

# Entrées/Sorties en java

## Introduction

Ce document présente un sous-ensemble minimal d'opérations d'entrées/sorties destinées à servir lors des TP de programmation en première année.

Si vous travaillez avec le poly 118 « Licence Sciences et Technologies. Module API Approche impérative. Méthodes et outils algorithmiques. » de J. Barré, P. Le Certen et L. Ungaro, les opérations ci-dessous remplacent celles présentées dans le poly au § 3.5.8.

## 1 Affichage à l'écran

La fonction

```
System.out.print(expression);
```

affiche à l'écran la valeur de l'expression passée en paramètre ; cette expression peut être :

**un nombre :**

- une constante : `System.out.print(15);` affiche **15**
- une variable : `double nb = -123.48 ; System.out.print(nb);` affiche la valeur de nb : **-123.48**

**une chaîne de caractères :**

- une constante : `System.out.print("Le soleil brille ");`
- une variable : `String nom = "Marc"; System.out.print(nom);`

**une combinaison** quelconque de nombres et de chaînes reliés par l'opérateur de concaténation + :

```
double nb = -123.48 ;
String nom = "Marc" ;
System.out.print("Le nombre vaut " + nb + " et le nom est " + nom);
```

Dans ce dernier cas java convertit tous les nombres en chaînes de caractères et concatène (« colle ») toutes les chaînes l'une à la suite de l'autre avec l'opérateur +. Les sous-expressions arithmétiques doivent être mises entre parenthèses.

**exemple :**

```
double nb1 = -123.48 ; int nb2 = 7;
String nom = "Marc" ;
System.out.print("Le nombre vaut " + (nb1 + nb2) + " et le nom est " + nom);
```

**résultat :** Le nombre vaut -116.48 et le nom est Marc

Pour faire un passage à la ligne en fin d'affichage, utiliser `System.out.println ( expression )`;

## 2 Lecture au clavier

On utilise les fonctionnalités de la classe `Scanner` ; il faut placer la directive

```
1 import java.util.Scanner;
```

au début du programme java.

Avant la première opération de lecture, il faut créer une variable de type `Scanner` :

```
1 Scanner entree = new Scanner(System.in);
```

Ici, la variable s'appelle `entree`, mais tout autre nom est acceptable.

Un scanner permet à un utilisateur de saisir au clavier des données de différents types, séparées par un ou plusieurs délimiteurs : espace, marque de fin de ligne.

## Exemples

```

int age = entree.nextInt();           // lit au clavier et renvoie un entier
// voir aussi entree.nextShort() et entree.nextLong()

double taux = entree.nextDouble();    // lit au clavier et renvoie un réel de type double
// voir aussi entree.nextFloat()

String mot = entree.next();           // lit au clavier et renvoie une chaîne
// Attention, la lecture s'arrête au premier délimiteur

String phrase = entree.nextLine();    // lit jusqu'à la fin de la ligne et renvoie une chaîne

```

saisie	variable	valeur
14	age	14
13,55	taux	13.55
Un joli poisson rouge !	mot	Un
Un joli poisson rouge !	phrase	Un joli poisson rouge !

## Remarques :

- Lors de la saisie de nombre réels, le séparateur décimal dépend de la langue de travail : dans un système français c'est la virgule, dans un système anglo-saxon, c'est un point ;
- Il est possible de saisir un nombre en notation scientifique : pour entrer  $1,25 \times 10^{-7}$ , il faudra saisir  $1,25e-7$ .
- le symbole `␣` représente un espace, le symbole `↵` représente l'appui de la touche « Entrée » et provoque un passage à la ligne ;
- les fonctions de lecture (sauf `nextLine`) ignorent les délimiteurs de tête et arrêtent la lecture au premier délimiteur qui suit la donnée lue.
- Il n'y a pas de fonction de lecture d'un caractère unique : il faut lire une chaîne avec `next` ou `nextLine` puis traiter la chaîne obtenue avec les opérations de chaînes (`length`, `charAt`, ...).

**Fin des opérations de lecture** Après la dernière opération de lecture (en général à la fin du programme), il est nécessaire de « fermer » le scanner ainsi :

```
entree.close();
```