(java.lang.String[] args) {  
    System.out.println(" un nombre aléatoire = " + Math.random());  
}
```

Résultat :

un nombre aléatoire = 0.8178819778125899



## 9 Exemples de programme Java

### 9.1 ValeursMathematiques.java

```
class ValeursMathematiques2
{
    public static void main (String[] args)
    {
        // Le caractere Unicode de la lettre grecque pi est \u03c0
        String texte = "\u03c0 = " + Math.PI + "\n";

        texte += "e = " + Math.E + "\n";

        double sinPiSur2 = Math.cos (Math.PI / 2.);
        texte += "cos (\u03c0 / 2) = " + sinPiSur2 + "\n";

        double sinPiSur4 = Math.sin (Math.PI / 4.);
        texte += "sin (\u03c0 / 4) = " + sinPiSur4 + "\n";

        double racineDe4 = Math.sqrt (4.);
        // Le caractere Unicode du symbole de la racine carree est \u221a
        texte += "\u221a 4 = " + racineDe4 + "\n";

        double racineCarreeDe2Sur2 = Math.sqrt (2.) / 2.;
        // Le caractere Unicode du symbole de la racine carree est \u221a
        texte += "\u221a 2 / 2 = " + racineCarreeDe2Sur2 + "\n";

        double logNeperienDeE = Math.log (Math.E);
        texte += "ln (e) = " + logNeperienDeE + "\n";

        texte += "5 nombres aléatoires entre 1 et 10 = "
            + ((int )(Math.random() * 10 ) + 1) + " "
            + ((int )(Math.random() * 10 ) + 1) + " "
            + ((int )(Math.random() * 10 ) + 1) + " "
            + ((int )(Math.random() * 10 ) + 1) + " "
            + ((int )(Math.random() * 10 ) + 1)
            + "\n";

        javax.swing.JOptionPane.showMessageDialog (null, texte);
        System.exit(0);
    }
}
```

### 9.2 CalculerEmprunt.java

```
package gnu.d06_04_Math.ouils;
public class CalculerEmprunt
{
    /**
     * Calcule le montant des intérêts d'un emprunt.
     * @param mensualite
     * @param nbAnnees
     * @param capital
     * @return Retourne les intérêts
     */
    public static double calculerInteret(double mensualite, int nbAnnees,
double capital) {
```

```
        return (mensualite * nbAnnees * 12) - capital;
    }

    /**
     * Calcule le montant des mensualites d'un emprunt.
     */
    public static double calculerMensualite(double taux, int nbMensualite,
double capital)
    {
        double mensualite = capital * taux / (1 - Math.pow (1 + taux, -
nbMensualite));
        return Math rint(mensualite * 100) / 100.; // Suppression des décimales
superflues
    }
}
```

### 9.3 CalculerInteretTest.java

```
package gnu.d06_04_Math.tests;
import gnu.d06_04_Math.outils.CalculerEmprunt;
import javax.swing.JOptionPane;
class CalculerInteretTest
{
    public static void main (String[] args)
    {
        // La methode showInputDialog affiche une boite de dialogue de saisie textuelle
        // avec le texte passe en parametre comme commentaire
        String texteCapital = JOptionPane.showInputDialog("Capital emprunt\u00e9 :");
        String texteMensualite = JOptionPane.showInputDialog("La mensualit\u00e9 :");
        String texteNbAnnees = JOptionPane.showInputDialog("Dur\u00e9e (en ann\u00e9es)
:");

        // Appel des methodes parse... qui convertissent les valeurs saisies
        double capital = Double.parseDouble (texteCapital);
        float mensualite = Float.parseFloat (texteMensualite);
        int nbAnnees = Integer.parseInt (texteNbAnnees);

        //On calcule avec des doubles et on affiche le résultat sur la console
        double interets = CalculerEmprunt.calculerInteret (mensualite, nbAnnees,
capital);

        // Construction du message
        String message = "Pour un capital emprunt\u00e9 de " + texteCapital + "
\u20ac"
                        + " sur " + texteNbAnnees + " ann\u00e9es"
                        + "\nvos mensualit\u00e9s sont de " + texteMensualite + "
\u20ac"
                        + " et ce cr\u00e9dit vous co\u00fbtera " + interets + "
\u20ac";
        JOptionPane.showMessageDialog(null, message);
        System.exit (0);
    }
}
```

## 9.4 CalculerMensualiteTest.java.java

```
package gnu.d06_04_Math.tests;
import gnu.d06_04_Math.ouils.CalculerEmprunt;
import javax.swing.JOptionPane;
class CalculerMensualiteTest
{
    public static void main (String[] args)
    {
        // La methode showInputDialog affiche une boite de dialogue de saisie
        // avec le texte passe en parametre comme commentaire
        String texteCapital = JOptionPane.showInputDialog("Capital
        emprunt\u00e9 :");
        String texteTaux = JOptionPane.showInputDialog("Taux
        d'int\u00e9r\u00eat (en %) :");
        String texteNbAnnees = JOptionPane.showInputDialog("Dur\u00e9e (en
        ann\u00e9es) :");

        // Appel des methodes parse... qui convertissent les valeurs saisies
        double capital      = Double.parseDouble (texteCapital);
        float  taux         = Float.parseFloat (texteTaux) / 100.f / 12f;
        int    nbMensualite = Integer.parseInt (texteNbAnnees) * 12;

        // Calcul des mensualites et des interets
        double mensualite = CalculerEmprunt.calculerMensualite (taux,
        nbMensualite, capital);
        double interets   = mensualite * nbMensualite - capital;
        // Construction d'un message d'information
        String message =  "Pour un capital emprunt\u00e9 de " + texteCapital +
        " \u20ac"
                        + " sur " + texteNbAnnees + " ann\u00e9es"
                        + " \u00e0 un taux d'int\u00e9r\u00eat de " +
        texteTaux + " %, "
                        + "\nvos mensualit\u00e9s seront de " + mensualite + "
        \u20ac"
                        + " et ce cr\u00e9dit vous co\u00fbtera " + interets +
        " \u20ac";
        JOptionPane.showMessageDialog(null, message);
        System.exit (0);
    }
}
```