Les tableaux

Table des matières

Introduction

La notion de variable présente dans les langages est étendue avec les tableaux : on peut, grâce à eux, regrouper plusieurs variables dans une même structure. Par exemple, une liste d'utilisateurs. Les tableaux permettent d'effectuer un grand nombre de traitements sur l'ensemble de leurs éléments : ajout, suppression, tri ou encore parcours.

Ce module a pour objectif de présenter le concept de tableaux et leur utilisation : nous étudierons leur création, leur modification, leur parcours, ainsi que les tableaux multidimensionnels et quelques fonctions de bases.

I Déclaration

Objectifs

- Savoir stocker des éléments dans un tableau ;
- Savoir récupérer ou modifier la case d'un tableau.

Mise en situation

À côté des variables classiques comme les chaînes de caractères, les nombres ou les booléens, il existe des variables permettant de stocker plusieurs éléments en même temps.

Ces variables, appelées **tableaux** sont très utilisées, car elles permettent de manipuler un ensemble de données similaires. Imaginez par exemple un logiciel de consultation d'e-mails, il est fort probable que, en interne, chaque dossier soit un tableau ayant pour éléments chaque mail du dossier.

Tableaux

Az Définition

Les tableaux sont des structures pouvant contenir plusieurs éléments, comme des nombres, des chaînes de caractères, etc. Les tableaux stockent le plus souvent un seul type d'éléments, mais les langages comme JavaScript et Python autorisent le mélange de plusieurs types dans un même tableau.

Tableaux de chaînes de caractères

Appellation

Remarque

En JavaScript, ces structures s'appellent des **tableaux** et dépendant d'ailleurs de la classe *Array* (tableau en anglais).

En Python, on utilise plutôt le mot **liste**.

Indexation

Les cases d'un tableau sont identifiées par un index : la première case à l'index 0, la deuxième à l'index 1, la troisième à l'index 2, etc. Les index permettent d'accéder à une case précise d'un tableau pour lire son contenu ou le modifier.

L'index commençant à 0, les cases d'un tableau de 7 éléments seront donc numérotées de l'index 0 à 6.

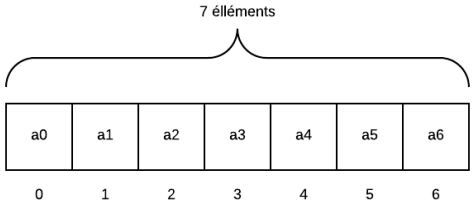

Tableau indexé

Déclaration

Syntaxe

En JavaScript comme en Python, un tableau se déclare avec des **crochets** et les éléments qu'il contient sont séparés par des virgules : `[element1, element2, element3]`.

Un tableau peut aussi être créé vide. Ses éléments peuvent être des nombres, des booléens, des chaînes de caractères, etc.

Exemple

Les tableaux sont affichables en sortie.

```python
"""Python: tableau de 5 cases."""
playlist = ['Remy', 'Népal', 'Kery James', 'August D.', 'Damso']

print(playlist)
```

```javascript
/** JavaScript: tableau de 5 cases */
const playlist = ['Remy', 'Népal', 'Kery James', 'August D.', 'Damso']

console.log(playlist)
```

Lecture des cases

Syntaxe

On accède aux éléments d'un tableau par leur index entre crochets. La syntaxe est identique en Python et JavaScript : `array[0]` renvoie le premier élément du tableau `array`, `array[1]` le deuxième, `array[5]` le quatrième, etc.

Accéder à un index qui n'existe pas dans le tableau cause une erreur en Python ou renvoie la valeur `undefined` en JavaScript.

Exemple

```python
"""Python."""
playlist = ['Remy', 'Népal', 'Kery', 'August D.', 'Damso']

print(playlist[0]) # Remy
print(playlist[2]) # Kery
print(playlist[5]) # error
```

```javascript
/** JavaScript */
const playlist = ['Remy', 'Népal', 'Kery', 'August D.', 'Damso']

console.log(playlist[0]) // Remy
console.log(playlist[2]) // Kery
console.log(playlist[5]) // undefined
```

Écriture dans un case

Syntaxe

On peut écrire dans la case d'un tableau à partir de son index en utilisant la même syntaxe `tableau[0] = 'nouvelle valeur'`.

En Python, seule les cases déjà affectées dans le tableau peuvent être modifiées.

En JavaScript, on peut aussi utiliser cette écriture pour ajouter des éléments au tableau (à un index qui n'existe pas encore).

Exemple

```python
"""Python."""
playlist = ['Remy', 'Damso', 'Kery', 'August D.', 'Sniper']
print(playlist[3]) # August D.

playlist[3] = 'Orelsan'
playlist[5] = 'Keny Arkana' # error

print(playlist[3]) # Orelsan

```

```javascript
/** JavaScript */
const playlist = ['Remy', 'Népal', 'Kery', 'August D.', 'Damso']
console.log(names[3]) // August D.

playlist[3] = 'Orelsan'
playlist[5] = 'Keny Arkana'

console.log(playlist[3]) // Orelsan
```

Types

Complément

D'autres types de tableaux existent avec des caractéristiques légèrement différentes pour s'adapter à plusieurs types de situation. Parmi eux, les dictionnaires qui utilisent le principe de clé-valeur : chaque case du tableau est nommé par une clé (et non un index) qui permet d'accéder à la valeur stockée.

À retenir

Les tableaux servent à stocker plusieurs éléments en les rangeant dans des cases, accessibles et modifiables via leur index.

II Exercice : Appliquer la notion

On veut réaliser un riz au lait. On dispose pour cela de la liste des ingrédients :

- 1L de lait,
- 100g de riz blanc rond,
- 5 cuillères à soupe de sucre,
- 1 sachet de sucre vanillé,
- 1 zeste de citron.

Question 1

[solution n°1 p. 25]

Réaliser un programme JavaScript qui déclare un tableau et stocke les éléments dedans.

Question 2

[solution n°2 p. 25]

Pour alléger la recette, on souhaite réduire la quantité de sucre. Reprendre le programme précédent et ajouter une instruction permettant de remplacer la case *« 5 cuillères à soupe de sucre »* par *« 2,5 cuillères à soupe de sucre »*.

III Parcours de tableaux

Objectif

- Savoir parcourir les éléments d'un tableau.

Mise en situation

Le parcours d'un tableau permet de passer sur ses éléments un à un, en combinant notamment un index et une boucle.

Taille d'un tableau

Syntaxe

Le nombre d'éléments d'un tableau définit sa **taille**.

En Python, la taille d'un tableau est calculée grâce à la fonction `len()`.

En JavaScript on utilise l'attribut `length`.

Afficher le nombre de jours

Exemple

```python
"""Python."""
week = ['Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi', 'Samedi', 'Dimanche']
print('Une semaine =', len(week), 'jours')
```

```javascript
/** JavaScript */
const week = ['Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi', 'Samedi',
 'Dimanche']
console.log('Une semaine =', week.length, 'jours')
```

Boucle for

Rappel

La structure itérative `for` est une boucle qui s'utilise avec un compteur incrémenté à chaque tour.

La boucle for permet d'itérer un **nombre connu** de fois.

Parcours par index

Syntaxe

La taille du tableau étant connue à l'avance, on utilise la boucle `for` avec un compteur allant de 0 au dernier index du tableau (soit taille - 1).

En Python, la boucle `for` utilise `range(taille)` pour représenter les valeurs allant de 0 à `taille - 1`.

En JavaScript, on utilise la condition `compteur < taille` pour finir la boucle quand `compteur` dépasse le dernier index.

Il faut bien penser au fait que le dernier index d'un tableau est égal à sa **taille - 1** puisque l'indexation commence à 0 et non à 1.

Exemple

```python
"""Python."""
week = ['Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi', 'Samedi', 'Dimanche']

for i in range(len(week)):
  print('Jour', i, week[i])
```

```javascript
/** JavaScript */
const week = ['Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi', 'Samedi',
'Dimanche']

for (let i = 0; i < week.length; i++) {
  console.log('Jour', i, week[i])
}
```

Boucles « pour chaque »

Complément

Pour parcourir un tableau du début à la fin, on peut se passer d'utiliser les index et la taille du tableau.

Il existe un type de boucle fait pour prendre un à un chaque élément du tableau, du premier au dernier.

En Python, on utilise `for...in`, et son équivalent en JavaScript est `for...of`.

Afficher chaque jour de la semaine

Exemple

```python
"""Python."""
week = ['Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi', 'Samedi', 'Dimanche']

for day in week:
  print(day)
```

```javascript
/** JavaScript */
const week = ['Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi', 'Samedi',
'Dimanche']

for(const day of week) {
  console.log(day)
}
```

À retenir

Les tableaux indexés sont parcourus grâce à un compteur d'index avec une boucle `for`.

IV Exercice : Appliquer la notion

Question 1

[solution n°3 p. 25]

Un emploi du temps vous est transmis sous la forme de deux listes :

- Une première liste contient les matières,
- Une seconde contient les horaires de chaque matière.

Chaque matière de la première liste va de pair avec l'horaire de la seconde liste qui se trouve au même index : la première matière correspond au premier horaire, la seconde matière au deuxième horaire, etc.

```
/** JavaScript */
const subjects = ['Maths', 'Anglais', 'Sport', 'Sciences économiques']
const schedule = ['8h30', '10h30', '14h00', '17h00']
```

Question 2

[solution n°4 p. 25]

Compléter le programme pour afficher une à une les matières associées à leur horaire.

Indice :

Un seul index est nécessaire pour accéder à l'élément de `subjects` et de `schedule`.

Question 3

[solution n°5 p. 25]

Une erreur s'est glissée dans l'emploi du temps et les matières sont en fait dans l'ordre inverse. Modifier la boucle pour afficher les matières dans l'ordre inverse.

Indice :

On peut accéder à l'élément d'index opposé en soustrayant la valeur du compteur au dernier index du tableau, soit `tableau.length - 1 - compteur`.

V Tableaux multidimensionnels

Objectifs

- Savoir créer et utiliser un tableau à deux dimensions ;
- Savoir parcourir un tableau à deux dimensions.

Mise en situation

Un tableau classique contient une seule ligne de plusieurs colonnes : il peut être assimilé à une liste. Or un tableau peut contenir des variables, mais aussi d'autres tableaux. Dans ce cas on parle de tableau **multidimensionnel**, ayant plusieurs lignes et plusieurs colonnes. Le plus courant est le tableau à 2 dimensions, qui permet de très bien représenter les tableaux que l'on manipule dans le monde physique. Par exemple la grille de pixels d'un écran est un tableau à 2 dimensions.

Nous allons voir qu'ils sont assez simple à utiliser, bien qu'ils nécessitent un usage vigilant de boucles imbriquées pour être parcourus.

Tableau 2D

Syntaxe

Un tableau multidimensionnel est un tableau qui contient lui même des tableaux. Un tableau en 2 dimensions est donc un tableau dont les éléments sont eux-même des tableaux de valeurs.

En Python et en JavaScript, on ajoute une deuxième dimension en utilisant la même notation entre crochets à l'intérieur d'un tableau : `tableau = [[a1, a2], [b1, b2], [c1, c2]]`.

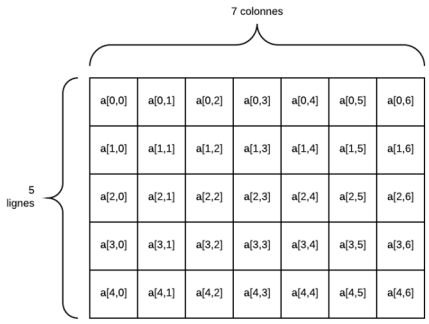

Tableau à 2 dimensions

Exemple

Une classe est divisée en 4 groupes de 3 étudiants chacun.

```python
"""Python: tableau de 4 groupes de 3 élèves."""
groups = [
  ['Manon', 'Lou', 'Alexandre'],
  ['Lucas', 'Antonin', 'Ayman'],
  ['Elliot', 'Kylian', 'Alix'],
  ['Cassandre', 'Tom', 'Erika']
]
```

```javascript
/** JavaScript: tableau de 4 groupes de 3 élèves */
const groups = [
  ['Manon', 'Lou', 'Alexandre'],
  ['Lucas', 'Antonin', 'Ayman'],
  ['Elliot', 'Kylian', 'Alix'],
  ['Cassandre', 'Tom', 'Erika']
]
```

Parcours d'un tableau à deux dimensions

En deux dimensions, un tableau doit être parcouru **doublement** : on utilise donc une boucle imbriquée, utilisant deux compteurs différents, un premier pour le parcours en **longueur** (ensemble des lignes), un deuxième pour le parcours en **profondeur** (éléments d'une ligne).

Lecture d'un élément dans un tableau à deux dimensions

Syntaxe

Pour lire un tableau bi-dimensionnel, on doit utiliser deux index : l'index de la colonne et l'index de la ligne. Le premier index permet de récupérer un tableau correspondant à la ligne, et le deuxième index accède enfin à l'élément.

On utilise donc deux paires de crochets successives : `tableau[i][j]` avec « *i* » l'index de la ligne et « *j* » l'index de la colonne.

Exemple

Le programme parcours un à un chaque groupe et affiche ses éléments.

```python
"""Python: tableau 2 dimensions de taille 4."""
groups = [
  ['Manon', 'Lou', 'Alexandre'],
  ['Lucas', 'Antonin', 'Ayman'],
  ['Elliot', 'Kylian', 'Alix'],
  ['Cassandre', 'Tom', 'Erika']
]

for i in range(len(groups)):
  print('\nGroupe', i)
  for j in range(len(groups[i])):
    print(groups[i][j])
```

```javascript
/** JavaScript: tableau 2 dimensions de taille 4 */
const groups = [
  ['Manon', 'Lou', 'Alexandre'],
  ['Lucas', 'Antonin', 'Ayman'],
  ['Elliot', 'Kylian', 'Alix'],
  ['Cassandre', 'Tom', 'Erika']
]

for (let i = 0; i < groups.length; i++) {
  console.log('\nGroupe', i)
  for (let j = 0; j < groups[i].length; j++) {
    console.log(groups[i][j])
  }
}
```

Nombre de dimensions

Complément

On peut créer des tableaux de plus de 2 dimensions : 3 dimensions si un tableau contient des tableaux qui eux-même contiennent des tableaux de valeurs, 4 dimensions si un tableau contient des tableaux qui eux-même contiennent des tableaux qui contiennent des tableaux de valeurs, etc.

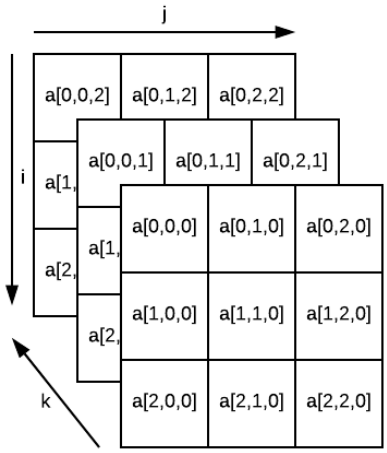

Tableau à 3 dimensions

À retenir

Les tableaux multidimensionnels permettent de stocker d'autres tableaux au lieu de simples valeurs.

VI Exercice : Appliquer la notion

Une enseignante conserve la liste des notes obtenues par chacune de ses classes dans un tableau. Elle veut calculer la moyenne des notes pour chaque classe et l'afficher.

```
[
  [15, 8, 11, 16], // classe 1
  [11, 10, 9, 13], // classe 2
  [12, 15, 15, 9], // classe 3
  [17, 8, 13, 16], // classe 4
  [13, 7, 15, 11]  // classe 5
]
```

Question

[solution n°6 p. 25]

Réaliser le programme JavaScript qui parcours le tableau pour calculer la moyenne de chaque classe. Afficher cette moyenne à côté du numéro de la classe et son numéro.

Indice :

La moyenne se calcule en faisant la somme des notes de toute une classe divisée par le nombre de notes.

Le nombre de notes correspond au nombre d'éléments de chaque sous-tableau, accessible avec `.length`.

VII Opérations sur les tableaux

Objectifs

- Savoir utiliser les fonctions de manipulation tableaux ;
- Savoir retrouver un élément sans son index ;
- Ajouter/retirer des éléments d'un tableau.

Mise en situation

Nous avons vu comment parcourir différents types de tableaux, mais il y a d'autres opérations intéressantes à réaliser. En effet les tableaux ne sont pas des structures fixes, leur taille et les éléments qui les composent sont susceptibles d'évoluer dans le temps. On dit que les tableaux sont **mutables**. Il est possible de trier un tableau, rechercher un élément précis, ou y ajouter des éléments. Pour réaliser ces opérations, la plupart des langages proposent des fonctions clé en main, que nous allons étudier.

Trier un tableau

Syntaxe

En Python et en JavaScript :

- la fonction `sort()` applique un tri par défaut à un tableau (croissant),
- `reverse()` inverse l'ordre des éléments.

Attention

En JavaScript, `sort()` trie les nombres dans l'ordre *« alphabétique »* et non pas numérique : 11 est devant 7 car le premier caractère 1 vient avant 7.

Exemple

```python
"""Python."""
names = ['Maria', 'Tarek', 'Anaïs', 'Benjamin']
notes = [12, 7, 18, 14]

# modifie les listes
names.sort() # ['Anaïs', 'Benjamin', 'Maria', 'Tarek']
notes.sort() # [7, 12, 14, 18]
notes.reverse() # [18, 144, 12, 7]

print('Participants:', names)
print('Notes:', notes)
```

```javascript
/** JavaScript */
const names = ['Maria', 'Tarek', 'Anaïs', 'Benjamin']
const notes = [12, 7, 18, 14]

names.sort() // ['Anaïs', 'Benjamin', 'Maria', 'Tarek']
notes.sort() // [12, 14, 18, 7]
notes.reverse() // [7, 18, 14, 12]
```

```
8
9 console.log('Participants:', names)
10 console.log('Notes:', notes)
```

Ajouter et retirer un élément

Syntaxe

En Python, `append()` permet d'ajouter un élément à la fin et `pop()` retire le dernier élément.

En JavaScript, on ajoute à la fin avec `push()` et on retire le dernier avec `pop()`.

Exemple

```python
"""Python."""
travel_plan = ['Paris', 'Berlin', 'Osaka', 'Busan']
print('Escales prévues:', travel_plan)

travel_plan.append('Pekin') # ['Paris', 'Berlin', 'Osaka', 'Busan', 'Pekin']
print('Nouvelle destination:', travel_plan)

travel_plan.pop() # ['Paris', 'Berlin', 'Osaka', 'Busan']
print('Annulation:', travel_plan)
```

```javascript
/** JavaScript */
const travelPlan = ['Paris', 'Berlin', 'Osaka', 'Busan']
console.log('Escales prévues:', travelPlan)

travelPlan.push('Pekin') // ['Paris', 'Berlin', 'Osaka', 'Busan', 'Pekin']
console.log('Nouvelle destination:', travelPlan)

travelPlan.pop() // ['Paris', 'Berlin', 'Osaka', 'Busan']
console.log('Annulation:', travelPlan)
```

Rechercher un élément

Syntaxe

On peut retrouver l'index d'un élément donné dans un tableau.

En Python, la fonction `index()` retourne l'index du premier élément correspondant à la recherche.

En JavaScript, la fonction `indexOf()` fait la même chose.

Exemple

```python
"""Python."""
months = ['Janv', 'Fev', 'Mars', 'Avr', 'Mai', 'Juin', 'Juil', 'Août', 'Sept',
 'Nov', 'Dec' ]

index = months.index('Août') # retourne 7
print('Août est le', index + 1, 'e mois de l\'année')
```

```javascript
/** JavaScript */
const months = ['Janv', 'Fev', 'Mars', 'Avr', 'Mai', 'Juin', 'Juil', 'Août',
'Sept', 'Nov', 'Dec']

const index = months.indexOf('Août') // retourne 7
console.log('Août est le', index + 1, 'e mois de l\'année')
```

Documentation Complément

De nombreuses autres fonctions existent et sont documentées ici :

- Listes Python : https://docs.python.org/fr/3/tutorial/datastructures.html?highlight=liste
- Tableaux JavaScript : https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/Array

À retenir

Les tableaux sont des structures **mutables** : des fonctions propres à chaque langage permettent d'y appliquer un grand nombre d'opérations pour manipuler ses éléments.

VIII Exercice : Appliquer la notion

On souhaite informatiser l'appel des élèves dans une classe. Pour cela, on a besoin de récupérer les noms des élèves dans un tableaux pour les envoyer à l'application.

Question 1

[solution n°7 p. 26]

Réaliser un programme JavaScript qui déclare un tableau `students`.

Il doit ensuite réaliser autant d'itérations qu'il y a d'élèves et afficher *« Élève n°x : »* à chaque itération, avec le bon numéro.

On connaît le nombre d'élèves au total : 3.

Indice :

Pour l'instant, comme les élèves ne sont pas connus, on n'accède pas aux éléments du tableau `students`, qui reste vide.

Question 2

[solution n°8 p. 26]

Il faut maintenant demander le nom de l'élève à chaque itération pour l'ajouter au tableau (si le nom donné n'est pas vide). Modifier la boucle pour cela. Le programme finira par afficher le nombre total d'élèves réellement présents et le contenu du tableau d'appel par ordre alphabétique.

IX Quiz

Exercice 1 : Quiz - Culture

[solution n°9 p. 26]

Exercice

À quoi sert l'index dans un tableau ?

- **A** À lire le contenu d'une case
- **B** À parcourir un tableau dans une boucle
- **C** À écrire dans le contenu d'une case
- **D** À créer une copie d'un tableau
- **E** À connaître la taille d'un tableau

Exercice

Que peut contenir un tableau en JavaScript ?

- **A** Des chaînes de caractères
- **B** Des nombres
- **C** Des entiers et des décimaux
- **D** Des tableaux
- **E** Tout en même temps

Exercice 4 : Quiz - Méthode

[solution n°10 p. 27]

Exercice

Quelle syntaxe permet, en JavaScript, d'accéder au 5^{e} élément du tableau rangé à l'index 8 d'un autre tableau nommé `inception` ?

Exercice

Comment copier le premier élément d'un tableau `todo` et l'ajouter à la fin d'un tableau `done` en JavaScript?

A `done.append(todo[1])`

B `done.push(todo[0])`

C `done[done.length - 1] = todo[0]`

D `const tmp = todo[0] done.push(tmp)`

Exercice 7 : Quiz - Code

[solution n°11 p. 28]

Exercice

Sans exécuter le code, déterminer l'affichage du programme suivant.

```
1 const brothers = ['Issa', 'Amaury', 'Nicolo', '']
2 brothers.push('Mateo')
3 console.log(brothers.length)
```

A 3

B 4

C 5

Exercice

Quelle instruction renvoie *« Le Caire »* ?

```
1 const route = ['Paris', 'Marseille', 'Tunis', 'Le Caire', 'Ankara', 'Le Caire',
  'Paris']
```

A `console.log(route[3])`

B `console.log(route[route.length - 2])`

C `console.log(route.indexOf(5))`

D `console.log(route[route.length - 1])`

E `console.log(route[4])`

Exercice

Quelle structure permettra de parcourir tout le tableau `array` en JavaScript ?

A `for(let i = 0; i < array.size; i++) {...}`

B `for(let i = 1 ; i < array.length; i++) {...}`

C `for(let i = array.length - 1; i >= 0; i--) {...}`

D `for(let i = 0; i <= array.length; i++) {...}`

X Exercice : Défi

Vous avez listé vos films d'animation préférés en renseignant les caractéristiques suivantes : *« titre »*, *« titre original »*, *« durée »*, *« année »*, *« réalisateur »*, *« studio »*, *« origine »*, *« personnage principal »*, *« distinction majeure »* :

- Le voyage de Chihiro ; Sen to Chihiro no kamikakushi ; 124 min ; 2001 ; Hayao Miyazaki ; Studio Ghibli ; Japonais ; Chihiro ; 10e sur les 100 meilleurs films de tous les temps en langue non-anglaise ;
- Le Château ambulant ; Hauru no ugoku shiro ; 119 min ; 2004 ; Studio Ghibli ; Hayao Miyazaki ; Japonais ; Hauru ; Nomination à l'Oscar du meilleur film d'animation en 2006 ;
- Anastasia ; Anastasia ; 94 min ; 1997 ; Fox Animation Studios ; Don Bluth et Gary Goldman ; Américain ; Anastasia ;
- Spirit, l'étalon des plaines ; Spirit: Stallion of the Cimarron ; 84 min, 2002 ; DreamWorks Animation ; Kelly Asbury ; Américain ; Spirit ; Nomination à l'Oscar du meilleur film d'animation en 2003 ;
- Ratatouille ; Ratatouille ; 111 min ; 2007 ; Brad Bird ; Pixar ; Américain ; Rémy; Oscar du meilleur film d'animation en 2007.

On souhaite présenter nos favoris grâce à un programme qui permettra d'afficher les détails sur chaque film.

Question 1

[solution n°12 p. 29]

Initialiser le programme en stockant ces films dans un tableau.

Indice :

Le tableau est en deux dimensions : un tableau de films, chacun étant lui-même un tableau dont les cases stockent ses caractéristiques.

Question 2

[solution n°13 p. 30]

Compléter le programme pour qu'il affiche la liste des titres de chaque film et leur date de sortie, précédés du numéro du film. Les numéros commencent à 1.

Ex : *1 Le voyage de Chihiro (2001)*

Question 3

[solution n°14 p. 30]

Vous voulez maintenant pouvoir sélectionner un numéro de film et afficher ses informations. Compléter le programme pour récupérer un numéro de film et l'afficher, en répétant l'opération tant que l'utilisateur le souhaite. Il faut pour cela ajouter une option *« X »* pour sortir du menu.

Il faut également s'assurer que l'entrée corresponde soit à l'option *« Sortir »*, soit à un numéro de film.

Voici un exemple d'affichage que le code doit produire :

```
X Sortir
1 Le voyage de Chihiro (2001)
2 Le Château ambulant (2004)
3 Anastasia  (1997)
4 Spirit, l'étalon des plaines (2002)
5 Ratatouille (2007)
```

```
Sélection > 1
Titre: Le voyage de Chihiro
Titre original: Sen to Chihiro no kamikakushi
Durée: 124 min
Année de sortie: 2001
Réalisateur: Hayao Miyazaki
Studio: Studio Ghibli
Origine: Japonais
Personnage principal: Chihiro
Distinction: 10e sur les 100 meilleurs films de tous les temps en langue non-anglaise
```

Indice :

Il faut tester l'entrée en vérifiant qu'elle est soit égale à *« X »*, soit comprise entre 1 et la taille du tableau.

Question 4

[solution n°15 p. 31]

Finalement, vous voulez ajouter au tableau les films préférés des utilisateurs. Il faut donc ajouter au programme une option *« A »* pour pouvoir ajouter un film, qui demande chaque information, crée le film et l'ajoute au tableau.

Il n'est pas utile de vérifier les informations entrées pour l'instant.

Indice :

Pour créer un nouveau film, il suffit de créer un nouveau tableau vide et d'y ajouter les caractéristiques une à une.

Indice :

Utiliser la méthode `films.push` pour rajouter le tableau nouvellement créé à la liste des films.

Conclusion

Nous avons appris à utiliser les tableaux, des structures permettant de stocker différents éléments sous forme de liste.

Il est même possible d'utiliser des tableaux pour stocker d'autres tableaux, et ainsi former des tableaux multidimensionnels.

Nous avons aussi constaté que la boucle `for` et son compteur se sont révélés fort utiles pour le parcours d'un tableau.

Enfin, les tableaux étant directement implémentés dans les langages de programmation, un grand nombre d'opérations peuvent être réalisées dessus.

Solutions des exercices

Solution n°1

[exercice p. 7]

```
/** JavaScript */
const ingredients = ['1L de lait', '100g de riz blanc rond', '5 cuillères à soupe de sucre', '1 sachet de sucre vanillé', '1 zeste de citron']
```

Solution n°2

[exercice p. 7]

```
/** JavaScript */
const ingredients = ['1L de lait', '100g de riz blanc rond', '5 cuillères à soupe de sucre', '1 sachet de sucre vanillé', '1 zeste de citron']

ingredients[2] = '2,5 cuillères à soupe de sucre'

```

Solution n°3

[exercice p. 10]

Solution n°4

[exercice p. 10]

```
/** JavaScript */
const subjects = ['Maths', 'Anglais', 'Sport', 'Sciences économiques']
const schedule = ['8h30', '10h30', '14h00', '17h00']

for(let i = 0; i < subjects.length; i++) {
  console.log(subjects[i], schedule[i])
}
```

Solution n°5

[exercice p. 10]

```
/** JavaScript */
const subjects = ['Maths', 'Anglais', 'Sport', 'Sciences économiques']
const schedule = ['8h30', '10h30', '14h00', '17h00']

for(let i = 0; i < subjects.length; i++) {
  console.log(subjects[subjects.length - 1 - i], schedule[i])
}
```

On aurait pu aussi partir de `i = subjects.length - 1` et décrémenter le compteur jusqu'à 0.

Solution n°6

[exercice p. 14]

```
/** JavaScript: tableau 2 dimensions de taille 5 */
const notes = [
  [15, 8, 11, 16], // classe 1
  [11, 10, 9, 13], // classe 2
  [12, 15, 15, 9], // classe 3
  [17, 8, 13, 16], // classe 4
  [13, 7, 15, 11]  // classe 5
]

```

```
for(let i = 0; i < notes.length; i++) {
  let sum = 0 // initialement nul
  for(let j = 0; j < notes[i].length; j++) {
    sum = sum + notes[i][j] // ajoute chaque note de la classe n°i
  }
  console.log('Classe', i+1, ':',  sum/notes[i].length) // moyenne = somme/nombre de notes
}
```

Solution n°7

[exercice p. 18]

```
/** JavaScript */
const students = []
const nbStudents = 3

for (let i = 0; i < nbStudents; i++) {
  console.log('Élève n°' + (i + 1) + ':')
}
```

Solution n°8

[exercice p. 18]

```
/** JavaScript */
let students = []
const nbStudents = 28

for(let i = 0; i < nbStudents; i++) {
  let name = prompt('Élève n°' + (i+1) + ':')
  if(name !== '')
    students.push(name)
}

console.log(students.length, 'élèves présents')
console.log(students.sort())
```

Solution n°9

[exercice p. 19]

Exercice

À quoi sert l'index dans un tableau ?

- **A** À lire le contenu d'une case
- **B** À parcourir un tableau dans une boucle
- **C** À écrire dans le contenu d'une case
- **D** À créer une copie d'un tableau
- **E** À connaître la taille d'un tableau

Exercice

Que peut contenir un tableau en JavaScript ?

A Des chaînes de caractères

B Des nombres

C Des entiers et des décimaux

D Des tableaux

E Tout en même temps

Un tableau peut contenir n'importe quel type, y compris des tableaux, et n'est pas limité à un type unique.

Solution n°10

[exercice p. 19]

Exercice

Quelle syntaxe permet, en JavaScript, d'accéder au 5e élément du tableau rangé à l'index 8 d'un autre tableau nommé `inception` ?

inception[8][4]

`inception` contient un tableau à l'index 8 (on y accède par `inception[8]`) qui lui même possède un 5e élément (rangé à l'index 4).

Exercice

Comment copier le premier élément d'un tableau `todo` et l'ajouter à la fin d'un tableau `done` en JavaScript?

A `done.append(todo[1])` `append` n'est pas une fonction JavaScript

B `done.push(todo[0])` `push` permet d'ajouter l'élément entre parenthèse à la fin du tableau

C `done[done.length - 1] = todo[0]` Cette instruction n'ajoute pas l'élément à la fin, mais modifie le dernier élément

D `const tmp = done.push(tmp) todo[0]` Équivalent à la réponse 2 en passant par une variable intermédiaire

Solution n°11

[exercice p. 20]

Exercice

Sans exécuter le code, déterminer l'affichage du programme suivant.

```
1 const brothers = ['Issa', 'Amaury', 'Nicolo', '']
2 brothers.push('Mateo')
3 console.log(brothers.length)
```

A 3

B 4

C 5

Le mot vide occupe quand même une case, et `push` ajoute un 5e élément.

Exercice

Quelle instruction renvoie « *Le Caire* » ?

```
1 const route = ['Paris', 'Marseille', 'Tunis', 'Le Caire', 'Ankara', 'Le Caire',
  'Paris']
```

A `console.log(route[3])`

B `console.log(route[route.length - 2])` Retourne bien l'avant dernier élément du tableau, soit le 6e élément, d'index 5.

C `console.log(route.indexOf(5))` `indexOf(5)` recherche l'élément 5 dans `route`, qui n'existe pas.

D `console.log(route[route.length - 1])` Retourne le dernier élément du tableau.

E `console.log(route[4])` L'index 4 désigne le 5e élément

Exercice

Quelle structure permettra de parcourir tout le tableau `array` en JavaScript ?

A `for(let i = 0; i < array.size; i++) {...}` `size` n'est pas le bon attribut en JavaScript.

B `for(let i = 1 ; i < array.length; i++) {...}`

C

`for(let i = array.length - 1; i >= 0; i--) {...}` La boucle va du dernier index à 0, soit un parcours à l'envers.

D

`for(let i = 0; i <= array.length; i++) {...}` Cette boucle ne s'arrête pas au dernier élément mais après, soit un élément indéfini (l'index = taille - 1).

Solution n°12

[exercice p. 22]

```
/** JavaScript */
const films = [
  // titre, titre original, durée (min), année, réalisateur, studio, origine, personnage principal, distinction
  [
    'Le voyage de Chihiro',
    'Sen to Chihiro no kamikakushi',
    124,
    2001,
    'Hayao Miyazaki',
    'Studio Ghibli',
    'Japonais',
    'Chihiro',
    '10e sur les 100 meilleurs films de tous les temps en langue non-anglaise'
  ],
  [
    'Le Château ambulant',
    'Hauru no ugoku shiro',
    119,
    2004,
    'Studio Ghibli',
    'Hayao Miyazaki',
    'Japonais',
    'Hauru',
    "Nomination à l'Oscar du meilleur film d'animation en 2006"
  ],
  [
    'Anastasia ',
    'Anastasia',
    94,
    1997,
    'Fox Animation Studios',
    'Don Bluth et Gary Goldman',
    'Américain',
    'Anastasia'
  ],
  [
    'Spirit, l\'étalon des plaines',
    'Spirit: Stallion of the Cimarron',
    84,
    2002,
    'DreamWorks Animation',
    'Kelly Asbury',
    'Américain',
    'Spirit',
```

```
    "Nomination à l'Oscar du meilleur film d'animation en 2003"
  ],
  [
    'Ratatouille',
    'Ratatouille',
    111,
    2007,
    'Brad Bird',
    'Pixar',
    'Américain',
    'Rémy',
    "Oscar du meilleur film d'animation en 2007"
  ]
]

```

Solution n°13

[exercice p. 22]

```
for (let i = 0; i < films.length; i++) {
  console.log(i + 1, films[i][0], '(' + films[i][3] + ')')
}
```

Solution n°14

[exercice p. 22]

```
/** JavaScript */
let input
let selectedFilm
let end = false

while (!end) {
  console.log('X Sortir')

  for (let i = 0; i < films.length; i++) {
    console.log(i + 1, films[i][0], '(' + films[i][3] + ')')
  }

  input = prompt('Sélection ')
  if (input === 'X') {
    end = true
  } else if (input > 0 && input <= films.length) {
    selectedFilm = films[input - 1]
    console.log('Titre:', selectedFilm[0])
    console.log('Titre original:', selectedFilm[1])
    console.log('Durée:', selectedFilm[2], 'min')
    console.log('Année de sortie:', selectedFilm[3])
    console.log('Réalisateur:', selectedFilm[4])
    console.log('Studio:', selectedFilm[5])
    console.log('Origine:', selectedFilm[6])
    console.log('Personnage principal:', selectedFilm[7])
    console.log('Distinction:', selectedFilm[8])
  }
}
```

Solution n°15

[exercice p. 23]

Un nouveau film est créé par un tableau vide puis, grâce à la fonction `push`, en lui ajoutant les caractéristiques une à une.

```
let input
let selectedFilm
let end = false

while (!end) {
  console.log('X Sortir')
  console.log('A Ajouter')

  for (let i = 0; i < films.length; i++) {
    console.log(i + 1, films[i][0], '(' + films[i][3] + ')')
  }

  input = prompt('Sélection ')
  // sortir du programme
  if (input === 'X') {
    end = true
  } else if (input === 'A') { // créer un film
    const newFilm = []
    console.log('Nouveau film:')
    newFilm.push(prompt('Titre'))
    newFilm.push(prompt('Titre original'))
    newFilm.push(prompt('Durée'))
    newFilm.push(prompt('Année de sortie'))
    newFilm.push(prompt('Réalisateur'))
    newFilm.push(prompt('Studio'))
    newFilm.push(prompt('Origine'))
    newFilm.push(prompt('Personnage principal'))
    newFilm.push(prompt('Distinction'))
    films.push(newFilm)
  } else if (input > 0 && input <= films.length) { // afficher le film
    selectedFilm = films[input - 1]
    console.log('Titre:', selectedFilm[0])
    console.log('Titre original:', selectedFilm[1])
    console.log('Durée:', selectedFilm[2], 'min')
    console.log('Année de sortie:', selectedFilm[3])
    console.log('Réalisateur:', selectedFilm[4])
    console.log('Studio:', selectedFilm[5])
    console.log('Origine:', selectedFilm[6])
    console.log('Personnage principal:', selectedFilm[7])
    console.log('Distinction:', selectedFilm[8])
  }
}
```

Crédits des ressources

Tableaux de chaînes de caractères p. 4
Universel - Transfert dans le Domaine Public

Tableau indexé p. 4
Universel - Transfert dans le Domaine Public

Tableau à 2 dimensions p. 11
Universel - Transfert dans le Domaine Public

Tableau à 3 dimensions p. 13
Universel - Transfert dans le Domaine Public