Bonnes pratiques et conseils :

Le site respecte les bonnes pratiques suivantes

**1 - Volume de données**

8 images sont chargées **100/100**

Ne chargez en priorité que les images nécessaires au bon fonctionnement de la page.

**Page web et images**Les images représentent un enjeu majeur pour les pages web. Elles sont trop souvent mal optimisées et affectent fortement le poids total de la page.

Il est donc nécessaire de charger uniquement les images en priorité. Le chargement des autres images doit être différé.

**Comment différer les images non prioritaires ?**Pour différer le chargement de contenu afin de ne pas augmenter le temps total de chargement de la page, vous pouvez par exemple vous renseigner sur le "lazy-loading".

Nous vous conseillons de **charger moins de 70 images** pour respecter cette bonne pratique.

Vos images sont diffusées à la bonne échelle **100/100**

Une page récupérant une image plus grande que celle réellement affichée charge inutilement un volume de données trop important.

**Redimensionnement d'images**L'utilisation des attributs width et height dans les balises img pour réduire la taille d'affichage d'une image n'est pas une alternative recommandée.

Fournissez directement vos images aux dimensions d'affichage utilisées sur votre site. Vous évitez ainsi l'envoi de données inutiles sur le réseau, ce qui diminue le temps de chargement de la page.

**Responsive design ? Images Retina ?**Le responsive design ou encore l'utilisation d'images retina ne jutifient pas un redimensionnement d'images. Même dans ces cas, différentes méthodes existent afin de délivrer vos images à la bonne taille. Nous vous conseillons de lire les ressources suivantes :

* [Introduction sur les images responsive](http://www.alsacreations.com/article/lire/1559-responsive-web-design-present-futur-adaptation-mobile.html)
* [Picturefill, pour supporter l'élément <picture> dès maintenant (EN)](http://scottjehl.github.io/picturefill/)
* [RICG, groupe de travail sur les images responsive (EN)](https://responsiveimages.org/)

Félicitations ! Aucun redimensionnement d'image n'a été détecté.

3 images utilisent le format PNG de façon adaptée **100/100**

Le choix d'un format adapté pour une image permet de réduire son poids.

**Le format PNG**Le format d'images PNG est destiné aux images nécessitant la fonctionnalité de transparence, ou encore aux petites images, disposant de peu de détails et de couleurs.

**Préférez le format JPEG...**Le principal défaut du format PNG est de ne pas supporter la perte de qualité. En effet, un format tel que JPEG propose de "dégrader" la qualité de l'image sans que cela ne soit perceptible par l'utilisateur. Vous pouvez ainsi diminuer la qualité de l'image d'[environ 25%](https://github.com/rflynn/imgmin) sans que vos utilisateurs ne s'en rendent compte.

Une image avec un poids important sera donc sans doute mieux compressée en utilisant le format JPEG.

**...ou PNG-8**Dans le cas ou votre image nécessite obligatoirement d'utiliser le mécanisme de transparence, non supporté par le format JPEG, vous devriez convertir votre image PNG "classique", en PNG-8. Ce format, basé sur une palette de 256 couleurs maximum permet de diminuer le poids de l'image sans altérer significativement sa qualité en général. Des outils tels [pngquant](http://pngquant.org/) ou encore [tinypng](https://tinypng.com/) vous aideront dans cette démarche.

Enfin, si la qualité proposée par le format PNG-8 n'est pas satisfaisante pour votre image, vous pouvez vous renseigner sur [les techniques avancées possibles (EN)](http://kaioa.com/node/104) pour obtenir ce comportement sans délivrer une image au format PNG. Par exemple, il est possible de séparer votre image en 2 images JPEG, l'une contenant les données de transparences et l'autre contenant les données liées aux couleurs, et de regrouper l'image côté client à l'aide d'un élément CANVAS.

Pour en savoir plus, n'hésitez pas à consulter [cet article sur la compression des images (EN)](http://www.html5rocks.com/en/tutorials/speed/img-compression/).

Le choix du format PNG sur les images de cette page n'a pas d'impact négatif.

Cette page n'envoie pas de cookies > 100 ko **100/100**

N'utilisez pas de cookies trop longs afin de minimiser leur impact sur le temps de chargement.

Les cookies HTTP sont utilisés pour le suivi d'un utilisateur pour personnaliser sa visite. Les cookies HTTP sont envoyés via des en-têtes HTTP par le serveur Web au navigateur. Ensuite, à chaque accès à la page par le navigateur, le cookie reçu lors de la première requête est renvoyé au serveur. Le client est alors reconnu. [En savoir plus](http://developer.yahoo.com/performance/rules.html#cookie_size).

Cette page n'envoie pas trop de cookies **100/100**

N'utilisez pas trop de cookies pour que leur impact sur le temps de chargement soit le plus faible possible.

Les cookies HTTP sont utilisés pour le suivi d'un utilisateur pour personnaliser sa visite. Les cookies HTTP sont envoyés via des en-têtes HTTP par le serveur Web au navigateur. Ensuite, à chaque accès à la page par le navigateur, le cookie reçu lors de la première requête est renvoyé au serveur. Le client est alors reconnu. [En savoir plus](http://developer.yahoo.com/performance/rules.html#cookie_size).

1 image utilise le format GIF de façon adaptée **100/100**

Le choix d'un format adapté pour une image permet de réduire son poids.

**Le format GIF**Le format d'images GIF est destiné aux images animées. Si vous n'utilisez pas d'animation, les formats PNG ou JPEG seront plus adaptés, et moins lourds une fois optimisés.

Le choix du format GIF sur les images de cette page n'a pas d'impact négatif.

Les réponses de redirection sont légères **100/100**

Le contenu d'une redirection ne sera en aucun cas utilisé par les navigateurs web. C'est donc poids inutile qu'il convient d'éliminer.

**Les redirections sur le web**Les redirections permettent d'indiquer que la ressource demandée est disponible à une autre URL. Le navigateur web va suivre automatiquement la redirection.

Les redirections peuvent être temporaire (code HTTP 302) ou permanente (301).

Ce mécanisme est par exemple utilisé sur des domaines secondaires (.fr vers .com) ou encore des mécanismes de détection de langue.

**Que faire ?**Tout d'abord s'assurer que la redirection est inévitable. Si c'est le cas il est nécessaire d'alléger, voire de supprimer totalement le contenu des réponses serveur qui correspondent à des redirections.

Une redirection doit idéalement être < 1ko.

Aucune de vos redirections ne dispose d'un poids trop important.

Les réponses en erreur sont légères **100/100**

Le contenu d'une réponse en erreur ne sera en aucun cas utilisé par les navigateurs web. C'est donc poids inutile qu'il convient d'éliminer.

**Les réponses d'erreur sur le web**Les ressources ne pouvant être accédées par le navigateur sont caractérisées par un code HTTP de retour compris entre 400 et 599.

**Que faire ?**Tout d'abord, il faut éviter à tout prix de faire face à une erreur. Comme cela reste possible, il est nécessaire d'alléger, voire de supprimer totalement le contenu des réponses serveur qui correspondent à des erreurs.

Une réponse en erreur doit idéalement faire moins de 10ko.

Aucune des réponses en erreur ne dispose d'un poids trop important.

Vos contenus sont bien appelés par des URLs uniques **100/100**

Si un même contenu est référencé plusieurs fois sur la page, il est préférable d'utiliser une seule URL. Cela permet d'éviter de télécharger plusieurs fois ce contenu.

**Servir le même contenu via plusieurs URLs**Si vous utilisez différentes URLs pour servir le même contenu, les navigateurs de vos visiteurs téléchargeront le contenu plusieurs fois. Ce qui implique des requêtes inutiles, du contenu dupliqué et donc un temps de chargement de votre page plus long. Si vous utilisez une seule URL, les navigateurs demanderont une seule fois le contenu au serveur. Il est également conseillé d'utiliser une seule URL pour un même contenu sur les différentes pages de votre site pour bénéficier du mécanisme de cache.

**Comment résoudre le problème ?**Utilisez une seule URL pour un même contenu. Si vous avez besoin d'un contenu identique plusieurs fois, assurez-vous qu'il est toujours référencé par la même URL.

Vous n'avez pas besoin de minifier vos ressources HTML **100/100**

En compressant votre code HTML (y compris le code JavaScript et CSS intégré), vous pouvez libérer de nombreux octets de données et réduire les délais de téléchargement et d'analyse.

[Réduisez la taille des ressources HTML](https://developers.google.com/speed/docs/insights/MinifyResources) suivantes afin de gagner 2.9 Ko (réduction de 11%).

* Une réduction de la taille de [www.clinique-esthe[...]r/v1/](http://www.clinique-esthetique.fr/v1/) pourrait libérer 2.9 Ko (réduction de 11%).

**Vous obtenez 100/100 alors que vos ressources HTML ne sont pas minifiées ici ?** Nous avons considéré que les gains apportés par la minification étaient trop faibles pour être remontés dans le rapport.

Aucun cookie n'est envoyé sur des ressources statiques **100/100**

Vous envoyez certaines données inutilement sur le réseau, à travers des cookies sur des ressources statiques.

**Cookies et ressources statiques**Les cookies sont utilisés pour personnaliser le contenu d'une page selon le profil de l'internaute. Il est inutile d'en envoyer sur des ressources renvoyant tout le temps le même contenu (ressources statiques).

Les requêtes vers des images, des ressources JavaScript ou encore sur des styles CSS ne doivent donc pas faire appel à des cookies.

**Que faire ?**N'utilisez plus de cookies sur les requêtes vers des ressources statiques. Cependant, certains noms de domaines nécessitent l'envoi de cookies sur toutes les ressources. Si possible, faites appel à d'autres domaines n'appliquant pas cette restriction. Vous pouvez par exemple configurer un nom de domaine spécifique pour servir vos ressources statiques : static.mondomaine.com

**2 - Politique de cache**

Cette page ne contient pas de scripts inline trop longs **100/100**

Vous devriez permettre la mise en cache de tout script ayant une taille non négligeable, afin de réduire le temps de chargement de vos pages.

**Scripts inline / mise en cache**Les scripts dits "inline" permettent d'intégrer facilement de petites portions de scripts directement dans le code HTML. Exemple :

|  |
| --- |
| <script type="text/javascript"> (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']...,'/analytics.js','ga'); ga('create', 'UA-11111111-1', 'dareboost.com');</script> |

En procédant ainsi, vous évitez d'effectuer une requête vers le serveur pour récupérer la ressource. Les scripts inline représentent alors un gain de performance pour intégrer de petits scripts.

Dès lors qu'un script dispose d'une taille assez conséquente, nous vous conseillons de l'externaliser et d'effectuer une requête pour le récupérer. Cela vous permettra de bénéficier des mécanismes de cache sur la ressource.

**Que faire ?**Externalisez vos scripts de plus de 1 500 caractères dans un ou des fichiers distincts.

Vous n'avez pas de script inline trop long sur cette page.

La durée du cache est supérieure à 48h **100/100**

En permettant la conservation de vos ressources statiques plusieurs jours en cache, vous diminuez la charge de votre serveur.

Certaines de vos ressources utilisent l'en-tête HTTP Expires pour disposer d'une politique de cache efficace, ce qui correspond à une bonne pratique. Cependant, vous devriez envisager d'améliorer sa configuration pour profiter au mieux des mécanismes de cache.

**Quelle configuration adopter ?**Nous vous conseillons de paramétrer l'en-tête Expires afin que la date mentionnée soit comprise entre 2 jours et 1 an par rapport à aujourd'hui.

Vos ressources sont mises en cache durant au moins 2 jours grâce à l'en-tête 'Expires'.

**3 – Compatibilité**

Le plugin Silverlight n'est pas utilisé **100/100**

Le plugin Silverlight est considéré comme obsolète en 2015, et n’apporte rien par rapport à ce qui peut être fait nativement avec le HTML5. Son utilisation vous expose à des soucis de compatibilité et peut renvoyer des signaux négatifs à vos utilisateurs (exemple : le navigateur qui indique qu’un contenu a été bloqué car susceptible d’être dangereux).

Félicitations, cette page ne nécessite pas le plugin Silverlight.

Aucune applet Java détectée **100/100**

Les applets Java sont considérées comme obsolètes en 2015, et n’apportent rien par rapport à ce qui peut être fait nativement avec le HTML5. Leur utilisation vous expose à des soucis de compatibilité et peut renvoyer des signaux négatifs à vos utilisateurs (exemple : le navigateur qui indique qu’un contenu a été bloqué car susceptible d’être dangereux).

Félicitations, cette page ne contient pas d'applet Java.

N'utilisez pas <bgsound> pour mettre un fond sonore **100/100**

Aucune balise bgsound detecté. Il s'agit d'une bonne pratique: cette balise n'est pas un standard HTML.

Cet élément ne fait pas partie des standards HTML. Son utilisation peut entraîner des problèmes de compatibilité. [Voir plus d'informations](https://developer.mozilla.org/fr/docs/Web/HTML/Element/bgsound).

Utilisez la balise [audio](https://developer.mozilla.org/fr/docs/Web/HTML/Element/audio) pour fournir du contenu audio sur votre page.

|  |
| --- |
| <audio src="my-audio-file.ogg" autoplay> Your browser doesn't support the audio element.</audio> |

Aucun lien vers un document Word détecté **100/100**

Les documents *.doc* et *.docx* ne garantissent pas la compatibilité avec les principaux systèmes d'exploitation. Il est recommandé d'utiliser des documents PDF.

Cette page contient uniquement des liens standards.

Le format de fonte EOT correctement déclaré **100/100**

Assurez-vous que le format EOT sera bien utilisé pour les internautes utilisant Internet Explorer <= 8.

**Indiquer le format EOT pour Internet Explorer <= 8**Les différents navigateurs web (Firefox, Chrome, Safari, Internet Explorer, Opera...) ne supportent pas tous les mêmes formats de fontes. Votre objectif est d'utiliser le format le plus léger possible pour chaque plateforme. Lorsque vous utilisez une fonte, spécifiez ainsi plusieurs formats, via l'instruction CSS @font-face : le navigateur utilisera le premier format compatible rencontré.

**Cela est vrai pour tous les navigateurs sauf Internet Explorer <= 8, qui n'interprète pas bien l'instruction @font-face** :
seul le premier format indiqué sera pris en compte par ces navigateurs. Or, le format EOT est le plus léger pour ces navigateurs.

**Que faire ?**Seul Internet Explorer supporte le format EOT. Indiquez le donc en premier au sein de la propriété src de @font-face : Internet Explorer sera capable de comprendre l'instruction sous cette forme, et les autres navigateurs, qui ne supportent pas EOT, utiliseront un autre format. Nous avons rédigé un article sur le sujet afin de vous aider à[déterminer le bon ordre à choisir pour vos fontes de caractères](http://blog.dareboost.com/fr/2015/08/comment-utiliser-fontes/#meilleur_format). Plus succinctement, **l'ordre à respecter pour vos formats est le suivant : EOT, WOFF2, WOFF, TTF, puis SVG**.

Toutes vos fontes au format EOT sont bien placées en première position dans l'instruction @font-face.

Aucun attribut déprécié n'est utilisé dans la balise <body> **100/100**

Certains attributs de présentation sont dépréciés en HTML 5 : alink / background / bgcolor / link / text / vlink

Félicitations, nous n'avons pas trouvé d'attributs dépréciés dans la balise body.

Préférez l'utilisation du CSS à la place.

Aucune expression CSS détectée **100/100**

Les expressions CSS sont dépréciées et vous ne devez pas les utiliser car elles sont très gourmandes en termes de ressources. [En savoir plus](http://developer.yahoo.com/performance/rules.html#css_expressions).

Félicitations. Vous n'utilisez pas d'expressions CSS.

Aucun BOM (Byte Order Mark) détecté **100/100**

Certains robots/analyseurs de textes ne sont pas capables d'interpréter une page contenant un BOM.

**Qu'est ce que le BOM ?**Le BOM est un caractère invisible placé en début de page, dont le but est d'aider à déterminer quel encodage est utilisé par la page. Or, les bonnes pratiques du web encouragent l'utilisation de l'en-tête HTTP Content-Type pour définir l'encodage utilisé par la page. Le BOM ne trouve donc pas son intérêt dans ce contexte.

Outre le fait que ce type d'indicateur est inutile sur le web, il peut impliquer des problèmes. C'est par exemple le cas pour la validation W3C, qui va tenter d'interpréter le premier caractère, qui correspond au BOM. Le document ne sera alors pas valide.

Aucune ressource n'utilise un indicateur d'ordre des octets (BOM). Il s'agit d'une bonne pratique.

**4 - Optimisation du rendu**

Vous interprétez et exécutez vos scripts une seule fois **100/100**

Une librairie ou un script externe sont le plus souvent destinés à être appelé une seule fois par page. Cependant l'utilisation de widgets par exemple peut conduire à plusieurs exécutions inutiles si aucune précaution n'est apportée.

**Duplication de scripts**Il est de plus en plus courant de voir des scripts utilisés plusieurs fois au sein d'une même page. Le cas le plus courant est l'intégration de widgets de réseaux sociaux, qu'il est parfois utile de retrouver à plusieurs endroits sur la page pour améliorer son taux d'engagement par exemple. Il ne s'agit pas d'une mauvaise pratique en soit, il faut cependant veiller à certains critères afin de ne pas voir le temps de chargement de la page s'allonger.

**Que se passe-t-il si un script est inclu 2 fois dans le code ? Combien de fois est-il chargé ? Interprété ? Exécuté ?**La plupart des navigateurs web récents ne téléchargent qu'une seule fois un script inclu 2 fois. Une exception subsiste : Firefox, qui chargera le fichier autant de fois qu'indiqué si aucune politique de cache efficace n'est configurée.

Outre cette exception, les problèmes de performances vont se ressentir au moment de l'interprétation et l'exécution des scripts. En effet, si le script est mentionné 3 fois dans le code, il sera exécuté 3 fois, et ce sur tous les navigateurs.

Pour plus de détails, n'hésitez pas à [consulter cet article sur le sujet (EN)](http://calendar.perfplanet.com/2014/the-pain-of-duplicate-scripts/).

**La solution**Il reste tout de même une solution pour pouvoir utiliser un script plusieurs fois sans l'interpréter et l'exécuter plus d'une fois. Vous devez écrire du code JavaScript, qui vérifie si le script est présent. Si oui, il l'utilise, sinon, il l'injecte afin de l'utiliser.

Prenons l'exemple du widget Facebook, toujours tiré de l'article ci-dessus. À chaque fois que vous voulez intégrer cette fonctionnalité dans votre page, il est nécessaire d'inclure le code suivant :

|  |
| --- |
| (function(d, s, id){ var js, fjs = d.getElementsByTagName(s)[0]; **if (d.getElementById(id)) {return;}** js = d.createElement(s); js.id = id; js.src = "//connect.facebook.net/en\_US/sdk.js"; fjs.parentNode.insertBefore(js, fjs);}(document, 'script', 'facebook-jssdk'));  |

Notez en gras la ligne de vérification de la présence du script dans le document. Le script est alors inclu, interprété et exécuté uniquement lors du premier appel dans la page. Les autres appels tomberont dans le cas de la ligne mise en valeur, et ne feront donc qu'utiliser le script déjà inclu et exécuté.

Vos scripts semblent être injectés efficacement **100/100**

L'injection de scripts via l'instruction JavaScript document.write retarde l'affichage d'une partie au moins de votre page et/ou l’interactivité de votre page pour l'utilisateur.

**Besoin d'injecter un script ?**document.write est parfois utilisé à tort pour injecter un script. Exemple :

|  |
| --- |
| document.write('<script src="' + src + '" type="text/javascript"><\/script>');  |

[Comme indiqué sur cet article (EN)](http://www.stevesouders.com/blog/2012/04/10/dont-docwrite-scripts/), vous devriez plutôt envisager l'utilisation du pattern "createElement-insertBefore" :

|  |
| --- |
| var sNew = document.createElement("script");sNew.async = true;sNew.src = "http://ajax.googleapis.com/ajax/libs/jquery/1.5.1/jquery.min.js";var s0 = document.getElementsByTagName('script')[0];s0.parentNode.insertBefore(sNew, s0);  |

Sur cette page, aucun document.write n'a été détecté.

Les en-têtes Keep-Alive sont activés **100/100**

Le keep-alive permet d'émettre et de recevoir plusieurs requêtes en utilisant la même connexion TCP (activé par défaut avec HTTP 1.1).

Aucune <meta> "http-equiv" inutile détectée **100/100**

Les en-têtes HTTP sont plus performants que les metas http-equiv.

**Les balises <meta http-equiv=""/>**Les metas http-equiv permettent de communiquer au navigateur web des informations équivalentes à celles d'en-têtes HTTP. Par exemple, la meta <meta http-equiv="content-type"/> aura la même portée que l'en-tête HTTP Content-Type.

Deux points n'encouragent pas à l'utilisation des metas http-equiv :

* passer par la meta nécessite d'interpréter le début de la page HTML, ce qui en terme de performances est plus lent que de passer par les en-têtes HTTP
* si l'en-tête HTTP est déjà présent, la meta est ignorée

**Dans quel cas les <meta http-equiv=""/> sont utiles ?**Un seul cas peut justifier la présence de ces metas : si vous n'avez pas accès à la configuration de votre serveur, et donc des en-têtes HTTP.

Nous vous conseillons cependant d'utiliser un serveur configurable, pour établir un site le plus performant possible.

Cette page ne contient pas de balise meta http-equiv. C'est une bonne pratique.

Aucun filtre AlphaImageLoader détecté **100/100**

Ce filtre propriétaire (destiné à IE < 7) permet de résoudre les problèmes liés à la semi-transparence des images PNG. Il empêche l'affichage et bloque le navigateur tant que l'image n'est pas téléchargée et traitée. Son utilisation n'est donc pas recommandée. [En savoir plus](http://developer.yahoo.com/performance/rules.html#no_filters).

Le DOM de cette page ne contient pas trop d'éléments **100/100**

Une structure complexe de la page implique plus d'octets à télécharger, et complexifie la recherche d'éléments précis.

Une page bien conçue peut présenter des contenus riches tout en conservant un nombre d'éléments raisonable. [En savoir plus (EN)](http://developer.yahoo.com/performance/rules.html#min_dom).

Nous vous conseillons de conserver une page avec moins de 1000 éléments dans le DOM.

Le nombre d'éléments du DOM est raisonnable (7 éléments).

**5 - Référencement (SEO)**

Une description est définie sur cette page **100/100**

La page doit disposer d'une description unique.

**Utilisation de la description par les moteurs de recherche**Lorsqu'elle est bien configurée, la description de la page se retrouve directement dans les résultats de recherche:

Cette description vous permet de maîtriser au maximum l'affichage sur les résultats des moteurs de recherche, et d'améliorer le taux de clics vers votre page. [En savoir plus.](https://support.google.com/webmasters/answer/79812?hl=fr)

**Comment définir la description d'une page ?**La description de la page est renseignée grâce à la balise <meta name="description" content="description de la page">, qui doit être placée dans la balise <head>, au début du code.

Cette page définit une <meta> description :

|  |
| --- |
| La chirurgie esthétique à Lyon possède une référence sérieuse depuis la création de la Clinique Esthétique du Grand LYON. Chaque chirurgien et médecin esthétique est expert dans son domaine, vous pouvez lui faire toute confiance.  |

Cette page utilise uniquement des formats d'images standards **100/100**

Les images au format non standard risquent de ne pas être indexées par les moteurs de recherche.

Seuls certains formats d'images sont considérés comme standards sur le web : jpeg, jpg, png, gif, svg, ico, webp. Vous devriez considérer une alternative à tout autre format.

Par ailleurs, pensez à soigner le texte entourant vos images: certains moteurs de recherche analysent les 10 mots (environ) précédents et suivants l'image, pour restituer son contexte.

Un titre est défini sur cette page **100/100**

La page doit disposer d'un titre unique.

**Utilisation du titre par les moteurs de recherche**Lorsqu'il est bien configuré, le titre de la page se retrouve directement dans les résultats de recherche :

L'utilisation d'un titre adapté est un critère important pour le référencement. De plus, il vous permet de maîtriser au maximum l'affichage sur les pages de résultats, et d'intégrer les mots clés sur lesquels vous souhaitez que votre site ressorte.

**Comment définir le titre d'une page ?**Le titre de la page est renseigné grâce à la balise <title>, qui doit être placée dans la balise <head>, au début du code.

Cette page définit bien un titre via la balise title.

Cette page définit un titre :

|  |
| --- |
| Clinique Esthétique du Grand LYON : peeling, Botox (toxine botulique), acide hyaluronique, mésolift, laser ,radio-fréquence mais aussi l'épilation longue durée par lumière pulsée ou laser |

Aucun paramètre de requête n'est détecté dans l'URL **100/100**

Cette URL ne contient aucun paramètre.

Une URL doit être la plus lisible possible. Par exemple, http://example.com?userId=332&group=MonGroupe sera moins bien lisible que http://example.com/mongroupe/moi.

La description de la page a une taille cohérente **100/100**

La page doit disposer d'une description comportant entre 20 et 300 caractères.

**Utilisation de la description par les moteurs de recherche**Lorsqu'elle est bien configurée, la description de la page se retrouve directement dans les résultats de recherche:

Mais les moteurs de recherche ne se servent pas nécessairement de la description que vous leur fournissez. Si celle-ci ne semble pas pertinente, ils se baseront sur d'autres critères pour présenter votre page, et vous perdrez la main sur le contenu affiché. Notez cependant que vous ne serez pas pénalisé directement par Google par exemple si votre texte est trop grand ou trop court. [En savoir plus.](https://support.google.com/webmasters/answer/79812?hl=fr)

**Comment définir la description d'une page ?**La description de la page est renseignée grâce à la balise <meta name="description" content="description de la page">, qui doit être placée dans la balise <head>, au début du code.

Cette page définit une <meta> description comportant entre 20 et 300 caractères:

|  |
| --- |
| La chirurgie esthétique à Lyon possède une référence sérieuse depuis la création de la Clinique Esthétique du Grand LYON. Chaque chirurgien et médecin esthétique est expert dans son domaine, vous pouvez lui faire toute confiance.  |

Vous n'abusez pas de duplication de liens **100/100**

Une page contenant un même lien dupliqué plusieurs fois perd en qualité, et risque de voir son rang diminuer dans les moteurs de recherche.

**Des liens naturels**Les algorithmes en charge du référencement veillent à la qualité et la pertinence de votre page. En répétant un même lien de façon abondante, vous risquez de vous faire pénaliser. De façon générale, une sur-optimisation de votre page est détectée par les algorithmes et peut lourdement pénaliser votre référencement.

**Comment éviter les pénalisations ?**Lorsque vous rédigez une page, établissez des liens cohérents, vers des contenus en relation avec vos propos, sans surcharge.

Les mots de cette URL sont bien délimités **100/100**

Les underscores dans l'URL freinent inutilement la remontée d'une page dans les résultats de Google.

**URL et référencement**Les mots contenus dans l'URL de la page font partie des nombreux facteurs impactant son référencement. Ainsi, lors d'une recherche sur les mots performance web, les URLs contenant les mots performance et web se donnent plus de chances de remonter dans les résultats.

Or pour Google, les underscores ne sont pas des séparateurs de mots : si votre URL contient performance\_web, elle n'aidera pas à ressortir la page sur la requête performance web (performance\_web est considéré comme étant un seul mot).

Ce comportement n'est cependant pas propre à tous les moteurs de recherche. Par exemple, Bing ne différencie pas tirets et underscores : l'URL aidera à la remontée de la page sur vos mots clés dans les deux cas.

Notez que cette bonne pratique souligne un frein à la remontée de votre page, mais qu'en aucun cas Google ne vous pénalisera pour avoir utilisé un underscore dans l'URL. [En savoir plus](http://www.webrankinfo.com/dossiers/techniques/tiret-underscore#test-2013).

**Comment résoudre le problème ?**Nous vous conseillons d'utiliser des - à la place des \_ sur vos nouvelles pages.

Le chantier à mener est plus complexe sur les pages déjà en place comme celle analysée ici : si vous changez l'URL, vous perdrez vos efforts de référencement effectués jusqu'à présent. Il faut alors mettre en place une redirection permanente (code HTTP 301) conservant l'ancienne URL, et redirigeant vers la nouvelle. Mais attention : un grand nombre de redirections sur votre site peut nuire à la visibilité de vos pages. N'utilisez donc des redirections que si le nombre des pages concernées reste restreint.

Cette URL est en accord avec la bonne pratique.

**6 – Sécurité**

**7 - Accessibilité**

Les attributs longdesc indiquent des URIs **100/100**

L'attribut longdesc permet de fournir une description plus longue d'un élément sur un autre endroit de la page, ou sur une page différente.

**Quelles valeurs pour l'attribut longdesc ?**

Vous devez indiquer une URI sur cet attribut. Les valeurs suivantes seront par exemples valables :

* #monAncre
* /mapage.html
* http://site-externe/mapage.html

Tous les attributs longdesc de la page définissent une URI.

Tous vos labels sont liés à un élément **100/100**

L'attribut « for »permet d'associer le label à un élément de la page, et aidera les lecteurs d’écran à mieux interpréter votre contenu.

**Label et attribut for**Un label a pour vocation de décrire un élément (texte à remplir, option à cocher, etc). Par exemple, lors du clic sur un label associé à un input de type radio, l'option associée au label sera directement sélectionnée, améliorant ainsi l'expérience utilisateur.

**Comment se servir d'un label ?**Pour lier le label à un élément de la page, vous devez passer par un identifiant. Exemple :

|  |
| --- |
| <form action="/action"> <label for="monIdentifiant"> <input type="radio" name="mesOptions" id="monIdentifiant" value="1" > |

Aucune <meta> 'refresh' détectée **100/100**

Il n'est pas recommandé de forcer automatiquement le rafraichissement d'une page. Ce comportement perturbe les utilisateurs, qui subissent l'action sans pouvoir la maîtriser.

Cette page n'utilise pas de balise meta refresh. Il s'agit d'une bonne pratique.

Aucun attribut "src" n'est vide **100/100**

L'attribut "src" permet d'accéder à une autre ressource. Rien ne justifie son utilisation avec une valeur vide.

Tous les attributs src de la page définissent une cible. Il s'agit d'une bonne pratique.

Vos liens ont des titres cohérents **100/100**

Un lien est plus attractif si son texte décrit ce qui se cache derrière. Vous pouvez par ailleurs en profiter pour travailler ces textes avec des mots clés bien choisis, dans une démarche d'amélioration de votre référencement.

Tâchez de mettre en place une description du lien au sein de vos balises <a>, plutôt que d'indiquer le lien lui même. Exemple : <a href="http://monlien.com/">Ma description</a>

Les contenus de vos balises <a> sont différents des liens eux-mêmes. Il s'agit d'une bonne pratique.

**8 - Nombre de requêtes**

Vous utilisez des sprites CSS pour vos images **100/100**

Combiner les petites images dans une sprite CSS réduit le nombre de fichiers que le navigateur doit télécharger et accélère donc le chargement de votre page.

**Les sprites CSS**Une sprite CSS est un fichier unique dans lequel plusieurs petites images sont regroupées, positionnées les unes à côté des autres. L'affichage de chaque petite image au sein de votre page se fait ensuite par l'application de styles CSS. Une seule requête est alors nécessaire pour récupérer l'ensemble des images. Cette technique est à utiliser uniquement pour les petites images, comme des icônes par exemple, afin que la sprite CSS ne soit pas trop lourde.

**Exemple**Voici un exemple de sprite CSS, regroupant plusieurs icônes :

La page applique par ailleurs les styles associés à la sprite, du type :

|  |
| --- |
| .sprite { background-image: url(img/sprite.png); background-repeat: no-repeat; display: block;}.sprite-browsers-firefox { width: 31px; height: 28px; background-position: -74px 0;} |

Il reste à définir la bonne classe dans votre fichier HTML, et l'icône apparaît :

|  |
| --- |
| <span class="sprite sprite-browsers-firefox"></span> |

**Comment créer une sprite CSS ?**Des outils facilitent leur création qui peut se révéler complexe. Par exemple :

* <https://draeton.github.io/stitches/>
* <http://spritegen.website-performance.org/>

Vous obtenez alors votre sprite CSS avec les styles à appliquer.

**9 - Qualité**

Les extensions de vos ressources sont bonnes **100/100**

L'extension d'une ressource permet de repérer facilement le type de contenu qui y est associé. Vous n'avez aucun intérêt à indiquer une extension différente du type réel d'un fichier.

Ainsi, une ressource nommée ressource.js devra être associée à un en-tête HTTP Content-Type ayant pour valeur application/javascript. Vous pouvez vérifier la valeur de l'en-tête Content-Type à l'aide d'un outil de debug, comme Firebug.

Toutes les ressources de la page définissent leur type **100/100**

Chaque ressource peut définir le type de son contenu afin de faciliter leur interprétation par les navigateurs web.

Vous n'avez aucun intérêt à dissimuler le type d'une ressource.

Nous vous conseillons de définir un en-tête HTTP Content-Type sur toutes les ressources de la page.

Les identifiants utilisées dans votre HTML sont uniques **100/100**

Utiliser un même identifiant pour 2 éléments différents peut entraîner des effets de bord, notamment lors de l'execution de JavaScript ou lors de l'application des règles CSS.

**Identifiant d'éléments**

Chaque élément d'une page web peut disposer d'un identifiant, défini par l'attribut id :

|  |
| --- |
| <p> <span id="mySpan1"></span></p> |

Ces identifiants vous permettent de manipuler les éléments de votre page avec des instructions CSS ou JavaScript.

**Que faire ?**Vous devez faire en sorte qu'aucun élément de la page ne dispose du même identifiant. Si vous souhaitez partager des propriétés/comportements entre plusieurs éléments, il faudra utiliser l'attribut class, dédié à cet effet :

|  |
| --- |
| <p> <span **class="mySpans"**></span><span **class="mySpans"**></span></p> |

Aucun sélecteur CSS n'est trop complexe **100/100**

Des sélecteurs CSS trop complexes nuisent à la lisibilité générale du code et constituent un gain possible en terme de performance.

**Les règles CSS**Les règles CSS permettent de sélectionner un ensemble d'éléments du DOM, et de leur appliquer des propriétés de mise en forme.

Elles sont composées d'un ensemble de sélecteurs (identifiants, classes, noms d’éléments, etc) permettant d’adresser les éléments à cibler. Une page bien structurée permet l’écriture de règles CSS simples, rapidement lisibles et performantes.

**Comment améliorer ce point ?**Simplifiez vos sélecteurs tant que possible, quitte à introduire de nouvelles classes dans votre code pour éviter des enchaînements trop longs.
Par exemple, la règle suivante :

|  |
| --- |
| body td .maClasse div .maClasse2 span .maClasse3 {} |

Pourrait être améliorée, en passant éventuellement par un identifiant :

|  |
| --- |
| #monIdentifiant .maClasse3 {} |

Vos styles sont bien séparés du HTML **100/100**

Dissocier les balises HTML de leur style améliore la maintenabilité du code et favorise sa factorisation.

**Définir les styles CSS**Les styles CSS permettent de mettre en forme la page. Pour les définir, vous disposez de 3 méthodes principales :

* la déclaration dans un fichier CSS
* la déclaration dans un bloc "inline" (balises <style>)
* la déclaration dans un attribut style d'un élément HTML

**Que faire ?**Nous vous conseillons de regrouper vos styles CSS au sein de balises <style> ou dans des fichiers CSS dédiés. Ainsi, la partie HTML se charge uniquement de fournir la structure de la page, et sa mise en forme est externalisée. L'attribut style devrait uniquement être généré en cas de besoin par du code JavaScript (exemple : nécessité de connaître la taille de l'écran).

Cette page n'utilise pas d'attribut style.

Vos images PNG ne sont pas compressées (gzip/deflate) **100/100**

La compression a un coût, du côté du serveur et du navigateur client. Il ne faut donc l'activer que si elle est efficace.

**PNG et compression**Les images au format png ne supportent pas la compression gzip. Il est donc inutile d'effectuer des opérations de compression au niveau de votre serveur, et de décompression au niveau client pour ce type de ressources.

Cette page n'essaie pas de compresser les images. C'est une bonne pratique.

**10 - Nombre de requêtes**

Aucune image n'utilise qu'une seule couleur **100/100**

Une requête vers une image à couleur unique augmente inutilement le volume de données qui transitent sur le réseau.

Il est inutile de récupérer une image composée uniquement d'une couleur de fond. Les styles CSS vous permettront de dessiner la même forme à moindre coût.

Par exemple, pour dessiner un rond, vous pouvez utiliser le code CSS suivant :

|  |
| --- |
| #myElement { background:#ff0000; border-radius:50%; width:160px; height:160px;} |

Sur cette page, toutes les images sont bien composées de plusieurs couleurs.

Aucune redirection trop longue n'a été détectée **100/100**

Les redirections déclenchent des allers-retours évitables sur le réseau, et allongent le temps de chargement de la page.

**Redirections http**Une redirection HTTP permet d'indiquer que le contenu souhaité est accessible depuis une autre URL. Plusieurs requêtes sont alors envoyées successivement pour récupérer la ressource cible. Les redirections HTTP se distingue grâce à leur code de retour, compris entre 300 et 399. [Vois les spécifications](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.3) des redirections HTTP.

**Que faire ?**Permettez à l'utilisateur d'accéder directement à votre contenu, sans passer par les redirections, ou déterminez et améliorez ce qui provoque ces temps de chargement trop importants sur vos redirections. [Voir les recommandations](https://developers.google.com/speed/docs/insights/AvoidRedirects) données par Google.

Cette page ne fait pas appel à des redirections avant d'accéder au bon contenu.

Cette page n'utilise que des ressources accessibles **100/100**

Toutes les ressources de la page sont accessibles.

Ces erreurs nuisent probablement à certains contenus ou comportements de votre site, et provoquent un trafic réseau inutile, ce qui impacte donc le temps de chargement de votre site.

Vous n'utilisez pas trop de "prefetch" sur vos liens **100/100**

Aucune ressource n'est pré-chargée sur cette page.

Nous vous conseillons de limiter le nombre de prefetch sur une seule page afin de ne pas surcharger inutilement le réseau. Analysez les comportements de vos utilisateurs afin d'identifier les ressources ayant le plus de chances d'être réclamées à la suite d'une action.

Sachez par ailleurs que certains navigateurs limitent automatiquement le nombre de ressources pré-chargées. C'est par exemple le cas d'Internet Explorer 11 (10 ressources au maximum).

**11 - Apache**

La version de votre serveur Apache n'est pas exposée **100/100**

Vous utilisez Apache, mais nous n'avons pas détecté quelle version. Une personne malveillante aura plus de mal à attaquer votre site, car elle ne saura pas quelle version utilise votre serveur Apache. C'est une bonne pratique.

Il y a deux valeurs à vérifier dans votre configuration pour que la version soit cachée: [ServerSignature](http://httpd.apache.org/docs/2.2/fr/mod/core.html#serversignature) et[ServerTokens](http://httpd.apache.org/docs/2.2/fr/mod/core.html#servertokens).

|  |
| --- |
| # Cacher la version dans l'en-tête HTTP 'Server'.# exemple: afficher uniquement "Server: Apache"ServerTokens Prod |

|  |
| --- |
| # Ne pas ajouter de ligne contenant# le nom et la version du serveur# à la fin de chaque réponse.ServerSignature Off |

Cependant, gardez en tête que le meilleur moyen de se prémunir d'attaques est de régulièrement mettre à jour votre serveur Apache.