

java02 : Communiquer et traiter une information

AFFICHER UNE INFORMATION

- **En mode console : System.out.println(...)**

```
System.out.print( expr ); // affiche à l'écran le contenu de expr
System.out.println( ); // saute une ligne
System.out.println( expr ); // affiche à l'écran le contenu de expr et saute une ligne
System.out.println( "Mon texte" ); // affiche à l'écran le texte et saute une ligne
System.out.println("Mon texte" + expr ); // affiche à l'écran le le texte et le contenu de expr
```

Remarque : println et print sont des méthodes (fonctions) qui ont été définies plusieurs fois de manière à fonctionner quelque soit le type de l'expression qu'on leur passe en paramètre.

- **En mode graphique : JOptionPane.showMessageDialog(...)**

```
import javax.swing.*;

public class Bonjour {

    public static void main(String[] args) {
        String nom = JOptionPane.showInputDialog (null,"Votre nom " ) ;
        JOptionPane.showMessageDialog (null,"Bonjour "+ nom) ;
    }
}
```

Remarque : La méthode showMessageDialog est définie dans la classe JOptionPane qui se situe dans le package swing qui lui même est dans un plus gros package appelé javax.

Remarque : un package est une unité (un fichier ou répertoire) regroupant des classes.

Ainsi pour utiliser, cette méthode, vous pouvez :

- o la préfixez par le nom des packages :

```
javax.swing.JOptionPane.showMessageDialog (null,"texte") ;
```

- o utilisez l'instruction d'import : **import nom du package**

Pour les méthodes print et println, aucun import n'est nécessaire car elles font parties des classes standards.

RECUPERER UNE INFORMATION (SAISIE)

- **En mode console :**

La classe **Scanner** de la version **java** 1.5.0 permet la lecture de données sur l'entrée standard (clavier) ou dans un fichier. Nous parlerons ici simplement de la lecture au clavier. Un objet de type **Scanner** va scanner les données de types primitifs ou de la classe String. Pour utiliser la classe **Scanner**, il faut d'abord l'importer : **import java.util.Scanner;**

Ensuite pour lire des données sur l'entrée standard, il faut créer un objet de la classe

```
Scanner sc = new Scanner(System.in);
```

Pour ensuite récupérer les données, il faut faire appel sur l'**objet sc** avec les méthodes décrites ci-dessous. Voir L'API java <http://java.sun.com/j2se/1.5.0/docs/api/>

```
import java.util.Scanner;
//affiche Bonjour suivi du nom saisi au clavier
Scanner sc = new Scanner(System.in);
System.out.println("entrez votre nom :");
String nom = sc.next();
System.out.println("Bonjour :"+nom);
```

Class Scanner : quelques méthodes

```
String    next()    renvoie la saisie de l'utilisateur sous forme d'une donnée de type String

float     nextFloat() renvoie la saisie de l'utilisateur sous forme d'une donnée de type float

int       nexInt()   ..... sous forme d'une donnée de type int

double    nextDouble() .....sous forme d'une donnée de type Double

etc.....
```

Pour les utiliser, vous devrez faire référence à la classe afin que le compilateur puisse savoir où trouver la définition de ces méthodes.

```
System.out.println(" Entrez votre âge ") ;
int age = sc.nextInt() ;
```

• En mode graphique : JOptionPane.showInputDialog(...)

```
String nom = JOptionPane.showInputDialog (null,"Votre nom ") ;
```

Remarque : cette méthode ne sait renvoyer que des variables de type String. Il faut alors utiliser des méthodes qui permettent de convertir les données. Exemple :

```
String sAge = JOptionPane.showInputDialog (null,"Votre age ") ;
int age = Integer.parseInt(sAge ) ;
```

Exemple de conversions

● Conversion de type **String** vers un type primitif :

```
...
String s1 = "123";
String s2 = ".23";

int    x1 = Integer.parseInt(s1);
double x2 = Double.parseDouble(s2);
float  x3 = Float.parseFloat(s2);
float  x4 = Float.valueOf(s2); // même effet
```

STRUCTURES CONDITIONNELLES ET RÉPÉTITIVES

Les instructions conditionnelles et répétitives sont généralement les mêmes quelque soit le langage.
Remarques :

- L'imbrication est toujours possible
- Pour former une condition, les règles sont aussi les mêmes quelque soit le langage, on utilise les opérateurs de comparaison (<, >, ==, !=...) et les opérateurs logiques (&&, |||)
- S'il n'y a qu'une seule instruction, vous n'êtes pas obligé de mettre les accolades.

FAIRE DES TESTS

Syntaxe		
<pre>if(condition){ //instructions }</pre>	<pre>if(condition){ //instructions } else{ //instructions }</pre>	<pre>if(condition){ //instructions } elseif{ //instructions } else{ //instructions }</pre>

REMARQUES :

- Le mot clef : elseif s'écrit en un seul mot !
- La condition doit porter sur des données de même type.
- Rappel : méfiez vous du sens du mot clef else lorsque vous utilisez des conditions multiples, méfiez vous du sens de votre condition si vous utilisez le mot clef elseif

RÉPÉTER DES ACTIONS

Syntaxe		
<pre>while (condition){ //instructions }</pre>	<pre>do { //instructions } while(condition);</pre>	<pre>for (int i=0;i<10;i++){ //instructions }</pre>

REMARQUE : la condition doit porter sur des variables déclarées en dehors de la boucle, excepté pour l'instruction for.