

Grille des critères Éco-conception web/les 100 bonnes pratiques

Grille des critères Éco-conception web/les 100 bonnes pratiques - version H.C. 1.0 du 04/09/2015

Page :			
n°	niv.	critère ou test	V.
Conception			
1.1		ARCHITECTURE	
1.1.1	Br.	Choisir les technologies les plus adaptées	
1.1.2	Ar.	Utiliser un framework ou développer sur mesure	
1.1.3	Ar.	Favoriser les pages statiques	
1.1.4	Br.	Utiliser certains forks applicatifs orientés « performance »	
1.2		DESIGN	
1.2.1	Br.	Favoriser un design simple, épuré et adapté au Web	
1.2.2	Ar.	Créer un site « responsive »	
Templating			
2.1		HTML	
2.1.1	Br.	Externaliser les CSS et JavaScript	
2.1.2	Ar.	Valider les pages auprès du W3C	
2.2		CSS	
2.2.1	Br.	Découper les CSS	
2.2.2	Br.	Limiter le nombre de CSS et les compresser	
2.2.3	Br.	Préférer les CSS aux images	
2.2.4	Br.	Minifier les fichiers CSS	
2.2.5	Br.	Écrire des sélecteurs CSS efficaces	
2.2.6	Br.	Grouper les déclarations CSS similaires	
2.2.7	Br.	Utiliser les notations CSS abrégées	
2.2.8	Br.	Toujours fournir une CSS print	
2.2.9	Br.	Utiliser les commentaires conditionnels	
2.2.10	Ar.	Générer des spritesheets CSS	
2.2		POLICES	
2.2.1	Br.	Favoriser les polices standards	
2.2.2	Ar.	Préférer les glyphes aux images	
2.3		IMAGES	
2.3.1	Br.	Optimiser les images	
2.3.2	Br.	Mettre en cache le favicon.ico	
2.3.3	Br.	Supprimer les balises images dont l'attribut SRC est vide	
Code client			
3.1		GÉNÉRAL	
3.1.1	Br.	Limiter le recours aux plug-ins	
3.1.2	Br.	Éviter d'utiliser eval()	
3.1.3	Br.	Préférer les fonctions aux strings, en argument à setTimeout() et setInterval()	
3.1.4	Ar.	Limiter l'utilisation de setTimeout()	
3.1.5	Ar.	Respecter le principe de navigation rapide dans l'historique	
3.1.6	Ar.	Éviter les boucles for...in	
3.1.7	Or	Éviter d'utiliser try...catch...finally	
3.1.8	Or	Utiliser les opérations primitives	
3.2		JAVASCRIPT	
3.2.1	Br.	Mettre en cache les objets souvent accédés en JavaScript	
3.2.2	Br.	Modifier plusieurs propriétés CSS en une seule fois	
3.2.3	Br.	Valider le code JavaScript avec JSLint	
3.2.4	Ar.	Privilégier les changements visuels instantanés	
3.2.5	Or	Privilégier les variables locales	
3.2.6	Or	Privilégier les fonctions anonymes	
3.2.7	Or	Utiliser le concaténateur de chaînes de façon optimale	
3.2.8	Or	Utiliser Ajax pour les zones de contenu souvent mises à jour	
3.3		DOM	
3.3.1	Br.	Utiliser la délégation d'événements	
3.3.2	Br.	Réduire au maximum le repaint (appearance) et le reflow (layout)	
3.3.3	Br.	Ne pas modifier le DOM lorsqu'on le traverse	
3.3.4	Ar.	Rendre les éléments du DOM invisibles lors de leur modification	
Code serveur			
4.1		GÉNÉRAL	
4.1.1	Br.	Éviter les redirections	
4.1.2	Br.	Ne pas générer de page 404	
4.1.3	Br.	Mettre en cache les données calculées souvent utilisées	
4.1.4	Br.	Utiliser des variables statiques	
4.1.5	Br.	Libérer de la mémoire les variables qui ne sont plus nécessaires	
4.1.6	Br.	Ne pas appeler de fonction dans la déclaration d'une boucle de type for	
4.1.7	Br.	Supprimer tous les warnings et toutes les notices	
4.1.8	Ar.	Utiliser la méthode GET pour les requêtes Ajax	
4.1.9	Or	Ne pas assigner inutilement de valeurs aux variables	
4.2		SQL	
4.2.1	Br.	Éviter d'effectuer des requêtes SQL à l'intérieur d'une boucle	
4.2.2	Br.	Ne se connecter à une base de données que si nécessaire	
4.2.3	Br.	Ne jamais écrire de SELECT * FROM	
4.2.4	Br.	Limiter le nombre de résultats	

4.2.5	Br.	Utiliser les procédures stockées
4.3		CMS
4.3.1	Br.	Utiliser tous les niveaux de cache du CMS
4.4		HTML
4.4.1	Or	Utiliser un moteur de templating
4.5		PHP
4.5.1	Or	Utiliser la simple quote (') au lieu du guillemet (")
4.5.2	Or	Remplacer les \$i++ par des ++\$i
4.5.3	Br.	Mettre en cache le bytecode
4.6		COOKIES
4.6.1	Or	Optimiser la taille des cookies
Hébergement		
5.1		INFRASTRUCTURE
5.1.1	Br.	Choisir un hébergeur « vert »
5.1.2	Br.	Utiliser une électricité « verte »
5.1.3	Br.	Optimiser l'efficacité énergétique des serveurs
5.1.4	Br.	Privilégier les serveurs équipés de mémoires SSD
5.1.5	Or	Utiliser des serveurs virtualisés
5.1.6	Or	Stocker les données dans le cloud
5.2		ARCHITECTURE
5.2.1	Br.	Utiliser un reverse proxy
5.2.2	Br.	Héberger les ressources sur un domaine sans cookies
5.2.3	Br.	Installer uniquement les services indispensables sur le serveur
5.2.4	Ar.	Utiliser un CDN
5.2.5	Ar.	Utiliser un serveur asynchrone
5.3		PARAMÉTRAGE
5.3.1	Br.	Ajouter des en-têtes Expires ou Cache-Control
5.3.2	Br.	Compresser la sortie HTML
5.3.3	Br.	Utiliser les ETags
5.3.4	Br.	Désactiver la directive Allowoverride d'Apache
5.3.5	Ar.	Désactiver certains logs d'accès du serveur web
5.3.6	Ar.	Désactiver les logs binaires de MySQL ou MariaDB
5.3.7	Ar.	Désactiver le DNSLookup d'Apache
5.4		OPTIMISATION
5.4.1	Br.	Monter les caches entièrement en RAM
5.4.2	Br.	Compresser les feuilles de style CSS et les bibliothèques JavaScript
5.4.3	Br.	Minifier les fichiers JavaScript
5.4.4	Br.	Combiner les fichiers CSS et les fichiers JavaScript
5.4.5	Ar.	Limiter le recours aux certificats SSL
5.4.6	Or	Mettre en cache les réponses Ajax
Contenu		
6.1		VIDÉOS
6.1.1	Br.	Adapter les vidéos aux contextes de visualisation
6.2		IMAGES
6.2.1	Br.	Redimensionner les images en dehors du code HTML
6.2.2	Br.	Redimensionner les images en dehors du CMS
6.3		SONS
6.3.1	Br.	Adapter les sons aux contextes d'écoute
6.3.2	Br.	Encoder les sons en dehors du CMS
6.4		DOCUMENTS
6.4.1	Br.	Compresser les documents
6.4.2	Br.	Optimiser les PDF
6.4.3	Br.	Générer les PDF en dehors du CMS
6.5		TEXTES
6.5.1	Ar.	Adapter les textes au Web
6.6		ANIMATIONS
6.6.1	Ar.	Limiter l'utilisation de Flash
6.7		E-MAILS
6.7.1	Br.	Dédoublonner les fichiers d'adresses e-mail avant envoi
6.7.2	Br.	N'utiliser que des adresses e-mail double opt-in
6.7.3	Br.	Préférer le texte brut au HTML

Organisation du contenu, mise en forme et support électronique du document présent : Copyright © 2015 Hervé CHUZEVILLE - Ce document est mis à disposition selon le Contrat "Attribution - Partage dans les Mêmes Conditions 3.0 France (CC BY-SA 3.0 FR)" disponible en ligne <http://creativecommons.org/licenses/by-sa/3.0/fr/> ou par courrier postal à Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
Document basé sur le livre "Eco-conception web : les 100 bonnes pratiques" (Copyright © 2013 GreenIT.fr / Frédéric Bordage / Stéphane Bordage / Jérémy Chatard / Eyrolles. La liste des critères d'éco-conception web est la propriété de GreenIT.fr / Frédéric Bordage).