

Déclaration d'un tableau de taille fixe

Syntaxe générale : Type des éléments + crochets [].

```
int[] scores;
```

Note : Java autorise la syntaxe équivalente suivante :

```
int scores[];
```

Les crochets indiquent que la variable peut contenir plusieurs éléments du type spécifié

Il existe **deux techniques pour initialiser** les éléments :

1. Dans l'instruction de déclaration
2. Dans des instructions séparées

Initialisation d'un tableau de taille fixe (1)

Si l'on connaît les valeurs de tous les éléments lors de la déclaration du tableau

☞ une seule instruction de **déclaration-initialisation**

1. Déclarer le type du tableau
2. Indiquer les éléments entre accolades
3. Séparer les éléments par des virgules

Exemple : avec des **littéraux** :

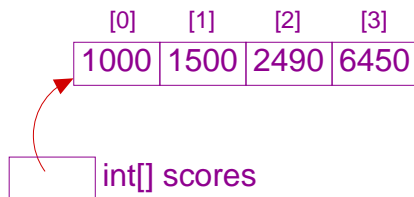
```
int[] scores = {1000, 1500, 2490, 6450};
```

Exemple : avec des **expressions** :

```
int[] scores = {(2500 * 10), (4200 * 10)};
```

Situation en mémoire

Important : Un tableau n'est pas de type de base, il est donc manipulé via une **référence** !



On dit que la variable **scores** **référence** (ou pointe vers) un tableau de **int**.

La variable **scores** contient une adresse : l'emplacement du tableau en mémoire !

Initialisation d'un tableau de taille fixe (2)

Dans le cas général, on ne connaît pas les valeurs de tous les éléments lors de la déclaration du tableau

On utilise alors plusieurs instructions pour **déclarer et initialiser** :

1. Déclarer le type du tableau
2. Construire le tableau avec :
`new type [taille]`
3. remplir le tableau **élément par élément**

La **déclaration-construction** d'un tableau peut se faire avec :

► deux instructions distinctes :

```
int[] scores;           // déclaration
scores = new int[2];    // construction
```

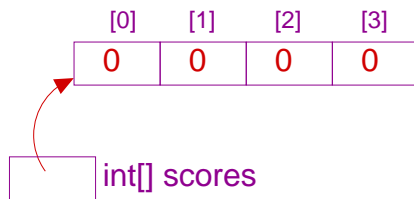
► une seule instruction :

```
int[] scores = new int[2]; // déclaration-
                          // -construction
```

Valeurs par défaut

Chaque élément d'un tableau reçoit une **valeur par défaut** lors de la construction avec `new`

int	0
double	0.0
boolean	false
char	'\u0000'
(objet quelconque)	(null)

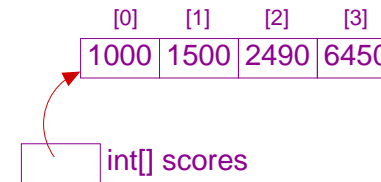


Initialisation d'un tableau de taille fixe (3)

Une fois le tableau déclaré et construit, il faut le **remplir élément par élément** :

```
int[] scores = new int[4];
scores[0] = 1000;
scores[1] = 1500;
scores[2] = 2490;
scores[3] = 6450;
```

Situation en mémoire : Comme pour le 1^{er} exemple d'initialisation




Accès direct aux éléments d'un tableau

Le $i+1^{\text{ème}}$ élément d'un tableau `tab` est accessible au moyen de l'indexation : `tab[i]`

Attention ! Les indices correspondant aux éléments d'un tableau de taille `T` **varient entre 0 et T-1**

Le 1^{er} élément d'un tableau `tab` précédemment déclaré est donc `tab[0]` et son 10^e élément est `tab[9]`

Attention ! En cas de débordement une exception est lancée par le programme  situation d'erreur provoquant l'arrêt du programme si on ne la traite pas (la gestion des exceptions n'est pas présentée dans ce cours d'introduction)

Il est impératif que l'élément auquel vous voulez accéder **existe** effectivement !