

Bien développer **pour le Web 2.0**

AJAX • Prototype • Scriptaculous
XHTML/Css • JavaScript • DOM

Christophe Porteneuve

**Préface de Tristan Nitot,
président de Mozilla Europe**

© Groupe Eyrolles, 2007,

ISBN : 2-212-12028-1, ISBN 13 : 978-2-212-12028-8

EYROLLES



Table des matières

Avant-propos	XXIII
À qui s'adresse ce livre ?	XXIII
Qu'allez-vous trouver dans ce livre ?	XXIV
Les standards du Web	XXV
De quelles technologies parle-t-on ?	XXV
Qui est à la barre, et où va-t-on ?	XXVI
À quoi servent les standards ?	XXIX
Qu'en pensent les concepteurs de navigateurs ?	XXXI
Quelques mots sur les dernières versions	XXXIII
Qu'est-ce que le « Web 2.0 » ?	XXXIV
Vue d'ensemble, chapitre par chapitre	XXXVI
Première partie : donner vie aux pages	XXXVI
Deuxième partie : Ajax ou l'art de chuchoter	XXXVII
Troisième partie : parler au reste du monde	XXXVII
Des annexes pour le débutant comme pour l'expert	XXXVIII
Aller plus loin... ..	XXXVIII
À propos des exemples de code	XXXIX
Remerciements	XXXIX
CHAPITRE 1	
Pourquoi et comment relever le défi du Web 2.0 ?	1
Avant/après : quelques scénarios frappants	1
La saisie assistée : complétion automatique de texte	1
Le chargement à la volée	4
La sauvegarde automatique	10
Bien maîtriser ses outils clés : XHTML, CSS, JS, DOM et Ajax	11
Faire la part des choses : Ajax, c'est quoi au juste ?	13
Plan d'actions pour deux objectifs : méthode et expertise	15

PREMIÈRE PARTIE

Donner vie aux pages 17

CHAPITRE 2

Ne prenez pas JavaScript pour ce qu'il n'est pas..... 19

Mythes et rumeurs sur JavaScript	20
JavaScript serait une version allégée de Java	20
JavaScript ne serait basé sur aucun standard	20
JavaScript serait lent	21
JavaScript serait un langage jouet, peu puissant	21
S'y retrouver entre JavaScript, EcmaScript, JScript et ActiveScript	22
Tout ce que vous ne soupçonniez pas : les recoins du langage	22
Variables déclarées ou non déclarées ?	23
Types de données	25
Fonctions et valeurs disponibles partout	27
<i>Les mystères de parseFloat et parseInt</i>	28
Rappels sur les structures de contrôle	31
<i>Les grands classiques</i>	31
<i>Labélisation de boucles</i>	31
<i>Simplifier l'accès répétitif à un objet avec with</i>	33
Opérateurs méconnus	34
<i>Retour rapide sur les grands classiques</i>	34
<i>Opérateurs plus exotiques</i>	35
<i>Comportements particuliers</i>	35
Prise en charge des exceptions	36
<i>Les types d'exceptions prédéfinis</i>	36
<i>Capter une exception : try/catch</i>	37
<i>Garantir un traitement : finally</i>	39
<i>Lancer sa propre exception : throw</i>	40
Améliorer les objets existants	41
<i>Un peu de théorie sur les langages à prototypes</i>	41
<i>Mise en pratique</i>	42
Arguments des fonctions	45
Le binding des fonctions : mais qui est « this » ?	47
Idiomes intéressants	51
<i>Initialisation et valeur par défaut avec </i>	52
<i>Sélectionner une propriété (donc une méthode) sur condition</i>	53
<i>Tester l'absence d'une propriété dans un objet</i>	53
<i>Fonctions anonymes : jamais new Function !</i>	54

<i>Objets anonymes comme hasbes d'options</i>	54
<i>Simuler des espaces de noms</i>	55
« Unobstrusive JavaScript » : bien associer code JS et page web	56
Astuces pour l'écriture du code	58
Déjouer les pièges classiques	58
Améliorer la lisibilité	60
Mieux déboguer du JavaScript	62
La console JavaScript	63
<i>Dans Mozilla Firefox</i>	63
<i>Dans Opera</i>	64
<i>Dans Safari</i>	64
<i>Dans Konqueror</i>	65
<i>Dans MSIE</i>	65
Venkman, le débogueur JavaScript	66
Firebug, le couteau suisse du développeur Web 2.0	70
Pour aller plus loin	74
Livres	74
Sites	75

CHAPITRE 3

Manipuler dynamiquement la page avec le DOM 77

Pourquoi faut-il maîtriser le DOM ?	78
La pierre angulaire des pages vivantes	78
Maîtriser la base pour utiliser les frameworks	78
Comprendre les détails pour pouvoir déboguer	78
Le DOM et ses niveaux 1, 2 et 3	79
Vue d'ensemble des niveaux	79
Support au sein des principaux navigateurs	80
Les aspects du DOM : HTML, noyau, événements, styles...	81
Maîtriser les concepts : document, nœud, élément, texte et collection	81
Le DOM de votre document : une arborescence d'objets	82
Node	84
Document	88
Element	90
Text	91
NodeList et NamedNodeMap	92
DOMImplementation	94
HTMLDocument	95
HTMLElement	96
Quelques bonnes habitudes	97
Détecer le niveau de DOM disponible	97

Créer les nœuds dans le bon ordre	98
Ne scripter qu'après que le DOM voulu soit construit	99
Ne jamais utiliser d'extension propriétaire	101
Utiliser un inspecteur DOM	101
<i>L'inspecteur DOM de Firefox/Mozilla</i>	101
<i>L'inspecteur DOM de Firebug</i>	104
Répondre aux événements	108
Les truands : les attributs d'événement dans HTML	108
La brute : les propriétés d'événement dans le DOM niveau 0	109
Le bon : addEventListener	110
<i>Accommoder MSIE</i>	112
La propagation : capture ou bouillonnement ?	113
<i>Le modèle le plus courant : le bouillonnement</i>	114
<i>La capture, ou comment jouer les censeurs</i>	115
L'objet Event	116
<i>Récupérer l'élément déclencheur</i>	116
<i>Stopper la propagation</i>	116
<i>Annuler le traitement par défaut</i>	117
<i>JavaScript, événements et accessibilité</i>	117
Besoins fréquents et solutions concrètes	119
Décoration automatique de labels	120
Validation automatique de formulaires	125
Résoudre les écueils classiques	128
MSIE et la gestion événementielle	129
MSIE et le DOM de select/option	130
Les principaux points problématiques	130
Pour aller plus loin	131
Livres	131
Sites	131

CHAPITRE 4

Prototype : simple, pratique, élégant, portable !..... 133

Avant de commencer... ..	134
Un mot sur les versions	134
L'objet global Prototype	135
Vocabulaire et concepts	136
Espaces de noms et modules	136
Itérateurs	136
Élément étendu	137
Alias	138
Comment utiliser Prototype ?	138

Vous allez aimer les dollars	139
La fonction \$ facilite l'accès aux éléments	139
La fonction \$A joue sur plusieurs tableaux	140
La fonction \$H, pour créer un Hash	141
La fonction \$F, des valeurs qui sont les vôtres	142
La fonction \$R et les intervalles	142
La fonction \$\$ et les règles CSS	143
Jouer sur les itérations avec \$break et \$continue	143
Extensions aux objets existants	145
Un Object introspectif	145
Gérer correctement le binding	146
Des drôles de numéros	148
Un objet String très enrichi	149
<i>Retraits de caractères : strip, stripTags, stripScripts, truncate</i>	150
<i>Transformations : sub, gsub, escapeHTML, unescapeHTML, camelize</i>	151
<i>Fragments de scripts : extractScripts, evalScripts</i>	153
<i>Conversions et extractions : scan, toQueryParams, parseQuery, toArray, inspect</i>	153
Des tableaux surpuissants !	154
<i>Conversions : from, inspect</i>	154
<i>Extractions : first, last, indexOf</i>	155
<i>Transformations : clear, compact, flatten, without, reverse, reduce, uniq</i>	156
Extraire les éléments ayant une classe précise	157
Modules et objets génériques	158
Enumerable, ce héros	158
<i>L'itération elle-même : each</i>	159
<i>Tests sur le contenu : all, any, include, member</i>	159
<i>Extractions : detect, find, findAll, select, grep, max, min, pluck, reject</i>	160
<i>Transformations et calculs : collect, map, inject, invoke, partition, sortBy</i>	162
<i>Conversions : toArray, entries, inspect</i>	164
Tableaux associatifs avec Hash	165
ObjectRange : intervalles d'objets	166
PeriodicalExecuter ne se lasse jamais	167
Vous devriez réévaluer vos modèles	167
<i>Utiliser un motif de détection personnalisé</i>	168
Try.these cherche une méthode valide	169
Manipulation d'éléments	170
Element, votre nouveau meilleur ami	170
<i>Element.Methods et les éléments étendus</i>	170
<i>Valeur de retour des méthodes</i>	172
<i>Élément es-tu là ? hide, show, toggle et visible</i>	172
<i>Gestion du contenu : cleanWhitespace, empty, remove, replace et update</i>	173

<i>Styles et classes : addClassName, classNames, getElementsByClassName, getElementsBySelector, getStyle, hasClassName, match, removeClassName et setStyle</i>	174
<i>Les copains d'abord : ancestors, descendants, nextSiblings, previousSiblings, siblings</i>	175
<i>Bougez ! down, next, previous et up</i>	176
<i>Positionnement : getDimensions, getHeight, makePositioned, undoPositioned</i>	177
<i>Défilement et troncature : makeClipping, scrollTo, undoClipping</i>	178
<i>childOf le mal nommé</i>	179
Selector, l'objet classieux	179
Manipulation de formulaires	180
Field / Form.Element	180
<i>Un mot sur Form.Element.Serializers</i>	181
Form	182
Form.Observer	184
Form.Element.Observer	184
Gestion unifiée des événements	185
Event	185
<i>Pister un événement</i>	186
<i>Démasquer l'élément qui a reçu l'événement</i>	187
<i>Étouffer la propagation de l'événement</i>	188
<i>Déterminer l'arme du crime : souris ou clavier</i>	188
Form.EventObserver	190
Form.Element.EventObserver	190
Insertions dynamiques	190
Pour aller plus loin...	192
Sites	192
Groupe de discussion	192
Canal IRC	192

DEUXIÈME PARTIE

Ajax, ou l'art de chuchoter 193

CHAPITRE 5

Les mains dans le cambouis avec XMLHttpRequest 195

Anatomie d'une conversation Ajax	195
Un petit serveur pour nos tests	196
Installation de Ruby	198
<i>Sous Windows</i>	199
<i>Sous Linux/BSD</i>	200
<i>Sous Mac OS X</i>	201
<i>Un mot sur le cache</i>	201

Un petit serveur HTTP et un code dynamique simple	202
La petite histoire de XMLHttpRequest	204
Origines et historique	204
Bien préparer un échange asynchrone	205
ActiveX versus objet natif JavaScript	205
Créer l'objet requêteur	205
Décrire notre requête	207
Envoyer la requête	208
Recevoir et traiter la réponse	209
Une utilisation complète de notre petit serveur d'exemple	210
Comment surveiller les échanges Ajax de nos pages ?	214
Types de réponse : XHTML, XML, JS, JSON...	215
Bien choisir son type de réponse	216
Une réponse textuelle simple : renvoyer une donnée basique	217
<i>Exemple 1 : sauvegarde automatique</i>	<i>217</i>
<i>Exemple 2 : barre de progression d'un traitement serveur</i>	<i>225</i>
Fragments de page prêts à l'emploi : réponse XHTML	231
<i>Un aperçu des coulisses de l'insertion...</i>	<i>236</i>
Dans la cour des grands : XPath pour traiter des données XML complexes ..	237
<i>Vite et bien : utilisation de DOM niveau 3 XPath</i>	<i>238</i>
<i>En simulant : utilisation de GoogleAJAXSLT</i>	<i>243</i>
Piloter la page en renvoyant du JavaScript	245
JSON : l'idéal pour des données structurées spécifiques	248

CHAPITRE 6

Ajax tout en souplesse avec Prototype	255
Prototype encore à la rescousse	256
Ajax.Request, c'est tellement plus simple !	256
Plus de détails sur Ajax.Request	258
Ajax.Updater : mettre à jour un fragment XHTML, exécuter un script	260
<i>Différencier la mise à jour entre succès et échec</i>	<i>263</i>
Presque magique : Ajax.PeriodicalUpdater	264
<i>Comprendre l'option decay</i>	<i>265</i>
Petits secrets supplémentaires	265
Pour aller plus loin...	268
Livres	268
Sites	268
Groupe de discussion	268
Canal IRC	268

CHAPITRE 7

Une ergonomie de rêve avec script.aculo.us..... 269

Une ergonomie haut de gamme avec script.aculo.us	270
Charger script.aculo.us	270
Les effets visuels	271
Les effets noyau	271
<i>Invocation de l'effet</i>	272
<i>Options communes à tous les effets noyau</i>	272
<i>Fonctions de rappel</i>	273
<i>Qu'y a-t-il dans un objet d'effet ?</i>	273
<i>Et si on essayait quelques effets ?</i>	274
<i>Les effets combinés</i>	280
<i>Files d'effets</i>	284
Glisser-déplacer	288
<i>Faire glisser un élément avec Draggable</i>	288
<i>Gérer le dépôt d'un élément avec Droppables</i>	294
Tri de listes par glisser-déplacer	303
<i>Que peut-on trier ?</i>	304
<i>Activer les fonctions d'ordonnement</i>	304
<i>Désactiver l'ordonnement</i>	313
<i>Envoyer l'ordre au serveur</i>	313
Complétion automatique de texte	314
<i>Création d'un champ de saisie à complétion automatique</i>	314
<i>Interaction clavier et souris</i>	317
<i>Un premier exemple</i>	317
<i>Personnalisation des contenus renvoyés</i>	323
Et ce n'est pas tout ! Il y a d'autres services	331
Avoir le bon recul : les cas où Ajax est une mauvaise idée	331
Ajax et l'accessibilité	332
<i>Considérations techniques</i>	332
<i>Considérations ergonomiques</i>	333
Utilisations pertinentes et non pertinentes	334
Pratiques recommandées	335
<i>Principes généraux</i>	335
<i>Ergonomie et attentes de l'utilisateur</i>	336
<i>Cognitif/Lecteurs d'écran</i>	336
Autres frameworks reconnus	338
Dojo	338
Mochikit	338
OpenRico	338

Pour aller plus loin...	339
Site	339
Groupe de discussion	339
Canal IRC	339

TROISIÈME PARTIE

Interagir avec le reste du monde341

CHAPITRE 8

Services web et REST : nous ne sommes plus seuls..... 343

Pourquoi la page ne parlerait-elle qu'à son propre site ?	344
Contraintes de sécurité sur le navigateur	344
Une « couche proxy » sur votre serveur	345
Architecture de nos exemples	346
Comprendre les services web	347
Qu'est-ce qu'une API REST ?	348
Cherchons des livres sur Amazon.fr	349
Obtenir une clé pour utiliser l'API	350
L'appel REST à Amazon.fr	351
<i>Anatomie de la requête</i>	351
<i>Le document XML de réponse</i>	353
Notre formulaire de recherche	353
Passer par Ajax	356
<i>La couche serveur, intermédiaire de téléchargement</i>	356
<i>Intercepter le formulaire</i>	358
De XML à XHTML : la transformation XSLT	360
<i>Notre feuille XSLT</i>	360
<i>Apprendre XSLT à notre page</i>	364
<i>Charger la feuille XSLT au démarrage</i>	364
<i>Effectuer la transformation</i>	365
<i>Embellir le résultat</i>	367
Rhumatismes 2.0 : prévisions météo	371
Préparer le projet	373
Récupérer les prévisions d'un lieu	374
<i>Requête et réponse REST</i>	375
<i>Initialisation de la page et obtention des prévisions</i>	377
<i>Et la feuille XSLT ?</i>	380
<i>Les petites touches finales</i>	384
Rechercher un lieu	389
<i>Préparons le terrain</i>	389

<i>Éblouissez vos amis avec Ajax.XSLTCompleter</i>	392
<i>Brancher les composants ensemble</i>	394
Gérer des images chez Flickr	398
Obtenir une clé API chez Flickr	398
Format général des requêtes et réponses REST chez Flickr	399
Préparer le terrain	400
Chargement centralisé parallèle des feuilles XSLT	403
Obtenir les informations du jeu de photos	404
Récupérer les photos du jeu	413
Afficher une photo et ses informations	419
Pour aller plus loin...	424

CHAPITRE 9

L'information à la carte : flux RSS et Atom 425

Aperçu des formats	426
Une histoire mouvementée	426
<i>RSS 0.9x et 2.0 : les « bébés » de Dave Winer</i>	426
<i>RSS 1.0 : une approche radicalement différente</i>	426
<i>Atom, le fruit de la maturité</i>	427
Informations génériques	428
Le casse-tête du contenu HTML	428
Récupérer et afficher un flux RSS 2.0	429
Format du flux	430
Préparer le terrain	431
<i>La feuille XSLT</i>	433
Chargement et formatage du flux	435
Ajustements des dates et titres	438
Affichage plus avancé et flux Atom 1.0	439
Notre flux Atom	440
Préparation de notre lecteur de flux	442
La feuille XSLT et le formatage	444
<i>Charger la feuille et le flux</i>	446
<i>Afficher dynamiquement les billets complets</i>	449
Les mains dans le cambouis : interpréter le HTML encodé	452
<i>Traiter des quantités massives de HTML encodé</i>	452
<i>Les dates W3DTF</i>	454
Pour aller plus loin...	456
Livres	456
Sites	457

ANNEXE A

Bien baliser votre contenu : XHTML sémantique 461

Les avantages insoupçonnés	462
Pour le site et ses propriétaires	462
Pour le développeur web	463
Règles syntaxiques et sémantiques	463
La DTD et le prologue	465
XHTML, oui mais lequel ?	466
Le balisage sémantique	466
Les balises XHTML 1 Strict par catégorie	467
Balises structurelles	467
Balises sémantiques	468
Balises de liaison	470
Balises de métadonnées	471
Balises de présentation	471
Balises de formulaires et d'interaction	473
Balises dépréciées	474
Attributs incontournables	475
Besoins fréquents, solutions concrètes	476
Un formulaire complexe mais impeccable	476
Un tableau de données à en-têtes groupés	483
Un didacticiel technique	485
Pour aller plus loin...	488
Livres	488
Sites	488

ANNEXE B

Aspect irréprochable et flexible : CSS 2.1 491

Statut, état et vocabulaire	492
Les versions de CSS	492
Prise en charge actuelle	492
Le jargon CSS	493
Bien comprendre la « cascade » et l'héritage	494
Le sens de la cascade	494
Calcul de la spécificité d'une règle	495
Que se passe-t-il avec la présentation dans HTML ?	496
L'héritage	496
<i>De la valeur spécifiée à la valeur concrète</i>	497
Les modèles des boîtes et de mise en forme visuelle	497
Les côtés d'une boîte : ordre et syntaxes courtes	498

Unités absolues et relatives	498
Marge, bordure et espacement	500
Éléments en ligne et de type bloc	501
Éléments remplacés	501
Fusion des marges	501
Le modèle W3C et le modèle Microsoft	502
Tour d'horizon des sélecteurs	503
Groupement	504
Pseudo-éléments	504
Tour d'horizon des propriétés	505
De l'art de réaliser des CSS légères	505
Propriétés du modèle des boîtes : marges, espacements et bordures	506
Propriétés de formatage visuel :	
positionnement, largeur, hauteur, baseline	506
Propriétés de contenu généré automatiquement	507
Propriétés de pagination	508
Propriétés de couleurs et d'arrière-plan	509
Propriétés de gestion de la police de caractères	509
<i>Taille de police</i>	510
<i>Famille de polices</i>	510
<i>Tout spécifier d'un coup !</i>	510
Propriétés de gestion du corps du texte	512
<i>L'espacement dans le corps du texte</i>	512
Propriétés des tableaux	513
Propriétés de l'interface utilisateur	513
Pour aller plus loin...	514
Livres	514
Sites	514

ANNEXE C

Le « plus » de l'expert : savoir lire une spécification 515

Intérêts d'aller chercher l'information à la source	516
Certitude et précision	516
« On m'a dit que là-dessus, c'est toi qui sais tout » : l'expertise	516
Les principaux formats de spécifications web	517
Les recommandations du W3C	517
Les grammaires formelles de langages à balises : DTD et schémas XML	517
Les RFC de l'IETF : protocoles et formats d'Internet	518
S'y retrouver dans une recommandation W3C	518
URL et raccourcis	518

Structure générale d'une recommandation	519
Recours à des syntaxes formelles	523
Les descriptions de propriétés CSS	525
Les descriptions de propriétés et méthodes DOM	526
Déchiffrer une DTD	528
Naviguer dans un schéma XML	532
Parcourir une RFC	535
Format général d'une RFC	535
Structure générale d'une RFC	537
Vos spécifications phares	539
ANNEXE D	
Développer avec son navigateur web	541
Le cache peut être votre pire ennemi	542
Le rafraîchissement strict	542
Vider le cache	542
Configurer le cache	543
Firefox, favori du développeur grâce aux extensions	544
Les trésors du menu Debug caché dans Safari	545
MSIE et la Internet Explorer Developer Toolbar	546
Et Opera, qu'a-t-il pour nous ?	546
Index.....	547