



Ressourcesinformatiques

Apprendre à développer avec JavaScript

3^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 **RI3JASAP** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Partie 1
Présentation de JavaScript
et programmation à partir d'algorithmes

Chapitre 1
Présentation du langage JavaScript

- 1. Définition et rapide historique 19
- 2. Prérequis pour un apprentissage aisé du langage 21
- 3. Outillage nécessaire 22
- 4. Positionnement du JavaScript face à d'autres technologies de développement web (HTML, CSS, PHP...) 23

Chapitre 2
Développement à partir d'algorithmes

- 1. Présentation de la notion d'algorithme 25
- 2. Notion de variable 27
 - 2.1 Présentation des notions de variable et de type 27
 - 2.2 Types de base et opérations associées 28
 - 2.3 Intérêt des types 30
 - 2.4 Utilisation des variables dans des expressions 31
 - 2.5 Tableau récapitulatif des opérateurs 31
- 3. Manipulation des variables 33
 - 3.1 Nommage des variables 33
 - 3.2 Affectation 34

2 _____ Apprendre à développer

avec JavaScript

3.3	Exercice n°1 : Inversion du contenu de deux variables mémoire	35
3.4	Affichage des résultats	36
3.5	Exercice n°2 : Surfaces de cercles	39
3.6	Saisie au clavier	40
3.7	Exercice n°3 : Surface et volume d'une sphère	41
4.	Fonctions prédéfinies	42
4.1	Exercice n°4 : Affichage de la longueur d'un nom	42
4.2	Exercice n°5 : Détermination des initiales	43
5.	Traitements conditionnés	46
5.1	Exercice n°6 : Polynôme du second degré	48
5.2	Exercice n°7 : Libellé du mois en clair	50
5.3	Exercice n°8 : Libellé du mois en clair (Suivant ... Finsuivant)	53
6.	Structures itératives	54
6.1	Principe des itérations	54
6.2	Structures itératives de base	54
6.3	Exercice n°9 : Moyenne de 10 nombres	57
6.4	Exercice n°10 : Moyenne d'une série de n nombres	58
6.5	Exercice n°11 : Plus Grand Commun Diviseur par la méthode des divisions successives	61
6.6	Structure itérative Pour	63
6.7	Exercice n°12 : Calcul de la moyenne de 10 nombres	65
6.8	Exercice n°13 : Décompte du nombre de voyelles dans un mot	66
7.	Tableaux à dimension unique	68
7.1	Exercice n°14 : Décompte des nombres pairs dans un tableau	69
8.	Tableaux à dimensions multiples	71
8.1	Exercice n°15 : Mini-tableau	71

9. Procédures, fonctions et passage de paramètres	73
9.1 Les objectifs.	73
9.2 Les procédures.	74
9.3 Exercice n°16 : Appel d'une procédure avec passage de paramètres	74
9.4 Les fonctions	77
9.5 Exercice n°17 : Appel d'une fonction avec passage de paramètres	78

Partie 2

Bien débiter avec JavaScript

Chapitre 3

Bases du langage JavaScript

1. Méthodologie d'apprentissage	81
2. Variables (déclaration et typage)	82
2.1 Exercice n°2 : Surfaces de cercles	82
2.2 Exercice n°3 : Surface et volume d'une sphère	89
2.3 Exercice n°4 : Nombre de lettres d'un mot	92
2.4 Exercice n°5 : Détermination des initiales	92

Chapitre 4

Conditionnement des traitements

1. Présentation de la syntaxe	95
2. Exemples	97
2.1 Exercice n°6 : Polynôme du second degré	97
2.2 Exercice n°8 : Impression du libellé d'un mois	98

4 _____ Apprendre à développer

avec JavaScript

Chapitre 5 Traitements itératifs (boucles)

1. Présentation de la syntaxe des boucles	101
2. Boucle while	102
2.1 Syntaxe	102
2.2 Exercice n°9 : Moyenne de 10 nombres saisis au clavier	103
2.3 Exercice n°10 : Moyenne d'une série de n nombres saisis au clavier	104
3. Boucle do while	105
3.1 Syntaxe	105
3.2 Exercice n°11 : Moyenne d'une série de n nombres saisis au clavier	106
4. Boucle for	107
4.1 Syntaxe	107
4.2 Exercice n°12 : Moyenne d'une série de 10 nombres saisis au clavier	108
4.3 Exercice n°13 : Décompte du nombre de voyelles dans un mot	109

Chapitre 6 Tableaux

1. Tableaux à dimension unique	111
1.1 Syntaxe	111
1.2 Exercice n°14 : Décompte des nombres pairs dans un tableau	113
2. Tableaux à dimensions multiples	114
2.1 Syntaxe	114
2.2 Exercice n°15 : Mini-tableau	115

Chapitre 7
Procédures et fonctions

- 1. Les procédures..... 119
 - 1.1 Syntaxe 120
 - 1.2 Exercice n°16 : Appel d'une procédure
avec passage de paramètres 120
- 2. Les fonctions..... 122
 - 2.1 Syntaxe 122
 - 2.2 Exercice n°17 : Appel d'une fonction
avec passage de paramètres 123

Partie 3
Approche POO sous JavaScript

Chapitre 8
Approche "objet" en JavaScript

- 1. Introduction 125
- 2. Programmation orientée objet au travers d'exemples 126
 - 2.1 Séquence 1 : Déclaration des objets
JavaScript en méthode "Inline" 126
 - 2.2 Séquence 2 : Création des objets JavaScript par constructeur. 127
 - 2.3 Séquence 3 : Variables privées dans une instance d'objet 129
 - 2.4 Séquence 4 : Passage de paramètre(s) à un constructeur 130
 - 2.5 Séquence 5 : Non-partage des méthodes
par les instances d'objets 131
 - 2.6 Séquence 6 : Notion de prototype..... 132
 - 2.7 Séquence 7 : Surcharge d'une méthode 134
 - 2.8 Séquence 8 : Extension d'un prototype..... 135
 - 2.9 Séquence 9 : Mécanisme de l'héritage 136
 - 2.10 Séquence 10 : Limite de l'héritage de la séquence n°9 138
 - 2.11 Séquence 11 : Une seconde limite à notre héritage 139

6 _____ Apprendre à développer

avec JavaScript

Chapitre 9 Objets de base de JavaScript

1. Présentation	141
2. Les objets de base	142
2.1 Objet Array	142
2.2 Objet Date	142
2.3 Objet Math	151
2.4 Objet window	154
2.5 Objet navigator	164
2.6 Objet String	166

Chapitre 10 Les nouveautés d'EcmaScript 6

1. Présentation générale	173
2. Apports au niveau de la Programmation Orientée Objet	174
2.1 Notion de prototype	174
2.2 Surcharge d'une méthode	177
2.3 Extension de prototype	180
2.4 Héritage	183
2.5 Premier exemple POO en EcmaScript 6	188
2.6 Héritage en EcmaScript	191
2.7 Méthodes getter, setter et static en EcmaScript 6	194
3. Fonctions fléchées (arrow functions)	199
3.1 Avantages des fonctions fléchées	199
3.2 Exemple	200
4. Structures Map, Set et boucle for of	205
4.1 Présentation générale	205
4.2 Exemple	206
5. Portée des variables (var ou let)	215
5.1 Présentation générale	215
5.2 Exemple	215

- 6. Promesses (promise) 219
 - 6.1 Présentation générale 219
 - 6.2 Exemple 219
- 7. Déstructuration 224
 - 7.1 Présentation générale 224
 - 7.2 Exemple 224

Partie 4

Gestion de formulaire et modèle DOM

Chapitre 11

Saisie de données via des formulaires

- 1. Pilotage des contrôles de saisie via JavaScript 229
 - 1.1 Contrôle de saisie sur un champ texte 229
 - 1.2 Contrôle de numéricité d'une saisie dans un champ texte ... 236
 - 1.3 Contrôle de caractères alphabétiques d'une saisie dans un champ texte 240
 - 1.4 Contrôle de caractères alphabétiques et numériques d'une saisie dans un champ texte 240
 - 1.5 Contrôle de longueur d'une saisie dans un champ texte 241
 - 1.6 Contrôle de saisie sur une adresse e-mail 242
 - 1.7 Contrôle d'un choix dans une liste déroulante (version simplifiée) 243
 - 1.8 Contrôle d'un choix dans une liste déroulante (version étendue) 246
 - 1.9 Contrôle d'un choix par bouton radio 249
 - 1.10 Contrôle d'un choix par case à cocher 253

8 _____ Apprendre à développer

avec JavaScript

Chapitre 12 Modèle DOM

1. Introduction	257
1.1 Définition de DOM	257
1.2 Définition de l'arborescence	258
2. Apprentissage du modèle DOM	261
2.1 Script "Hello World!"	261
2.2 Différence entre write et writeln	262
2.3 Gestion des liens hypertextes	264
2.4 Gestion des images	265
2.5 Gestion des formulaires et de leurs balises	268
2.6 Gestion des ancres	270
2.7 Gestion de la navigation entre pages web	273
2.8 Affichage de caractéristiques générales du document	278
2.9 Gestion des boutons dans les formulaires	280
2.10 Gestion des tableaux (balise HTML table)	290

Chapitre 13 Exploration de flux XML via DOM

1. Notion de flux XML	319
2. Exemples	320
2.1 Exemple 1 : Affichage d'un contenu d'e-mail codé en XML	320
2.2 Exemple 2 : Liste des marques des voitures (fichier voitures.xml)	324
2.3 Exemple 3 : Liste des marques des voitures avec une boucle	326
2.4 Exemple 4 : Liste des nœuds rattachés à la racine	328
2.5 Exemple 5 : Liste des champs (nœuds) de chaque voiture	330
2.6 Exemple 6 : Remplacement d'une valeur de nœud	333
2.7 Exemple 7 : Accès aux attributs	334
2.8 Exemple 8 : Accès à un nœud parent	335
2.9 Exemple 9 : Parcours arrière des nœuds	336

- 2.10 Exemple 10 : Remplacement systématique d'une valeur d'attribut 337
- 2.11 Exemple 11 : Conversion XML en HTML 339
- 2.12 Exemple 12 : Suppression d'un nœud dans un flux XML 341

Partie 5
Cookie et mécanismes de persistance

Chapitre 14
Gestion des cookies en JavaScript

- 1. Notion de cookie. 345
- 2. Écriture d'un cookie 346
- 3. Lecture d'un cookie. 348
- 4. Suppression d'un cookie 350

Chapitre 15
Stockage local de données

- 1. Présentation générale des solutions. 353
 - 1.1 Stockage par sessionStorage. 354
 - 1.2 Stockage par localStorage. 354
- 2. Mise en œuvre du Web Storage au travers d'exemples 355
 - 2.1 Exemple 1 : Stockage par localStorage de chaînes de caractères 355
 - 2.2 Exemple 2 : Stockage dans le localStorage d'un objet JavaScript. 365

Chapitre 16

Stockage distant (Ajax - PHP - MySQL - XML)

1. Présentation générale de la solution375
2. Mise en œuvre du stockage distant au travers d'exemples376
 - 2.1 Exemple 1 : Accès Ajax sur BDD MySQL
(liste de l'ensemble des voitures)377
 - 2.2 Exemple 2 : Accès MySQL via Ajax.403

Chapitre 17

Stockage distant (Ajax - PHP - MySQL - JSON)

1. Présentation générale de la solution411
2. Mise en œuvre du stockage distant au travers d'exemples413
 - 2.1 Exemple 1 : Présentation du système de notation JSON.413
 - 2.2 Exemple 2 : Lecture d'un fichier JSON via XMLHttpRequest.418
 - 2.3 Exemple 3 : Lecture d'un fichier JSON
via XMLHttpRequest et un script serveur en PHP424
 - 2.4 Exemple 4 : Lecture d'une table MySQL
via XMLHttpRequest (serveur PHP et flux JSON)426
 - 2.5 Exemple 5 : Recodage de l'exemple 4
avec une liste déroulante432

Partie 6
Géolocalisation, dessin et graphiques de gestion

Chapitre 18
Géolocalisation

- 1. Principe de la géolocalisation 437
- 2. Exemples d'applications de géolocalisation 438
 - 2.1 Exemple 1 : Affichage de la carte de l'Ouest de la France 438
 - 2.2 Exemple 2 : Affichage de la carte de l'Ouest de la France (informations météorologiques) 455
 - 2.3 Exemple 3 : Affichage de la carte de Rennes Centre-Sud (couche panorama) 458
 - 2.4 Exemple 4 : Affichage de la carte de Rennes (Street View) . . . 462

Chapitre 19
Dessin (HTML5 CANVAS)

- 1. Présentation de l'API HTML5 CANVAS 465
- 2. Exemples d'applications de l'élément <canvas> 466
 - 2.1 Exemple 1 : Tracé d'un simple carré 466
 - 2.2 Exemple 2 : Tracé d'une grille de TicTacToe 470
 - 2.3 Améliorations possibles sur le jeu du TicTacToe 477

Chapitre 20
Graphiques de gestion

- 1. Différentes solutions de conception de graphiques de gestion 479
- 2. Exemples d'utilisation des API Google Charts 480
 - 2.1 Exemple 1 : Tracé d'un histogramme 480
 - 2.2 Exemple 2 : Tracé d'un graphique en secteurs 486
 - 2.3 Exemple 3 : Tracé d'une carte 489

Partie 7 Frameworks JavaScript et serveur/client

Chapitre 21 Framework AngularJS

1. Présentation générale	493
2. Notions de base.	494
2.1 Premier exemple	494
2.2 Directive ng-bind	497
2.3 Directive ng-init	499
2.4 Évaluation d'une expression	500
2.5 Exemple de synthèse	502
3. Manipulation de tableaux mémoire et d'objets	505
3.1 Utilisation d'un tableau	505
3.2 Utilisation d'un objet	507
4. Module et contrôleur	509
4.1 Exemple utilisant un modèle et un contrôleur.	511
4.2 Directive personnelle	513
4.3 Méthode personnelle	515
4.4 Contrôleur dans un fichier externe	517
4.5 Saisie de données par zones de texte.	519
5. Boucles et affichage en mode tableau	522
5.1 Itération sur un tableau de données	522
5.2 Itération sur un tableau de données et un filtre.	524
5.3 Itération sur un tableau de données et un tri.	527
5.4 Itération sur un tableau de données et un tableau HTML	529
5.5 Filtrage d'un tableau via une zone de texte	532
6. Accès à un serveur de données distant (serveur PHP)	535
6.1 Liste simple à partir d'une table MySQL	535
6.2 Liste filtrée à partir d'une table MySQL	545
6.3 Insertion d'un enregistrement dans une table MySQL	551

- 7. Contrôles de saisie dans les formulaires 557
 - 7.1 Contrôle de saisie sur une zone de texte 557
 - 7.2 Contrôle de saisie sur adresse mail 559
 - 7.3 Contrôle de saisie sur une zone de texte requise 560
 - 7.4 Liste déroulante pour choisir une marque 562
 - 7.5 Liste déroulante plus évoluée pour choisir une marque 564
 - 7.6 Liste déroulante encore plus évoluée
pour choisir une marque 566
 - 7.7 Liste déroulante couplée à une recherche MySQL 567
 - 7.8 Directives ng-click et ng-mousemove 571
 - 7.9 Directives ng-show et ng-mouseleave 575
 - 7.10 Gestion du temps (temporisation) 577
 - 7.11 Gestion de choix par boutons radio 581
 - 7.12 Gestion de choix par cases à cocher 583

Chapitre 22
Framework Node.js

- 1. Présentation générale 589
- 2. Installation de Node.js 590
- 3. Lancement de l’environnement Node.js 592
- 4. Premier script Node.js affichant « Hello World » 594
 - 4.1 Présentation générale 594
 - 4.2 Code source 595
 - 4.3 Exécution 597
- 5. Deuxième script affichant « Hello World »
(version avec Content-Type) 598
 - 5.1 Présentation générale 598
 - 5.2 Code source 598
 - 5.3 Exécution 599
- 6. Troisième script affichant « Hello World » (version HTML) 600
 - 6.1 Présentation générale 600

14 _____ Apprendre à développer

avec JavaScript

6.2	Code source	600
6.3	Exécution	602
7.	Gestion des URL	602
7.1	Présentation générale	602
7.2	Code source	602
7.3	Exécution	604
8.	Récupération de paramètre dans l'URL	605
8.1	Présentation générale	605
8.2	Code source	605
8.3	Exécution	607
9.	Détection de l'événement close sur le serveur	608
9.1	Présentation générale	608
9.2	Code source	608
9.3	Exécution du serveur node_06.js	609
10.	Création d'un objet avec son propre événement	610
10.1	Présentation générale	610
10.2	Code source	610
10.3	Exécution	612
11.	Affichage du premier enregistrement d'une table MySQL (version 1)	612
11.1	Présentation générale	612
11.2	Code source	613
11.3	Exécution	616
12.	Affichage de l'ensemble des enregistrements d'une table MySQL	618
12.1	Présentation générale	618
12.2	Code source	618
12.3	Exécution	620
13.	Affichage du premier enregistrement d'une table MySQL (version 2)	621
13.1	Présentation générale	621
13.2	Code source	621
13.3	Exécution	623

Partie 8
Développement hybride avec Ionic 3

Chapitre 23
Développement hybride en JavaScript

- 1. Approches de développement pour les applications mobiles. 625
 - 1.1 Développements web, natif et hybride 626
 - 1.1.1 Applications web 626
 - 1.1.2 Applications natives 627
 - 1.1.3 Applications hybrides 628
 - 1.2 Les trois principales plateformes 629
 - 1.2.1 Apple iOS 629
 - 1.2.2 Android 629
 - 1.2.3 Windows Phone, Windows 10 Mobile 630
- 2. Panorama des principales plateformes « hybrides » 630
 - 2.1 Ionic 630
 - 2.2 React Native 631
 - 2.3 Autres solutions 631

Chapitre 24
Ionic 3, JavaScript en mode hybride

- 1. Présentation détaillée de l’environnement Ionic 633
- 2. Installation de l’environnement Ionic 3 634
 - 2.1 L’installation de Node.js 635
 - 2.2 L’installation des frameworks Ionic et Cordova 636
 - 2.3 Une première application de test 636
 - 2.4 Un IDE pour modifier les scripts,
Microsoft Visual Studio Code 639
 - 2.5 Le code du projet ionic3-blank en détail 640

Chapitre 25 Gestion du Hardware sous Ionic 3

1. Introduction	659
2. Application ioni3-infosHardware	659
2.1 Compte rendu d'exécution de l'application	659
2.2 Analyse des scripts principaux.	660
2.3 Déploiement de l'application sous Android	668
2.3.1 Téléchargement d'Android Studio	670
2.3.2 Préparation de l'application Ionic pour transfert sous Android Studio	670
2.3.3 Importation de l'application Ionic sous Android Studio	672
2.3.4 Test de l'application Ionic sous Android Studio en mode AVD	675
2.3.5 Test de l'application Ionic sur le périphérique « physique »	678
3. Application ionic3-battery_status	680
3.1 Compte rendu d'exécution de l'application	680
3.2 Analyse des scripts principaux.	681
4. Application ionic3-screen_orientation	686
4.1 Compte rendu d'exécution de l'application	686
4.2 Analyse des scripts principaux.	687

Chapitre 26 Composants de base sous Ionic 3

1. Introduction	695
2. Application ionic3-passage_parametres	696
2.1 Compte rendu d'exécution de l'application	696
2.2 Analyse des scripts principaux.	698

- 3. Applications intégrant des fenêtres modales 706
 - 3.1 Application ionic3-alert_basic 708
 - 3.2 Application ionic3-alert_confirm 712
 - 3.3 Application ionic3-alert_checkbox 716
 - 3.4 Application ionic3-alert_prompt 723
 - 3.5 Application ionic3-alert_radio 727

Chapitre 27

Applications basiques sous Ionic 3

- 1. Introduction 733
- 2. Application ionic3-racines_polynome 733
 - 2.1 Compte rendu d'exécution de l'application 733
 - 2.2 Analyse des scripts principaux 735
- 3. Application ionic3-pgcd 743
 - 3.1 Compte rendu d'exécution de l'application 743
 - 3.2 Analyse des scripts principaux 745

Chapitre 28

Gestion de la persistance sous Ionic 3

- 1. Introduction 747
- 2. Application ionic3-clipboard 748
 - 2.1 Compte rendu d'exécution de l'application 748
 - 2.2 Analyse des scripts principaux 749
- 3. Application ionic3-local_storage 753
 - 3.1 Compte rendu d'exécution de l'application 753
 - 3.2 Analyse des scripts principaux 754
- 4. Application ionic3-sqlite 758
 - 4.1 Compte rendu d'exécution de l'application 758
 - 4.2 Analyse des scripts principaux 761

Chapitre 29

Gestion des listes sous Ionic 3

1. Introduction	769
2. Application ionic3-list_basic	770
2.1 Compte rendu d'exécution de l'application	770
2.2 Analyse des scripts principaux.	771
3. Application ionic3-list_avatar	773
3.1 Compte rendu d'exécution de l'application	773
3.2 Analyse des scripts principaux.	774
4. Application ionic3-list_slider	775
4.1 Compte rendu d'exécution de l'application	775
4.2 Analyse des scripts principaux.	778

Chapitre 30

Graphiques de gestion sous Ionic 3

1. Introduction	787
2. Application ionic3-chartjs_bar.	788
2.1 Compte rendu d'exécution de l'application	788
2.2 Analyse des scripts principaux.	789
3. Application ionic3-chartjs_pie.	794
3.1 Compte rendu d'exécution de l'application	794
3.2 Analyse des scripts principaux.	796
4. Application ionic3-chartjs_doughnut_json	800
4.1 Compte rendu d'exécution de l'application	800
4.2 Analyse des scripts principaux.	801

Index	811
-----------------	-----



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 (3 lignes * 4 colonnes) par une saisie clavier */
for (numLigne = 1; numLigne <= 3; numLigne++) {
    for (numColonne = 1; numColonne <= 4; numColonne++) {
        valeur = parseInt(prompt("tableau[" + numLigne + "][" + numColonne + "] = "));
        tb[numLigne][numColonne] = valeur;
    }
}

/* 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]);

```

Commentaires du code JavaScript

Le tableau `tb` est dans un premier temps déclaré comme étant un tableau à une seule dimension avec cinq cellules, implicitement numérotées de 0 à 4 comme suit :

```
var tb = new Array(5);
```

Vous noterez que la ligne de numéro zéro ne sera pas utilisée par la suite. Ce choix a été fait car l'utilisation ultérieure de cette ligne ne serait pas très intuitive.

Dans un second temps les lignes de ce tableau sont elles-mêmes subdivisées en colonnes, comme ceci :

```

/* 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);
}

```