

Table des matières

Avant-propos	XXV
À qui s'adresse ce livre ?	XXV
Qu'allez-vous trouver dans ce livre ?	XXVI
Qu'apporte cette deuxième édition ?	XXVII
Les standards du Web	XXVII
De quelles technologies parle-t-on ?	XXVIII
Qui est à la barre, et où va-t-on ?	XXIX
À quoi servent les standards du Web ?	XXXII
Et les navigateurs, qu'en pensent-ils ?	XXXIV
Quelques mots sur les dernières versions	XXXVI
Qu'est-ce que le « Web 2.0 » ?	XXXVII
Vue d'ensemble, chapitre par chapitre	XXXIX
Première partie : donner vie aux pages	XXXIX
Deuxième partie : Ajax, ou l'art de chuchoter discrètement	XL
Troisième partie : Parler au reste du monde	XL
Des annexes pour le débutant comme pour l'expert	XLI
Aller plus loin...	XLII
À propos des exemples de code	XLII
Remerciements	XLII
Et pour cette deuxième édition...	XLIII

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 clefs : XHTML, CSS, JS, DOM et Ajax	11

Faire la part des choses : Ajax, c'est quoi au juste ?	13
Plan d'action 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	23
Variables déclarées ou non déclarées ?	23
Types de données	25
Fonctions et valeurs disponibles partout	27
<i>Les mystères de parseFloat et parseInt</i>	<i>28</i>
Rappels sur les structures de contrôle	31
<i>Les grands classiques</i>	<i>31</i>
<i>Labélisation de boucles</i>	<i>31</i>
<i>Employer with, mais avec circonspection</i>	<i>33</i>
Opérateurs méconnus	34
<i>Retour rapide sur les grands classiques</i>	<i>34</i>
<i>Opérateurs plus exotiques</i>	<i>35</i>
<i>Comportements particuliers</i>	<i>36</i>
Prise en charge des exceptions	36
<i>Les types d'exceptions prédéfinis</i>	<i>36</i>
<i>Capturer une exception : try/catch</i>	<i>37</i>
<i>Garantir un traitement : finally</i>	<i>39</i>
<i>Lancer sa propre exception : throw</i>	<i>40</i>
Améliorer les objets existants	41
<i>Un peu de théorie sur les langages à prototypes</i>	<i>41</i>
<i>Mise en pratique</i>	<i>42</i>
Arguments des fonctions	46
Le binding des fonctions : mais qui est « this » ?	47
Idiomes intéressants	50
<i>Initialisation et valeur par défaut avec </i>	<i>51</i>

<i>Sélectionner une propriété (donc une méthode) sur condition</i>	52
<i>Tester l'absence d'une propriété dans un objet</i>	52
<i>Fonctions anonymes : jamais new Function !</i>	53
<i>Objets anonymes comme hasbes d'options</i>	53
<i>Simuler des espaces de noms</i>	54
« Unobtrusive JavaScript » : bien associer code JS et page web	55
Astuces pour l'écriture du code	57
Déjouer les pièges classiques	57
Améliorer la lisibilité	59
Pour aller plus loin	61
Livres	61
Sites	62

CHAPITRE 3

Manipuler dynamiquement la page avec le DOM 63

Pourquoi faut-il maîtriser le DOM ?	64
La pierre angulaire des pages vivantes	64
Maîtriser la base pour utiliser les frameworks	64
Comprendre les détails pour pouvoir déboguer	64
Le DOM et ses niveaux 1, 2 et 3	65
Vue d'ensemble des niveaux	65
Support au sein des principaux navigateurs	66
Les aspects du DOM : HTML, noyau, événements, styles...	67
Maîtriser les concepts : document, nœud, élément, texte et collection	67
Le DOM de votre document : une arborescence d'objets	68
Node	70
Document	74
Element	76
Text	77
NodeList et NamedNodeMap	78
DOMImplementation	80
HTMLDocument	81
HTMLDocument	82
Quelques bonnes habitudes	83
Détecer le niveau de DOM disponible	83
Créer les nœuds dans le bon ordre	84
Ne scripter qu'après que le DOM voulu est construit	85
Ne jamais utiliser d'extension propriétaire	87
Utiliser un inspecteur DOM	87
<i>L'inspecteur DOM de Firefox/Mozilla</i>	87
<i>L'inspecteur DOM de Firebug</i>	90

Répondre aux événements	94
Les truands : les attributs d'événement dans HTML	94
La brute : les propriétés d'événement dans le DOM niveau 0	95
Le bon : addEventListener	96
<i>Accommoder MSIE</i>	98
La propagation : capture ou bouillonnement ?	99
<i>Le modèle le plus courant : le bouillonnement</i>	100
<i>La capture, ou comment jouer les censeurs</i>	101
L'objet Event	102
<i>Récupérer l'élément déclencheur</i>	102
<i>Stopper la propagation</i>	102
<i>Annuler le traitement par défaut</i>	103
<i>JavaScript, événements et accessibilité</i>	103
Besoins fréquents et solutions concrètes	105
Décoration automatique de libellés	106
Validation automatique de formulaires	111
Résoudre les écueils classiques	115
MSIE et la gestion événementielle	115
MSIE et le DOM de select/option	116
Les principaux points problématiques	116
Pour aller plus loin	117
Livres	117
Sites	117

CHAPITRE 4

Prototype : simple, pratique, élégant, portable !..... 119

Avant de commencer...	120
Un mot sur les versions	120
Navigateurs pris en charge	121
Vocabulaire et concepts	121
Espaces de noms et modules	121
Itérateurs	122
Élément étendu	122
Alias	123
Comment utiliser Prototype ?	124
Vous allez aimer les dollars	124
La fonction \$ facilite l'accès aux éléments	125
La fonction \$A joue sur plusieurs tableaux	126
La fonction \$H, pour créer un Hash	127
La fonction \$F, des valeurs qui sont les vôtres	128
La fonction \$R et les intervalles	128

La fonction \$\$ et les sélecteurs CSS	129
La fonction \$w joue sur les mots	129
Jouer sur les itérations avec \$break et return	130
Extensions aux objets existants	131
Un Object introspectif	131
Gérer correctement le binding	133
Plus de puissance pour les fonctions	135
De drôles de numéros	139
Prototype en a dans le String	140
<i>Suppression de caractères : strip, stripTags, stripScripts, truncate</i>	141
<i>Transformations : camelize, capitalize, dasherize, escapeHTML, gsub, interpret, sub, underscore, unescapeHTML</i>	142
<i>Fragments de scripts : extractScripts, evalScripts</i>	144
<i>Conversions et extractions : scan, toQueryParams, parseQuery, toArray, inspect, succ</i>	145
<i>Inspection : blank, empty, endsWidth, include, startsWith</i>	146
Des tableaux surpuissants !	147
<i>Conversions : from, inspect</i>	147
<i>Extractions : first, last, indexOf, lastIndexOf</i>	147
<i>Transformations : clear, compact, flatten, without, reverse, reduce, uniq, intersect</i>	148
Modules et objets génériques	151
Enumerable, ce héros	151
<i>L'itération elle-même : each</i>	151
<i>Le contexte d'itération</i>	152
<i>Tests sur le contenu : all, every, any, some, include, member, size</i>	153
<i>Extractions : detect, find, filter, findAll, select, grep, max, min, pluck, reject, partition</i>	154
<i>Transformations et calculs : collect, map, inGroupsOf, inject, invoke, sortBy, zip</i>	156
<i>Conversions : toArray, entries, inspect</i>	159
Il suffit parfois d'un peu de Hash	159
ObjectRange : intervalles d'objets	162
PeriodicalExecuter ne se lasse jamais	162
Vous devriez ré-évaluer vos modèles	163
<i>Utiliser un motif de détection personnalisé</i>	165
Prise en charge de JSON	166
Plus de POO classique : classes et héritage	168
L'héritage et la surcharge de méthode	170
Manipulation d'éléments	171
Element, votre nouveau meilleur ami	171

<i>Element.Methods et les éléments étendus</i>	171
<i>Valeur de retour des méthodes</i>	173
<i>Enfin manipuler librement les attributs : hasAttribute, readAttribute et writeAttribute</i>	173
<i>Élément es-tu là ? hide, show, toggle et visible</i>	174
<i>Gestion du contenu : cleanWhitespace, empty, remove, replace, update, insert, wrap et la syntaxe constructeur</i>	174
<i>Styles et classes : addClassName, getOpacity, getStyle, hasClassName, match, removeClassName, setOpacity, setStyle et toggleClassName</i>	177
<i>Les copains d'abord : ancestors, childElements, descendantOf, descendants, firstDescendant, immediateDescendants, previousSiblings, nextSiblings, siblings</i>	179
<i>Bougez ! adjacent, down, next, previous, select et up</i>	180
<i>Positionnement : absolutize, clonePosition, cumulativeOffset, cumulativeScrollOffset, getDimensions, getHeight, getOffsetParent, getWidth, makePositioned, positionedOffset, relativize, undoPositioned et viewportOffset</i>	182
<i>Défilement et troncature : makeClipping, scrollTo, undoClipping</i>	185
<i>Ajouter des méthodes aux éléments étendus</i>	185
<i>Quelques détails pour finir...</i>	186
La gestion du viewport	187
Selector, l'objet classieux	188
Manipulation de formulaires	190
Field / Form.Element	191
<i>Un mot sur Field.Serializers</i>	192
Form	193
Form.Observer	195
Field.Observer	196
Gestion unifiée des événements	196
Event	197
Observer (ou cesser d'observer) un événement	197
Examiner un événement déclenché	199
<i>Coincer le coupable</i>	199
<i>Étouffer l'affaire</i>	202
<i>Déterminer l'arme du crime</i>	202
Utiliser les événements personnalisés	205
<i>L'événement personnalisé dom:loaded de l'objet document</i>	206
Form.EventObserver	206
Field.EventObserver	207
L'objet global Prototype	207

Pour aller plus loin.....	208
Sites	208
Groupe de discussion	208
Canal IRC	209

CHAPITRE 5

Débugger vos scripts..... 211

Débugger d'abord sur Firefox avec Firebug	212
La console : rapide, efficace, et... très sous-utilisée	213
Débugger des scripts chargés	217
Regarder sous le capot d'Ajax	220
Examiner les performances avec le profileur	221
Peaufiner sur IE6 et IE7	222
L'utopie d'un Windows pour deux	222
Le poids plume : Microsoft Script Debugger	224
<i>Préparer le terrain avec une page à échec garanti</i>	224
<i>Configurer MSIE pour autoriser le débogage</i>	225
<i>Les possibilités du débogueur</i>	227
Le poids lourd : Visual Web Developer 2008 Express Edition	230
Peaufiner sur IE8	236
Peaufiner sur Safari	238
Peaufiner sur Opera	239
Pour aller plus loin	242
Sites	242

DEUXIÈME PARTIE

Ajax, ou l'art de chuchoter243

CHAPITRE 6

Les mains dans le cambouis avec XMLHttpRequest 245

Anatomie d'une conversation Ajax	245
Un petit serveur pour nos tests	246
Installation de Ruby	248
<i>Sous Windows</i>	249
<i>Sous Linux/BSD</i>	250
<i>Sous Mac OS X</i>	251
<i>Un mot sur le cache</i>	251
Un petit serveur HTTP et un code dynamique simple	252

La petite histoire de XMLHttpRequest	254
Origines et historique	254
Bien préparer un échange asynchrone	255
ActiveX versus objet natif JavaScript	255
Créer l'objet requêteur	255
Décrire notre requête	257
Envoyer la requête	258
Recevoir et traiter la réponse	259
Une utilisation complète de notre petit serveur d'exemple	260
Comment surveiller les échanges Ajax de nos pages ?	263
Types de réponse : XHTML, XML, JS, JSON...	265
Bien choisir son type de réponse	266
Une réponse textuelle simple : renvoyer une donnée basique	267
<i>Exemple 1 : sauvegarde automatique</i>	267
<i>Exemple 2 : barre de progression d'un traitement serveur</i>	275
Fragments de page prêts à l'emploi : réponse XHTML	280
Dans la cour des grands : XPath pour traiter des données XML complexes	286
<i>Vite et bien : utilisation de DOM niveau 3 XPath</i>	287
<i>En simulant : utilisation de GoogleAJAXSLT</i>	291
Piloter la page en renvoyant du JavaScript	293
JSON : l'idéal pour des données structurées spécifiques	296

CHAPITRE 7

Ajax tout en souplesse avec Prototype 303

Prototype encore à la rescousse	304
Ajax.Request, c'est tellement plus simple !	304
Plus de détails sur Ajax.Request	306
Quelques mots sur les fonctions de rappel	308
Interpréter la réponse avec Ajax.Response	309
Ajax.Updater : mettre à jour un fragment XHTML, exécuter un script	310
Différencier la mise à jour entre succès et échec	313
Presque magique : Ajax.PeriodicalUpdater	314
Comprendre l'option decay	315
Petits secrets supplémentaires	316
Pour aller plus loin...	318
Livres	318
Sites	318
Groupe de discussion	319
Canal IRC	319

CHAPITRE 8

Une ergonomie de rêve avec script.aculo.us..... 321

Une ergonomie haut de gamme avec script.aculo.us	322
Les effets visuels	322
Les effets noyau	323
<i>Invocation de l'effet</i>	323
<i>Options communes à tous les effets noyau</i>	324
<i>Fonctions de rappel</i>	325
<i>Et si on essayait quelques effets ?</i>	325
<i>Les effets combinés</i>	338
<i>Files d'effets</i>	342
Glisser-déplacer	346
<i>Faire glisser un élément avec Draggable</i>	347
<i>Gérer le dépôt d'un élément avec Droppables</i>	352
Tri de listes par glisser-déplacer	362
<i>Que peut-on trier ?</i>	363
<i>Activer les fonctions d'ordonnancement</i>	363
<i>Désactiver l'ordonnancement</i>	372
<i>Envoyer l'ordre au serveur</i>	372
Complétion automatique de texte	373
<i>Création d'un champ de saisie à complétion automatique</i>	373
<i>Interaction clavier et souris</i>	375
<i>Un premier exemple</i>	376
<i>Personnalisation des contenus renvoyés</i>	382
Et ce n'est pas tout ! Il y a d'autres services	390
Avoir du recul : les cas où Ajax est une mauvaise idée	390
Ajax et l'accessibilité	391
<i>Considérations techniques</i>	391
<i>Considérations ergonomiques</i>	392
Utilisations pertinentes et non pertinentes	393
Pratiques recommandées	394
<i>Principes généraux</i>	394
<i>Ergonomie et attentes de l'utilisateur</i>	395
<i>Cognitif/Lecteurs d'écran</i>	396
Pour aller plus loin...	397
Site	397
Groupe de discussion	397
Canal IRC	397

TROISIÈME PARTIE

Interagir avec le reste du monde 399

CHAPITRE 9

Services web et REST : nous ne sommes plus seuls..... 401

Pourquoi la page ne parlerait-elle qu'à son propre site ?	402
Contraintes de sécurité sur le navigateur	402
Une « couche proxy » sur votre serveur	403
Architecture de nos exemples	404
Comprendre les services web	405
Qu'est-ce qu'une API REST ?	406
Cherchons des livres sur Amazon.fr	407
Obtenir une clef pour utiliser l'API	408
L'appel REST à Amazon.fr	409
<i>Anatomie de la requête</i>	409
<i>Le document XML de réponse</i>	411
Notre formulaire de recherche	411
Passer par Ajax	414
<i>La couche serveur, intermédiaire de téléchargement</i>	414
<i>Intercepter le formulaire</i>	416
De XML à XHTML : la transformation XSLT	418
<i>Notre feuille XSLT</i>	418
<i>Apprendre XSLT à notre page</i>	422
<i>Charger la feuille XSLT au démarrage</i>	422
<i>Effectuer la transformation</i>	423
<i>Embellir le résultat</i>	425
Rhumatismes 2.0 : prévisions météo	430
Préparer le projet	431
Récupérer les prévisions d'un lieu	433
<i>Requête et réponse REST</i>	433
<i>Initialisation de la page et obtention des prévisions</i>	436
<i>Et la feuille XSLT ?</i>	439
<i>Les petites touches finales</i>	443
Rechercher un lieu	447
<i>Préparons le terrain</i>	448
<i>Éblouissez vos amis avec Ajax.XSLTCompleter</i>	451
<i>Brancher les composants ensemble</i>	453
Gérer des images chez Flickr	456
Obtenir une clef API chez Flickr	457
Format général des requêtes et réponses REST chez Flickr	458

Préparer le terrain	459
Chargement centralisé parallèle des feuilles XSLT	461
Obtenir les informations du jeu de photos	463
Récupérer les photos du jeu	471
Afficher une photo et ses informations	477
Pour aller plus loin...	483

CHAPITRE 10

L'information à la carte : flux RSS et Atom 485

Aperçu des formats	486
Une histoire mouvementée	486
<i>RSS 0.9x et 2.0 : les « bébés » de Dave Winer</i>	486
<i>RSS 1.0 : une approche radicalement différente</i>	486
<i>Atom, le fruit de la maturité</i>	487
Informations génériques	488
Le casse-tête du contenu HTML	488
Récupérer et afficher un flux RSS 2.0	489
Format du flux	490
Préparer le terrain	491
<i>La feuille XSLT</i>	493
Chargement et formatage du flux	495
Ajustements des dates et titres	498
Affichage plus avancé et flux Atom 1.0	499
Notre flux Atom	500
Préparation de notre lecteur de flux	502
La feuille XSLT et le formatage	504
<i>Charger la feuille et le flux</i>	506
<i>Afficher dynamiquement les billets complets</i>	509
Les mains dans le cambouis : interpréter le HTML encodé	512
<i>Traiter des quantités massives de HTML encodé</i>	512
<i>Les dates W3DTF</i>	514
Pour aller plus loin...	516
Livres	516
Sites	517

CHAPITRE 11

Mashups et API 100 % JavaScript..... 519

Des cartes avec Google Maps	520
Exigences techniques de l'API	520
Créer une carte	522
Ajouter des contrôles et des marqueurs	524

<i>Des contrôles pour bouger et s'y retrouver</i>	524
<i>Des marqueurs pour identifier des points spécifiques</i>	525
Et tellement d'autres choses...	530
Et si l'on veut d'autres types de cartes ?	530
Des graphiques avec Google Chart	531
Le principe	531
Créer un premier graphique	532
Les grands axes de personnalisation	534
<i>La taille</i>	534
<i>Les données</i>	534
<i>Le type de graphique</i>	539
<i>Les couleurs</i>	541
<i>Mais aussi...</i>	542
Carte ou graphique ? Les cartographies	543
Alternatives suivant les besoins	545
Pour aller plus loin	547
Sites	547

ANNEXE A

Bien baliser votre contenu : XHTML sémantique 551

Les avantages insoupçonnés	552
Pour le site et ses propriétaires	552
Pour le développeur web	553
Règles syntaxiques et sémantiques	553
La DTD et le prologue	555
XHTML, oui mais lequel ?	556
Le balisage sémantique	556
Les balises XHTML 1 Strict par catégorie	557
Balises structurelles	557
Balises sémantiques	558
Balises de liaison	560
Balises de métadonnées	561
Balises de présentation	561
Balises de formulaires et d'interaction	563
Balises dépréciées	564
Attributs incontournables	565
Besoins fréquents, solutions concrètes	566
Un formulaire complexe mais impeccable	566
Un tableau de données à en-têtes groupés	573
Un didacticiel technique	575

Pour aller plus loin.....	578
Livres	578
Sites	578

ANNEXE B

Aspect irréprochable et flexible : CSS 2.1 581

Statut, état et vocabulaire	582
Les versions de CSS	582
Prise en charge actuelle	582
Le jargon CSS	583
Bien comprendre la « cascade » et l'héritage	584
Le sens de la cascade	584
Calcul de la spécificité d'une règle	585
Que se passe-t-il avec la présentation dans HTML ?	586
L'héritage	586
<i>De la valeur spécifiée à la valeur concrète</i>	587
Les modèles de boîte et de mise en forme visuelle	587
Les côtés d'une boîte : ordre et syntaxes courtes	588
Unités absolues et relatives	588
Marge, bordure et espacement	590
Éléments en ligne et de type bloc	591
Éléments remplacés	591
Fusion des marges	591
Le modèle W3C et le modèle Microsoft	592
Tour d'horizon des sélecteurs	593
Groupement	594
Pseudo-éléments	594
Tour d'horizon des propriétés	595
De l'art de réaliser des CSS légères	595
Propriétés du modèle de boîte : marges, espacements et bordures	596
Propriétés de formatage visuel : positionnement, largeur, hauteur, baseline ..	596
Propriétés de contenu généré automatiquement	597
Propriétés de pagination	598
Propriétés de couleurs et d'arrière-plan	599
Propriétés de gestion de la police de caractères	599
<i>Taille de police</i>	600
<i>Famille de polices</i>	600
<i>Tout spécifier d'un coup !</i>	600
Propriétés de gestion du corps du texte	602
<i>L'espacement dans le corps du texte</i>	602
Propriétés des tableaux	603

Propriétés de l'interface utilisateur	603
Pour aller plus loin...	604
Livres	604
Sites	604

ANNEXE C

Le « plus » de l'expert : savoir lire une spécification 605

De Intérêt d'aller chercher l'information à la source	606
Certitude et précision	606
« On m'a dit que là-dessus, c'est toi qui sais tout » : l'expertise	606
Les principaux formats de spécifications web	607
Les recommandations du W3C	607
Les grammaires formelles de langages à balises : DTD et schémas XML ...	607
Les RFC de l'IETF : protocoles et formats d'Internet	608
S'y retrouver dans une recommandation W3C	608
URL et raccourcis	608
Structure générale d'une recommandation	609
Recours à des syntaxes formelles	613
Les descriptions de propriétés CSS	615
Les descriptions de propriétés et méthodes DOM	616
Déchiffrer une DTD	618
Naviguer dans un schéma XML	622
Parcourir une RFC	625
Format général d'une RFC	625
Structure générale d'une RFC	627
Vos spécifications phares	629

ANNEXE D

Développer avec son navigateur web 631

Le cache peut être votre pire ennemi	632
Le rafraîchissement strict	632
Vider le cache	632
Configurer le cache	633
Firefox, favori du développeur grâce aux extensions	634
Firebug	634
Web Developer Toolbar	636
Les trésors du menu Développement caché dans Safari	639
MSIE 6-7 et la Internet Explorer Developer Toolbar	646
MSIE 8b1 et son outil de développement intégré	647
Opera et ses nouveaux outils de développement	650

ANNEXE E

Tour d'horizon des autres frameworks JavaScript	653
jQuery	653
YUI (Yahoo! User Interface)	654
Dōjō	655
Ext JS	656
Base2	657
MooTools	658
Et deux autres, très liés à la couche serveur...	659
GWT	659
ASP.NET AJAX (ex-Atlas)	660
Petit tableau récapitulatif	661
La vraie fausse question de la taille...	661
Index.....	663