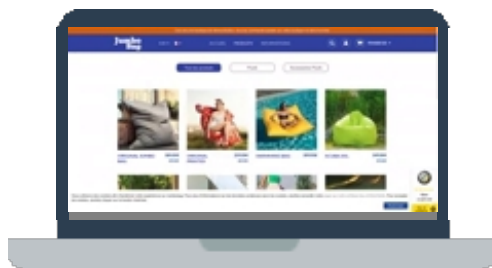




Rapport d'analyse performance et qualité



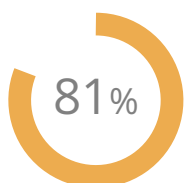
<https://dev.jumbobag.fr/products.html>

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.

Sommaire

Résumé	3
Conseils et bonnes pratiques	4
Accessibilité	4
Nombre de requêtes	5
Optimisation du rendu	7
Politique de cache	8
Qualité	10
Référencement (SEO)	13
Sécurité	16
Volume de données	19
Apache	21
jQuery	22

Résumé



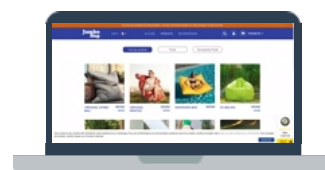
Problèmes



Améliorations



Succès



VISITEUR SIMULÉ :



Chrome



Paris

8.0/1.5Mbps (Latence : 50 ms)

Premier octet



Début affichage



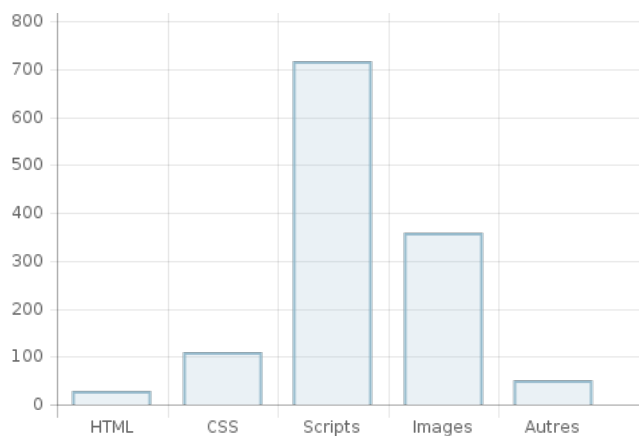
Fin chargement



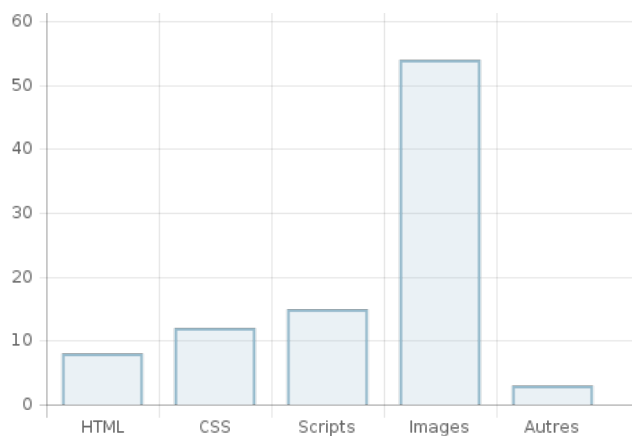
Speed Index



Répartition du poids



Répartition du nombre de requêtes



Technologies détectées



Apache



Font Awesome



Google Analytics



Google Font API



Google Tag Manager



Magento



Modernizr



Prototype



Twitter Bootstrap



jQuery



PHP

Conseils et bonnes pratiques

Accessibilité

Vos principaux axes d'optimisation

0/100

#2523

! 3 labels ne sont pas 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" >
```

Les labels suivants ne définissent pas d'attribut `for` :

- `<label>`
- `<label>`
- `<label>`

0/100

#2381

! Expliquez le but de chaque champ de formulaire

Améliorez l'expérience utilisateur sur votre site en spécifiant une étiquette descriptive pour chaque champ de formulaire.

Un formulaire est composé de plusieurs champs qui doivent être les plus explicites possibles pour que l'utilisateur comprenne rapidement leur fonction.

Définir un label

Pour indiquer l'objectif d'un champ, nous vous recommandons l'utilisation de la balise `label` :

```
<label for="nom">Renseignez votre nom :</label>
<input id="nom" type="text" name="nom">
```

Exemple

Si vous ne souhaitez pas afficher de label, vous pouvez utiliser les attributs `aria-label` ou `title` (ce dernier n'est cependant pas supporté par certaines liseuses d'écran).

En savoir plus sur [l'association de texte à un champ de formulaire](#).

Cette page contient **2 champs sans explications sur leur objectif** :

- `<select onchange="setLocation(this.value)">`
- `<select onchange="setLocation(this.value)">`

Vos principaux axes d'optimisation

20/100

#2442

! Économisez 13 requêtes à l'aide de sprites CSS

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;  
}
```

Exemple

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

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

Exemple

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.

14 images peuvent être regroupées au sein d'une sprite CSS. Le nom de domaine jumbobag devrait utiliser cette technique pour les ressources suivantes :

- <https://dev.jumbobag.fr/skin/frontend/jumbobag/default/images/twitter-footer.png>
- [dev.jumbobag.fr/media/wysiwyg/1416523839_payment\[...\]method_card_visa-128.png](https://dev.jumbobag.fr/media/wysiwyg/1416523839_payment[...]method_card_visa-128.png)
- [dev.jumbobag.fr/skin/frontend/jumbobag/default/i\[...\]ages/facebook-footer.png](https://dev.jumbobag.fr/skin/frontend/jumbobag/default/i[...]ages/facebook-footer.png)
- <https://dev.jumbobag.fr/skin/frontend/jumbobag/default/images/new-label.png>
- [dev.jumbobag.fr/media/wysiwyg/1416523842_payment\[...\]thod_master_card-128.png](https://dev.jumbobag.fr/media/wysiwyg/1416523842_payment[...]thod_master_card-128.png)
- https://dev.jumbobag.fr/skin/frontend/jumbobag/default/images/icon_myaccout.png
- <https://dev.jumbobag.fr/skin/frontend/jumbobag/default/images/mail-footer.png>
- https://dev.jumbobag.fr/skin/frontend/jumbobag/default/images/icon_search.png
- https://dev.jumbobag.fr/skin/frontend/jumbobag/default/images/arrow_switch.png
- https://dev.jumbobag.fr/skin/frontend/jumbobag/default/images/btn_overlay.png
- et 4 autres

88/100

#2419

Regroupez 4 fichiers CSS

Chaque requête HTTP représente un coût en terme de performance (roundtrip time, utilisation de bande passante...).

Il est ainsi préférable de faire une requête vers un fichier de 50ko de données plutôt que 10 requêtes vers des fichiers de 5ko.

Quelle répartition adopter ?

Répartissez vos styles en les intégrant directement dans votre code HTML ou en les regroupant dans des fichiers de taille plus conséquente. Nous vous conseillons l'emploi de cette dernière méthode pour profiter des mécanismes de cache.

Vous devriez reconsidérer les ressources suivantes :

- [dev.jumbobag.fr/skin/frontend/jumbobag/\[...\]jasonry-gallery.css](http://dev.jumbobag.fr/skin/frontend/jumbobag/[...]jasonry-gallery.css)
- [dev.jumbobag.fr/skin/frontend/jumbobag/\[...\]lt/css/style-r.css](http://dev.jumbobag.fr/skin/frontend/jumbobag/[...]lt/css/style-r.css)
- <https://dev.jumbobag.fr/options.css.php?store=default&cat=3>
- [dev.jumbobag.fr/media/css_secure/7d27f0\[...\]31dbd424fd4c3b.css](http://dev.jumbobag.fr/media/css_secure/7d27f0[...]31dbd424fd4c3b.css)

92/100

#2418

Regroupez 3 fichiers JavaScript

Chaque requête HTTP représente un coût en terme de performance (roundtrip time, utilisation de bande passante...).

Il est ainsi préférable de faire une requête vers un fichier de 50ko de données plutôt que 10 requêtes vers des fichiers de 5ko.

Quelle répartition adopter ?

Répartissez vos scripts en les intégrant directement dans votre code HTML ou en les regroupant dans des fichiers de taille plus conséquente. Nous vous conseillons l'emploi de cette dernière méthode pour profiter des mécanismes de cache.

Vous devriez reconsidérer les ressources suivantes :

- [dev.jumbobag.fr/skin/frontend/jumbobag/\[...\]jquery-2.0.2.min.js](http://dev.jumbobag.fr/skin/frontend/jumbobag/[...]jquery-2.0.2.min.js)
- [dev.jumbobag.fr/skin/frontend/jumbobag/\[...\]Jery.isotope.min.js](http://dev.jumbobag.fr/skin/frontend/jumbobag/[...]Jery.isotope.min.js)
- [dev.jumbobag.fr/skin/frontend/jumbobag/\[...\]lt/js/categorie.js](http://dev.jumbobag.fr/skin/frontend/jumbobag/[...]lt/js/categorie.js)

Vos principaux axes d'optimisation

0/100

#2353

! Différez l'utilisation du code JavaScript

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 ?

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>)

768.0 Ko du code JavaScript sont analysés lors du chargement initial de la page. Différez l'analyse de ce code pour éviter de bloquer l'affichage de la page.

- [dev.jumbobag.fr/me\[...\].js](#) (663.3 Ko)
- [dev.jumbobag.fr/sk\[...\].js](#) (75.5 Ko)
- [dev.jumbobag.fr/pr\[...\].html](#) (15.2 Ko de code JavaScript intégré)
- [dev.jumbobag.fr/sk\[...\].js](#) (14.0 Ko)

94/100

#2416

✓ Vous devriez réduire le nombre d'éléments du DOM

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 DOM de cette page contient trop d'éléments (1019 éléments).

Vos principaux axes d'optimisation

0/100

#70

! N'utilisez 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){i['GoogleAnalyticsObject']=r;
  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 devriez **regrouper les scripts suivants dans un/des fichiers distincts** :

- ```
(function(i, s, o, g, r, a, m) {
 i["GoogleAnalyticsObject"] = r;
 i[r] = i[r] || function() {
 (i[r].q = i[r].q || []).push(arguments)
 };
 g = s.createElement('script');
 g.async = true;
 g.src = 'https://www.google-analytics.com/analytics.js';
 m = s.getElementsByTagName('script')[0];
 m.parentNode.insertBefore(g, m);
})(window, document, 'script', 'https://www.google-analytics.com/analytics.js', 'ga');
```
- ```
//
var Translator = new Translate({"HTML tags are not allowed":"Les balises HTML ne sont pas autoris\u00e9es", "Please select an option":"Veuillez sélectionner une option"});
//]]&gt;</pre></li><li><pre>jQuery(window).load(function () {
  var size = 1;
  var button = 1;
  var button_class = "gallery-header-center-right-links-current";
  var normal_size_class = "normal-size";
  // ...
});</pre></li></ul></div><div data-bbox="90 927 277 966" data-label="Page-Footer"><p>Rapport généré<br/>par <b>dareboost</b></p></div><div data-bbox="333 940 660 959" data-label="Page-Footer"><p>Rapport d'analyse performance web</p></div><div data-bbox="701 940 901 958" data-label="Page-Footer"><p>23 nov. 2017 - 8 / 22</p></div>
```


✓ 8 requêtes provenant de partis tiers n'adoptent pas une politique de cache assez longue

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 :

```
maresource.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 8 ressources **n'ayant pas une date d'expiration assez lointaine** :

Les ressources suivantes sont hébergées par un parti tiers, il se peut donc qu'elles ne soient pas sous votre responsabilité. Vous devriez cependant considérer toute alternative possible à ces fichiers pour rester en accord avec la bonne pratique.

- [static-eu.payments-amazon.com/OffAmazon\[...\]/lpa/js/Widgets.js](#)
- [https://www.googletagmanager.com/gtm.js?id=GTM-T55T8X](#)
- [https://www.google-analytics.com/analytics.js](#)
- [widgets.trustedshops.com/js/X3B73FD19E1\[...\]/DD7DD1C76F880D7.js](#)
- [https://www.google-analytics.com/plugins/ua/ec.js](#)
- [https://eu.account.amazon.com/lwa/js/sdk/login1.js](#)
- [widgets.trustedshops.com/css/style.css\[...\]/elementId=&channel=](#)
- [widgets.trustedshops.com/css/tbAuthent\[...\]/F81DD7DD1C76F880D7](#)

Vos principaux axes d'optimisation

0/100

#2491

! Plusieurs éléments utilisent le même identifiant

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

L'identifiant suivant est utilisé plusieurs fois dans la page :

- `gallery-header-center-right`, utilisé 2 fois

! 52 sélecteurs CSS sont dupliqués

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.

Les règles CSS

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

Comment améliorer ce point ?

Il vous suffit de regrouper au sein d'une même règle les sélecteurs.
Par exemple, si votre fichier dispose des règles suivantes :

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

Vous devriez envisager leur réécriture de cette manière :

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

Si la duplication des sélecteurs est volontaire (architecture propre à votre code, etc), vous pouvez envisager l'utilisation d'un outil de minification CSS capable de faire cette transformation automatiquement.

3271 caractères peuvent être économisés. Les fichiers suivants définissent des sélecteurs identiques dans des règles différentes :

[dev.jumbobag.fr/media/css_secure/1ba70e6d1\[...\]3eeb34d16e31ad42.css](https://dev.jumbobag.fr/media/css_secure/1ba70e6d1[...]3eeb34d16e31ad42.css)

- `.toolbar` (3 fois)
- `footer .liens-footer ul` (3 fois)
- `footer .liens-footer` (2 fois)
- `.form-list .customer-name-prefix-suffix .name-lastname` (2 fois)
- `.form-list .customer-nam[...].fix .name-lastname input.input-text` (2 fois)
- `.breadcrumbs` (2 fois)
- `@media (min-width: 768px) .col-sm-3` (2 fois)
- `.toolbar .pager` (2 fois)
- et 25 autres

<https://dev.jumbobag.fr/options.css.php?store=default&cat=3>

- `.slider-container` (2 fois)
- `.footer-info` (2 fois)

<https://dev.jumbobag.fr/skin/frontend/jumbobag/default/css/style-r.css>

- `.stars-p` (10 fois)
- `.catalog-product-view .i[...].ntainer .swatch-img:not(:last-child)` (10 fois)
- `.product-view .add-to-cart button.button span span` (10 fois)
- `@media (max-width: 480px) .uvmg1` (10 fois)
- `@media (max-width: 480px) .thin-part` (10 fois)
- `@media (min-width: 1024px) .uvmg1` (10 fois)
- `.marg-plus-tissu` (10 fois)
- `.product-original-printe[...].issu .container .wheelmark h3:before` (2 fois)
- et 9 autres

! Le mot clé !important est utilisé 277 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 clé.

Voici la liste des !important détectés :

[dev.jumbobag.fr/media/css_secure/1ba70e6d1\[...\]3eeb34d16e31ad42.css](https://dev.jumbobag.fr/media/css_secure/1ba70e6d1[...]3eeb34d16e31ad42.css)

- `h1 {letter-spacing: -1px !important}` (ligne 22, col 55)
- `h2 {letter-spacing: -1px !important}` (ligne 23, col 55)
- `h3 {letter-spacing: -1px !important}` (ligne 24, col 53)
- `h4 {letter-spacing: 1px !important}` (ligne 25, col 36)
- `h5 {letter-spacing: -1px !important}` (ligne 26, col 36)
- `h6 {letter-spacing: -1px !important}` (ligne 27, col 36)
- `.hidden {display: block !important}` (ligne 43, col 9)
- `.hidden {border: 0 !important}` (ligne 43, col 34)
- et 100 autres

<https://dev.jumbobag.fr/options.css.php?store=default&cat=3>

- `.fullwidthbanner-container .fullwidthbanner {max-height: 460...` (ligne 187, col 47)

<https://dev.jumbobag.fr/skin/frontend/jumbobag/default/css/style-r.css>

- `.product-original .presentation-produit-wrapper {background:...` (ligne 5, col 2)
- `.product-original .entretien-wrapper {background: transparen...` (ligne 9, col 2)
- `.product-original .wrapper-en-savoir-plus .inner-en-savoir-p...` (ligne 13, col 2)
- `.product-original .informations-produit .product-infos .prod...` (ligne 27, col 2)
- `.product-original .height-centered h1 .title-product-h1 {dis...` (ligne 34, col 2)
- `.product-original .informations-produit .product-images {mar...` (ligne 44, col 2)
- `.product-original .presentation-produit {padding-bottom: 110...` (ligne 53, col 2)
- `.product-original .informations-produit .product-infos .prod...` (ligne 69, col 2)
- et 160 autres

Les autres conseils

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="sub-wrapper" style="width:485px">`

Vos principaux axes d'optimisation

0/100

#78

! Définissez au moins 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:

- 4 balise(s) <h3>

0/100

#69

! Ajoutez l'attribut alt sur vos balises

L'attribut `alt` est un critère important en terme de référencement (SEO). En effet, les robots d'indexation des moteurs de recherche ne peuvent pas analyser de contenu graphique. Il 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 41 balises `img`, mais la balise suivante ne définit pas d'attribut `alt` :

- ``

Attention, notez cependant que 1 de vos images contient un attribut `alt`, **mais il est vide** :

- ``

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 \(EN\)](#).

! Cette page n'indique pas de 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 :

```
<div id="a" itemscope itemtype="http://data-vocabulary.org/Breadcrumb" itemref="b">  
  <a href="http://www.example.com/macategorie" itemprop="url">  
    <span itemprop="title">Ma Catégorie</span>  
  </a> >  
</div>  
<div id="b" itemscope itemtype="http://data-vocabulary.org/Breadcrumb" itemprop="child">  
  <a href="http://www.example.com/macategorie/monproduit" itemprop="url">  
    <span itemprop="title">Mon Produit</span>  
  </a>  
</div>
```

Exemple

Sur la page analysée ici, nous n'avons pas trouvé de fil d'Ariane.

Les autres conseils

0/100

#2475

! Indiquez la présence d'au moins un produit sur la page

Les moteurs de recherche se servent des données structurées dans le but de mettre en valeur les informations utiles de la page.

Les données structurées

Vous pouvez ajouter des informations sur le sens de certains éléments de votre page (données sémantiques). Ainsi, vous pouvez indiquer que telle section parle d'un produit, une autre parle d'une personne, d'une entreprise, etc. Un moteur de recherche pourra alors extraire ces informations et les mettre en avant sur ses pages de résultats (par exemple la note attribuée à un produit, son prix, ...).

Comment les définir ?

Ces données sémantiques peuvent être définies au choix de 3 façons différentes : les microdonnées, JSON-LD, et RDFa (l'utilisation des 2 premières reste préférée pour leur simplicité).

Voici un exemple d'utilisation des microdonnées pour spécifier la présence d'un produit :

```
<div itemscope="" itemtype="http://schema.org/Product">
  
  <a itemprop="url" href="http://me.com/product1/">
    <span itemprop="name">BRAND</span>
  </a>
  <div itemprop="offers" itemscope itemtype="http://schema.org/Offer">
    <span itemprop="price">4399 p.</span>
  </div>
</div>
```

Exemple

Vous pouvez consulter [la liste des propriétés fournies par Schema.org](#). Une page ecommerce peut par exemple se servir des propriétés **Product**, **Offer**, **Order**, ou encore **Rating**.

Nous vous conseillons d'utiliser des données structurées liées au ecommerce sur cette page (toutes les pages de votre site doivent notamment faire référence à un produit).

0/100

#2432

! Google risque de ne pas utiliser votre description de la page

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 pénalisé directement par Google par exemple si votre texte est trop grand ou trop court. [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 utilise Magento. Vous pouvez configurer la <meta> description via l'interface d'administration du site.

La balise `<meta> description` doit faire entre 20 et 300 caractères. Celle-ci en fait 873:

Légendaire et incontournable, voilà un pouf qui sait se plier à toutes vos envies !

Accessoire indispensable d'une déco design, le jumbo bag s'accorde à l'humeur du moment : ambiance cosy pour recevoir vos amis, relax sur la terrasse ou cocooning dans un coin de la chambre, il est le compagnon idéal...

Vos principaux axes d'optimisation

0/100

#2434

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

Il est primordial de restreindre la provenance des contenus d'une page en vous prémunissant d'attaques de type XSS (Cross-Site Scripting).

Les attaques XSS

Une attaque de type XSS (Cross-Site scripting) a pour objectif d'injecter un contenu sur la page.

Pour accroître votre protection contre ces attaques, mettez en place une politique de sécurité qui va expliciter au navigateur web les serveurs qui sont autorisés à délivrer des ressources pour former la page. Si le navigateur effectue une requête vers un serveur non autorisé, il doit en informer l'internaute.

La solution : configurer un en-tête HTTP "Content-Security-Policy" (CSP)

Pour spécifier une politique de sécurité sur la provenance de vos ressources, 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, la page se charge correctement à condition que tous les scripts proviennent de l'hôte courant ou de `https://apis.google.com`.

[Le lien suivant](#) vous aidera à en savoir plus sur cet en-tête HTTP. Vous serez alors amené à utiliser les directives exposées [sur ce lien](#).

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.

La CSP peut se configurer avec Apache. Assurez vous tout d'abord que le module `mod_headers` est bien activé. Ensuite, dans votre configuration (fichier `.htaccess` par exemple), vous pouvez indiquer votre politique de sécurité en adéquation avec votre cas. Ci-dessous un exemple :

```
<IfModule mod_headers.c>
Header set Content-Security-Policy "script-src 'self' https://www.google.com"
</IfModule>
```

Exemple

Dans cet exemple, la page n'accepte que les scripts provenant du même nom de domaine et de google.com.

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

0/100

#2521

! 9 cookies ne sont pas sécurisés

Un cookie transmis au navigateur via le protocole HTTP 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 retransmet 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. [Lire l'article dédié à la sécurisation de cookies](#) pour en savoir plus.

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.

L'en-tête HTTP `Set-Cookie` peut se configurer avec Apache. Assurez vous tout d'abord que le module

L'en-tête HTTP **Set-Cookie** peut se configurer avec Apache. Assurez vous tout d'abord que le module **mod_headers** est bien activé. Ensuite, vous pouvez indiquer l'en-tête dans votre configuration (fichier .htaccess par exemple), comme sur l'exemple ci-dessous :

```
<IfModule mod_headers.c>
# seulement pour Apache > 2.2.4:
Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure

# versions antérieures:
Header set Set-Cookie HttpOnly;Secure
</IfModule>
```

Exemple

Les cookies suivants ne sont pas sécurisés, vous devriez ajouter l'instruction **Secure** lors du **Set-Cookie** :

<https://dev.jumbobag.fr/products.html>

- Set-Cookie: frontend=d451389c5bf5523eebf74a16dc3ca994; expires=Thu, 23-Nov-2017 14:34:20 GMT; Max-Age=3600; path=/; domain=dev.jumbobag.fr; HttpOnly

[payments-de.amazon.com/gp/apa/csm/jsonp?sellerId\[...\]27entries%27%3A%5B%5D%7D](payments-de.amazon.com/gp/apa/csm/jsonp?sellerId[...]27entries%27%3A%5B%5D%7D)

- Set-Cookie: ubid-main=259-6673689-2272560; Domain=.amazon.com; Expires=Wed, 18-Nov-2037 13:34:23 GMT; Path=/

[payments-de.amazon.com/gp/widgets/abtests?mercha\[...\]lback=jsonp1511444063217](payments-de.amazon.com/gp/widgets/abtests?mercha[...]lback=jsonp1511444063217)

- Set-Cookie: ubid-main=259-6673689-2272560; Domain=.amazon.com; Expires=Wed, 18-Nov-2037 13:34:23 GMT; Path=/

[payments-de.amazon.com/gp/apa/csm/jsonp?sellerId\[...\]27entries%27%3A%5B%5D%7D](payments-de.amazon.com/gp/apa/csm/jsonp?sellerId[...]27entries%27%3A%5B%5D%7D)

- Set-Cookie: ubid-main=259-6673689-2272560; Domain=.amazon.com; Expires=Wed, 18-Nov-2037 13:34:23 GMT; Path=/

<https://payments-de.amazon.com/gp/widgets/sessionstabilizer>

- Set-Cookie: ubid-main=259-6673689-2272560; Domain=.amazon.com; Expires=Wed, 18-Nov-2037 13:34:23 GMT; Path=/
- Set-Cookie: session-token=+1xvFRidq8DajfjCy0F4R2tf9dFoAn2taArMDBoucIOAotNecuNwZQkX90pEduHY6BHpr3w5KLx3uGciph7g5JVLLv55QK5WMM5pss23SaeLLtmcCUfgmpYlrfjo1+dVYwF1mZ5LWMrUewqxhk9ibIUxvG0efjjATbnO7LqnRVKPk0kPaVuxheehta793Jl3; Domain=.amazon.com; Expires=Wed, 18-Nov-2037 13:34:23 GMT; Path=/
- Set-Cookie: session-id=262-1825764-8113837; Domain=.amazon.com; Expires=Wed, 18-Nov-2037 13:34:23 GMT; Path=/
- Set-Cookie: session-id-time=2142164063l; Domain=.amazon.com; Expires=Wed, 18-Nov-2037 13:34:23 GMT; Path=/

[payments-de.amazon.com/gp/apa/csm/jsonp?sellerId\[...\]27entries%27%3A%5B%5D%7D](payments-de.amazon.com/gp/apa/csm/jsonp?sellerId[...]27entries%27%3A%5B%5D%7D)

- Set-Cookie: ubid-main=259-6673689-2272560; Domain=.amazon.com; Expires=Wed, 18-Nov-2037 13:34:23 GMT; Path=/

✓ 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.

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.

 **L'en-tête HTTP "X-Content-Type-Options" peut se configurer avec Apache. Assurez vous tout d'abord que le module `mod_headers` est bien activé. Ensuite, vous pouvez indiquer l'en-tête dans votre configuration (fichier `.htaccess` par exemple), comme sur l'exemple ci-dessous :**

```
<IfModule mod_headers.c>
Header always set X-Content-Type-Options "nosniff"
</IfModule>
```

Exemple

Les ressources suivantes sont hébergées par un parti tiers, il se peut donc qu'elles ne soient pas sous votre responsabilité. Vous devriez cependant considérer toute alternative possible à ces fichiers pour rester en accord avec la bonne pratique.

- cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.1/css/font-awesome.min.css
- <https://fonts.googleapis.com/css?family=Lato:100,300,400,700,900>
- <https://fonts.googleapis.com/css?family=Open+Sans:300,400,700>
- <https://static-eu.payments-amazon.com/OffAmazonPayments/de/lpa/js/Widgets.js>
- <https://www.googletagmanager.com/gtm.js?id=GTM-T55T8X>
- <https://widgets.trustedshops.com/js/X3B73FD19E154D0F81DD7DD1C76F880D7.js>
- <https://payments-de.amazon.com/gp/widgets/sessionstabilizer>
- <https://eu.account.amazon.com/lwa/js/sdk/login1.js>
- [payments-de.amazon.com/gp/apa/csm/jsonp?sellerId\[...\].27entries%27%3A%5B%5D%7D](https://payments-de.amazon.com/gp/apa/csm/jsonp?sellerId[...].27entries%27%3A%5B%5D%7D)
- [payments-de.amazon.com/gp/widgets/abtests?mercha\[...\].back=jsonp1511444063217](https://payments-de.amazon.com/gp/widgets/abtests?mercha[...].back=jsonp1511444063217)
- [widgets.trustedshops.com/css/style.css?version=\[...\]ckoutElementId=&channel=](https://widgets.trustedshops.com/css/style.css?version=[...]ckoutElementId=&channel=)
- [payments-de.amazon.com/gp/apa/csm/jsonp?sellerId\[...\].27entries%27%3A%5B%5D%7D](https://payments-de.amazon.com/gp/apa/csm/jsonp?sellerId[...].27entries%27%3A%5B%5D%7D)
- [payments-de.amazon.com/gp/apa/csm/jsonp?sellerId\[...\].27entries%27%3A%5B%5D%7D](https://payments-de.amazon.com/gp/apa/csm/jsonp?sellerId[...].27entries%27%3A%5B%5D%7D)
- <https://static-eu.payments-amazon.com/de/live/bridge.html>
- [images-eu.ssl-images-amazon.com/images/G/01/EP/o\[...\].e/prod/css/donations.css](https://images-eu.ssl-images-amazon.com/images/G/01/EP/o[...].e/prod/css/donations.css)
- et 10 autres

Vos principaux axes d'optimisation

75/100

#2388

! Vous pouvez réduire certains fichiers JavaScript (minification)

En compressant votre code JavaScript, vous pouvez libérer de nombreux octets de données et réduire les délais de téléchargement, d'analyse et d'exécution.

Réduisez la taille des ressources JavaScript suivantes afin de gagner 26.6 Ko (réduction de 6%).

Ressources provenant de "jumbobag"

- Une réduction de la taille de [dev.jumbobag.fr/me\[...\].js](http://dev.jumbobag.fr/me[...].js) pourrait libérer 25.4 Ko (réduction de 11%).

Ressources hébergées par un parti tiers

Les ressources suivantes sont hébergées par un parti tiers, il se peut donc qu'elles ne soient pas sous votre responsabilité. Vous devriez cependant considérer toute alternative possible à ces fichiers pour rester en accord avec la bonne pratique.

- Une réduction de la taille de [static-eu.payments\[...\].js](http://static-eu.payments[...].js) pourrait libérer 1.3 Ko (réduction de 1%).

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

85/100

#2500

! Vos fontes de caractères peuvent être allégées

Veillez à ce que les formats de fontes utilisés par le navigateur soient les plus légers.

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 ?

Repassez sur les instructions `@font-face` qui font défaut, et vérifiez si l'ordre indiqué est cohérent, ou encore si aucune erreur de syntaxe n'est présente. Voici un article sur le sujet afin de vous aider à [déterminer le bon ordre à choisir pour vos fontes de caractères](#). Plus succinctement, **l'ordre à respecter pour vos formats est le suivant : EOT, WOFF2, WOFF, TTF, puis SVG.**

La ressource suivante est récupérée, cependant vous devriez envisager l'utilisation du format WOFF2 à la place :

- <https://widgets.trustedshops.com/fonts/tsfonts.woff>

95/100

#2387

✓ Vous pouvez réduire certains fichiers CSS (minification)

En compressant votre code CSS, vous pouvez libérer de nombreux octets de données et réduire les délais de téléchargement et d'analyse.

Réduisez la taille des ressources CSS suivantes afin de gagner 6.5 Ko (réduction de 11%).

- Une réduction de la taille de [dev.jumbobag.fr/me\[...\].css](http://dev.jumbobag.fr/me[...].css) pourrait libérer 6.5 Ko (réduction de 11%).

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

Si vous souhaitez en savoir plus sur [la minification CSS](#), n'hésitez pas à consulter cet article sur le sujet !

Les autres conseils

99/100

#2384

✓ **Activez la compression des ressources**

En compressant vos ressources avec "gzip" ou "deflate", vous pouvez réduire le nombre d'octets envoyés sur le réseau.

Autorisez la compression des ressources suivantes afin de réduire le volume de données transférées de 169.6 Ko (réduction de 70%).

Les ressources suivantes sont hébergées par un parti tiers, il se peut donc qu'elles ne soient pas sous votre responsabilité. Vous devriez cependant considérer toute alternative possible à ces fichiers pour rester en accord avec la bonne pratique.

- La compression de [static-eu.payments\[...\].js](#) pourrait libérer 169.6 Ko (réduction de 70%).

🌈 Cette page est retournée par un serveur Apache. Vérifiez s'il utilise bien le module [mod_deflate](#).

Vos principaux axes d'optimisation

0/100

#74

! Évitez d'exposer votre version d'Apache

Les réponses HTTP de votre serveur contiennent un en-tête HTTP `Server` qui indique la version de votre serveur Apache: `Apache/2.2`. Il sera plus difficile pour une personne malveillante d'attaquer votre site si elle ne connaît pas la version de votre serveur Apache. Pensez à modifier cette valeur.

Par exemple, si vous êtes sur un serveur Linux, éditez votre fichier `/etc/apache2/conf.d/security`. Vous devez alors modifier les valeurs de `ServerSignature` et de `ServerTokens`, comme indiqué ci-dessous:

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

Exemple

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

Exemple

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

Vos principaux axes d'optimisation

75/100

#2299

! Évitez de manipuler le DOM dans des boucles avec jQuery

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.

Vous avez **1 manipulation(s) du DOM dans des boucles** :

[dev.jumbobag.fr/media/js/920b708c8832ff\[...\]d90080133074fa7.js](https://dev.jumbobag.fr/media/js/920b708c8832ff[...]d90080133074fa7.js)

- ```
for(var e=0;e<b.slideamount;e++){var f=Math.round(Math.random()*b.slideamount);b.ul.find(">li:eq("+f+")").prependTo(b.ul)}
```