



Le langage Java

Tableau & Structure de contrôle

Programme détaillé ou sommaire

Les tableaux

- Les tableaux de dimension 1

- Les tableaux de dimension 2

Les structures de contrôle

- if / else if / else

- switch / case / default

- for

- while

- do / while

Normes de codage

Chapitre 1

Les tableaux

Tableaux

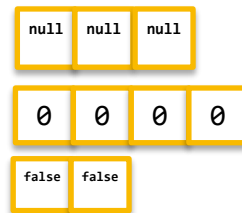
Structure de taille fixe qui contient des composants de même type
types primitifs ou objets

Création du tableau vide

```
String[] prenoms = new String[3];
```

```
int[] data = new int[4];
```

```
boolean[] values = new boolean[2];
```



Tableaux

Le remplissage d'un tableau se fait grâce à un index.

Attention la numérotation commence à 0

opérateur d'accès : []

Création du tableau vide puis remplissage

```
String[] prenom = new String[3];
```

```
prenom[0] = "A";
```

```
prenom[1] = "B";
```

```
prenom[2] = "C";
```

null	null	null
A	null	null
A	B	null
A	B	C

Tableaux

Référencé par des variables

```
String[] prenom = {"a", "b"};
```

```
int[] notes = {12, 20, 3, 14};
```

a	b
---	---

12	20	3	14
----	----	---	----

Tableaux

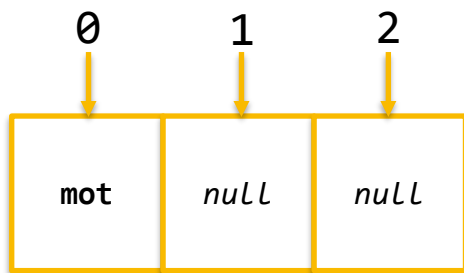
Accès aux éléments par leur **index** (l'index est un entier positif)

commence à l'index 0

opérateur d'accès : []

```
String[] chaines = new String[3];  
chaines[0] = "mot";  
System.out.print(chaines[0]);
```

Index



Longueur = 3

Tableaux

longueur du tableau : **length**

permet de faire des boucles

```
String[] chaines = new String[3];  
chaines[0] = "1ers mots";  
System.out.print(chaines[0]);
```

`chaines.length`

Si accès hors des limites du tableau, erreur à l'exécution

`ArrayIndexOutOfBoundsException`

Exemple: `chaines[3]`

Atelier (TP)

OBJECTIFS :

Utiliser des tableaux.

DESCRIPTION : Dans ce TP vous allez déclarer et utiliser des tableaux.

Chapitre 2

Les structures de contrôle

Introduction

2 types: exécution conditionnelle et itération

Exécutions conditionnelles

if / else if / else

switch / case / default

Itérations

for

while

do / while

if-else

la condition :

```
if (condition) {  
    instructions;  
}
```

```
if (condition) {  
    instructions;  
}  
else {  
    instructions;  
}
```

```
if (condition) {  
    instructions;  
} else if (condition){  
    instructions;  
} else {  
    instructions;  
}
```

Exemple sur 'if-else'

Exemple:

```
if (i == 0) { // retourne un boolean
    System.out.println("i est nul");
} else if (i > 0) {
    System.out.println("i est positif");
} else {
    System.out.println("i est négatif");
}
```

switch

mot-clef **break** : sortie du niveau de traitement.

En cas d'absence de break le « case » suivant est exécuté

valeur peut être:

Un char, short, byte, int

Une String

Une énumération

Note: default peut être avant les « case »

```
switch (valeur) {  
    case (valeur1) : instructions;  
    break;  
    case (valeur2) :  
    instructions;  
    break;  
    default :instructions;  
    break;  
}
```

switch

Switch avec « possibilités multiples »:

```
int index = 0;
int c = getValue();
switch (c) {
    case 1:
    case 2:
        index = 1;
        break;
    case 3:
    case 4:
        index = 2;
        break;
}
```

for

*initialisation, condition de continuation et incrémentation sont **facultatives**.*

La boucle n'est pas exécutée si la condition est fausse au départ mais l'*initialisation* est toujours effectuée.

L'*incrémentation* est une expression qui est exécutée après chaque itération (en fin de bloc).

```
for (initialisation; condition; incrémentation ) {  
    instructions  
}
```


Exemple de boucle 'for'

Exemple

```
for (int i = 0; i < 10; i++) {  
    // calcul du carré de i;  
    int carre = i * i;  
    String chaine = "le carré de " + i + " est " +    carre;  
    System.out.println(chaine);  
}
```

Exemple de boucle 'for'

Exemple équivalent

```
int i = 0;
for (i = 0; i<10; i++) {
    // calcul du carré de i;
    int carre = i * i;
    String chaine = "le carré de " + i + " est " +   carre;
    System.out.println(chaine);
}
```

Exemple de boucle 'while'

Exemple

```
for (int i=0; i<100; i++) {  
  
    if (i == 0) {  
        break;  
    }  
    int div = i/i;  
}
```

break dans une boucle for

Il est possible d'agir sur le déroulement d'une boucle :
le mot-clef **break** permet de sortir de la boucle.

Attention : break n'est pas utilisable dans une condition

while et do-while

Différence entre while et do-while :

Les instructions dans le do-while sont exécutées au moins une fois.

```
while (condition) {  
    instructions;  
}  
  
do {  
    instructions;  
} while (condition);
```

Exemple de boucle 'while'

Exemple

```
int i = 10;
while (i > 0) {
    // calcul du carré de i;
    int carre = i * i;
    String chaine = "le carré de " + i + " est " + carre;
    System.out.println(chaine);
    i--;
}
```

Exemple de boucle 'while'

Exemple

```
for (int i=0; i<2; i++) {  
  
    if (i == 0) {  
        continue;  
    }  
    int div = i/i;  
}
```

Chapitre 3

Agrandir un tableau

Agrandissement de tableau en Java

En Java les tableaux ont une taille fixe.

Pour agrandir un tableau :

```
// tableau initiale avec des valeurs
```

```
int[] tab = { -2, 13, 4, 0 };
```

```
// L'expression à droite du = crée un nouveau tableau à partir de tab mais de  
// taille +1. Ensuite on affecte ce nouveau tableau à tab.
```

```
tab = Arrays.copyOf(tab, tab.length + 1);
```

```
// On alimente la dernière case de tab, désormais avec une case de plus, avec la  
// nouvelle valeur
```

```
tab[tab.length - 1] = 12;
```