

# Corrigé des exercices du premier chapitre

## Exercice 1.1 Conversion de centimètre en pouces

En vous inspirant du programme Conversion du cours, écrivez un programme qui demande à l'utilisateur une longueur en centimètres et la convertit en une longueur en pouces (inches en anglais). Un centimètre correspond à 0,393701 pouces.

---

```
public class CentimetresPouces {
    public static void main (String[] args) {
        double enCm;
        double enPouces;
        System.out.println("Longueur_en_centimetres?_");
        enCm = Terminal.lireDouble();
        enPouces = enCm *0.393701;
        System.out.print ("La_longueur_en_pouces:_");
        System.out.println(enPouces);
    }
}
```

---

## Exercice 1.2 Terminologie, exécution

Voici un programme :

---

```
public class LeProg{
    public static void main(String[] args){
        double nombre;
        System.out.println("Entrez_un_nombre:_");
        nombre = Terminal.lireDouble();
        System.out.print(nombre);
        System.out.println("_est_un_beau_nombre");
    }
}
```

---

### Question 1

- Quel est le nom de ce programme ? **LeProg**
- Quel terme utilise-t-on en Java pour désigner la catégorie de nombre ? **Une variable**

- 
- Quel terme utilise-t-on en Java pour désigner la catégorie de `Terminal.lireDouble`?  
**Une méthode**
  - Quel terme utilise-t-on en Java pour désigner le type de `" est un beau nombre"`?  
**Une chaîne de caractères**
  - Pourquoi y a-t-il des guillemets autour de `" est un beau nombre"` et pas de guillemets autour de `nombre` à la ligne précédente? **Il y a des guillemets autour de " est un beau nombre" parce qu'il s'agit d'un message à afficher tel quel alors qu'il n'y en a pas autour de nombre parce qu'il ne s'agit pas d'un mot à afficher tel quel mais du nom d'une variable dont il faut afficher le contenu.**

## Question 2

Expliquez ligne par ligne ce qui se passe à l'exécution du programme.

- ligne 3 : déclaration de la variable `nombre`, allocation d'un espace en mémoire pour y stocker un nombre à virgule (type double).
- ligne 4 : affichage à l'écran du message `"Entrez un nombre: "` avec passage à la ligne.
- ligne 5 : lecture d'un nombre à virgule au clavier et stockage de ce nombre en mémoire dans l'espace alloué à la variable `nombre`.
- ligne 6 : affichage du nombre à virgule stocké dans la variable `nombre` sans passage à la ligne.
- ligne 7 : affichage du message `" est un beau nombre"` avec passage à la ligne.

## Exercice 1.3 Erreurs de compilation

Faites un copier-coller de ce programme et corrigez chacune des erreurs de compilation qui se produit.

---

```
public classe Conversion {  
    public static void main (String[] args) {  
        double euros;  
        double dollars  
        System.out.println("Somme_en_euros?_");  
        euros = Terminal.lireDouble();  
        dollars = euros x 1,118;  
        System.out.println(La somme en dollars: );  
        System.out.println(dollars);  
    }  
}
```

---

- File: /home/barthe/prog/java/Conversion.java [line: 1]  
Error: class, interface, or enum expected  
**Le mot-clé class s'écrit sans e.**
- File: /home/barthe/prog/java/Conversion.java [line: 4]  
Error: ';' expected  
**Il manque un point-virgule à la fin de la ligne 4.**
- File: /home/barthe/prog/java/Conversion.java [line: 7]  
Error: ';' expected  
**Ligne 7 : la multiplication s'écrit avec \* et non pas avec x.**

- 
- Ligne 7 : le nombre 1,118 doit s'écrire avec un point et non pas une virgule.
  - File: /home/barthe/prog/java/Conversion.java [line: 8]  
Error: ')' expected  
Ligne 8 : le message à afficher doit être entre guillemets.  
System.out.println(La somme en dollars: );