

# Attributs (variables), Attributs statiques, constantes - JAVA

Date de dernière mise à jour : 27/06/2007 à 19:36

Source : <http://www.vulgarisation-informatique.com/java-attributs.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)

Les attributs :

Un attribut permet de stocker des données à l'intérieur d'une classe. Un attribut se note comme ceci dans la classe (ou dans une méthode) :

[droits d'accès] [type de l'attribut] [nom de l'attribut] [éventuellement une définition ici]

Droits d'accès :

Les droits d'accès à un attribut peuvent être de trois types :

- private : seules les méthodes de la classe dans laquelle est défini l'attribut peuvent y accéder et le modifier
- protected : seules les méthodes de la classe et des éventuelles classes héritées peuvent accéder à l'attribut et le modifier
- public : tout le monde peut accéder à l'attribut en question.

Type de l'attribut :

Les types correspondent aux [types de données](#) vus auparavant.

Nom de l'attribut :

Le nom que vous souhaitez donner à votre attribut. Il vaut mieux qu'il se compose de caractères alphanumériques. L'underscore est autorisé, toutefois évitez de le placer en premier.

Définition de l'attribut :

La définition d'un attribut indique que vous lui avez attribué une valeur. Ceci est facultatif. Si vous n'attribuez pas de valeur à votre attribut, il y aura simplement une zone mémoire réservée, cette zone ne contiendra aucune valeur tant que vous n'aurez pas attribué de valeur à votre attribut.

Exemple :

```
public
class
Test{
    private f
loat
nombre; public double
attribut_public; protected int
nb; public
Test()
{
    nombre =
3.14f; nb =
(
int
)
```

```
8.14; }  
}
```

Pour l'instant rien ne change ou presque de ce que vous avez eu l'occasion de voir jusqu'à présent.

Les tableaux :

Un tableau va vous permettre de stocker plusieurs données au sein d'une seule variable. Les données sont accessibles via des indices numériques.

Pour dire que telle variable est un tableau, on fait succéder le type de la variable par deux crochets : []

Si vous souhaitez attribuer une taille fixe à votre tableau (pour qu'il ne puisse contenir plus d'éléments que spécifié), vous devez utiliser la syntaxe suivante (on prendra un tableau d'entiers pour notre exemple) : **public int[] tableau; tableau = new int[10];**

Ce tableau aura 10 cases numérotées de 0 à 9 (le premier indice étant 0).

Exemple de tableau ayant une taille indéfinie :

```
public  
class  
Test{  
    public  
int  
[] compteur;    public  
Test(  
)  
{  
    compteur[0] =  
0;    compteur[1] =  
1;    System.out.println(  
compteur.length)  
; }  
}
```

Si vous souhaitez récupérer la taille d'un tableau, il faut utiliser l'attribut **length** comme ceci : `System.out.println(`

`&quot;taille : &quot;`

`+`

`tableau.length)`

`;`

Constantes :

Les constantes diffèrent des attributs &quot;normaux&quot; par le fait qu'elles sont définies une fois pour toutes. C'est pourquoi vous devez

obligatoirement sur la même ligne déclarer et définir la constante qui ne pourra avoir une valeur variable au cours du temps. On ajoute le mot clé **final**

pour dire que l'attribut en question est une constante :

```
public  
class
```

```

Test{
    private final float
    NOMBRE =
    3.14f; public
    Test()
    {
        System.out.println(
        &quot;Le nombre flottant vaut : &quot;
        +
        NOMBRE)
    ; }
}

```

Attributs statiques :

Un attribut statique est un attribut qui est commun à tous les objets que vous pourrez créer. On peut par exemple citer un compteur du nombre d'instances de classe que vous aurez lancées. Si vous souhaitez compter le nombre de fois où vous avez instancié la classe &quot;Test&quot;, vous pourrez écrire ceci :

```

public
class
Test{
    public static int
    nombre; public static final int
    nb =
    5; public
    Test(
    )
    {
        nombre++
    ; System.out.println(
    &quot;Nombre d'instances créées : &quot;
    +
    nombre)
    ; }
}

```

L'avantage des attributs statiques est que vous pouvez y accéder même si vous n'avez pas créé d'instance de votre classe. Ainsi, vous pourrez écrire si vous le souhaitez n'importe où dans une méthode ceci :

```

System.out.println(
&quot;Nombre d'instances créées : &quot;
+
Test.nb)
;

```

Ce code affichera 5.

Source : <http://www.vulgarisation-informatique.com/java-attributs.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)