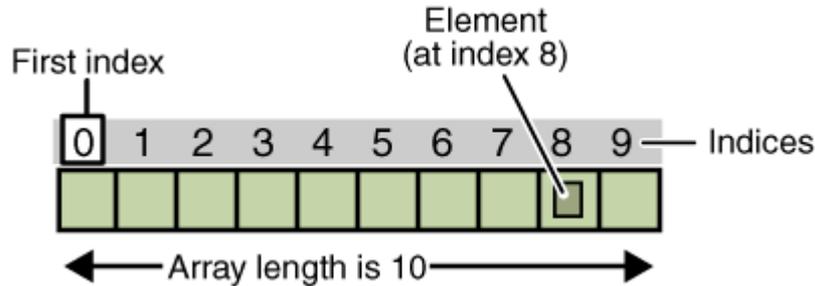


Séquence 5 : Les tableaux

I. Généralités

Les tableaux sont des variables qui permettent de ranger ce qu'on veut (ou presque), des données de tous types : int, String, Boolean, Objet, Sprite et même des tableaux...



Un tableau en ActionScript se dit Array

II. Création d'un tableau

Un tableau est donc une **variable** de **type Array**...

```
var tabEssai : Array = new Array();
var tabEntier : Array = new Array( 0 , 2 , 5 , 3);
var tabString : Array = new Array( "Paris" , "Rennes" , "Carhaix" );
```

Nous voilà avec un tableau vide, ou plus exactement, un tableau sans tiroirs.

Ce qui est pratique, c'est que les tiroirs se fabriquent au fur et à mesure des besoins.

On **ajoute** un élément dans un tableau à l'aide de la méthode **push**.

```
tabEssai.push("Pierre");
```

...vous venez d'ajouter la chaîne "Pierre" dans tabEssai.

```
tabEssai.push("Paul");
```

Qu'est ce que ça donne ? Regardons :

```
trace(tabEssai);
```

Résultat dans Output : _____

```
tabEssai.push("Jacques");
```

```
trace(tabEssai);
```

Résultat dans Output : _____

```
// plusieurs d'un coup :
```

```
tabEssai.push("Marie","Madeleine","Claude")
```

```
trace(tabEssai);
```

Résultat dans Output : _____

III. Lecture écriture dans un tableau

Un tiroir, c'est un emplacement et un emplacement est identifié par un **numéro d'index**.

Ça donne : Pierre est au n° 0 (Attention on commence à 0 !), Paul au 1, et Jacques au 2.

Du coup, c'est très simple de lire ou récupérer l'élément de son choix dès lors que l'on connaît son numéro d'index.

On **accède** à un élément d'un tableau (lecture/écriture) avec la **syntaxe à crochets**, soit :

S.ISN
leTableau [**n° Index**]

```
var tabEssai:Array = new Array();  
tabEssai.push("Pierre");  
tabEssai.push("Paul");  
tabEssai.push("Jacques");
```

```
// lecture :  
trace(tabEssai[0]);  
Résultat dans Output : _____
```

```
// écriture :  
tabEssai[0]="Manon";  
trace(tabEssai);  
Résultat dans Output : _____
```

L'autre truc utile, c'est la propriété **length** qui renvoie le **nombre d'éléments du tableau**.

```
trace(tabEssai.length);  
Résultat dans Output : _____
```

IV. Les fonctions de tableau les plus courantes

Vous pouvez ajouter de nouveaux éléments aux tableaux, leur retirer des éléments, les trier et faire porter une recherche sur leur contenu :

Fonction	Exemple	Description
push	myArray.push("Wizard")	Ajoute une valeur à la fin d'un tableau
pop	myArray.pop()	Supprime la dernière valeur d'un tableau et la retourne
unshift	myArray.unshift("Wizard")	Ajoute une valeur au début d'un tableau
shift	myArray.shift("Wizard")	Supprime la première valeur d'un tableau et la retourne
splice	myArray.splice(7,2,"Wizard","Bard")	Supprime les éléments d'un emplacement du tableau et y insère de nouveaux éléments
indexOf	myArray.indexOf("Rogue")	Retourne l'emplacement d'un élément ou -1 si l'élément n'est pas trouvé
sort	myArray.sort()	Trie un tableau

V. Exercices

Énoncé 1 :

```
var tab1 : Array = new Array( 0 , 2 , 5 , -3 , 19 , -6 , 7 , 4 , 12 , -8 );
```

Écrire une boucle for qui calcule la somme des éléments du tableau tab1.

Énoncé 2 :

```
var tab1 : Array = new Array( 0 , 2 , 5 , -3 , 19 , -6 , 7 , 4 , 12 , -8 );
```

Écrire un programme qui lit 10 nombres entiers dans le tableau tab1 avant d'en rechercher le plus grand et le plus petit.

Énoncé 3 : Soient deux tableaux tab1 et tab2 déclarés ainsi :

```
var tab1 : Array = new Array( 0 , 2 , 5 , -3 , 19 , -6 , 7 , 4 , 12 , -8 );
```

```
var tab2 : Array = new Array(10);
```

Écrire les instructions permettant de recopier, dans tab2, tous les éléments positifs de tab1, en complétant éventuellement tab2 par des zéros.

Énoncé 4 :

Écrire une **fonction** qui fournit en valeur de retour la somme des éléments d'un tableau.

Énoncé 5 :

Écrire un programme qui demande à l'utilisateur de lui fournir un nombre entier entre 1 et 7 et qui affiche le nom du jour de la semaine ayant le numéro indiqué (lundi pour 1, mardi pour 2, ... dimanche pour 7).

VI. Conclusion
