

## À propos

Le projet Opquast (Open Quality Standards) fait avancer le Web à travers des check-lists, des outils d'évaluation, une reconnaissance. Opquast élabore et diffuse des référentiels publics de bonnes pratiques pour la qualité Web. La première version de la liste "écoconception Web" mise à votre disposition dans ce

## Crédits et remerciements

Cette checklist est un extrait du référentiel publié par Frédéric Bordage / GreenIT.fr chez Eyrolles sous le titre "écoconception web : les 100 bonnes pratiques" en octobre 2012, puis sous le titre "écoconception web : les 115 bonnes pratiques, 2ème édition" en septembre 2015 (voir : [www.ecoconceptionweb.com](http://www.ecoconceptionweb.com)). Nous remercions chaleureusement l'ensemble des contributeurs qui ont participé à la mise au point de ce référentiel et le font évoluer depuis plusieurs années, dans le désordre :

Christian Meixenberger (Banque Cantonale de Fribourg), Christian Marchand (Green IT Consulting), Stéphane Bordage (Breek), Jérémy Chatard (Breek), Frédéric Lohier (GreenIT.fr), l'Agence pour le Développement Et la Maîtrise de l'Energie (ADEME), l'Alliance Green IT (AGIT), l'Association Française Des Editeurs de Logiciel (AFDEL), le CIGREF réseau de Grandes Entreprises, EcoInfo (CNRS), agenceNTIC, Agence Régionale de Développement de l'Innovation et de l'Economie (ARDIE) Bourgogne, UCM, IT-CE – groupe BPCE , La Poste, Solocal Group, Jacques Bourdos (Renault), Frédéric Cerbelaud (SNCF), Thierry Vonck (SNCF), Jean-Christophe Chaussat (Pôle Emploi), Philippe Derouette (IT-CE - groupe BPCE), Blueight, D2SI, Émeraude Créative, European Service Network (ESN), Logomotion, Neoma Interactive, Neutreo, Nxtweb, Sébastien Delorme (Atalan), Nicolas Hoizey (Clever Age), Armel Fauveau (Globalis Media Systems), Christophe Clouzeau (Neoma Interactive), Elie Sloïm (Opquast), Paul Souche (Sfeir), Aurélien Levy (Temesis), Luc Poupard (W3Qualite.net), Thomas Broyer (Atol CD).

La liste des contributeurs est remise à jour au fil de l'eau.

## Licence

Cette checklist est publiée sous licence Creative Commons BY-NC. Cela signifie que vous pouvez l'utiliser librement pour des utilisations non commerciales, à la condition de maintenir la paternité du contenu, via un lien vers <http://checklists.opquast.com/ecoconception/> et [www.ecoconceptionweb.com](http://www.ecoconceptionweb.com). Opquast est une marque française déposée à l'INPI. Le contenu du référentiel publié chez Eyrolles est protégé par le droit d'auteur

<http://creativecommons.org/licenses/by-nc/4.0/>

## Bonnes pratiques écoconception Web : version 1.0 – 2015

N°	Rubrique	Libellé
1	Conception fonctionnelle	Éliminer les fonctionnalités non essentielles
2	Conception fonctionnelle	Quantifier précisément le besoin
3	Conception fonctionnelle	Fluidifier le processus
4	Ergonomie	Préférer la saisie assistée à l'autocomplétion
5	Ergonomie	Respecter le principe de navigation rapide dans l'historique
6	Ergonomie	Favoriser un design simple, épuré, adapté au web
7	Ergonomie	Créer un site responsive
8	Conception fonctionnelle	Proposer un traitement asynchrone lorsque c'est possible
9	Conception technique	Limiter le nombre de requêtes HTTP
10	Conception technique	Stocker les données statiques localement
11	Conception technique	Choisir les technologies les plus adaptées
12	Conception technique	Utiliser un framework ou développer sur mesure
13	Conception technique	Limiter le recours aux plugins
14	Conception technique	Limiter l'utilisation du Flash
15	Développement	Valider les pages auprès du W3C
16	Développement	Externaliser les CSS et JavaScript
17	Conception graphique / IHM	Favoriser les polices standard
18	Conception graphique / IHM	Préférer les glyphs aux images
19	Conception graphique / IHM	Supprimer les balises images dont l'attribut SRC est vide
20	Conception graphique / IHM	Redimensionner les images en dehors du HTML
21	Conception graphique / IHM	Éviter d'utiliser des images bitmap pour l'interface
22	Conception graphique / IHM	Optimiser les images vectorielles
23	Conception graphique > CSS	Générer des spritesheets CSS
24	Conception graphique > CSS	Découper les CSS
25	Conception graphique > CSS	Limiter le nombre de CSS
26	Conception graphique > CSS	Préférer les CSS aux images
27	Conception graphique > CSS	Écrire des sélecteurs CSS efficaces
28	Conception graphique > CSS	Grouper les déclarations CSS similaires
29	Conception graphique > CSS	Utiliser les notations CSS abrégées
30	Conception graphique > CSS	Fournir une CSS print
31	Conception graphique / IHM	Utiliser les commentaires conditionnels
32	Conception graphique > CSS	Modifier plusieurs propriétés CSS en 1 seule fois
33	Développement > Javascript	Valider le code JavaScript
34	Développement > Javascript	Éviter d'utiliser try-catch-finally
35	Développement > Javascript	Utiliser les opérations primitives
36	Développement > Javascript	Mettre en cache les objets souvent accédés en JavaScript
37	Développement > Javascript	Privilégier les variables locales
38	Développement > Javascript	Privilégier les fonctions inline
39	Développement > Javascript	Utiliser le concatéateur de chaînes de façon optimale
40	Développement > Javascript	Donner des fonctions en paramètre à setTimeout() et setInterval() plutôt que des string
41	Développement > Javascript	Éviter les for - in
42	Développement > Javascript	Réduire les accès au DOM via JavaScript
43	Développement > Javascript	Ne pas faire de modification du DOM lorsqu'on le traverse
44	Développement > Javascript	Rendre les éléments du DOM invisibles lors de leur modification
45	Développement > Javascript	Réduire au maximum le repaint (appearance) et le reflow (layout)
46	Développement > Javascript	Utiliser la délégation d'événements
47	Développement > Javascript	Privilégier les changements visuels instantanés
48	Développement > Javascript	Éviter les animations Javascript / CSS coûteuses
49	Développement > Javascript	Utiliser Ajax pour certaines zones de contenu
50	Développement > Javascript	Utiliser la méthode GET pour les requêtes Ajax

## Bonnes pratiques écoconception Web : version 1.0 – 2015

N°	Rubrique	Libellé
51	Conception technique > Serveur	Favoriser les pages statiques
52	Conception technique > Serveur	Créer une architecture applicative modulaire
53	Conception technique > Serveur	Utiliser certains forks applicatifs orientés "performance"
54	Conception technique > Serveur	Choisir un format de données adapté
55	Conception technique > Serveur	Limiter le nombre de domaine servant les ressources
56	Conception technique > Serveur	Utiliser un système de templating (Smarty, Twig)
57	Conception technique > Serveur	Utiliser tous les niveaux de cache du CMS
58	Conception technique > Serveur	Générer les PDF en dehors du CMS
59	Conception technique > Serveur	Redimensionner les images en dehors du CMS
60	Conception technique > Serveur	Encoder les sons en dehors du CMS
61	Conception technique > Serveur	Mettre en cache le Bytecode
62	Conception technique > Serveur	Mettre en cache les données calculées souvent utilisées
63	Développement > Serveur > Serveur d'applications	Libérer de la mémoire les variables qui ne sont plus nécessaires
64	Développement > Serveur > Serveur d'applications	Ne pas appeler de fonction dans la déclaration d'une boucle de type for
65	Développement > Serveur > Serveur d'applications	Supprimer tous les warning et toutes les notices
66	Développement > Serveur > Serveur d'applications	Utiliser des variables statiques
67	Développement > Serveur > Serveur d'applications	Eviter la ré-écriture des getter / setter natifs
68	Développement > Serveur > Serveur d'applications	Ne pas assigner inutilement de valeur aux variables
69	Développement > Serveur > Serveur d'applications	Utiliser la simple côte (') au lieu du guillemet (")
70	Développement > Serveur > Serveur d'applications	Remplacer les \$i++ par ++\$i
71	Développement > Serveur > SQL	Eviter d'effectuer des requêtes SQL à l'intérieur d'une boucle
72	Développement > Serveur > SQL	Ne se connecter à une base de données que si nécessaire
73	Développement > Serveur > SQL	Ne jamais faire de SELECT * FROM
74	Développement > Serveur > SQL	Limiter le nombre de résultats (clause LIMIT)
75	Développement > Serveur > SQL	Utiliser les procédures stockées
76	Hébergement > Optimisation post-production	Optimiser les images bitmap
77	Hébergement > Optimisation post-production	N'utilisez que les portions indispensables des librairies Javascript et frameworks CSS
78	Hébergement > Optimisation post-production	Minifier les fichiers CSS
79	Hébergement > Optimisation post-production	Minifier les fichiers JavaScript
80	Hébergement > Optimisation post-production	Compresser les librairies CSS et Javascript
81	Hébergement > Optimisation post-production	Combiner les fichiers CSS et JavaScript

## Bonnes pratiques écoconception Web : version 1.0 – 2015

N°	Rubrique	Libellé
82	Hébergement > Optimisation post-production	Optimiser la taille des cookies
83	Hébergement > Optimisation en production	Compresser la sortie HTML
84	Conception technique > Infrastructure	Choisir un hébergeur "vert"
85	Conception technique > Infrastructure	Utiliser une électricité "verte"
86	Conception technique > Infrastructure	Adapter la qualité de service et le niveau de disponibilité
87	Conception technique > Infrastructure	Utiliser des serveurs virtualisés
88	Conception technique > Infrastructure	Optimiser l'efficacité énergétique des serveurs
89	Conception technique > Infrastructure	Installer le minimum requis sur le serveur
90	Conception technique > Infrastructure	Mettre les caches entièrement en RAM (opcode et kvs)
91	Conception technique > Infrastructure	Privilégier les serveurs équipés de mémoires SSD
92	Conception technique > Infrastructure	Stocker les données dans le cloud
93	Hébergement > Optimisation en production	Désactiver les logs binaires de MySQL / MariaDb
94	Hébergement > Optimisation en production	Utiliser un serveur asynchrone
95	Hébergement > Optimisation en production	Limiter le recours au SSL
96	Conception technique > Infrastructure	Héberger les ressources (CSS/JS) sur un domaine sans cookie
97	Conception technique > Infrastructure	Eviter les redirections
98	Conception technique > Infrastructure	Ne pas générer de page 404
99	Hébergement > Optimisation en production	Désactiver certains logs d'accès du serveur web
100	Hébergement > Optimisation en production	Désactiver le DNS lookup d'Apache
101	Hébergement > Optimisation en production	Apache Vhost : désactiver le AllowOverride
102	Hébergement > Optimisation en production	Utiliser un CDN
103	Hébergement > Optimisation en production	Utiliser un reverse proxy
104	Hébergement > Optimisation en production	Mettre en cache le favicon.ico
105	Hébergement > Optimisation en production	Ajouter des entêtes Expires ou Cache-Control
106	Hébergement > Optimisation en production	Configurer les ETags
107	Hébergement > Optimisation en production	Mettre en cache les réponses Ajax
108	Contenu	Compresser les documents
109	Contenu	Optimiser les PDF
110	Contenu	Dédoublonner systématiquement les fichiers avant envoi
111	Contenu	N'utiliser que des fichiers double opt-in
112	Contenu	Préférer le texte brut au HTML
113	Contenu	Adapter les sons aux contextes d'écoute
114	Contenu	Adapter les textes au web
115	Contenu	Adapter les vidéos aux contextes de visualisation