

```

1  /*
2  *  Chapitre 08
3  *  Sujet : Les boucles en Java
4  */
5
6  public class Boucles {
7      public static void main(String[] args) {
8          /*
9              Les boucles permettent de répéter un bloc d'instruction plusieurs fois
10             Il y a 4 sortes de boucles en Java :
11             - la boucle while      : while (true) {...}
12             - la boucle do while  : do {...} while (true)
13             - la boucle for       : for ([inits]; (true); [post-inc]) {...}
14             - la boucle for-each  : for (variable : tableau) {...}
15          */
16
17          /*
18              1. La boucle while
19              -----
20              Il s'agit d'une boucle 'tant que condition est vraie, exécute...'
21              Cette boucle est utilisée lorsqu'on ne connaît pas le nombre d'itérations à l
22              'avance et que le test de continuation de la boucle est fait avant chaque exécution du
23              bloc d'instructions.
24          */
25          int valeurCalculee = 1;
26
27          while (valeurCalculee < 100) {
28              System.out.println("Valeur calculée = " + valeurCalculee);
29              valeurCalculee = valeurCalculee * 2;
30          }
31
32          /*
33              2. La boucle do while
34              -----
35              Il s'agit d'une boucle 'exécute ..., tant que condition est vraie'
36              Cette boucle est utilisée lorsqu'on ne connaît pas le nombre d'itérations à l
37              'avance et que le test de continuation de la boucle est fait après chaque exécution du
38              bloc d'instructions.
39              Contrairement à la boucle while, les instructions sont exécutées au moins une
40              fois avant le test
41          */
42          do {
43              System.out.println("Valeur calculée = " + valeurCalculee);
44              valeurCalculee = valeurCalculee / 2;
45          } while (valeurCalculee >= 1);
46
47          /*
48              3. La boucle for
49              -----
50              Il s'agit d'une boucle 'tant que condition est vraie, exécute...'
51              Cette boucle est utilisée principalement lorsqu'on connaît le nombre d'
52              itérations à l'avance et que le test de continuation de la boucle est fait avant chaque exécution du
53              bloc d'instructions.
54              La syntaxe est :
55              for ( [initialisation de variables locales]; condition d'itération; [
56              instructions post itération] )
57              { bloc d'instructions itératives }
58          */
59
60          for (int var1 = 0; var1 < 10; var1++) {
61              System.out.println("Var1 = " + var1);
62          }
63      }
64  }

```

```

61     for (int var1 = 0, var2 = 10; var1 < var2; var1++, var2--) {
62         System.out.println("Var1 = " + var1 + "; var2 = " + var2);
63     }
64
65     // Remplissage d'un tableau
66     int[] tableau = new int[10];
67     for (int var1 = 0; var1 < 10; var1++) {
68         tableau[var1] = var1;
69     }
70
71     /*
72         4. La boucle for-each
73         -----
74         Il s'agit d'une boucle 'tant qu'il y a des éléments dans le tableau, exécute
75         ...'
76         Cette boucle est utilisée principalement lorsqu'on veut faire des actions
77         avec chaque élément d'un tableau
78         La syntaxe est :
79         for ( variable locale : tableau )
80             { bloc d'instructions pour chaque élément du tableau }
81     */
82     for (int element : tableau) {
83         System.out.println("Valeur dans le tableau " + element);
84     }
85     // A chaque itération, 'element' prend la valeur de l'élément suivant dans le '
86     // tableau'
87     // Le code précédent correspond à
88     for (int indice = 0; indice < tableau.length; indice++) {
89         int element = tableau[indice];
90         System.out.println("Valeur dans le tableau " + element);
91     }
92     // Pour connaître le nombre de colonnes et de lignes dans un tableau à 2
93     // dimensions
94     char[][] tableau2Dimensions = new char[2][10];
95     int line = 1;
96     System.out.printf("Il y a %d lignes dans le tableau %n", tableau2Dimensions.
97     length);
98     for (var ligne : tableau2Dimensions) {
99         System.out.printf("Ligne : %d, il y a %d colonnes%n", line++, ligne.length);
100     }
101     // Il est possible de quitter une boucle (for, while, do while, for-each) en
102     // utilisant la clause 'break'
103     for (int i = 0; i < 15; i++) {
104         if (i == 10) {
105             break;
106         }
107         System.out.println("i = " + i);
108     }
109     // Il est possible de continuer une boucle (for, while, do while, for-each) en
110     // utilisant la clause 'continue'
111     for (int i = 0; i < 15; i++) {
112         if (i == 10) {
113             continue;
114         }
115         System.out.println("i = " + i);
116     }
117 }

```