

Rapport d'analyse performance et qualité



https://gflahaut.github.io/GwendolineFlahaut_4_03122020/

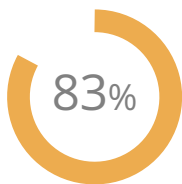
Ce rapport a été généré sur [Dareboost.com](https://dareboost.com), **service en ligne d'analyse et de surveillance de la performance et de la qualité des sites web.**

N'hésitez pas à découvrir [nos offres](#) ou à nous contacter :
contact@dareboost.com.

Sommaire

Résumé	3
Conseils et bonnes pratiques	4
Accessibilité	4
Compatibilité	6
Nombre de requêtes	9
Optimisation du rendu	11
Comment spécifier un jeu de caractères dans l'en-tête Content-Type ?	12
Politique de cache	16
Qualité	20
Référencement (SEO)	25
Sécurité	31
Volume de données	36
jQuery	41

Résumé



Problèmes



Améliorations



Succès



VISITEUR SIMULÉ :



Chrome



Paris

8.0/1.5Mbps (Latence : 50 ms)

Largest Contentful Paint



Total Blocking Time



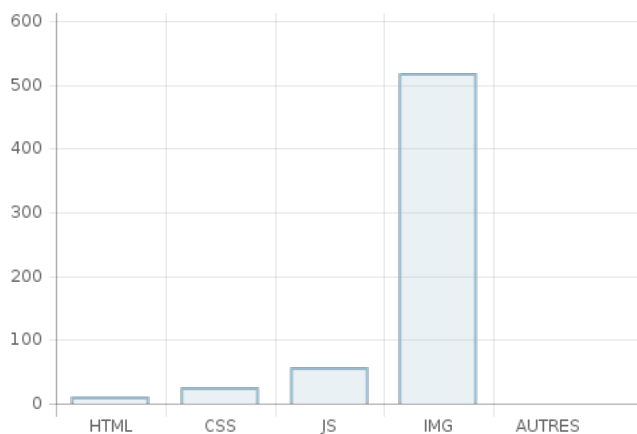
Cumulative Layout Shift



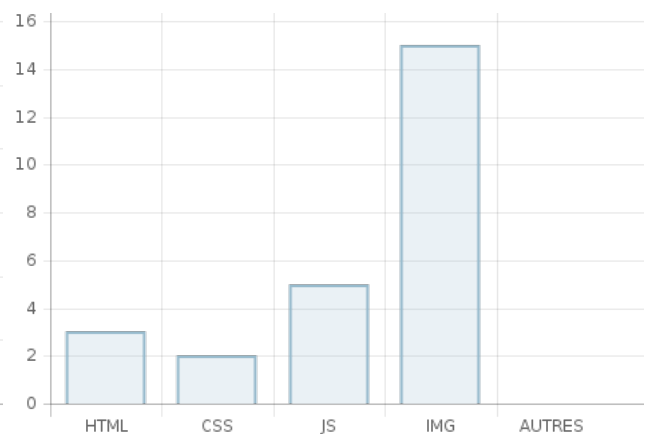
Speed Index



Répartition du poids



Répartition du nombre de requêtes



Technologies détectées



Twitter Bootstrap



Varnish



jQuery

Conseils et bonnes pratiques

Accessibilité

Vos principaux axes d'optimisation

0/100

#2530

! Rendez la page accessible avec et sans www. au début de l'URL

Certains internautes ont pour habitude d'accéder directement à votre site, parfois avec et parfois sans `www.` au début de l'URL.

Pour éviter des problèmes d'accès au contenu de la page, il est nécessaire d'utiliser une redirection (permanente) pour faire pointer l'une des adresses vers l'autre.

Que faire ?

Vous devez mettre en place une redirection permanente (301) de `https://gflahaut.github.io/GwendolineFlahaut_4_03122020/` vers `https://www.gflahaut.github.io/GwendolineFlahaut_4_03122020/`

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#2569

✓ Aucun élément vide détecté

Les éléments `<p>`, ``, `<button>`, `<legend>`, `<caption>`, `<figcaption>` and `<quote>` ne doivent pas être vides car s'ils le sont, certains lecteurs d'écran auront des difficultés à interpréter leur présence.

Supprimez ces éléments vides de votre code ou décochez-les avec l'attribut `aria-hidden` pour que les lecteurs d'écran les ignorent.

```
<p aria-hidden="true"></p>
```

Exemple

100/100

#72

✓ La balise `<noscript>` est définie

Cette page utilise la balise `<noscript>`. Celle-ci permet d'afficher un message lorsque le JavaScript est désactivé par l'utilisateur.

100/100

#2334

✓ La langue de la page est spécifiée

Votre page possède un attribut `lang` qui permet aux liseuses d'écran de prendre correctement en charge votre site.

100/100

#2395

✓ Aucune `<meta>` 'refresh' détectée

Il n'est pas recommandé de forcer automatiquement le rafraîchissement 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.

100/100

#2410

✓ **Aucun attribut "src" n'est vide**

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.

100/100

#2423

✓ **Vos liens ont des titres cohérents**

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 : `Ma description`

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

100/100

#2468

✓ **Votre titre principal est bien le premier titre spécifié**

Facilitez le travail des liseuses d'écran en indiquant votre titre principal (`<h1>`) avant tout autre titre.

Il est conseillé de respecter une hiérarchie cohérente entre vos titres. Le titre principal de la page doit ainsi être spécifié avant tout autre titre `<hX>`.

100/100

#2523

✓ **Tous vos labels sont liés à un élément**

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" >
```

Exemple

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#80

✓ **Aucune balise frameset, frame ou noframes détectée**

Ces balises sont aujourd'hui à proscrire, car elles présentent des problèmes pour la cohérence de navigation, le référencement, ou encore la sauvegarde de pages dans les favoris d'un navigateur.

Aucune de ces balises n'est détectée sur cette page.

L'utilisation de la balise `iframe` est préférée.

100/100

#82

✓ **Aucune applet Java détectée**

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.

100/100

#89

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

Aucune balise `bgsound` détecté. 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](#).

Utilisez la balise `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>
```

Exemple

100/100

#99

✓ **Aucun lien vers un document Word détecté**

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.

100/100

#2382

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

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.

100/100

#2424

✓ Aucun BOM (Byte Order Mark) détecté

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.

100/100

#2454

✓ Vos caractères sont encodés en UTF-8

Votre contenu est lisible par le plus grand nombre possible d'internautes.

Codage de caractères

Le codage de caractères permet d'indiquer au navigateur web comment interpréter les octets de la page web pour les convertir en caractère lisible par l'utilisateur. De nombreux codages existent afin de pouvoir représenter tous les caractères spécifiques à chaque langue. Notez que les panels de codages supportés sont différents d'une machine à une autre. Si vous utilisez un codage trop spécifique, de nombreux internautes ne pourront donc pas interpréter correctement la page.

Pourquoi choisir l'UTF-8 ?

L'UTF-8 est reconnu pour être un codage supporté par quasiment tous les internautes, et prenant en compte une grande variété de caractères. Son universalité est donc particulièrement bien adaptée au web.

100/100

#2488

✓ Aucune ressource Flash détectée

Flash 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).

Par ailleurs, Google indique directement dans ses résultats de recherche sur mobile que [le site risque de pas fonctionner \(EN\)](#).

Félicitations, cette page ne contient pas de ressources utilisant Flash.

100/100

#2493

✓ Le plugin Silverlight n'est pas utilisé

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.

100/100

#2494

✓ Le format de fonte EOT correctement déclaré

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.

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

100/100

#2496

✓ Vos fontes de caractères sont optimisées

Veillez à délivrer le format de fonte le plus adapté pour chaque plateforme.

Indiquer plusieurs types de fontes

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 sur lequel il tombe.

Que faire ?

Assurez vous que vos fontes de caractères sont déclarées dans le bon ordre au sein de la propriété `src` de `@font-face` .
L'ordre à respecter pour vos formats est le suivant : EOT, WOFF2, WOFF, TTF, puis SVG.

Toutes vos fontes sont bien placées dans l'instruction `@font-face` .

Saviez-vous que ?

#2543

Répartition des ressources par domaine

Cette page charge ses ressources depuis 2 domaines différents. Nous avons extrait pour chaque domaine :

- Le temps de chargement (cumulé) : délai nécessaire au chargement de toutes les ressources du domaine
- Le temps de réponse serveur (cumulé) : délai des réponses du serveur (éventuelle connexion TCP + attente de la réception du premier octet)
- Le poids total des données qui ont été chargées
- Le nombre de requêtes

Ci-dessous les domaines utilisés lors du chargement de la page :

Domaine	Temps total (ms)	Temps niveau serveur (ms)	Poids (ko)	Requêtes
gflahaut.github.io	5875	2301	604	23
github.com	675	639	6	2

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#2339

✓ **Aucune redirection trop longue n'a été détectée**

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 distinguent grâce à leur code de retour, compris entre 300 et 399. [Voyez les spécifications](#) 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. [Voyez les recommandations](#) données par Google.

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

100/100

#2344

✓ **Cette page n'utilise que des ressources accessibles**

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.

100/100

#2403

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

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).

100/100

#2466

✓ **Aucune image n'utilise qu'une seule couleur**

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;  
}
```

Exemple

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

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#2566

✓ **Votre réponse HTML n'est pas trop lourde**

Pourquoi réduire la quantité de code présente dans une page web ?

Avant de pouvoir être affichée, une page web doit, entre autres, être téléchargée, analysée et modélisée en un document compréhensible par le moteur de rendu. Si la quantité de code présenté dans la page est trop importante, ces étapes sont ralenties et le rendu, retardé.

Comment réduire la quantité de code ?

Votre réponse HTML ne devrait contenir que les ressources immédiatement nécessaires à l'affichage de la zone visible de la page. Déplacez les informations inline dans des fichiers externes (JS pour les scripts, CSS pour les styles, requêtes asynchrones pour des contenus additionnels) et simplifier la structure de votre page web.

100/100

#2575

✓ **Aucun Mutation Events détecté dans vos scripts**

Pour capturer les événements relatifs au DOM, n'utilisez pas les Mutation Events. Des alternatives existent.

Une bonne idée, mais une mauvais implémentation

Lors du développement d'applications JavaScript complexes, vous pouvez avoir besoin de déterminer quand l'arbre des nœuds DOM a été modifié. Introduit en 2000 dans la [spécification DOM, Level 2](#) pour répondre à ce besoin, les Mutations Events sont des événements initiés par le navigateur qui vous permettent de savoir quand un nœud DOM est ajouté, retiré ou supprimé.

Les Mutations Events, cependant, présentent d'importants problèmes de performance. D'abord, ils sont synchrones, c'est-à-dire qu'ils empêchent que d'autres événements de la file d'attente soient déclenchés (si ces événements sont utilisés pour mettre à jour l'interface, cela entraînera des lenteurs). Ensuite, ils sont implémentés en tant qu'événements de navigateur, et traversent donc l'arbre DOM depuis l'élément HTML ciblé vers l'élément parent qui écoute l'événement, encombrant le fil d'exécution JavaScript pendant ce temps.

Les Mutations Events ont été dépréciés en 2016 dans la [spécification DOM, Level 3](#).

Les Mutation Observers à la rescousse

Si vous devez surveiller les changements apportés à l'arbre DOM, préférez l'utilisation des interfaces `MutationObserver` ([DOM4 Living Standard](#)). Les Mutation Observers sont asynchrones, traités par lots, et sont capable d'observer tout ou partie des changements apportés à un nœud. Ils sont moins gourmand en utilisation du processeur que les événements du navigateur et provoquent donc moins de gels de l'interface utilisateur, voire aucun.

[Apprenez à utiliser les Mutation Observers \(Mozilla Developer Network\)](#).

100/100

#2353

✓ Vos ressources JavaScript ne bloquent pas le chargement de la page

Lorsque le navigateur web rencontre du code JavaScript en interprétant le code source d'une page web, cela peut ralentir considérablement l'affichage de la page, surtout s'il est nécessaire de télécharger un script externe.

Différez au maximum l'utilisation du JavaScript pour assurer un début rapide de l'affichage de la page.

Que faire ?

Avant toute chose, distinguez les portions de votre code JavaScript qui sont critiques au chargement de la page, et doivent être chargées le plus tôt possible. Placez-les dans un fichier externe JavaScript et chargez-le au plus tôt. Gardez ce fichier aussi petit que possible et différer les chargements ou l'exécution de tous les autres scripts JS.

Utilisez si possible l'une des techniques suivantes pour des appels à des fichiers externes :

- utilisation de l'attribut `async`
- utilisation de l'attribut `defer`
- ajout du script dans le DOM en JavaScript, lors de l'évènement `onload`
- placer les scripts à la fin de votre code source (idéalement à la fin du `<body>`)

100/100

#2356

✓ Vous spécifiez un jeu de caractères dans vos en-têtes HTTP de réponse

En spécifiant le jeu de caractères utilisé dans l'en-tête HTTP **Content-Type**, vous permettez au navigateur d'analyser immédiatement la page.

Lorsque le navigateur reçoit des octets, il doit identifier la collection de lettres et de symboles qui ont été utilisés pour écrire le texte qui a été converti en ces octets, ainsi que le codage utilisé pour cette conversion, afin de l'inverser. Si aucune information de ce type n'a été transmise, le navigateur tentera de trouver des motifs reconnaissables dans les octets pour déterminer l'encodage lui-même, et éventuellement essayer certains jeux de caractères courants, ce qui prendra du temps, retardant le traitement ultérieur de la page.

Comment spécifier un jeu de caractères dans l'en-tête Content-Type ?

Dans l'explication suivante, nous considérerons UTF-8 comme le jeu de caractères à définir, mais n'oubliez pas que le jeu de caractères déclaré dans votre en-tête HTTP Content-Type doit refléter le jeu de caractères utilisé pour encoder le fichier, qui peut ne pas être UTF-8.

Dans **Apache 2.2+**, la configuration de l'UTF-8 comme jeu de caractères par défaut pour les fichiers de types `text/plain` ou `text/html` implique d'utiliser la directive `AddDefaultCharset` :

```
AddDefaultCharset utf-8
```

Exemple

Pour les autres types de fichiers, vous aurez besoin d'utiliser la directive `AddCharset` :

```
AddCharset utf-8 .js .css ...
```

Exemple

Dans **nginx**, vous devrez vous assurer que le module `ngx_http_charset_module` est chargé, puis utiliser la directive `charset` :

```
charset utf-8;
```

Exemple

Ici aussi, il est possible d'affiner le champs d'application pour que d'autres types de fichiers que `text/html` soient livrés en UTF-8, en utilisant la directive `charset_types` :

100/100

#2370

✓ Vous évitez judicieusement les @import CSS

L'utilisation de l'instruction CSS @import permet d'ajouter des feuilles de style. En réalité, les navigateurs internet ne peuvent pas les télécharger en parallèle, et cela peut ajouter un délai dans l'affichage de la page. Il faut plutôt utiliser la balise `link`. [En savoir plus](#).

100/100

#2405

✓ Aucune <meta> "http-equiv" inutile détectée

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 termes 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.

100/100

#2416

✓ Le DOM de cette page ne contient pas trop d'éléments

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 raisonnable. [En savoir plus \(EN\)](#).

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

Le nombre d'éléments du DOM est raisonnable (216 éléments).

100/100

#2445

✓ Vous interprétez et exécutez vos scripts une seule fois

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 \(EN\)](#).

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'));
```

Exemple

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.

100/100

#2539

✓ Cette page n'utilise pas de redirection côté client

Aucune redirection n'a été effectuée côté client (window.location, meta refresh...) pour l'affichage de cette page. Les redirections côté client sont à éviter absolument.

100/100

#2550

✓ Vos scripts semblent être injectés efficacement

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>');
```

Exemple

Comme indiqué sur [cet article \(EN\)](#), 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);
```

Exemple

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

Vos principaux axes d'optimisation

0/100

#2437

! Adoptez une politique de cache plus longue dans 19 requêtes

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

L'entête expires

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. L'en-tête HTTP **Expires** se présente sous la forme suivante :

Expires: Thu, 25 Dec 2014 20:00:00 GMT

Exemple

Lorsque vous mettez en production **une nouvelle version de votre site, pensez bien à renommer les ressources statiques** ayant été modifiées (versioning). En effet, si vous ne changez pas leur nom, vos utilisateurs garderont les ressources stockées en cache, correspondant à l'ancienne version. Ils risquent alors de se retrouver dans une version instable de votre page. Par exemple :

maressource.min.20140101.js

Exemple

Voir [les recommandations de Yahoo!](#) sur le sujet.

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.

Cette page contient 19 ressources **n'ayant pas une date d'expiration assez lointaine** :

- [gflahaut.github.io/GwendolineFlahaut_4_.../bootstrap.min.css](#)
- [gflahaut.github.io/GwendolineFlahaut_4_.../houette-agence.jpg](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]2020/Sass/main.css](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]0/img/jpg/logo.jpg](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]2020/img/jpg/1.jpg](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]2020/img/jpg/2.jpg](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]2020/img/jpg/3.jpg](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]2020/img/jpg/4.jpg](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]query-2.1.0.min.js](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]s/bootstrap.min.js](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]20/js/blocs.min.js](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]20/js/gmaps.min.js](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\].touchSwipe.min.js](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]g/presentation.svg](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]gence-banniere.jpg](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]0/img/svg/edit.svg](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]e-presentation.jpg](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]ribbble-brands.svg](#)
- [gflahaut.github.io/GwendolineFlahaut_4_...\]stagram-brands.svg](#)

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#70

✅ Cette page ne contient pas de scripts inline trop longs

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){if('GoogleAnalyticsObject' in _ga){return;}
  ga=('create', 'UA-11111111-1', 'monsite.com');
}</script>
```

Exemple

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.

100/100

#2352

✅ Un en-tête "Vary: Accept-Encoding" est défini

L'en-tête HTTP **Vary: Accept-Encoding** permet de mettre en cache deux versions d'une ressource sur les proxys parcourus : l'une est compressée, l'autre non. Ainsi, les clients ne pouvant pas décompresser les ressources peuvent utiliser la version décompressée, et les autres la version compressée.

100/100

#2430

✓ Vous spécifiez des en-têtes d'expiration

Les en-têtes de cache (`Cache-Control` , `ETag` , anciennement `Expires`) sont indispensables pour une politique de cache efficace, et vont grandement impacter le temps de chargement de vos pages lors des prochaines visites.

L'en-tête `Cache-Control`

Chaque ressource peut définir ses règles de mises en cache via l'en-tête HTTP `Cache-Control` . La propriété `max-age` permet de définir la durée de la mise en cache (en secondes), et peut être accompagnée d'instructions concernant la mise en cache des ressources sur des serveurs intermédiaires, situés entre le navigateur et le serveur délivrant la ressource.

L'en-tête suivant indique que la réponse peut être mise en cache sur des serveurs intermédiaires et sur le navigateur (`public` , par opposition à `private` lorsque seul le navigateur est autorisé à mettre en cache) pendant deux heures :

```
Cache-Control: private, max-age=7300
```

Exemple

Si certaines de vos ressources ne doivent pas être mises en cache, vous pouvez également l'explicitier :

```
Cache-Control: no-store
```

Exemple

L'en-tête `Expires`

L'en-tête `Expires` est l'en-tête le plus ancien permettant de gérer la mise en cache des ressources, et vous permettra de gérer le cache pour des [navigateurs ne supportant pas `Cache-Control`](#) .

Grâce à l'en-tête `Expires` , vous pouvez définir une date d'expiration pour chaque ressource : tant que la date n'est pas dépassée, l'utilisateur stocke et utilise la ressource en cache.

La date d'expiration des ressources se paramètre à l'aide de l'en-tête HTTP `Expires` :

```
Expires: Thu, 25 Dec 2014 20:00:00 GMT
```

Exemple

Vous pouvez indiquer une date d'expiration lointaine pour les ressources statiques (1 an maximum), et plus proche pour des ressources susceptibles d'être modifiées (48h minimum).

Revalidation des ressources à expiration du cache

Si aucune politique de cache n'est définie pour une ressource ou que la durée de sa mise cache est dépassée, le navigateur procède à une nouvelle requête pour télécharger la ressource.

Pour éviter au navigateur de télécharger une ressource qui n'aurait pas été modifiée depuis sa mise en cache, utilisez l'en-tête HTTP `ETag` . Il permet d'associer à chaque version d'une ressource un jeton de validation. Lorsque le cache d'une ressource expire, le navigateur demandera à nouveau au serveur la ressource, en lui passant ce jeton avec l'en-tête `If-None-Match` contenant la valeur du jeton. Le serveur comparera sa version du jeton avec celle fournie. Si la ressource n'a pas été modifiée, le serveur autorisera le navigateur à renouveler la mise en cache de la ressource sans la télécharger à nouveau, via une réponse HTTP 304 Not Modified.

Invalidation forcée du cache

Lorsque vous mettez en production **une nouvelle version de votre site, pensez bien à renommer les ressources statiques** ayant été modifiées (versioning), afin de forcer le téléchargement de ces nouvelles versions par vos utilisateurs, qui risque sinon de conserver les ressources stockées en cache dans leurs navigateurs. Ils risquent alors de se retrouver dans une version instable de votre page. Par exemple :

```
maressource.min.20140101.js
```

Exemple

Pour en savoir plus sur la mise en cache HTTP, n'hésitez pas à consulter [les recommandations de Google](#).

Félicitations! Vos ressources sont mises en cache.

Vos principaux axes d'optimisation

0/100

#2517

! Le mot clé !important est utilisé 33 fois

Le mot clé !important s'approche d'un hack permettant d'interdire toute surcharge d'une propriété définie. Si parfois il peut s'avérer utile, cela doit rester exceptionnel !

Si vous en abusez, songez à revoir votre structure de code CSS. Dans ce conseil, vous êtes pénalisé au-delà de 10 utilisations de ce mot clef.

Voici la liste des !important détectés :

https://gflahaut.github.io/GwendolineFlahaut_4_03122020/Sass/main.css

- `.col-sm-4 a {color: #fafafa !important}` (ligne 11, col 3)
- `a, button {outline: none !important}` (ligne 91, col 3)
- `.bg-center, .bg-l-edge, .bg-r-edge, .bg-t-edge, .bg-b-edge, ...` (ligne 147, col 3)
- `.bg-center, .bg-l-edge, .bg-r-edge, .bg-t-edge, .bg-b-edge, ...` (ligne 148, col 3)
- `.bg-center, .bg-l-edge, .bg-r-edge, .bg-t-edge, .bg-b-edge, ...` (ligne 149, col 3)
- `.bg-center, .bg-l-edge, .bg-r-edge, .bg-t-edge, .bg-b-edge, ...` (ligne 150, col 3)
- `.navbar-toggle:hover {background: transparent !important}` (ligne 403, col 3)
- `.nav > li > .dropdown a {background: none !important}` (ligne 434, col 3)
- et 25 autres

92/100

#2383

✓ Séparez vos styles de votre code HTML

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 utilise 1 attribut(s) `style` :

- `<div class="keywords" style="color: #d43900">`

Saviez-vous que ?

#71

Attention aux commentaires au sein du code source

Ne transmettez à vos internautes que les données dont ils ont besoin.

Pages web et commentaires dans le code source

Les commentaires dans le code source sont indispensables aux développeurs : ils permettent de détailler une portion de code complexe, et aident à naviguer plus efficacement dans le DOM. Cependant, il n'y a aucun intérêt à communiquer ces informations à vos utilisateurs. Les seuls commentaires dont la présence est justifiée sur le navigateur de l'internaute sont ceux permettant la compatibilité de votre site sur toutes les plateformes (exemple : rendre le site compatible avec Internet Explorer 8).

Veillez donc à ne pas y exposer des données sensibles pouvant permettre à une personne malveillante de corrompre votre système. Par ailleurs, rien ne justifie que des blocs de code soient transmis à vos visiteurs.

#2512

1 propriété CSS est surchargée

Il est souvent inutile de définir 2 fois la même propriété au sein d'une même règle : la seconde surchargera la première. Cela risque de nuire à la lisibilité du code CSS. Supprimer les propriétés surchargées inutilement réduira également le poids du fichier.

Comment améliorer ce point ?

Supprimez l'une des occurrences de la propriété en cause. Par exemple, si votre fichier dispose des propriétés suivantes :

```
.maClasse {  
  margin: 20px;  
  ...  
  margin: 10px;  
}
```

Exemple

Vous devriez envisager leur réécriture de la manière suivante :

```
.maClasse {  
  margin: 10px;  
}
```

Exemple

Les fichiers suivants définissent plusieurs fois la même propriété dans une règle. La [gestion de fallbacks \(EN\)](#) peut justifier une duplication.

Assurez-vous que les propriétés remontées ici sont maîtrisées :

https://gflahaut.github.io/GwendolineFlahaut_4_03122020/Sass/main.css

- `.navbar-brand {font-weight: 600}` (ligne 374, col 3)

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#2379

Une favicon est définie

Vous avez défini une favicon sur cette page. C'est une bonne pratique.

Une Favicon est une petite image fournissant une icône à un site web. Elle est située à la racine du serveur et le navigateur la demandera toujours. Il est préférable de ne pas répondre par un code HTTP 404 (ressource non trouvée).

De plus ce fichier sera demandé pour chaque nouvelle page: il faut le mettre en cache pour qu'il ne soit demandé qu'une seule fois. [En savoir plus](#).

100/100

#2398

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

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.

100/100

#2448

✓ Les extensions de vos ressources sont bonnes

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 de l'outil développeur de votre navigateur préféré, par exemple DevTools sur Chrome.

100/100

#2449

✓ Toutes les ressources de la page définissent leur type

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.

100/100

#2491

✓ Les identifiants utilisés dans votre HTML sont uniques

Utiliser un même identifiant pour 2 éléments différents peut entraîner des effets de bord, notamment lors de l'exécution 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>
```

Exemple

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>
```

Exemple

100/100

#2515

✓ Aucune propriété CSS n'est surchargée par des shorthand

Le CSS comporte [quelques pièges](#) (EN) pouvant engendrer des effets de bords indésirables sur vos styles.

La surcharge de propriétés en CSS

Pour une même règle CSS, si 2 propriétés identiques sont spécifiées, la dernière surchargera toutes les autres. La possibilité de définir des propriétés "raccourcies" (shorthand) peut parfois prêter à confusion dans ce contexte. Par exemple, "border" se réfère à toutes les bordures d'un élément, et revient à définir "border-top", "border-right", "border-left" et "border-bottom". Ainsi, avec le code suivant :

```
.maClasse {  
  border-color:red;  
  border:5px solid; // border reprend la couleur par défaut  
}
```

Exemple

La propriété "border-color" se retrouve surchargée par "border", la couleur reprend donc la valeur par défaut.

Évitez les effets de bord

Avoir de telles surcharges en place présente un risque, puisque la surcharge n'est pas explicite et peut donc être un effet de bord. Si le style obtenu est celui désiré, alors la propriété qui est ainsi surchargée est donc inutile, supprimez-la (cela réduira également le poids du CSS).

100/100

#2516

✓ Aucun sélecteur CSS n'est trop complexe

Des sélecteurs CSS trop complexes nuisent à la lisibilité générale du code et constituent un gain possible en termes 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 {}
```

Exemple

Pourrait être améliorée, en passant éventuellement par un identifiant :

```
#monIdentifiant .maClasse3 {}
```

Exemple

100/100

#2527

✓ Vos propriétés CSS ne sont pas dupliquées

Les propriétés dupliquées au sein d'une même règle CSS nuisent à la lisibilité générale du code CSS. C'est également une source d'optimisation, puisque supprimer les éléments dupliqués réduira le poids du fichier.

Les propriétés CSS

Les propriétés CSS permettent d'appliquer une mise en forme pour un ensemble d'éléments donnés. Il est inutile de définir 2 fois une même propriété sur une règle : la seconde surchargera automatiquement la première.

Comment améliorer ce point ?

Supprimez l'une des occurrences de la propriété en cause. Par exemple, si votre fichier dispose des propriétés suivantes :

```
.maClasse {  
  margin: 10px;  
  ...  
  margin: 10px;  
}
```

Exemple

Vous devriez envisager leur réécriture de la manière suivante :

```
.maClasse {  
  margin: 10px;  
}
```

Exemple

100/100

#2556

✓ Tous vos sélecteurs CSS sont uniques

Utiliser plusieurs fois le même sélecteur sur des blocs déclaratifs différents au sein d'un même fichier CSS peut nuire à la lisibilité et à la maintenabilité du code. C'est également une source d'optimisation, puisque regrouper ces éléments au sein d'une même règle réduit le poids du fichier et réduit le temps nécessaire avant le rendu.

Le modèle objet CSS (CSSOM)

Les règles CSS permettent d'appliquer des propriétés de mise en forme sur des éléments ciblés de votre code HTML.

Pour ce faire, le navigateur construit sa propre modélisation avec le code CSS qu'il a récupéré : le modèle objet CSS (CSSOM). Cette transformation se fait en plusieurs étapes: lecture du fichier, conversion des chaînes de texte en jetons, transformation de ces jetons en objets avec propriétés et règles et organisation des éléments dans un modèle arborescent. Ces opérations se dérouleront d'autant plus vite que le code CSS aura été rédigé de manière simple et non-redondante.

Comment améliorer mes règles CSS ?

Pour des raisons de maintenance autant que de performance, regroupez au sein d'une même règle les sélecteurs CSS dupliqués.

Par exemple, si votre fichier dispose des règles suivantes :

```
.maClasse {  
  margin: 0;  
}  
...  
.maClasse {  
  border: 1px solid black;  
}
```

Exemple

Vous devriez envisager leur réécriture de cette manière :

```
.maClasse {  
  margin: 0;  
  border: 1px solid black;  
}
```

Exemple

Si la duplication des sélecteurs est héritée d'un processus historique ou qu'elle est justifiée par un besoin de maintenance, vous pouvez envisager l'utilisation d'un outil de minification CSS capable de faire cette transformation automatiquement.

Vos principaux axes d'optimisation

0/100

#2503

! Le fichier robots.txt doit être défini

Facilitez au maximum la découverte de votre site par les robots en leur indiquant quelles URLs ne doivent pas être explorées.

Fichier robots.txt

Ce fichier est placé à la racine du site et est interprété par les robots en charge du référencement de votre site. Il délivre des instructions pour indiquer les pages à explorer par les robots.

Notez que ces instructions sont données à titre indicatives. Un robot quelconque ne sera pas bloqué par les restrictions du fichier.

Nous n'avons pas détecté de fichier robots.txt sur ce site, vous devriez le mettre en place à l'adresse suivante :

- <https://gflahaut.github.io/robots.txt>

0/100

#2345

! Évitez l'utilisation d'underscores ou d'espaces pour délimiter les mots de l'URL

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'aidra 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 aidra 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.](#)

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.

Dans cette URL, nous avons détecté 2 underscores `_`.

Saviez-vous que ?

#2457

Cette page comporte 36 liens

Il existe 2 grands types de liens :

- **les liens internes**, qui font référence aux pages du même nom de domaine
- **les liens externes**, qui pointent vers d'autres sites (doivent être pertinents et pointer vers des contenus de qualité)

Si vous faites référence à de nombreux liens, vous pouvez demander aux algorithmes de référencement de ne s'intéresser qu'à certains d'entre eux, en ajoutant un attribut `rel=nofollow` sur ceux qui ne sont pas pertinents (ex: publicités).

Voici la répartition des 36 liens contenus dans la page :

- Aucun lien interne (0,00%)
- 36 liens externes "follow" (100,00%)
- Aucun lien externe "nofollow" (0,00%)

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#69

Vos balises `` contiennent toutes un attribut `alt`

L'attribut `alt` est un critère important en termes de référencement (SEO). En effet, les robots d'indexation des moteurs de recherche ne peuvent pas analyser de contenu graphique. Ils se servent alors de leur texte alternatif pour répondre aux requêtes des internautes. [C'est par exemple le cas pour Google Images](#).

```

```

Exemple

De plus, l'attribut `alt` s'avère utile dans d'autres cas :

- utilisation d'un lecteur d'écran (personnes malvoyantes)
- connexion trop lentes pour charger l'image
- affichage d'un contenu dans le cas d'une erreur dans l'attribut `src`

Vous avez 6 balises `img` et elles ont toutes un attribut `alt`.

Il reste possible de définir un texte alternatif vide si aucune description ne semble cohérente pour l'image, mais soyez vigilant. Il est préférable de veiller à ce que la majorité de vos images indiquent un texte cohérent. Voir [les recommandations du W3C \(EN\)](#).

100/100

#78

Cette page définit des balises `<h1>` et `<h2>`

Il est recommandé de placer vos mots clés dans les balises de titres (h1 et h2 au minimum). Les moteurs de recherche utilisent les balises h1, h2 et h3 pour le référencement (SEO). Cette page contient:

- 1 balise(s) `<h1>`
- 2 balise(s) `<h2>`
- 4 balise(s) `<h3>`

100/100

#81

Cette page utilise uniquement des formats d'images standards

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.

100/100

#83

✓ Un titre est défini sur cette page

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 :

Amazon.fr : livres, DVD, jeux vidéo, MP3, high-tech, informatique ...

<https://www.amazon.fr/> ▼

Achat et vente en ligne parmi des millions de produits en stock. Livraison gratuite à partir de 25€. Vos articles à petits prix : culture, high-tech, mode, jouets, sport ...

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`.

Voici le titre défini :

La Chouette Agence Webdesign

100/100

#84

✓ Une description est définie sur cette page

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:

Amazon.fr : livres, DVD, jeux vidéo, MP3, high-tech, informatique ...

<https://www.amazon.fr/> ▼

Achat et vente en ligne parmi des millions de produits en stock. Livraison gratuite à partir de 25€. Vos articles à petits prix : culture, high-tech, mode, jouets, sport ...

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.](#)

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 chouette agence, une grande agence de web design basée à Lyon.

100/100

#90

✓ Aucun paramètre de requête n'est détecté dans l'URL

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`.

100/100

#2399

✓ Vous facilitez la lecture de la page par les réseaux sociaux

Vous pouvez aider les réseaux sociaux à comprendre votre page en utilisant les propriétés Open Graph.

Les propriétés de Open Graph

Plusieurs propriétés permettent aux réseaux sociaux d'en savoir plus sur le contenu de la page. Nous recommandons d'utiliser au moins les propriétés requises :

- `<meta property="og:title" content="Le titre" />` [Exemple](#)
- `<meta property="og:type" content="Le type" />` [Exemple](#)
- `<meta property="og:url" content="http://url.com/" />` [Exemple](#)
- `<meta property="og:image" content="http://image.jpg" />` [Exemple](#)

Ces informations permettent d'améliorer l'interprétation de la page par les réseaux sociaux, y compris Facebook. [En savoir plus sur Open Graph.](#)

Cette page indique bien ses principales caractéristiques aux réseaux sociaux.

100/100

#2432

✓ La description de la page a une taille cohérente

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 :

Amazon.fr : livres, DVD, jeux vidéo, MP3, high-tech, informatique ...
<https://www.amazon.fr/> ▼
Achat et vente en ligne parmi des millions de produits en stock. Livraison gratuite à partir de 25€. Vos articles à petits prix : culture, high-tech, mode, jouets, sport ...

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 nécessairement pénalisé si votre texte est trop grand ou trop court. [En savoir plus sur le fonctionnement de Google sur ce sujet.](#)

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 chouette agence, une grande agence de web design basée à Lyon.

100/100

#2444

✓ Le titre de la page a une taille adéquate

Indiquez un titre assez court pour maximiser les chances qu'il soit utilisé sur les SERPs (Search Engine Results Page).

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 :

Amazon.fr : livres, DVD, jeux vidéo, MP3, high-tech, informatique ...

<https://www.amazon.fr/> ▼

Achat et vente en ligne parmi des millions de produits en stock. Livraison gratuite à partir de 25€. Vos articles à petits prix : culture, high-tech, mode, jouets, sport ...

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.

Définir un titre assez court

Le titre de la page est renseigné grâce à la balise `<title>`, qui doit être placée dans la balise `<head>`, au début du code.

Celui-ci doit rester assez court afin de pouvoir être affiché au sein d'une page de résultats. Nous vous conseillons d'**établir un titre en moins de 75 caractères** ([voir plus d'informations \(EN\)](#)).

Le titre de cette page comporte 28 caractères :

La Chouette Agence Webdesign

100/100

#2467

✓ Vos titres ne sont pas vides

Les balises `<h1>`, `<h2>` et `<h3>` envoient des signaux aux robots destinés au référencement de votre page, et doivent donc contenir des mots-clés.

Les titres de cette page renseignent bien un contenu.

✓ Cette page indique un fil d'Ariane aux moteurs de recherche

Vous pouvez mettre en valeur vos pages sur les moteurs de recherche en indiquant un fil d'Ariane via des données structurées.

Fil d'Ariane et moteurs de recherche

Le fil d'Ariane permet d'améliorer visuellement les résultats correspondant à votre site lors d'une recherche. Sur Google, vous pouvez par exemple le retrouver à la place de l'URL :

Tim Minchin: how my beat poem took the world by Storm ...
www.theguardian.com > **Arts > Books > Poetry** ▾ The Guardian ▾
Oct 6, 2014 - Join **Tim Minchin** for a live streamed Q&A on Monday 6 October (7 ...
extract from the introduction to *Storm* by **Tim Minchin** with DC Turner and ...

Sur une recherche effectuée via un appareil fixe, le fil d'Ariane remplace l'URL uniquement s'il est défini sur la page.

Pour la recherche mobile, Google affiche systématiquement les fils d'Ariane depuis avril 2015. Si vous n'avez pas défini de fil d'Ariane sur votre site, il sera généré automatiquement (à partir de l'URL notamment). Nous vous conseillons donc d'explicitier cette information dans votre code source, pour vous permettre de garder la main sur son affichage.

Définir son fil d'Ariane

Veillez à définir le fil d'Ariane [selon l'une des manières indiquées par Google](#). Exemple :

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "BreadcrumbList",
  "itemListElement": [{
    "@type": "ListItem",
    "position": 1,
    "name": "Books",
    "item": "https://example.com/books"
  }, {
    "@type": "ListItem",
    "position": 2,
    "name": "Authors",
    "item": "https://example.com/books/authors"
  }, {
    "@type": "ListItem",
    "position": 3,
    "name": "Ann Leckie",
    "item": "https://example.com/books/authors/annleckie"
  }, {
    "@type": "ListItem",
    "position": 4,
    "name": "Ancillary Justice",
    "item": "https://example.com/books/authors/ancillaryjustice"
  }
]
}
</script>
```

Exemple

Sur la page analysée ici, le fil d'Ariane est bien présent.

Vos principaux axes d'optimisation

0/100

#2433

! Il manque une politique de sécurité sur la provenance de vos ressources

Protégez votre site Web contre les attaques de type XSS (Cross-Site Scripting) en mettant en place une politique restrictive de sécurité du contenu.

Les attaques XSS

Les attaques XSS sont un type d'attaque dans laquelle des données malveillantes sont malicieusement ajoutées aux sites Web. Le nombre de vulnérabilités permettant ces attaques est assez important, c'est pourquoi il est aussi utile de les prévenir que de limiter leurs effets néfastes.

Vous pouvez protéger vos pages contre ces attaques en limitant l'exécution à des portions de code identifiées par le domaine auquel elles appartiennent, ou légitimées par un identifiant d'intégrité. Le code ne correspondant pas à cette politique de sécurité ne sera pas exécuté et l'utilisateur sera informé.

Configurer un en-tête HTTP "Content-Security-Policy" (CSP)

Pour prévenir ou limiter les dégâts potentiels d'une attaque XSS, vous devez configurer votre serveur afin que la réponse de la ressource principale contienne l'en-tête HTTP "Content-Security-Policy".

Voici un exemple d'utilisation de cet en-tête :

```
Content-Security-Policy: script-src 'self' https://apis.google.com
```

Exemple

Dans ce cas, seuls les scripts provenant de l'hôte courant ou de <https://apis.google.com> seront exécutés.

Découvrez-en davantage sur les Content Security Policy dans [les directives du W3C](#).

Soyez vigilant, si l'en-tête est mal configuré, certains de vos contenus, scripts, ou encore styles pourront être bloqués, ce qui pourrait engendrer des effets de bords non souhaités. De plus, les restrictions s'appliquent à toutes les pages du site. Nous vous conseillons de tester les différentes pages de votre site avant de déployer l'instruction dans votre environnement de production.

Aucune CSP n'a été détectée sur cette page : elle est plus facilement exposée à des attaques de type XSS.

! Cette page est exposée à des attaques du type "clickjacking"

Ne permettez pas à des personnes malveillantes d'intégrer vos pages sur leur site.

Clickjacking

Ce type d'attaque consiste à intégrer votre page sur un site malveillant via des balises <frame> ou <iframe>. Ainsi il est possible de faire croire à un internaute qu'il est sur votre propre page. L'internaute peu averti sera en confiance et sera potentiellement amené à saisir des informations que le site malveillant sera à même d'intercepter.

Vous devez donc indiquer quels domaines sont autorisés à intégrer votre page.

Comment se prémunir du Clickjacking ?

Il y a deux méthodes permettant de se protéger contre cette attaque.

1/ Utilisez un en-tête HTTP "X-Frame-Options". Configurez votre serveur de telle sorte que la réponse de la ressource principale contienne l'en-tête HTTP "X-Frame-Options".

Trois types de valeurs peuvent être définies :

- **DENY** pour refuser toute frame ou iframe intégrant la page
- **SAMEORIGIN** pour n'autoriser que les frames provenant du même nom de domaine
- **ALLOW-FROM uri** pour préciser les domaines pouvant intégrer la page dans une frame ([non compatible avec tous les navigateurs \(EN\)](#))

2/ Définissez une directive `frame-ancestors` explicite dans votre politique de sécurité. Cette approche, plus récente et supportée par moins de navigateurs, vous permettra cependant d'autoriser plusieurs domaines plutôt qu'uniquement le domaine courant. Configurer cette directive avec la valeur 'none' revient à mettre en place l'en-tête **X-Frame-Options: DENY**.

Quelle approche choisir ? Si vous n'avez que le domaine courant à autoriser, mettez en place les deux politiques de sécurité pour une meilleure compatibilité avec les anciens navigateurs. Si vous souhaitez autoriser plusieurs domaines à intégrer le vôtre, privilégiez la directive de sécurité "frame-ancestors".

Ni l'en-tête "X-Frame-Options" ni la directive "frame-ancestors" ne sont définis sur cette page, vous êtes plus facilement exposé au clickjacking.

! Désactivez la détection automatique du type des ressources

Protégez-vous de l'exploitation malveillante du MIME Sniffing.

Le MIME-Type sniffing

Les navigateurs Internet Explorer et Chrome disposent d'une fonctionnalité dite de "MIME-Type sniffing", qui consiste à détecter automatiquement le type d'une ressource web. Ainsi, une ressource déclarée comme étant une image pourra être interprétée comme un script si tel est son contenu.

Une personne malveillante pourrait profiter d'un envoi de fichier sur votre site par exemple pour injecter du code malveillant. Nous vous conseillons de désactiver le MIME-Type sniffing pour limiter les effets de l'envoi d'un tel fichier.

Chrome a travaillé sur une fonctionnalité appelée [Isolation de Site \(EN\)](#) qui permet d'atténuer considérablement l'exploitation de ces types de vulnérabilités. L'isolation du site est plus efficace lorsque les types MIME sont corrects.

La solution : configurer un en-tête HTTP "X-Content-Type-Options"

Ajoutez **dans les réponses de chacune de vos ressources** l'en-tête HTTP "X-Content-Type-Options", associé à la valeur "nosniff". Cela vous permettra de vous prémunir du risque d'une mauvaise interprétation de vos ressources.

Sur cette page, **vous devriez configurer les ressources suivantes**, qui risquent d'être mal interprétées :

- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/css/bootstrap.min.css
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/Sass/main.css
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/js/jquery-2.1.0.min.js
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/js/bootstrap.min.js
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/js/blocs.min.js
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/js/gmaps.min.js
- [https://gflahaut.github.io/GwendolineFlahaut_4_03122020/\[...\]jquery.touchSwipe.min.js](https://gflahaut.github.io/GwendolineFlahaut_4_03122020/[...]jquery.touchSwipe.min.js)

Saviez-vous que ?

#2567

Votre certificat SSL

Votre certificat SSL expirera le 14/04/2022. Mettez à jour votre certificat avant cette date.

Que se passera-t-il si mon certificat expire ?

Laisser expirer un certificat peut avoir des conséquences pour les utilisateurs finaux qui verront alors de nombreux messages d'erreur ou d'alerte durant leur navigation sur le site, les avertissant à juste titre de risques de fraude, de vol d'identité ou d'interception du trafic. Ces alertes peuvent avoir un impact très négatif sur la perception par l'utilisateur du domaine visité.

#2474

23 ressources de cette page sont destinées à une utilisation publique

Par défaut, le navigateur n'accepte d'effectuer des requêtes AJAX, ou encore de récupérer des polices web, que sur le nom de domaine de la page. Une police de caractère définie sur toto.com ne pourra donc être utilisée que par les pages de toto.com. Cela évite une mauvaise utilisation de vos ressources par n'importe quel site.

Certaines ressources sont publiques, et veulent explicitement être mise à disposition de chacun (ex: les Google Fonts). Dans ce cas, l'en-tête HTTP `Access-Control-Allow-Origin` peut-être utilisé, avec la valeur `"*"`. Vous ne devez cependant utiliser cette propriété que si votre ressource a bien pour objectif d'être utilisée par le plus grand nombre. Sinon, nous vous conseillons de garder la valeur par défaut, ou de définir un nom de domaine spécifique dans l'en-tête `Access-Control-Allow-Origin`.

Vous devriez porter votre vigilance sur les ressources suivantes, qui utilisent un en-tête HTTP `Access-Control-Allow-Origin:*`. Assurez vous qu'elles sont effectivement destinées à être utilisées par des pages de tous horizons :

- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/css/bootstrap.min.css
- [gflahaut.github.io/GwendolineFlahaut_4_03122020/\[...\]g/la-chouette-agence.jpg](https://gflahaut.github.io/GwendolineFlahaut_4_03122020/[...]g/la-chouette-agence.jpg)
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/Sass/main.css
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/img/jpg/logo.jpg
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/img/jpg/1.jpg
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/img/jpg/2.jpg
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/img/jpg/3.jpg
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/img/jpg/4.jpg
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/js/jquery-2.1.0.min.js
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/js/bootstrap.min.js
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/js/blocs.min.js
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/js/gmaps.min.js
- [gflahaut.github.io/GwendolineFlahaut_4_03122020/\[...\]jquery.touchSwipe.min.js](https://gflahaut.github.io/GwendolineFlahaut_4_03122020/[...]jquery.touchSwipe.min.js)
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/img/svg/presentation.svg
- et 8 autres

#2528

i 2 cookies peuvent être altérés par le client

Un cookie doit uniquement être manipulé par le serveur. Il n'est pas recommandé de l'utiliser au niveau du navigateur web.

Les cookies

Les cookies sont placés par le serveur web sur le navigateur web via l'en-tête HTTP `Set-Cookie`. Le navigateur retransmet ensuite ces informations lors des prochaines requêtes au serveur via l'en-tête HTTP `Cookie`.

Vous devez garantir que ces cookies ne peuvent être altérés que par le serveur.

L'instruction `HttpOnly`

En ajoutant l'instruction `HttpOnly` au niveau de l'en-tête HTTP `Set-Cookie`, le serveur informe le navigateur que le cookie ne peut être modifié. Le client se contente de récupérer et retourner le cookie envoyé par le serveur : il transite donc sur le protocole HTTP mais ne peut pas être modifié via du JavaScript par exemple.

Les cookies pourraient être exploités en cas de faille XSS, vous devriez considérer l'ajout de l'instruction `HttpOnly` lors du `Set-Cookie` :

https://github.com/gflahaut/GwendolineFlahaut_4_03122020/blob/main/favicon.jpg

- `set-cookie: _octo=GH1.1.1575861168.1619631641; Path=/; Domain=github.com; Expires=Thu, 28 Apr 2022 17:40:41 GMT; Secure; SameSite=Lax`

<https://github.com/AndroidprojectIUT4/blob/main/favicon.jpg>

- `set-cookie: _octo=GH1.1.1661570030.1619631641; Path=/; Domain=github.com; Expires=Thu, 28 Apr 2022 17:40:41 GMT; Secure; SameSite=Lax`

#2553

i Vos ressources tierces sont-elles sécurisées par des contrôles d'intégrité ?

Si cette page charge des contenus provenant de tiers, assurez-vous de l'intégrité de ces données.

Contrôle d'intégrité SRI

Pour vous assurer qu'une ressource provenant d'une tierce partie n'a pas été altérée, utilisez un contrôle d'intégrité SRI (SubResource Integrity). Pour cela, ajoutez l'attribut `integrity` sur les balises `<script>` et `<link>` qui chargent ces ressources. Exemple :

```
<script src="https://exemple.com/exemple-framework.js"
  integrity="sha384-oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/uxy9rx7HNQIGY1kPzQho1wx4JwY8wC"
  crossorigin="anonymous">
</script>
```

Exemple

L'attribut `integrity` a pour valeur une empreinte, correspondant au SHA du contenu souhaité, encodé en base64. Le navigateur comparera alors ce SHA avec celui du contenu téléchargé, pour déterminer si la ressource correspond bien à celle attendue.

Plusieurs outils vous aideront dans la création de cette empreinte. En ligne de commande, vous pouvez utiliser openssl. Des outils en ligne existent par ailleurs, tels que srihash.org ou report-uri.io. [En savoir plus sur SubResource Integrity.](#)

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#2397

✓ Cette page utilise des contenus sécurisés

Cette page a été transmise via le protocole HTTPS, et toutes ses ressources sont récupérées en utilisant ce même protocole. Il n'y a ainsi pas de vulnérabilité liée à des contenus mixtes.

100/100

#2463

✓ **Votre serveur communique uniquement en HTTPS avec vos internautes**

Prémunissez vous d'attaques du type "[man in the middle](#)" en vous assurant de communiquer uniquement en HTTPS avec le serveur.

L'en-tête HTTP Strict Transport Security (HSTS)

Lorsque vous communiquez avec un serveur via une connexion sécurisée, il est recommandé que toutes les requêtes envoyées vers ce serveur utilisent le protocole HTTPS. L'en-tête HTTP HSTS permet d'indiquer au navigateur que toutes les requêtes envoyées vers le domaine concerné doivent se faire via HTTPS. Ainsi si l'URL est présentée sous la forme "http://...", le navigateur web va automatiquement la remplacer par "https://...".

Nous vous conseillons cependant de ne mettre en place cet en-tête à l'unique condition que votre site entier desserve ses ressources en HTTPS.

Cette page définit un en-tête HSTS.

100/100

#2509

✓ **La version sécurisée est systématiquement utilisée**

Redirigez les utilisateurs utilisant la version HTTP de la page vers HTTPS.

Redirection vers HTTPS

Même si une page est sécurisée, il arrive que les visiteurs continuent à utiliser la version HTTP (via un lien externe, ou parce qu'ils l'ont ajouté en HTTP à leurs favoris, par exemple). Mettez toujours en place une redirection pour que les utilisateurs accédant à la page en utilisant le protocole HTTP soit redirigés vers une version sécurisée.

Utilisez l'interface d'administration de votre hébergement Web ou contactez l'hébergeur pour mettre en place une redirection automatique de la version HTTP vers la version sécurisée.

Vos utilisateurs accédant à cette page via le protocole HTTP sont systématiquement redirigés vers la version sécurisée de la page.

100/100

#2521

✓ **Les cookies de la page sont sécurisés**

Un cookie transmis au navigateur via le protocole HTTPS doit pouvoir transiter uniquement sur une connexion sécurisée (sauf cas particulier).

Les cookies

Les cookies sont placés par le serveur web sur le navigateur web via l'en-tête HTTP **Set-Cookie**. Le navigateur renvoie ensuite ces informations lors des prochaines requêtes au serveur via l'en-tête HTTP **Cookie**. Si votre serveur fournit un cookie sur une connexion sécurisée (HTTPS), c'est probablement que ce cookie contient des informations sensibles : vous devez alors garantir qu'il ne pourra pas être exploité sur une connexion non sécurisée.

L'instruction Secure

En ajoutant l'instruction **Secure** au niveau de l'en-tête HTTP **Set-Cookie**, le serveur informe au navigateur qu'il n'est autorisé à retransmettre le cookie que sur des requêtes sécurisées.

Attention : vérifiez que la redirection HTTP vers HTTPS est bien en place sur votre site. Dans le cas contraire, le cookie Secure pourrait ne pas être envoyé lors d'une requête HTTP.

Cookies d'infrastructures mutualisées

Avec certains hébergeurs, notamment sur des infrastructures mutualisées, il est possible que des cookies soient placés sans aucun contrôle de votre part. Les informations qu'ils contiennent ne sont normalement pas sensibles, vous pouvez le vérifier et ignorer cet avertissement si tel est votre cas.

Vos principaux axes d'optimisation

0/100

#2436

! 3 images sont redimensionnées côté navigateur

Si vos images sont plus grandes que leur zone d'affichage, le navigateur téléchargera des données inutiles et procédera à des redimensionnements disgracieux.

Éviter le redimensionnement d'images côté navigateur

Lorsqu'il souhaite afficher une image dans votre page, le navigateur fait tout son possible pour l'adapter à sa surface de rendu. Si l'image est trop grande, il la réduit.

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.

Comme les algorithmes embarqués dans les navigateurs ne sont pas aussi bons que ceux des outils dédiés à la manipulation d'images, vous obtiendrez un résultat visuel plus satisfaisant en redimensionnant vos images en amont, plutôt qu'en laissant le navigateur le faire.

Utiliser des Images Adaptatives (Responsive)

Plusieurs méthodes existent, permettant de servir des images adaptées au navigateur quelle que soit la résolution de l'écran ou la densité de pixels du matériel. Nous vous conseillons de lire les ressources suivantes :

- ["Images adaptatives", sur le Mozilla Developer Network \(MDN\)](#)
- [Picturefill, permet d'utiliser l'élément <picture> dans des navigateurs qui ne le supportent pas encore \(EN\)](#)
- [RICG, groupe de travail sur les images responsive \(EN\)](#)

Ne redimensionnez pas les images suivantes :

- [gflahaut.github.io/Gwendoli\[/...\]/jpg/1.jpg](https://gflahaut.github.io/Gwendoli[/...]/jpg/1.jpg) (taille d'affichage : 270x270)
- [gflahaut.github.io/Gwendoli\[/...\]/g/logo.jpg](https://gflahaut.github.io/Gwendoli[/...]/g/logo.jpg) (taille d'affichage : 100x100)
- [gflahaut.github.io/Gwendoli\[/...\]/jpg/2.jpg](https://gflahaut.github.io/Gwendoli[/...]/jpg/2.jpg) (taille d'affichage : 270x270)

Sur cette page, 2 images sont redimensionnées de moins de 40 % de leur taille. Plusieurs techniques de Responsive Web Design reposent sur le redimensionnement d'images dont les tailles sont proches de leurs surfaces de rendu respectives. Il est donc possible que des redimensionnements de ce type soit désirés, et nous ne les pénalisons pas. Cependant, nous vous invitons à vous assurer qu'il n'affecte pas les performances de la page :

- [gflahaut.github.io/Gwendoli\[/...\]/jpg/4.jpg](https://gflahaut.github.io/Gwendoli[/...]/jpg/4.jpg) (taille d'affichage : 270x270)
- [gflahaut.github.io/Gwendoli\[/...\]/jpg/3.jpg](https://gflahaut.github.io/Gwendoli[/...]/jpg/3.jpg) (taille d'affichage : 270x270)

Saviez-vous que ?

#2443

Cette page charge 607 ko de données

Une page chargeant trop de données ralentit sa vitesse d'affichage, d'autant plus sur les connexions bas débit. Cela peut également générer de la frustration pour vos internautes qui sont limités par des forfaits data (voir whatdoesmysitecost.com).

Situer le poids de ma page web

Selon HTTPArchive, en juillet 2019, [le poids moyen d'une page web était de 1,95Mo](#).

Comment réduire le poids de ma page ?

Vous pouvez vous reporter à notre catégorie "Volume de données" pour voir quelles sont les optimisations possibles dans votre cas. De manière générale, les images sont souvent à pointer du doigt.

Veillez aussi à construire vos pages web de manière à charger d'abord ce qui est essentiel à l'internaute en priorité (optimisation du chemin critique de rendu).

Pour tout autre contenu (plugins de réseaux sociaux, publicités, contenus situés en bas de page...), il est préférable de différer le chargement (asynchrone, lazy-loading...), pour ne pas pénaliser les contenus prioritaires.

Afin de vous aider à établir vos priorités d'optimisation, voici la répartition du poids de la page par type de ressources :

- **Images** : 85,16% du poids total
- **JavaScript** : 9,22% du poids total
- **CSS** : 4,03% du poids total
- **Textes** : 1,60% du poids total

Par ailleurs, voici les 10 ressources les plus lourdes qui sont téléchargées lors du chargement de votre page :

- [gflahaut.github.io/GwendolineFlahaut\[...\]/nce-banniere.jpg](https://gflahaut.github.io/GwendolineFlahaut[...]/nce-banniere.jpg) (184 ko)
- [gflahaut.github.io/GwendolineFlahaut\[...\]/20/img/jpg/1.jpg](https://gflahaut.github.io/GwendolineFlahaut[...]/20/img/jpg/1.jpg) (88 ko)
- [gflahaut.github.io/GwendolineFlahaut\[...\]/presentation.jpg](https://gflahaut.github.io/GwendolineFlahaut[...]/presentation.jpg) (85 ko)
- [gflahaut.github.io/GwendolineFlahaut\[...\]/20/img/jpg/2.jpg](https://gflahaut.github.io/GwendolineFlahaut[...]/20/img/jpg/2.jpg) (57 ko)
- [gflahaut.github.io/GwendolineFlahaut\[...\]/20/img/jpg/3.jpg](https://gflahaut.github.io/GwendolineFlahaut[...]/20/img/jpg/3.jpg) (41 ko)
- [gflahaut.github.io/GwendolineFlahaut\[...\]/20/img/jpg/4.jpg](https://gflahaut.github.io/GwendolineFlahaut[...]/20/img/jpg/4.jpg) (38 ko)
- [gflahaut.github.io/GwendolineFlahaut\[...\]/jery-2.1.0.min.js](https://gflahaut.github.io/GwendolineFlahaut[...]/jery-2.1.0.min.js) (30 ko)
- [gflahaut.github.io/GwendolineFlahaut\[...\]/bootstrap.min.css](https://gflahaut.github.io/GwendolineFlahaut[...]/bootstrap.min.css) (20 ko)
- [gflahaut.github.io/GwendolineFlahaut\[...\]/bootstrap.min.js](https://gflahaut.github.io/GwendolineFlahaut[...]/bootstrap.min.js) (10 ko)
- [gflahaut.github.io/GwendolineFlahaut\[...\]/js/gmaps.min.js](https://gflahaut.github.io/GwendolineFlahaut[...]/js/gmaps.min.js) (10 ko)

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#2325

Vos contenus sont bien appelés par des URLs uniques

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.

100/100

#2364

✓ Vous n'avez pas besoin de minifier vos ressources HTML

Vous avez réduit la taille de vos ressources HTML. En savoir plus sur la [réduction de la taille des ressources HTML](#).

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.

100/100

#2384

✓ La compression des ressources est activée

Vous avez activé la compression. En savoir plus sur l'[activation de la compression](#).

100/100

#2387

✓ Vos ressources CSS sont minifiées ou n'ont pas besoin de l'être

Vous avez réduit la taille de vos ressources CSS. En savoir plus sur la [réduction de la taille des ressources CSS](#).

De nombreux outils existent pour minifier des fichiers CSS. C'est le cas de [YUI Compressor](#) ou [cssmin.js](#), recommandés par Google.

100/100

#2388

✓ Vos ressources JavaScript sont minifiées

Vous avez réduit la taille de votre contenu JavaScript. En savoir plus sur la [réduction de la taille des ressources JavaScript](#).

De nombreux outils existent pour minifier des fichiers JavaScript. C'est le cas de [YUI Compressor](#) ou [JSMIn](#), recommandés par Google.

100/100

#2389

✓ Vos images sont optimisées

Vous avez optimisé vos images. En savoir plus sur l'[optimisation des images](#). Les images peuvent comporter des données inutiles à leur utilisation sur le web, ce qui peut augmenter considérablement leur poids. Des outils existent pour supprimer automatiquement ces données, sans perte de qualité, et ainsi réduire le poids de vos images.

Voici quelques formats de fichiers et des outils d'optimisation correspondants, parfois inclus dans les outils graphiques comme Photoshop ou GIMP :

- PNG: Zopfli-png, PNGOUT, OptiPNG, AdvPNG, PNGCrush, PNGQuant...
- JPG: JPEGOptim, MozJPEG, Jpegtran, Guetzli...

Ces algorithmes sont utilisés par des logiciels comme [FileOptimizer](#) (Windows), [ImageOptim](#) (Mac) ou [Trimimage](#) (Linux), qui trouveront pour chaque image la meilleure optimisation et le meilleur encodage, avec ou sans perte de qualité.

100/100

#2421

✓ Toutes vos ressources ont un poids inférieur à 1Mo

Les fichiers trop lourds sont à proscrire sur une page web. Assurez vous que tous les éléments du fichier sont nécessaires au chargement de la page.

100/100

#2446

✓ 4 images (223 ko) sont chargées trop tôt

Chargez en priorité les images au-dessus de la ligne de flottaison, c'est à dire celles qui sont visibles sans défilement (scroll) de l'utilisateur.

Page web et images

Les images représentent en moyenne plus de 60 % du poids d'une page web. En chargeant uniquement les images visibles sans scroll de l'utilisateur au chargement initial de la page, vous limitez la bande passante consommée, pour votre serveur comme pour les internautes et mobinautes. Si une image n'est visible que suite à une action de l'utilisateur comme le scroll, il est alors préférable de la charger à la demande, uniquement si elle devient nécessaire pour l'internaute (lazyloading).

Comment mettre en place le lazyloading ?

Vous pouvez vous informer sur les fonctionnalités et plugins proposés par les frameworks/CMS utilisés par votre page, qui vous proposeront probablement une solution clef en main.

Si vous devez implémenter le Lazy Loading vous-même, le moyen le plus simple est peut-être d'utiliser la fonctionnalité native introduite par Chrome en 2019 : sur toutes les images ciblées, ajoutez `loading="lazy"`. Cette fonctionnalité est [supportée par tous les navigateurs récents \(ou le sera dans un avenir proche\)](#). En attendant : sur les navigateurs qui ne prennent pas en charge le Lazy Loading natif, l'attribut n'a aucun effet.

```

```

Exemple

Le Lazy Loading natif est cependant la solution qui offre le moins de contrôle. De plus, la navigation est parfois trop zélée et charge des images qui n'ont pas besoin d'être chargées si rapidement ([en savoir plus \(en anglais\)](#)).

Certaines bibliothèques JavaScript peuvent vous aider à implémenter le Lazy Loading avec plus de granularité.

🔗 Avec jQuery vous pouvez disposer d'un plugin comme [dédié au lazyloading d'images](#).

Sur cette page, le chargement de 4 images peut être différé.

- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/img/jpg/1.jpg
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/img/jpg/2.jpg
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/img/jpg/4.jpg
- https://gflahaut.github.io/GwendolineFlahaut_4_03122020/img/jpg/3.jpg

Si le chargement anticipé de certaines de ces images est justifié, bien qu'elles ne soient pas immédiatement visibles, vous pouvez leur ajouter l'attribut `loading="eager"`, qui précise clairement ce comportement.

```

```

Exemple

100/100

#2450

✓ Cette page n'envoie pas de cookies > 100 ko

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](#).

100/100

#2453

✓ Cette page n'envoie pas trop de cookies

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](#).

100/100

#2486

✓ Les réponses de redirection sont légères

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.

100/100

#2487

✓ Les réponses en erreur sont légères

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.

100/100

#2490

✓ La favicon est légère

La favicon se doit d'être la plus légère possible.

La favicon sur le web

Cette petite image permet de représenter votre page via une petite icône, sur l'onglet du navigateur par exemple.

Que faire ?

Une favicon doit idéalement être < 10ko.

Le poids de la favicon (https://github.com/gflahaut/GwendolineFlahaut_4_03122020/blob/main/favicon.jpg) n'est pas trop important (-1octets).

Vos principaux axes d'optimisation

75/100

#2302

! Évitez les sélecteurs jQuery trop spécifiques

Vous utilisez des sélecteurs jQuery trop spécifiques qui diminuent les performances de votre site web : [en savoir plus](#).
Voici un exemple de bonne utilisation de jQuery:

```
$( ".data table.firstClass td.secondClass" );  
// Plus performant: supprimer le sélecteur  
// intermédiaire si possible  
$( ".data td.secondClass" );
```

Exemple

Si possible, **simplifiez les sélecteurs suivants** :

[gflahaut.github.io/GwendolineFlahaut_4_\[...\]20/js/blocs.min.js](https://github.com/gflahaut/GwendolineFlahaut_4_[...]20/js/blocs.min.js)

- `$(".blocsapp-special-menu blocsnav li")`

Saviez-vous que ?

#2303

i En savoir plus sur la performance avec jQuery

jQuery est la librairie JavaScript la plus utilisée aujourd'hui. Rendez votre site plus performant en respectant ses bonnes pratiques. Pour connaître les fondamentaux de la performance jQuery, nous vous conseillons de lire le lien suivant:
<http://learn.jquery.com/performance/>.

Bravo ! Vous respectez les bonnes pratiques suivantes

100/100

#2299

✓ Aucune manipulation de DOM dans des boucles détectée

Travailler avec le DOM a un coût. Si vous avez besoin d'ajouter des éléments à un nœud, il est préférable de les ajouter tous d'un coup plutôt qu'un par un.

Aucune de vos boucles ne contient de méthode jQuery de modification du DOM.

100/100

#2470

✓ Une seule version de jQuery est chargée

Charger plusieurs fois jQuery revient à alourdir le poids de la page et complexifie sa maintenabilité.

Une seule version de jQuery a été détectée, la page est en accord avec la bonne pratique.