

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **RI4JASAP** 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 1-1

Présentation du langage JavaScript

- 1. Définition et rapide historique 15
- 2. Prérequis pour un apprentissage aisé du langage 17
- 3. Outillage nécessaire 18
- 4. Positionnement du JavaScript face à d'autres technologies de développement web (HTML, CSS, PHP...) 19

Chapitre 1-2

Développement à partir d'algorithmes

- 1. Présentation de la notion d'algorithme 21
- 2. Notion de variable 23
 - 2.1 Présentation des notions de variable et de type 23
 - 2.2 Types de base et opérations associées 24
 - 2.3 Intérêt des types 26
 - 2.4 Utilisation des variables dans des expressions 27
 - 2.5 Tableau récapitulatif des opérateurs 27
- 3. Manipulation des variables 29
 - 3.1 Nommage des variables 29
 - 3.2 Affectation 30
 - 3.3 Exercice n°1 : Inversion du contenu de deux variables mémoire 31
 - 3.4 Affichage des résultats 32

2 _____ Apprendre à développer

avec JavaScript

3.5	Exercice n°2 : Surfaces de cercles	35
3.6	Saisie au clavier	36
3.7	Exercice n°3 : Surface et volume d'une sphère	37
4.	Fonctions prédéfinies	38
4.1	Exercice n°4 : Affichage de la longueur d'un nom	38
4.2	Exercice n°5 : Détermination des initiales	39
5.	Traitements conditionnés	42
5.1	Exercice n°6 : Polynôme du second degré	44
5.2	Exercice n°7 : Libellé du mois en clair	46
5.3	Exercice n°8 : Libellé du mois en clair (Suivant ... Finsuivant)	49
6.	Structures itératives	50
6.1	Principe des itérations	50
6.2	Structures itératives de base	50
6.3	Exercice n°9 : Moyenne de 10 nombres	53
6.4	Exercice n°10 : Moyenne d'une série de n nombres	54
6.5	Exercice n°11 : Plus Grand Commun Diviseur par la méthode des divisions successives	57
6.6	Structure itérative Pour	59
6.7	Exercice n°12 : Calcul de la moyenne de 10 nombres	61
6.8	Exercice n°13 : Décompte du nombre de voyelles dans un mot	62
7.	Tableaux à dimension unique	64
7.1	Exercice n°14 : Décompte des nombres pairs dans un tableau	65
8.	Tableaux à dimensions multiples	67
8.1	Exercice n°15 : Mini-tableau	67
9.	Procédures, fonctions et passage de paramètres	69
9.1	Les objectifs	69
9.2	Les procédures	70
9.3	Exercice n°16 : Appel d'une procédure avec passage de paramètres	70
9.4	Les fonctions	73
9.5	Exercice n°17 : Appel d'une fonction avec passage de paramètres	74

Partie 2 : Bien débiter avec JavaScript

Chapitre 2-1

Bases du langage JavaScript

1. Méthodologie d'apprentissage	77
2. Variables (déclaration et typage)	78
2.1 Exercice n°2 : Surfaces de cercles	78
2.2 Exercice n°3 : Surface et volume d'une sphère	85
2.3 Exercice n°4 : Nombre de lettres d'un mot	88
2.4 Exercice n°5 : Détermination des initiales	88

Chapitre 2-2

Conditionnement des traitements

1. Présentation de la syntaxe	91
2. Exemples	93
2.1 Exercice n°6 : Polynôme du second degré	93
2.2 Exercice n°8 : Impression du libellé d'un mois	94

Chapitre 2-3

Traitements itératifs (boucles)

1. Présentation de la syntaxe des boucles	97
2. Boucle while	98
2.1 Syntaxe	98
2.2 Exercice n°9 : Moyenne de 10 nombres saisis au clavier	99
2.3 Exercice n°10 : Moyenne d'une série de n nombres saisis au clavier	100
3. Boucle do while	101
3.1 Syntaxe	101
3.2 Exercice n°11 : Moyenne d'une série de n nombres saisis au clavier	102

4 _____ Apprendre à développer

avec JavaScript

4. Boucle for	103
4.1 Syntaxe	103
4.2 Exercice n°12 : Moyenne d'une série de 10 nombres saisis au clavier	104
4.3 Exercice n°13 : Décompte du nombre de voyelles dans un mot	105

Chapitre 2-4 Tableaux

1. Tableaux à dimension unique	107
1.1 Syntaxe	107
1.2 Exercice n°14 : Décompte des nombres pairs dans un tableau	109
2. Tableaux à dimensions multiples	110
2.1 Syntaxe	110
2.2 Exercice n°15 : Mini-tableur	111

Chapitre 2-5 Procédures et fonctions

1. Les procédures	115
1.1 Syntaxe	116
1.2 Exercice n°16 : Appel d'une procédure avec passage de paramètres	116
2. Les fonctions	118
2.1 Syntaxe	118
2.2 Exercice n°17 : Appel d'une fonction avec passage de paramètres	119

Partie 3 : Approche POO sous JavaScript

Chapitre 3-1

Approche "objet" en JavaScript

- 1. Introduction 121
- 2. Programmation orientée objet au travers d'exemples 122
 - 2.1 Séquence 1 : Déclaration des objets
JavaScript en méthode "Inline" 122
 - 2.2 Séquence 2 : Création des objets JavaScript par constructeur. 123
 - 2.3 Séquence 3 : Variables privées dans une instance d'objet 125
 - 2.4 Séquence 4 : Passage de paramètre(s) à un constructeur 126
 - 2.5 Séquence 5 : Non-partage des méthodes
par les instances d'objets 127
 - 2.6 Séquence 6 : Notion de prototype. 128
 - 2.7 Séquence 7 : Surchage d'une méthode 130
 - 2.8 Séquence 8 : Extension d'un prototype..... 131
 - 2.9 Séquence 9 : Mécanisme de l'héritage 132
 - 2.10 Séquence 10 : Limite de l'héritage de la séquence n°9 134
 - 2.11 Séquence 11 : Une seconde limite à notre héritage 135

Chapitre 3-2

Objets de base de JavaScript

- 1. Présentation 137
- 2. Les objets de base 138
 - 2.1 Objet Array 138
 - 2.2 Objet Date. 138
 - 2.3 Objet Math 147
 - 2.4 Objet window. 150
 - 2.5 Objet navigator. 160
 - 2.6 Objet String 162

6 _____ Apprendre à développer

avec JavaScript

Chapitre 3-3 Les nouveautés d'EcmaScript 6

1. Présentation générale	169
2. Apports au niveau de la Programmation Orientée Objet	170
2.1 Notion de prototype	170
2.2 Surcharge d'une méthode	173
2.3 Extension de prototype	176
2.4 Héritage	179
2.5 Premier exemple POO en EcmaScript 6	184
2.6 Héritage en EcmaScript	187
2.7 Méthodes getter, setter et static en EcmaScript 6	190
3. Fonctions fléchées (arrow functions)	195
3.1 Avantages des fonctions fléchées	195
3.2 Exemple	196
4. Structures Map, Set et boucle for of	201
4.1 Présentation générale	201
4.2 Exemple	202
5. Portée des variables (var ou let)	211
5.1 Présentation générale	211
5.2 Exemple	211
6. Promesses (promise)	215
6.1 Présentation générale	215
6.2 Exemple	215
7. Déstructuration	220
7.1 Présentation générale	220
7.2 Exemple	220

Partie 4 : Gestion de formulaire et modèle DOM

Chapitre 4-1

Saisie de données via des formulaires

- 1. Pilotage des contrôles de saisie via JavaScript 225
 - 1.1 Contrôle de saisie sur un champ texte 225
 - 1.2 Contrôle de numéricité d'une saisie dans un champ texte . . . 232
 - 1.3 Contrôle de caractères alphabétiques d'une saisie dans un champ texte 236
 - 1.4 Contrôle de caractères alphabétiques et numériques d'une saisie dans un champ texte 236
 - 1.5 Contrôle de longueur d'une saisie dans un champ texte 237
 - 1.6 Contrôle de saisie sur une adresse e-mail 238
 - 1.7 Contrôle d'un choix dans une liste déroulante (version simplifiée) 239
 - 1.8 Contrôle d'un choix dans une liste déroulante (version étendue) 242
 - 1.9 Contrôle d'un choix par bouton radio 245
 - 1.10 Contrôle d'un choix par case à cocher 249

Chapitre 4-2

Modèle DOM

- 1. Introduction 253
 - 1.1 Définition de DOM 253
 - 1.2 Définition de l'arborescence 254
- 2. Apprentissage du modèle DOM 257
 - 2.1 Script "Hello World!" 257
 - 2.2 Différence entre write et writeln 258
 - 2.3 Gestion des liens hypertextes 260
 - 2.4 Gestion des images 261
 - 2.5 Gestion des formulaires et de leurs balises 264
 - 2.6 Gestion des ancrs 266

8 _____ Apprendre à développer

avec JavaScript

2.7	Gestion de la navigation entre pages web.	269
2.8	Affichage de caractéristiques générales du document	274
2.9	Gestion des boutons dans les formulaires.	276
2.10	Gestion des tableaux (balise HTML table)	286

Chapitre 4-3 Exploration de flux XML via DOM

1.	Notion de flux XML	315
2.	Exemples	316
2.1	Exemple 1 : Affichage d'un contenu d'e-mail codé en XML . .	316
2.2	Exemple 2 : Liste des marques des voitures (fichier voitures.xml)	320
2.3	Exemple 3 : Liste des marques des voitures avec une boucle .	322
2.4	Exemple 4 : Liste des nœuds rattachés à la racine	324
2.5	Exemple 5 : Liste des champs (nœuds) de chaque voiture. . .	326
2.6	Exemple 6 : Remplacement d'une valeur de nœud.	329
2.7	Exemple 7 : Accès aux attributs.	330
2.8	Exemple 8 : Accès à un nœud parent	331
2.9	Exemple 9 : Parcours arrière des nœuds	332
2.10	Exemple 10 : Remplacement systématique d'une valeur d'attribut	333
2.11	Exemple 11 : Conversion XML en HTML	335
2.12	Exemple 12 : Suppression d'un nœud dans un flux XML . . .	337

Partie 5 : Cookies et mécanismes de persistance

Chapitre 5-1 Gestion des cookies en JavaScript

1.	Notion de cookie.	341
2.	Écriture d'un cookie	342
3.	Lecture d'un cookie.	344

4. Suppression d'un cookie 346

Chapitre 5-2
Stockage local de données

1. Présentation générale des solutions. 349
1.1 Stockage par sessionStorage. 350
1.2 Stockage par localStorage. 350
2. Mise en œuvre du Web Storage au travers d'exemples 351
2.1 Exemple 1 : Stockage par localStorage
de chaînes de caractères 351
2.2 Exemple 2 : Stockage dans le localStorage
d'un objet JavaScript. 361

Chapitre 5-3
Stockage distant (Ajax - PHP - MySQL - XML)

1. Présentation générale de la solution 371
2. Mise en œuvre du stockage distant au travers d'exemples 372
2.1 Exemple 1 : Accès Ajax sur BDD MySQL
(liste de l'ensemble des voitures) 373
2.2 Exemple 2 : Accès MySQL via Ajax. 399

Chapitre 5-4
Stockage distant (Ajax - PHP - MySQL - JSON)

1. Présentation générale de la solution 407
2. Mise en œuvre du stockage distant au travers d'exemples 409
2.1 Exemple 1 : Présentation du système de notation JSON. 409
2.2 Exemple 2 : Lecture d'un fichier JSON
via XMLHttpRequest. 414
2.3 Exemple 3 : Lecture d'un fichier JSON
via XMLHttpRequest et un script serveur en PHP 420

10 _____ Apprendre à développer

avec JavaScript

2.4 Exemple 4 : Lecture d'une table MySQL via XMLHttpRequest (serveur PHP et flux JSON)	422
2.5 Exemple 5 : Recodage de l'exemple 4 avec une liste déroulante	428

Partie 6 : Géolocalisation, dessin et graphiques de gestion

Chapitre 6-1

Géolocalisation

1. Principe de la géolocalisation	433
2. Exemples d'applications de géolocalisation.	434
2.1 Exemple 1 : Affichage de la carte de l'Ouest de la France	434
2.2 Exemple 2 : Affichage de la carte de l'Ouest de la France (marqueur)	441
2.3 Exemple 3 : Affichage de la carte de l'Ouest de la France (marqueur et cercles de population)	444
2.4 Exemple 4 : Affichage de la carte de l'Ouest de la France (informations météorologiques)	447
2.5 Exemple 5 : Affichage de la carte de Rennes Centre-Sud (avec photo des sites importants)	454
2.6 Exemple 6 : Affichage de la carte de Rennes (Street View)	456

Chapitre 6-2

Dessin (HTML5 CANVAS)

1. Présentation de l'API HTML5 CANVAS.	461
2. Exemples d'applications de l'élément <canvas>	462
2.1 Exemple 1 : Tracé d'un simple carré	462
2.2 Exemple 2 : Tracé d'une grille de TicTacToe	466
2.3 Améliorations possibles sur le jeu du TicTacToe	473

Chapitre 6-3
Graphiques de gestion

- 1. Différentes solutions de conception de graphiques de gestion 475
- 2. Exemples d'utilisation des API Google Charts 476
 - 2.1 Exemple 1 : Tracé d'un histogramme 476
 - 2.2 Exemple 2 : Tracé d'un graphique en secteurs 482
 - 2.3 Exemple 3 : Tracé d'une carte 485

Partie 7 : Frameworks JavaScript

Chapitre 7-1
Positionnement des frameworks JavaScript

- 1. Présentation générale des frameworks JavaScript 489
 - 1.1 Frameworks « front-end » 490
 - 1.2 Frameworks « back-end » 490
 - 1.3 Solutions de développement « hybride » 491
- 2. Les frameworks Node.js, Svelte, React et React Native. 491

Chapitre 7-2
Installation de Node.js

- 1. Présentation du framework Node.js 493
- 2. Installation du framework Node.js 495

Chapitre 7-3
Framework Svelte

- 1. Présentation du framework Svelte 503
- 2. Site svelte.dev 504
- 3. Création locale d'un projet Svelte 507

12 _____ Apprendre à développer

avec JavaScript

4. Installation de Microsoft Visual Studio Code	509
5. Projets Svelte.	510
5.1 « eni_svelte_01 » - Première application	510
5.2 « eni_svelte_02 » - Importance des commentaires	517
5.3 « eni_svelte_03 » - Mise en place d'une image.	519
5.4 « eni_svelte_04 » - Intégration de balises HTML	521
5.5 « eni_svelte_05 » - Un premier bouton	523
5.6 « eni_svelte_06 » - Le rôle du \$ dans les formules	526
5.7 « eni_svelte_07 » - Champs de saisie	528
5.8 « eni_svelte_08 » - Saisie avec curseur	531
5.9 « eni_svelte_09 » - Cases à cocher	534
5.10 « eni_svelte_10 » - Boutons « radio »	537
5.11 « eni_svelte_11 » - Liste.	539
5.12 « eni_svelte_12 » - Liste et boutons « radio »	541
5.13 « eni_svelte_13 » - Boutons Svelte Materialify	544
5.14 « eni_svelte_14 » - Liste déroulante Svelte Materialify.	546
5.15 « eni_svelte_15 » - Switches Svelte Materialify	549
5.16 « eni_svelte_16 » - Première application Sapper-MySQL	551
5.17 « eni_svelte_17 » - Liste déroulante Sapper-MySQL.	568
5.18 « eni_svelte_18 » - Ajout d'un sport via Sapper-MySQL	571
5.19 « eni_svelte_19 » - Mise à jour Sapper-MySQL.	578
5.20 « eni_svelte_20 » - Suppression Sapper-MySQL	584

Chapitre 7-4 Framework React

1. Présentation de React	589
2. Création locale d'un projet React	590
3. Projets React basiques	591
3.1 « eni_react_bases_01 »	592
3.2 « eni_react_bases_02 »	598
3.3 « eni_react_bases_03 »	600

- 3.4 « eni_react_bases_04 » 603
- 3.5 « eni_react_bases_05 » 606
- 3.6 « eni_react_bases_06 » 611
- 4. Les props React 614
 - 4.1 « eni_react_props_01 » 614
 - 4.2 « eni_react_props_02 » 616
 - 4.3 « eni_react_props_03 » 618
- 5. Les librairies tierces pour React 621
 - 5.1 « eni_react_material-ui_table_01 » 622
 - 5.2 « eni_react_material-ui_table_02 » 625
 - 5.3 « eni_react_material-ui_table_03 » 629
 - 5.4 « eni_react_recharts_01 » 632
 - 5.5 « eni_react_recharts_02 » 637
 - 5.6 « eni_react_recharts_03 » 641
- 6. Lecture de fichiers JSON sous React 643
 - 6.1 Différents types de flux JSON 643
 - 6.2 « eni_react_lecture_json_local » 644
- 7. Interactions avec un serveur PHP-MySQL 653
 - 7.1 « eni_react_php_mysql_01 » 654
 - 7.2 « eni_react_php_mysql_02 » 661
 - 7.3 « eni_react_php_mysql_03 » 679
 - 7.4 « eni_react_php_mysql_04 » 683
 - 7.5 « eni_react_php_mysql_05 » 694
- 8. La navigation sous React (routage) 700

14 _____ Apprendre à développer

avec JavaScript

Chapitre 7-5 Framework React Native

1. Approches de développement pour périphériques mobiles	705
1.1 Développements web, natif et hybride	706
1.1.1 Applications web	706
1.1.2 Applications natives	707
1.1.3 Applications hybrides	708
1.2 Les trois principales plateformes	709
1.2.1 Apple iOS	709
1.2.2 Android	709
1.2.3 Windows Phone, Windows 10 Mobile	710
2. Présentation du framework React Native	710
3. Projets React Native	712
3.1 Application « eni_react_native_helloworld »	713
3.2 Application « eni_react_native_list_view »	730
3.3 Application « eni_react_native_input_text »	736
3.4 Application « eni_react_native_picker_basique »	741
3.5 Application « eni_react_native_hook »	746
3.6 Application « eni_react_native_php_mysql »	751
Index	757

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

Chapitre 1
Introduction

- 1. Quelles notions ce livre couvre-t-il ? 17
- 2. Prérequis 19
- 3. Historique de Vue.js 20
 - 3.1 Pourquoi Vue.js ? 20
 - 3.1.1 Ses origines 20
 - 3.1.2 Ses atouts 21
 - 3.2 Qu'est-ce que le Virtual DOM ? 24
 - 3.2.1 Définition 24
 - 3.2.2 API DOM pour mettre à jour une page web 24
 - 3.2.3 Limites de l'API DOM 26
 - 3.2.4 Apparition du concept de DOM virtuel 27
 - 3.2.5 Manipulation du DOM virtuel 29
 - 3.3 Le DOM virtuel avec Vue.js 30
- 4. Modèles d'architectures 32
 - 4.1 MVC (Modèle - Vue - Contrôleur) 32
 - 4.1.1 MVC pour toutes applications 32
 - 4.1.2 MVC adapté aux applications web 33
 - 4.2 MVVM (Model - View - ViewModel) 34
- 5. Comparatifs avec les autres frameworks 36
 - 5.1 Popularité entre Vue, React et Angular 36
 - 5.2 Avantages et inconvénients 38
 - 5.2.1 Angular 38
 - 5.2.2 React.js 39
 - 5.2.3 Vue.js 39
 - 5.2.4 Quel framework pour quel besoin ? 40
 - 5.2.5 Aller plus loin dans la comparaison
avec les autres frameworks 40
- 6. Futur de Vue.js 40

Chapitre 2

Notions essentielles de JavaScript

1. Introduction	43
2. Bases algorithmiques	45
2.1 Variables et types de valeurs	45
2.1.1 Déclaration d'une variable	45
2.1.2 Types de valeurs	46
2.1.3 Contextes d'exécution global et local	48
2.1.4 Portée de variable	50
2.1.5 Conversion de types de données	53
2.1.6 Affectation par décomposition	55
2.2 Structures de contrôle	55
2.2.1 Instructions conditionnelles	55
2.2.2 Boucles	58
3. Fonctions	60
3.1 Définition et utilisation des fonctions	60
3.2 Fermetures (closures)	61
3.3 Fonction en argument d'autres fonctions	63
3.4 Paramètres par défaut et paramètres du reste	64
3.5 Fonctions fléchées	64
3.6 Fonctions anonymes autoexécutantes	65
4. Manipulation de tableaux	67
4.1 Déclarer, lire, modifier, supprimer des éléments	67
4.2 Itérer sur un tableau	69
4.3 Décomposer avec l'opérateur spread	69
4.4 Filtrer un tableau	70
4.5 Trier un tableau	70
5. Manipulation d'objets	70
5.1 Définir un objet et son prototype	70
5.2 Instancier un objet	72
5.3 Lire, ajouter, modifier ou supprimer une propriété	75
5.4 Copier ou fusionner des objets	76
5.5 Itérer sur un objet	77
5.6 Chaîner les méthodes d'objet	77

- 6. Utilisation du mot-clé this 78
 - 6.1 En dehors d'une fonction 78
 - 6.2 Dans une fonction appelée de manière classique 78
 - 6.3 Dans une fonction appelée via call() et apply() 79
 - 6.4 Avec bind() pour associer un objet à this 79
 - 6.5 Dans une fonction appelée en tant que méthode d'objet. 80
 - 6.6 Dans un gestionnaire d'évènement 80
 - 6.7 Dans une fonction fléchée 80
- 7. Gestion des exceptions 81
 - 7.1 L'utilité 81
 - 7.2 La structure try...catch...finally 82
 - 7.3 L'objet Error et les erreurs personnalisées 84
- 8. Utilisation des promesses 85
 - 8.1 Objet natif Promise 85
 - 8.2 Méthode then() 86
 - 8.3 Gestionnaires de then() 87
 - 8.4 Méthode catch() 88
 - 8.5 Chaînage de traitements asynchrones avec les promesses. 90
 - 8.6 Création de promesses déjà tenues ou rompues 92
 - 8.7 Exécution de tâches asynchrones en parallèle 93
 - 8.8 Gestion de la pile d'appels 93
 - 8.9 Fonctions asynchrones avec async 94
- 9. Utilisation des modules JavaScript. 96
 - 9.1 L'historique des modules JavaScript 96
 - 9.1.1 Introduction. 96
 - 9.1.2 Application sans module 97
 - 9.1.3 Objets en tant que modules et fonctions auto-invoquées 99
 - 9.1.4 Modules CommonJS 100
 - 9.1.5 Modules AMD 102
 - 9.1.6 Chargeur de modules. 104
 - 9.1.7 Empaqueteur de modules 106
 - 9.2 Les modules ES 6. 108
 - 9.2.1 Introduction. 108
 - 9.2.2 Déclarer et utiliser un module 108
 - 9.2.3 Éviter les conflits de nommage 109

9.2.4	Agréger plusieurs modules	111
9.2.5	Importer dynamiquement	111

Chapitre 3

Comprendre les fondamentaux de Vue.js

1.	Installation	113
1.1	Une version par environnement	113
1.2	Via téléchargement manuel	114
1.3	Via l'inclusion d'un CDN (le plus simple)	114
1.4	Via npm ou yarn pour de plus gros projets	115
1.4.1	Téléchargement du package vue	115
1.4.2	Explication des différents builds	116
1.5	Via Vue-CLI	118
2.	Outils de développements	119
2.1	VS Code et ses plugins	119
2.1.1	Installer et configurer VS Code	119
2.1.2	Débuguer avec VS Code	121
2.2	Vue Devtools	127
2.3	Vue Performance Devtool	130
2.4	CodeSandbox	132
2.5	Git	132
3.	Instance Vue.js	133
3.1	Hello World	133
3.2	Liaison de données entre View et ViewModel	137
3.3	Cycle de vie d'une instance	139
3.4	Utilisation du mot-clé this	142
4.	Virtual DOM et réactivité	144
4.1	Le système de liaison de données	144
4.2	Le fonctionnement de la file d'attente de mise à jour asynchrone	147
4.3	La déclaration de propriétés réactives	150
4.4	Les limitations de la détection de changement	150
4.5	Les directives pour manipuler le DOM	152
4.5.1	Un attribut HTML spécial	152
4.5.2	Les arguments et les arguments dynamiques	155
4.5.3	Les modificateurs	155

5.	Liaison de données.	156
5.1	Liaison réactive uni-directionnelle.	156
5.1.1	Afficher des données réactives avec l'interpolation texte	156
5.1.2	Rendre un attribut réactif avec v-bind.	156
5.2	Liaison bidirectionnelle avec v-model	157
6.	Affichage des données dans la View.	158
6.1	L'interpolation avec des données plus complexes.	158
6.2	La directive v-html	159
6.3	La directive v-once	160
6.4	Le rendu conditionnel.	161
6.4.1	La directive v-show	161
6.4.2	Les directives v-if, v-else-if et v-else	162
6.5	La balise <template>.	164
7.	Traitement et formatage des données	165
7.1	Valeurs traitées avec les propriétés calculées	165
7.2	Valeurs traitées avec les méthodes	167
7.3	Propriétés calculées vs méthodes	168
7.4	Mutateurs calculés	171
7.5	Valeurs traitées avec les observateurs watch	173
7.6	Utilisation des filtres pour formater les données	175
8.	Affichage de listes de données	177
8.1	Utiliser la directive v-for.	177
8.2	Filtrer les éléments d'une liste	180
8.2.1	Ne pas utiliser v-if avec v-for	180
8.2.2	En utilisant une propriété calculée	182
8.3	Trier les éléments d'une liste	184
8.4	Muter un tableau ou un objet	185
9.	Capture des événements déclenchés par l'utilisateur	189
9.1	Directive v-on	189
9.1.1	Mettre à jour une donnée suite à un événement du DOM	189
9.1.2	Récupérer l'évènement dans la méthode	191
9.2	Modificateurs d'évènements	192
9.3	Évènement dynamique et syntaxe objet	193
10.	Gestion des styles CSS	193
10.1	Scoped styles.	193
10.2	Les styles dynamiques	194

10.3 Les styles dynamiques avec un objet	196
10.4 Les styles dynamiques avec un objet sans CSS	198

Chapitre 4 Utiliser les formulaires

1. Les éléments de formulaire	199
1.1 L'utilisation de bibliothèques tierces	199
1.2 Les champs text et textarea	201
1.3 Les cases à cocher et les boutons radio	204
1.4 Les listes déroulantes	206
2. Les champs particuliers de formulaire	209
2.1 Champs date	209
2.2 Champs de chargement de fichier	211
2.3 Sliders	214
3. Les modificateurs de la directive v-model	215
3.1 Formater une saisie en tant que nombre	215
3.2 Réagir sur l'évènement change plutôt qu'input	216
3.3 Retirer les espaces	217
4. La validation du formulaire	218
4.1 Les bibliothèques de validations de formulaire	218
4.2 Utiliser VeeValidate	218
4.2.1 Installer	218
4.2.2 Utiliser les règles de validation par défaut	220
4.2.3 Ajouter ses propres règles de validation	226
4.2.4 Gérer la soumission du formulaire	228
4.3 Utiliser Vuelidate	231
4.3.1 Installer	231
4.3.2 Utiliser les règles de validation par défaut	232
4.3.3 Ajouter ses propres règles de validation	240
4.3.4 Gérer la soumission de formulaire	246

Chapitre 5
Utiliser les composants

- 1. Qu'est-ce qu'un composant ? 249
 - 1.1 Définition 249
 - 1.2 Structure d'un composant 251
 - 1.3 Structuration d'une application en composants. 253
- 2. Création d'un composant 254
 - 2.1 Manière globale. 254
 - 2.2 Manière locale 254
- 3. Communication entre composants 255
 - 3.1 Parent - enfant 255
 - 3.1.1 Définition des props dans la Vue 255
 - 3.1.2 Types de données. 256
 - 3.1.3 Définition des props dans le ViewModel. 256
 - 3.1.4 Validation des props 257
 - 3.1.5 Utilisation des props dans le ViewModel 258
 - 3.2 Enfant - parent 260
 - 3.2.1 Émettre un évènement depuis un composant enfant 260
 - 3.2.2 Écouter un évènement dans le composant parent. 260
 - 3.3 Enfant - Enfant 262
 - 3.3.1 Limite de l'utilisation des props
et des évènements personnalisés 262
 - 3.3.2 Utilisation d'un bus d'évènements 263
- 4. Pour aller plus loin. 264
 - 4.1 Les différentes propriétés d'instance d'un composant 264
 - 4.2 L'utilisation de la directive v-model sur un composant 265
 - 4.2.1 V-model pour un composant contenant un champ texte. ... 265
 - 4.2.2 V-model pour un composant contenant une case à cocher
ou un bouton radio 266
 - 4.3 L'héritage des attributs du composant 267
 - 4.4 Les évènements natifs à relier au composant 269
 - 4.5 La liaison bidirectionnelle d'une prop avec le modificateur .sync ... 270
- 5. Plusieurs types de composants 271
 - 5.1 Les composants dynamiques 271
 - 5.1.1 L'élément <component> et l'attribut is 271
 - 5.1.2 L'élément <keep-alive> 272

5.2	Les composants monofichiers	273
5.3	Les composants basiques	276
5.4	Les composants asynchrones pour améliorer les performances	277
5.5	L'utilisation de la fonction <code>render()</code> pour le rendu	281
5.6	Les composants fonctionnels	285

Chapitre 6

Mutualiser des fonctionnalités

1.	Utiliser les slots pour injecter du contenu	289
1.1	Définition	289
1.2	Slots nommés	293
1.3	Props de slot	295
2.	Utiliser les composants sans rendu	296
2.1	Définition	296
2.2	Création d'un composant de captures d'erreurs	297
3.	Utiliser les mixins	302
3.1	Définition	302
3.2	Stratégies de fusion des options	304
4.	Utiliser les plugins	307
4.1	Définition	307
4.2	Optimisation des performances	311
4.2.1	Vérifier la taille des plugins et leur vitesse de chargement	311
4.2.2	Utiliser le tree-shaking pour les plugins qui le permettent	313
5.	Les directives personnalisées	315
5.1	Définition	315
5.2	Arguments et modificateurs	318

Chapitre 7

Créer et déployer une application avec Vue CLI

1.	Plusieurs types d'applications	321
1.1	Une SPA classique	321
1.1.1	Définition	321
1.1.2	Contraintes d'une SPA classique	322

- 1.2 Une application universelle (SPA + SSR) 323
 - 1.2.1 Définition 323
 - 1.2.2 Contraintes d'une application universelle 325
- 1.3 Un générateur de site statique 327
 - 1.3.1 Définition 327
 - 1.3.2 Architecture JAMstack 329
 - 1.3.3 Contraintes des sites statiques 332
- 1.4 Une PWA 332
 - 1.4.1 Définition 332
 - 1.4.2 Différents types de caches 333
 - 1.4.3 Contraintes d'une PWA 335
- 1.5 Quel type d'application pour quel besoin ? 336
- 2. Création d'un projet avec Vue CLI 337
 - 2.1 Présentation 337
 - 2.2 Installation 339
 - 2.2.1 Prérequis 339
 - 2.2.2 Prototypage rapide d'une application 339
 - 2.2.3 Création d'une application Vue.js complète 340
 - 2.3 Comprendre l'arborescence du projet 348
 - 2.3.1 Racine du projet 348
 - 2.3.2 Dossier src 349
 - 2.3.3 Modification de l'arborescence
pour un projet de taille moyenne 351
 - 2.3.4 Modification de l'arborescence pour un gros projet 356
- 3. Fonctionnalités de Vue CLI 358
 - 3.1 Les plugins et les presets 358
 - 3.1.1 Les plugins 358
 - 3.1.2 Les presets 360
 - 3.2 Le rechargement à chaud 361
 - 3.3 La configuration de Webpack 362
 - 3.3.1 Comprendre le fonctionnement 362
 - 3.3.2 Accéder aux fichiers de configurations des différents modes . 365
 - 3.3.3 Modifier la configuration 366
 - 3.3.4 Utiliser l'API de chaînage pour modifier
plus finement la configuration 369

3.4	Les modes et variables d'environnement	370
3.4.1	Les modes pour chaque environnement	370
3.4.2	Les variables d'environnement	371
3.5	Le dossier public	375
3.5.1	Interpolation dans les fichiers HTML	375
3.5.2	Fichiers statiques	376
3.6	La compatibilité des navigateurs	377
3.6.1	La liste des navigateurs compatibles	377
3.6.2	Le mode moderne	378
3.6.3	Les contrôles CORS en mode moderne pour les modules ES6	378
4.	Déploiement de votre application en production	380
4.1	Procédure de déploiement	380
4.1.1	Compiler votre application avec Webpack	380
4.1.2	Prévisualiser en local	381
4.1.3	Déployer votre application sur un serveur	381
4.2	Déploiement simple avec des plateformes PaaS	382
4.2.1	Déployer sur Netlify	382
4.2.2	Déployer sur Heroku	385
4.2.3	Créer une image Docker avec Nginx	387

Chapitre 8

Consommer des API REST et GraphQL

1.	Plusieurs types d'API (REST et GraphQL)	391
1.1	Définition et historique	391
1.2	API REST	392
1.2.1	Définition	392
1.2.2	Contraintes d'une API REST	395
1.3	API GraphQL	399
1.3.1	Définition	399
1.3.2	Description du fonctionnement	402
2.	Sécurité et modes d'authentification	403
2.1	Principes à adopter	403
2.2	Cookie d'authentification	405
2.2.1	Cookie de session	405
2.2.2	Attaques CSRF (Cross Site Request Forgery)	406

- 2.2.3 Attaques XSS (Cross Site Scripting) 407
- 2.3 Authentification basique (Basic) 408
- 2.4 Authentification avec jeton porteur (Bearer) 409
- 2.5 Authentification avec jeton signé (Bearer + JWT) 409
 - 2.5.1 Signature 409
 - 2.5.2 Principe clé privée - clé publique 410
 - 2.5.3 Composition d'un jeton JWT 410
- 2.6 Authentification avec une clé API 411
 - 2.6.1 Authentification de l'application par le fournisseur d'API 411
 - 2.6.2 Problématique de sécurité sur le réseau de l'utilisateur 412
 - 2.6.3 Où conserver les clés API ? 412
- 2.7 Authentification avec OAuth 2.0 413
 - 2.7.1 Un serveur d'authentification 413
 - 2.7.2 La problématique de conservation du jeton 415
- 3. Créer une API rapidement avec Strapi 416
 - 3.1 Créer le backend de votre API 416
 - 3.1.1 Présentation 416
 - 3.1.2 Installation 418
 - 3.2 Créer le schéma de base de données 421
 - 3.2.1 Présentation du tableau de bord 421
 - 3.2.2 Créer le schéma de base de données 423
 - 3.2.3 Gérer les rôles et les permissions 430
 - 3.3 Tester l'API REST 432
 - 3.3.1 Tester les requêtes GET, POST, PUT et DELETE 432
 - 3.3.2 Utiliser des paramètres pour filtrer une collection 437
 - 3.4 Tester l'API GraphQL 438
 - 3.4.1 Utiliser Postman ou l'éditeur GraphQL de Strapi 438
 - 3.4.2 Récupérer des données 442
 - 3.4.3 Créer un enregistrement 444
 - 3.4.4 Modifier un enregistrement 446
 - 3.4.5 Supprimer un enregistrement 447
 - 3.4.6 Utiliser des filtres 448
 - 3.4.7 Fonctions d'agrégation et clause groupBy 450
- 4. Fetch et Axios pour consommer des API REST 454
 - 4.1 Fetch 454
 - 4.1.1 Définition et usage 454
 - 4.1.2 Les objets Request et Response 454

4.1.3	Gérer les erreurs	459
4.1.4	Interrompre une requête	460
4.1.5	Limiter les appels trop fréquents avec debounce et throttle	461
4.1.6	Exécuter des requêtes en parallèle.	463
4.2	Création d'un catalogue de produits	464
4.2.1	Afficher la liste des produits avec fetch	464
4.2.2	Gérer les erreurs avec un composant sans rendu	469
4.2.3	Mutualiser la fonctionnalité d'appel API	475
4.2.4	Mettre en place une pagination	478
4.2.5	Ajouter un menu déroulant pour filtrer la liste par catégorie.	483
4.2.6	Utiliser une barre de recherche	490
4.3	Axios	493
4.3.1	Définition et usage.	493
4.3.2	L'objet config et l'objet response.	495
4.3.3	Gérer les erreurs	499
4.3.4	Interrompre une requête	501
4.3.5	Utiliser les intercepteurs	502
4.3.6	Centraliser la configuration axios dans l'application.	503
5.	Apollo pour consommer des API GraphQL	505
5.1	Installation	505
5.1.1	Installation avec Vue CLI	505
5.1.2	Installation manuelle pour consommer une API Strapi	505
5.1.3	Installation du plugin VS Code.	507
5.2	Récupérer des données	508
5.2.1	Les requêtes intelligentes.	508
5.2.2	Utiliser des paramètres dans la requête	513
5.2.3	Requête conditionnelle	515
5.2.4	Modifier les données reçues.	515
5.2.5	Intercepter les erreurs	516
5.2.6	Les options supplémentaires	517
5.3	Placer le code des requêtes dans des fichiers .gql.	518
5.3.1	Créer un fichier .gql	518
5.3.2	Importer un fichier .gql	519
5.3.3	Utiliser les fragments.	520
5.4	Créer, modifier ou supprimer une ressource	521
5.4.1	La méthode this.\$apollo.mutate.	521
5.4.2	Le composant ApolloMutation.	524

- 5.5 Mise à jour du catalogue de produits avec Apollo 525
 - 5.5.1 Nettoyer l'ancienne logique métier de l'API REST 525
 - 5.5.2 Définir les requêtes GraphQL 525
 - 5.5.3 Définition des requêtes intelligentes 528

Chapitre 9

Utiliser Vue Router pour la navigation

- 1. Définition et installation 533
 - 1.1 Définition 533
 - 1.2 Installation 534
 - 1.2.1 Télécharger le plugin ou utiliser un CDN 534
 - 1.2.2 Utiliser npm 534
 - 1.2.3 Utiliser Vue CLI 534
 - 1.3 Déclaration du routeur 535
- 2. Usage 536
 - 2.1 Définition d'une route 536
 - 2.2 Résolution d'une route 538
 - 2.3 Mode historique 541
 - 2.4 Affichage des composants dans des vues routeur 542
 - 2.4.1 La vue par défaut 542
 - 2.4.2 Les vues nommées 544
 - 2.5 Naviguer à l'aide des liens <router-link> 546
 - 2.5.1 Fonctionnement 546
 - 2.5.2 La classe active 546
 - 2.5.3 Les props de <router-link> 547
 - 2.5.4 Les routes nommées 548
 - 2.6 Routes dynamiques 548
 - 2.6.1 Utilisation des segments dynamiques 548
 - 2.6.2 Récupérer les segments dynamiques avec \$route 550
 - 2.6.3 Utiliser les segments dynamiques pour passer des valeurs
aux props d'un composant 551
 - 2.6.4 Passage entre deux chemins liés à la même route 553
 - 2.7 Les routes imbriquées 553

3.	Navigation programmée	555
3.1	Navigation programmée avec \$router	555
3.1.1	\$router.push() et \$router.replace()	555
3.1.2	\$router.forward(), \$router.back(), \$router.go()	556
3.2	Les intercepteurs de navigation	556
3.2.1	Des intercepteurs globaux, par route ou par composant	556
3.2.2	Flux de résolution lors d'un passage d'une route à l'autre	558
3.2.3	Déclaration d'un intercepteur	559
3.2.4	Exemple d'utilisation d'un intercepteur global	560
3.2.5	Exemple de chargement de données avant la navigation	561
3.3	Les redirections	562
4.	Pour aller plus loin	563
4.1	Les transitions	563
4.1.1	Le composant <transition>	563
4.1.2	Exemple d'une transition entrante et sortante en fondu	565
4.2	Optimisation des performances	566
4.2.1	Utilisation du lazy-loading avec les composants asynchrones	566
4.2.2	Utiliser le préchargement avec la méthode webpackPrefetch	568

Chapitre 10

Utiliser Vuex pour la gestion d'états

1.	Partager un état global entre plusieurs composants	569
1.1	Problématique	569
1.1.1	Le flux de donnée unidirectionnel	569
1.1.2	Utiliser les props et les événements pour partager et accéder à l'état global	570
1.1.3	Utiliser un bus d'évènement pour réagir aux actions	571
1.2	Utilisation d'un store pour centraliser les données	572
1.2.1	Déporter les états dans un objet partagé	572
1.2.2	Tracer les mutations	573
1.2.3	Utiliser Vuex	575

2.	Installer et utiliser le store Vuex.	577
2.1	Installation	577
2.1.1	Avec un CDN.	577
2.1.2	Avec npm ou yarn	577
2.1.3	Avec Vue CLI.	579
2.2	État	580
2.2.1	Définir et accéder à une donnée de l'état	580
2.2.2	Utiliser mapState() pour générer les propriétés calculées	581
2.3	Accesseurs	583
2.3.1	Accéder à une donnée d'état via un accesseur	583
2.3.2	Utiliser mapGetters() pour générer les propriétés calculées	585
2.4	Mutateurs	586
2.4.1	Muter une donnée de l'état du store.	586
2.4.2	Muter plusieurs données avec une mutation.	588
2.4.3	Muter un objet ou un tableau.	590
2.4.4	Utiliser un fichier de constantes pour lister les types de mutations.	591
2.4.5	Utiliser mapMutations() pour générer les propriétés calculées.	592
2.4.6	Les mutations avec la directive v-model	593
2.5	Actions.	595
2.5.1	Utiliser les actions pour des traitements asynchrones	595
2.5.2	Utiliser mapActions pour générer les méthodes	597
2.5.3	Chaîner des actions asynchrones	597
3.	Utiliser les modules pour organiser son store	599
3.1	Séparer le store en plusieurs fichiers	599
3.2	Utiliser les modules Vuex.	601
	Conclusion	603
	Index.	605