

Collections: Les *array*

- Qu'est-ce qu'un *array*?
- Déclarer un *array*
- Accéder aux éléments d'un *array*
- Parcourir un *array*
- Modifier un *array*
- Saviez-vous que?

Qu'est-ce qu'un *array*?

- Un *array* est une structure de données qui contient une collection d'éléments de même type.
- Chaque élément du *array* est identifié par un indice.
- Le nombre d'élément dans un array (sa taille) est défini à sa déclaration et ne peut plus être modifié par la suite.
- C'est un peu comme une variable, mais pour laquelle on réserve plusieurs emplacement mémoire d'un seul coup.

Déclaration d'un *array*

- On ajoute des [] après le type pour préciser qu'il s'agit d'une collection.
- On définit ensuite les valeurs initiales de la collection entre {}
- Par exemple:

```
ENTIER[] lotterieNumbers = [3, 15, 27, 12, 1, 21]  
STRING[] names = ["Alice", "Bruno", "Charlie"]
```

Accès aux éléments d'un *array*

- Les éléments sont accessibles via leur indice.
- L'indice commence toujours à 0 pour le premier élément.
- Par exemple:

```
ENTIER[] lotterieNumbers = [3, 15, 27, 12, 1, 21]  
STRING[] names = ["Alice", "Bruno", "Charlie"]
```

```
ECRIRE lotterieNumbers[0] // 3  
ECRIRE lotterieNumbers[1] // 15  
ECRIRE names[0] // "Alice"  
ECRIRE names[2] // "Charlie"  
ECRIRE lotterieNumbers[5] // 21
```

Parcourir un *array*

- La façon la plus simple est de parcourir le *array* avec une boucle POUR...À
- Pour déterminer la fin de la boucle, on peut stocker, au préalable, la taille du *array* dans une variable.

```
ENTIER[] lotterieNumbers = [3, 5, 7, 1]
POUR i = 0 À lotterieNumbers.taille
    AFFICHER lotterieNumbers[i]
FINPOUR
```

Modifier un *array*

- Les *array* sont des variables, on peut modifier les valeurs qu'il contient avec l'opérateur d'affectation.
- Par exemple:

```
STRING[] names = ["Alice","Bruno","Charlie"]  
ECRIRE names[1] // Bruno  
names[1] = "Benoît"  
ECRIRE names[1] // Benoît
```

Saviez-vous que?

- Saviez-vous que le type `string`, que vous utilisez depuis le début de la session, est en fait une collection de `char`.
- C'est pourquoi vous pouvez accéder à une lettre de la chaîne de caractères avec les opérateurs d'accès aux éléments `[]`.
 - Toutefois, il s'agit d'un *array* immuable, vous ne pouvez pas modifier une lettre directement.

```
STRING name = "Alice"  
Ecrire name[2] // i  
name[0] = 'E' // ERROR
```

Convention de nommage

- Les *array* suivent les mêmes conventions de nommage que les variables (lowerCamelCase, significatif, etc.)
- De plus, puisqu'elles représentent une collection, nous nommons ces variables au pluriel. Cela permet de les identifier rapidement.