



Ressourcesinformatiques

Apprendre à développer avec JavaScript

2^e édition

Christian VIGOUROUX

Fichiers complémentaires
à télécharger



Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **RI2JASAP** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Chapitre 1
Présentation du langage JavaScript

- 1. Définition et rapide historique 17
- 2. Prérequis pour un apprentissage aisé du langage 19
- 3. Outillage nécessaire 20
- 4. Positionnement du JavaScript face à d'autres technologies de développement web (HTML, CSS, PHP...) 21

Chapitre 2
Développement à partir d'algorithmes

- 1. Présentation de la notion d'algorithme 23
- 2. Notion de variable 25
 - 2.1 Présentation des notions de variable et de type 25
 - 2.2 Types de base et opérations associées 26
 - 2.3 Intérêt des types 28
 - 2.4 Utilisation des variables dans des expressions 29
 - 2.5 Tableau récapitulatif des opérateurs 29
- 3. Manipulation des variables 31
 - 3.1 Nommage des variables 31
 - 3.2 Affectation 32
 - 3.3 Exercice n°1 : Inversion du contenu de deux variables mémoire 33
 - 3.4 Affichage des résultats 34
 - 3.5 Exercice n°2 : Surfaces de cercles 37
 - 3.6 Saisie au clavier 38

2 _____ Apprendre à développer

avec JavaScript

3.7	Exercice n°3 : Surface et volume d'une sphère	39
4.	Fonctions prédéfinies	40
4.1	Exercice n°4 : Affichage de la longueur d'un nom	40
4.2	Exercice n°5 : Détermination des initiales	41
5.	Traitements conditionnés	44
5.1	Exercice n°6 : Polynôme du second degré	46
5.2	Exercice n°7 : Libellé du mois en clair	48
5.3	Exercice n°8 : Libellé du mois en clair (Suivant ... Finsuivant) .	51
6.	Structures itératives	52
6.1	Principe des itérations	52
6.2	Structures itératives de base	52
6.3	Exercice n°9 : Moyenne de 10 nombres.	55
6.4	Exercice n°10 : Moyenne d'une série de n nombres	56
6.5	Exercice n°11 : Plus Grand Commun Diviseur par la méthode des divisions successives.	59
6.6	Structure itérative Pour	61
6.7	Exercice n°12 : Calcul de la moyenne de 10 nombres.	63
6.8	Exercice n°13 : Décompte du nombre de voyelles dans un mot.	64
7.	Tableaux à dimension unique	66
7.1	Exercice n°14 : Décompte des nombres pairs dans un tableau .	67
8.	Tableaux à dimensions multiples	69
8.1	Exercice n°15 : Mini-tableau.	69
9.	Procédures, fonctions et passage de paramètres	71
9.1	Les objectifs.	71
9.2	Les procédures.	72
9.3	Exercice n°16 : Appel d'une procédure avec passage de paramètres	72
9.4	Les fonctions.	75
9.5	Exercice n°17 : Appel d'une fonction avec passage de paramètres	76

Chapitre 3
Bases du langage JavaScript

- 1. Méthodologie d'apprentissage 79
- 2. Variables (déclaration et typage) 80
 - 2.1 Exercice n°2 : Surfaces de cercles 80
 - 2.2 Exercice n°3 : Surface et volume d'une sphère 87
 - 2.3 Exercice n°4 : Nombre de lettres d'un mot 90
 - 2.4 Exercice n°5 : Détermination des initiales 90

Chapitre 4
Conditionnement des traitements

- 1. Présentation de la syntaxe 93
- 2. Exemples 95
 - 2.1 Exercice n°6 : Polynôme du second degré 95
 - 2.2 Exercice n°8 : Impression du libellé d'un mois 96

Chapitre 5
Traitements itératifs (boucles)

- 1. Présentation de la syntaxe des boucles 99
- 2. Boucle while 100
 - 2.1 Syntaxe 100
 - 2.2 Exercice n°9 : Moyenne de 10 nombres saisis au clavier 101
 - 2.3 Exercice n°10 : Moyenne d'une série de n nombres saisis au clavier 102
- 3. Boucle do while 103
 - 3.1 Syntaxe 103
 - 3.2 Exercice n°11 : Moyenne d'une série de n nombres saisis au clavier 104

4 _____ Apprendre à développer

avec JavaScript

4. Boucle for	105
4.1 Syntaxe	105
4.2 Exercice n°12 : Moyenne d'une série de 10 nombres saisis au clavier	106
4.3 Exercice n°13 : Décompte du nombre de voyelles dans un mot	107

Chapitre 6 Tableaux

1. Tableaux à dimension unique	109
1.1 Syntaxe	109
1.2 Exercice n°14 : Décompte des nombres pairs dans un tableau	111
2. Tableaux à dimensions multiples	112
2.1 Syntaxe	112
2.2 Exercice n°15 : Mini-tableur	113

Chapitre 7 Procédures et fonctions

1. Les procédures	117
1.1 Syntaxe	118
1.2 Exercice n°16 : Appel d'une procédure avec passage de paramètres	118
2. Les fonctions	120
2.1 Syntaxe	120
2.2 Exercice n°17 : Appel d'une fonction avec passage de paramètres	121

Chapitre 8
Approche "objet" en JavaScript

- 1. Introduction 123
- 2. Programmation orientée objet au travers d'exemples 124
 - 2.1 Séquence 1 : Déclaration des objets
JavaScript en méthode "Inline" 124
 - 2.2 Séquence 2 : Création des objets JavaScript par constructeur. 125
 - 2.3 Séquence 3 : Variables privées dans une instance d'objet 127
 - 2.4 Séquence 4 : Passage de paramètre(s) à un constructeur 128
 - 2.5 Séquence 5 : Non-partage des méthodes
par les instances d'objets 129
 - 2.6 Séquence 6 : Notion de prototype. 130
 - 2.7 Séquence 7 : Surcharge d'une méthode 132
 - 2.8 Séquence 8 : Extension d'un prototype. 133
 - 2.9 Séquence 9 : Mécanisme de l'héritage 134
 - 2.10 Séquence 10 : Limite de l'héritage de la séquence n°9 136
 - 2.11 Séquence 11 : Une seconde limite à notre héritage 137

Chapitre 9
Objets de base de JavaScript

- 1. Présentation 139
- 2. Les objets de base 140
 - 2.1 Objet Array 140
 - 2.2 Objet Date. 140
 - 2.3 Objet Math 149
 - 2.4 Objet window. 152
 - 2.5 Objet navigator. 162
 - 2.6 Objet String 164

6 _____ Apprendre à développer

avec JavaScript

Chapitre 10

Saisie de données via des formulaires

1. Pilotage des contrôles de saisie via JavaScript.	171
1.1 Contrôle de saisie sur un champ texte	171
1.2 Contrôle de numéricité d'une saisie dans un champ texte . . .	178
1.3 Contrôle de caractères alphabétiques d'une saisie dans un champ texte	182
1.4 Contrôle de caractères alphabétiques et numériques d'une saisie dans un champ texte	182
1.5 Contrôle de longueur d'une saisie dans un champ texte	183
1.6 Contrôle de saisie sur une adresse e-mail	184
1.7 Contrôle d'un choix dans une liste déroulante (version simplifiée)	185
1.8 Contrôle d'un choix dans une liste déroulante (version étendue)	188
1.9 Contrôle d'un choix par bouton radio	191
1.10 Contrôle d'un choix par case à cocher	195

Chapitre 11

Modèle DOM

1. Introduction	199
1.1 Définition de DOM	199
1.2 Définition de l'arborescence	200
2. Apprentissage du modèle DOM	203
2.1 Script "Hello World!"	203
2.2 Différence entre write et writeln	204
2.3 Gestion des liens hypertextes	206
2.4 Gestion des images	207
2.5 Gestion des formulaires et de leurs balises	210
2.6 Gestion des ancres	212
2.7 Gestion de la navigation entre pages web	215
2.8 Affichage de caractéristiques générales du document	220

2.9	Gestion des boutons dans les formulaires	222
2.10	Gestion des tableaux (balise HTML table)	232

Chapitre 12 Exploration de flux XML via DOM

1.	Notion de flux XML	261
2.	Exemples	262
2.1	Exemple 1 : Affichage d'un contenu d'e-mail codé en XML . .	262
2.2	Exemple 2 : Liste des marques des voitures (fichier voitures.xml)	266
2.3	Exemple 3 : Liste des marques des voitures avec une boucle .	268
2.4	Exemple 4 : Liste des nœuds rattachés à la racine	270
2.5	Exemple 5 : Liste des champs (nœuds) de chaque voiture . . .	272
2.6	Exemple 6 : Remplacement d'une valeur de nœud	275
2.7	Exemple 7 : Accès aux attributs	276
2.8	Exemple 8 : Accès à un nœud parent	277
2.9	Exemple 9 : Parcours arrière des nœuds	278
2.10	Exemple 10 : Remplacement systématique d'une valeur d'attribut	279
2.11	Exemple 11 : Conversion XML en HTML	281
2.12	Exemple 12 : Suppression d'un nœud dans un flux XML . . .	283

Chapitre 13 Gestion des cookies en JavaScript

1.	Notion de cookie	287
2.	Écriture d'un cookie	288
3.	Lecture d'un cookie	290
4.	Suppression d'un cookie	292

8 _____ Apprendre à développer

avec JavaScript

Chapitre 14

Stockage local de données

1. Présentation générale des solutions. 295
 - 1.1 Stockage par sessionStorage. 296
 - 1.2 Stockage par localStorage. 296
2. Mise en œuvre du Web Storage au travers d'exemples 297
 - 2.1 Exemple 1 : Stockage par localStorage
de chaînes de caractères 297
 - 2.2 Exemple 2 : Stockage dans le localStorage
d'un objet JavaScript 307

Chapitre 15

Stockage distant (Ajax - PHP - MySQL - XML)

1. Présentation générale de la solution 317
2. Mise en œuvre du stockage distant au travers d'exemples 318
 - 2.1 Exemple 1 : Accès Ajax sur BDD MySQL
(liste de l'ensemble des voitures) 319
 - 2.2 Exemple 2 : Accès MySQL via Ajax. 345

Chapitre 16

Stockage distant (Ajax - PHP - MySQL - JSON)

1. Présentation générale de la solution 353
2. Mise en œuvre du stockage distant au travers d'exemples 355
 - 2.1 Exemple 1 : Présentation du système de notation JSON. 355
 - 2.2 Exemple 2 : Lecture d'un fichier JSON via XMLHttpRequest. 360
 - 2.3 Exemple 3 : Lecture d'un fichier JSON via XMLHttpRequest
et un script serveur en PHP 366
 - 2.4 Exemple 4 : Lecture d'une table MySQL
via XMLHttpRequest (serveur PHP et flux JSON) 368
 - 2.5 Exemple 5 : Recodage de l'exemple 4
avec une liste déroulante 374

Chapitre 17
Géolocalisation

- 1. Principe de la géolocalisation 379
- 2. Exemples d'applications de géolocalisation 380
 - 2.1 Exemple 1 : Affichage de la carte de l'Ouest de la France 380
 - 2.2 Exemple 2 : Affichage de la carte de l'Ouest de la France (marqueur) 389
 - 2.3 Exemple 3 : Affichage de la carte de l'Ouest de la France (marqueur et cercles de population) 393
 - 2.4 Exemple 4 : Affichage de la carte de l'Ouest de la France (informations météorologiques) 397
 - 2.5 Exemple 5 : Affichage de la carte de Rennes Centre-Sud (couche panorama) 400
 - 2.6 Exemple 6 : Affichage de la carte de Rennes (Street View) . . . 404

Chapitre 18
Dessin (HTML5 CANVAS)

- 1. Présentation de l'API HTML5 CANVAS 407
- 2. Exemples d'applications de l'élément <canvas> 408
 - 2.1 Exemple 1 : Tracé d'un simple carré 408
 - 2.2 Exemple 2 : Tracé d'une grille de TicTacToe 412
 - 2.3 Exemple 3 : Positionnement de deux marques dans la grille du TicTacToe 416
 - 2.4 Améliorations possibles sur le jeu du TicTacToe 419

Chapitre 19 Graphiques de gestion

- 1. Différentes solutions de conception de graphiques de gestion 421
- 2. Exemples d'utilisation des API Google Charts 422
 - 2.1 Exemple 1 : Tracé d'un histogramme 422
 - 2.2 Exemple 2 : Tracé d'un graphique en secteurs 428
 - 2.3 Exemple 3 : Tracé d'une carte 431
 - 2.4 Exemple 4 : Tracé d'une jauge 434
 - 2.5 Exemple 5 : Tracé d'une timeline. 437

Chapitre 20 Framework AngularJS

- 1. Présentation générale 441
- 2. Notions de base. 442
 - 2.1 Premier exemple 442
 - 2.2 Directive ng-bind 445
 - 2.3 Directive ng-init 447
 - 2.4 Évaluation d'une expression 448
 - 2.5 Exemple de synthèse 450
- 3. Manipulation de tableaux mémoire et d'objets 453
 - 3.1 Utilisation d'un tableau 453
 - 3.2 Utilisation d'un objet 455
- 4. Module et contrôleur 457
 - 4.1 Exemple utilisant un modèle et un contrôleur. 459
 - 4.2 Directive personnelle 461
 - 4.3 Méthode personnelle 463
 - 4.4 Contrôleur dans un fichier externe 465
 - 4.5 Saisie de données par zones de texte. 467
- 5. Boucles et affichage en mode tableau 470
 - 5.1 Itération sur un tableau de données 470
 - 5.2 Itération sur un tableau de données et un filtre. 472

- 5.3 Itération sur un tableau de données et un tri 475
- 5.4 Itération sur un tableau de données et un tableau HTML . . . 477
- 5.5 Filtrage d'un tableau via une zone de texte 480
- 6. Accès à un serveur de données distant (serveur PHP) 483
 - 6.1 Liste simple à partir d'une table MySQL 483
 - 6.2 Liste filtrée à partir d'une table MySQL 493
 - 6.3 Insertion d'un enregistrement dans une table MySQL 499
- 7. Contrôles de saisie dans les formulaires 505
 - 7.1 Contrôle de saisie sur une zone de texte 505
 - 7.2 Contrôle de saisie sur adresse mail 507
 - 7.3 Contrôle de saisie sur une zone de texte requise 508
 - 7.4 Liste déroulante pour choisir une marque 510
 - 7.5 Liste déroulante plus évoluée pour choisir une marque 512
 - 7.6 Liste déroulante encore plus évoluée
pour choisir une marque 514
 - 7.7 Liste déroulante couplée à une recherche MySQL 515
 - 7.8 Directives ng-click et ng-mousemove 519
 - 7.9 Directives ng-show et ng-mouseleave 523
 - 7.10 Gestion du temps (temporisation) 525
 - 7.11 Gestion de choix par boutons radio 529
 - 7.12 Gestion de choix par cases à cocher 531

Chapitre 21
EcmaScript 6

- 1. Présentation générale 537
- 2. Apports au niveau de la Programmation Orientée Objet 538
 - 2.1 Notion de prototype 538
 - 2.2 Surcharge d'une méthode 541
 - 2.3 Extension de prototype 545
 - 2.4 Héritage 548
 - 2.5 Premier exemple POO en EcmaScript 6 552
 - 2.6 Héritage en EcmaScript 555

12 _____ Apprendre à développer

avec JavaScript

2.7	Méthodes getter, setter et static en EcmaScript 6	559
3.	Fonctions fléchées (arrow functions)	563
3.1	Avantages des fonctions fléchées	563
3.2	Exemple	564
4.	Structures Map, Set et boucle for of	569
4.1	Présentation générale	569
4.2	Exemple	570
5.	Portée des variables (var ou let)	579
5.1	Présentation générale	579
5.2	Exemple	579
6.	Promesses (promise)	583
6.1	Présentation générale	583
6.2	Exemple	583
7.	Déstructuration	588
7.1	Présentation générale	588
7.2	Exemple	588

Chapitre 22 Framework Node.js

1.	Présentation générale	593
2.	Installation de Node.js	594
3.	Lancement de l'environnement Node.js	596
4.	Premier script Node.js affichant « Hello World »	598
4.1	Présentation générale	598
4.2	Code source	599
4.3	Exécution	601
5.	Deuxième script affichant « Hello World » (version avec Content-Type)	602
5.1	Présentation générale	602
5.2	Code source	602

- 5.3 Exécution 603
- 6. Troisième script affichant « Hello World » (version HTML) 604
 - 6.1 Présentation générale 604
 - 6.2 Code source 604
 - 6.3 Exécution 606
- 7. Gestion des URL 606
 - 7.1 Présentation générale 606
 - 7.2 Code source 606
 - 7.3 Exécution 608
- 8. Récupération de paramètre dans l'URL 609
 - 8.1 Présentation générale 609
 - 8.2 Code source 609
 - 8.3 Exécution 611
- 9. Détection de l'événement close sur le serveur 612
 - 9.1 Présentation générale 612
 - 9.2 Code source 612
 - 9.3 Exécution du serveur node_06.js 613
- 10. Création d'un objet avec son propre événement 614
 - 10.1 Présentation générale 614
 - 10.2 Code source 614
 - 10.3 Exécution 616
- 11. Affichage du premier enregistrement d'une table MySQL (version 1) 616
 - 11.1 Présentation générale 616
 - 11.2 Code source 617
 - 11.3 Exécution 620
- 12. Affichage de l'ensemble des enregistrements d'une table MySQL 622
 - 12.1 Présentation générale 622
 - 12.2 Code source 622
 - 12.3 Exécution 624

14 _____ Apprendre à développer

avec JavaScript

13. Affichage du premier enregistrement d'une table MySQL (version 2)	625
13.1 Présentation générale	625
13.2 Code source	625
13.3 Exécution	627

Chapitre 23 **Serveur de données Mongo DB**

1. Présentation générale	629
2. Installation de Mongo DB	631
3. Lancement du serveur de données Mongo DB	632
4. Accès client au serveur de données Mongo DB	633
5. Opérations de base sur Mongo DB	634
5.1 Connexion à une base de données Mongo DB	634
5.2 Insertion de données dans une collection	635
5.3 Liste du contenu d'une collection	636
5.4 Liste du contenu d'une collection avec filtrage des données	637
5.5 Liste des noms de collections	638
5.6 Suppression d'un objet dans une collection	638
5.7 Mise à jour d'une valeur de propriété	639
5.8 Utilitaire d'exportation de données mongoexport	639
5.9 Utilitaire d'importation de données mongoimport	641

Chapitre 24 **Accès à un serveur Mongo DB depuis Node.js**

1. Présentation générale	643
2. Accès au serveur de données Mongo DB	644
2.1 Présentation générale	644
2.2 Code source	644
2.3 Exécution	646

- 3. Insertion de documents sous Mongo DB 646
 - 3.1 Présentation générale 646
 - 3.2 Code source..... 646
 - 3.3 Exécution 649
- 4. Listage des documents d'une collection Mongo DB..... 650
 - 4.1 Présentation générale 650
 - 4.2 Code source..... 650
 - 4.3 Exécution 652
- 5. Mise à jour d'un document d'une collection Mongo DB..... 653
 - 5.1 Présentation générale 653
 - 5.2 Code source..... 653
 - 5.3 Exécution 656
- 6. Parcours pas à pas d'une collection Mongo DB 657
 - 6.1 Présentation générale 657
 - 6.2 Code source..... 657
 - 6.3 Exécution 660
- 7. Parcours pas à pas et affichage d'une collection Mongo DB 661
 - 7.1 Présentation générale 661
 - 7.2 Code source..... 661
 - 7.3 Exécution 664

Chapitre 25

Accès à un serveur Node.js depuis AngularJS

- 1. Présentation générale 665
- 2. Premier serveur Express js 666
 - 2.1 Présentation générale 666
 - 2.2 Code source..... 666
- 3. Un second serveur Node js-Express js 670
 - 3.1 Présentation générale 670
 - 3.2 Code source..... 670

16 _____ Apprendre à développer

avec JavaScript

4. Un troisième serveur Node js-Express js	675
4.1 Présentation générale	675
4.2 Code source	676
5. Insertion de données dans Mongo DB via Express js	680
5.1 Présentation générale	680
5.2 Code source	680
6. Listage de données dans Mongo DB via Express js	691
6.1 Présentation générale	691
6.2 Code source	692
7. Insertion de données dans Mongo DB via Express js (AngularJS) .	697
7.1 Présentation générale	697
7.2 Code source	698
8. Insertion de données dans Mongo DB via Express js (mongoose) .	706
8.1 Présentation générale	706
8.2 Code source	707
Index	717

Chapitre 6

Tableaux

1. Tableaux à dimension unique

Nous avons entrevu dans le chapitre Développement à partir d'algorithmes le potentiel des tableaux à dimension unique et à dimensions multiples, voyons leur prise en compte sous JavaScript.

1.1 Syntaxe

En JavaScript, un tableau à dimension unique est une variable mémoire "composite" dans laquelle il va être possible de stocker plusieurs données indépendantes, y compris de types différents, avec une indexation de chacune des valeurs par un numéro (ou indice).

L'accès à chaque donnée du tableau se fera donc par l'intermédiaire de cette valeur d'indice.

Une particularité quant à cet indice, sa valeur pour la première cellule du tableau est 0.

Le langage JavaScript fournit plusieurs façons de créer un tableau :

- la syntaxe littérale,
- la syntaxe dite "Programmation orientée objet".

Avec une syntaxe littérale, la déclaration d'un tableau de nom `tabSemaine` de sept cellules contenant les libellés des jours d'une semaine se fait comme suit :

```
var tabSemaine = ["Lundi", "Mardi", "Mercredi", "Jeudi",  
"Vendredi", "Samedi", "Dimanche"];
```

Vous noterez que la déclaration s'est accompagnée de l'initialisation de chacune des cellules du tableau `tabSemaine` (de la cellule d'indice 0 à la cellule d'indice 6).

Avec une syntaxe "Programmation orientée objet", vous auriez :

```
var tabSemaine = new Array("Lundi", "Mardi", "Mercredi", "Jeudi",  
"Vendredi", "Samedi", "Dimanche");
```

Nous aurions pu déclarer le tableau `tabSemaine` sans lui affecter des valeurs. Des affectations ultérieures peuvent être envisagées, comme par exemple pour le Lundi :

```
tabSemaine[0] = "Lundi";
```

Ce qui est vraiment particulier dans la gestion des tableaux sous JavaScript est l'extrême souplesse autorisée :

- pas de dimensionnement a priori (il est toujours possible d'étendre la taille du tableau en fonction des besoins),
- possibilité de mélanger dans un même tableau des données de types différents,
- possibilité d'utiliser des tableaux associatifs (tableaux pour lesquels les indices sont remplacés par des valeurs textuelles).

Pour accéder dans un traitement au contenu d'une valeur de tableau rangée à une position d'indice particulière, la syntaxe sera :

```
document.write("Le 4ième jour de la semaine est " + tabSemaine[3]);
```

Remarque

Il faut toujours se rappeler que la numérotation des indices débute à zéro.

Enfin, sachez que JavaScript propose une multitude de méthodes s'appliquant sur les tableaux (`Array`). Vous pourrez facilement par ces méthodes insérer, supprimer, repérer des éléments d'un tableau. Il existe même des méthodes de tri (`sort`, `reverse`) pour classer facilement les valeurs contenues dans un tableau sans avoir recours à l'écriture fastidieuse d'un algorithme de tri.

1.2 Exercice n°14 : Décompte des nombres pairs dans un tableau

Sujet

Détermination du nombre de nombres pairs dans un tableau (saisie préalable des valeurs à prévoir au clavier)

Corrigé (partiel) en JavaScript

```
/* Déclaration de variables locales */
/*
i          : Compteur de boucle
nbPairs   : Cumul du nombre de nombres pairs
tableau   : Tableau des nombres
*/
var i, nb_pairs;
var tableau = new Array;

/* Initialisations */
nbPairs = 0;
for (i=1; i<=5; i++)
{
    tableau[i] = parseInt(prompt("tableau[" + i + "] : "));
}

/* Détermination du nombre de nombres pairs dans le tableau */
for (i=1; i<=5; i++)
{
    if (tableau[i]%2 == 0)
    {
        nbPairs = nbPairs + 1;
    }
}

/* Affichage du résultat */
document.write("Le tableau contient " + nbPairs + " nombres pairs");
```

Commentaires du code JavaScript

Rien de vraiment nouveau n'est présenté dans ce script hormis le calcul du modulo. Ce calcul sert ici à déterminer la parité de chaque contenu de cellules du tableau. Il est réalisé par l'intermédiaire l'opérateur %.

Vous aurez peut-être noté que dans ce script la cellule d'indice 0 n'a pas été utilisée (la numérotation par la boucle `for` débute à 1). Ce choix rend sans doute plus compréhensible l'algorithme (il n'y a que les informaticiens qui s'accommodent de la numérotation à partir de zéro !).

2. Tableaux à dimensions multiples

Il est fréquent que l'on ait besoin de tableau à dimensions multiples pour gérer des problématiques, notamment en mathématique, en statistique...

JavaScript offre cette possibilité.

2.1 Syntaxe

Comme pour les tableaux à dimension unique, JavaScript permet de déclarer les tableaux à dimensions multiples de plusieurs façons :

- avec une syntaxe littérale,
- avec une syntaxe dite "Programmation orientée objet".

Avec une syntaxe dite "Programmation orientée objet" (encore appelée *JSON - JavaScript Object Notation*), la déclaration d'un tableau de nom `tabMatrice` de deux lignes subdivisées en quatre colonnes avec initialisation se fait comme suit :

```
/* Déclaration du tableau tabMatrice */
var tabMatrice tableau = new Array();

/* Déclaration de la première "ligne" du tableau tabMatrice */
tabMatrice[0]=new Array()

/* Initialisation des 4 "colonnes" de la première "ligne" */
tabMatrice[0][0] = "Un";
```

```
tabMatrice[0][1] = "Deux";
tabMatrice[0][2] = "Trois";
tabMatrice[0][3] = "Quatre";

/* Déclaration de la deuxième "ligne" du tableau tabMatrice */
tabMatrice[1]=new Array()

/* Initialisation des 4 "colonnes" de la deuxième "ligne" */
tabMatrice[1][0] = "Onze";
tabMatrice[1][1] = "Douze";
tabMatrice[1][2] = "Treize";
tabMatrice[1][3] = "Quatorze";
```

2.2 Exercice n°15 : Mini-tableur

Sujet

Soit le tableau `tb` à deux dimensions comportant quatre lignes et cinq colonnes. Réaliser les traitements suivants :

- saisir au clavier des valeurs dans les trois premières lignes et les quatre premières colonnes (on conserve la dernière ligne et la dernière colonne libres pour des additions de lignes et de colonnes),
- additionner les colonnes en dernière ligne et les lignes en dernière colonne.

Corrigé (partiel) en JavaScript

```
/* Déclaration de variables locales */
var tb = new Array(5);
var numLigne, numColonne;
var valeur;

/* Déclaration de 5 "colonnes" par "ligne" pour le tableau tb */
for (var numLigne=1; numLigne<tb.length; numLigne++)
{
    /* Création des "colonnes" (numérotées de 0 à 5) */
    tb[numLigne]=new Array(6);
}

/* Initialisation du tableau tb */
valeur = 1;
for (numLigne=1; numLigne<=3; numLigne++)
{
```

```
        for (numColonne=1; numColonne<=4; numColonne++)
        {
            tb[numLigne][numColonne] = valeur;
            valeur = valeur + 1;
        }
    }

    /* Mise à zéro des totaux en ligne n°4 */
    for (numColonne=1; numColonne<=5; numColonne++)
    {
        tb[4][numColonne] = 0;
    }

    /* Mise à zéro des totaux en colonne n°5 */
    for (numLigne=1; numLigne<=4; numLigne++)
    {
        tb[numLigne][5] = 0;
    }

    /* Détermination des totaux en ligne n°4 et en colonne n°5 */
    for (numLigne=1; numLigne<=3; numLigne++)
    {
        for (numColonne=1; numColonne<=4; numColonne++)
        {
            /* Totalisation en ligne n°4 */
            tb[4][numColonne] = tb[4][numColonne]
            + tb[numLigne][numColonne];
            /* Totalisation en colonne n°5 */
            tb[numLigne][5] = tb[numLigne][5]
            + tb[numLigne][numColonne];
            /* Totalisation générale en ligne n°4-colonne n°5 */
            tb[4][5] = tb[4][5] + tb[numLigne][numColonne];
        }
    }

    /* Affichage du total général */
    /* NB : Total de 78 étant donné la technique de remplissage retenue
    du tableau tb */
    document.write("Total général en tb[4][5] = " + tb[4][5]);
```