**Qu'est ce qu'un bon site Internet - une bonne application ? 10 règles d'or**

Ces 10 règles d’or sont là pour vous donner des bonnes pratiques et des outils sur lesquels s’appuyer pour concevoir un bon site Internet - une bonne application. **Il faut cependant distinguer un site qui sera durable et maintenable d’un prototype que l’on réalise pour tester une idée, et sur lequel vous allez réitérer plusieurs fois.** Dans le premier cas, nous sommes face à une application ou un site dont l’utilité est confirmé et qui est plébiscité par les utilisateurs : il convient donc de respecter l’ensemble de ces règles. Dans le deuxième cas - par exemple lors d’un hackathon - nous cherchons à réaliser une version “jetable” où la seule règle d’or sera de faire au plus vite, en apportant un soin particulier à l’interface utilisateur. Nous ne cherchons alors pas à bien documenter le code ou bien le structurer, nous passerons à une version durable lorsque le projet aura été validé.



**Répond au besoin du client et de l'utilisateur**



**Respecte les grands principes d'un code bien structuré et bien documenté**



**Est bien référencé**



**Est ergonomique, facile à utiliser et intuitif**



**Respecte les règles clés d'un design efficace et responsive**



**Veille à l'accessibilité**

**Respecte les règles, mentions légales, CGU CGV, et mentions de la Wild Code School**



**Fait bonne utilisation des cookies**



**Est optimisé au niveau du chargement**

**Est protégé, ne présente pas de failles de sécurité majeures**

1. **Répond au besoin du client et de l'utilisateur**

Un bon site Internet est avant tout un site qui répond au besoin du client et de l’utilisateur. Cela semble évident, mais **un développeur peut être assez vite tenté de développer l’application dont il a envie ou qu’il pense être la bonne**, en mettant de côté les besoins du client et de l’utilisateur.

Pour éviter de tomber dans ce piège, la **posture** (avoir de l’empathie, ne pas être dans le jugement) et l’**écoute active** du client sont essentielles pour récolter les bonnes informations. Il est impératif de bien observer et analyser l’entreprise du client, son marché et son écosystème. Il faut notamment réaliser une étude de marché, comparer le produit du client avec ce qui existe déjà, pour lui apporter la meilleure solution possible. D’où l’importance du [questionnaire envoyé au client](https://docs.google.com/a/wildcodeschool.fr/forms/d/1qiVPr-KQsuXpxIRql4dP6I8MNPsJPYQnWgn2OUChdjo/edit), qui permet de mieux cerner son profil et de lui proposer une application qui répond à ses besoins et à ceux de ses utilisateurs.

**A ne pas confondre : client et utilisateur.** Le client est celui qui paie pour le projet, l’utilisateur est bien souvent le client du client. L’objectif d’un site Internet est de répondre aux besoins de l’utilisateur final. Le bon développeur est celui qui arrivera à comprendre ses besoins, et qui saura conseiller le client et l’aiguiller s’il se trompe.

1. **Respecte les grands principes d'un code bien structuré et bien documenté**

**Pourquoi doit-on bien structurer un code ?** La vie d’un site Internet ne s’arrête pas après son développement, il doit ensuite être maintenu. Cette maintenance n’est cependant pas forcément prise en charge par la personne qui a développé le site, l’objectif est donc de lui faciliter la vie en concevant dès le départ un code lisible dans lequel elle pourra facilement et rapidement s’y retrouver. Un code bien structuré est également beaucoup mieux référencé.

Le [W3C (World Wild Web Consortium)](http://www.w3c.fr/standards/), l’autorité des standards du web, a défini plusieurs bonnes pratiques d’un code propre qu’il faut chercher à respecter. Si un code n’est jamais parfait dans l’absolu, suivre les bonnes pratiques listées ci-dessous vous permettra de bien le structurer et de le rendre plus lisible :

* **Adopter une convention de style pour le code** en choisissant une “mise en page-type”. Par exemple, l’utilisation de “;” en fin de ligne constitue une convention de style. [Plus d’exemples de conventions en Javascript](https://github.com/airbnb/javascript).
* **Donner un nom explicite** aux variables et aux fonctions dans le code, pour harmoniser au sein d’une équipe la compréhension du code.
* **Séparer la structure du contenu** (HTML) **de sa mise en forme** (CSS), afin de faciliter la maintenance le jour où la charte graphique évolue. Par exemple, plutôt que de devoir modifier tous les éléments HTML qui utilisent la même couleur, il suffit de changer une seule ligne dans le CSS.
* **Extraire les fonctionnalités clés dans des fonctions ou des classes** pour faciliter leur réutilisation à plusieurs endroits dans l’application : **Don’t Repeat Yourself (DRY) !** Cela implique de ne pas réécrire deux fois un code ayant la même utilité.
* **Ecrire des tests sur les fonctionnalités clés** de l’application, comme lors des dojos.

Un code bien structuré est un prérequis mais cela ne fait pas tout, il restera difficilement maintenable sans son manuel d’utilisation. C’est pourquoi lors d’un projet **il est attendu des développeurs qu’ils documentent les éléments clés de l’application dans un** [**Read me**](https://github.com/jbehuet/mean-starter-es6/blob/master/README.md), permettant ensuite à d’autres développeurs de reprendre en main l’application si besoin. Les grands domaines que l’on retrouve généralement dans un Read me :

* Liste des prérequis : base de donnée avec le numéro de version, version du langage utilisé, services extérieurs (API d’envoi d’e-mails, hébergement d’images, etc.)
* Opérations à effectuer pour faire fonctionner l’application dans l’environnement de développement
* Opérations à effectuer pour déployer le projet en production
* Conventions adoptées par le projet (ex : la convention de style utilisée)
* Exécution des tests du projet

Pour aller plus loin : [Bien documenter un projet](https://drive.google.com/open?id=1R6mSUNZYU8D9jsYr44u2o0MiNjqzuGzdTf-x4s-_sb4)

1. **Bien référencé**

Créer un site internet pour s’assurer une visibilité sur le web, c’est beaucoup de boulot pour pas grand chose si vous oubliez ensuite de demander à Google de le rendre visible et de tenter de le faire apparaître dans les premiers résultats : il faut leréférencer !

**L’indexation**

De la même manière qu’un commerçant doit demander à un supermarché de faire apparaître son produit dans les rayons de la grande surface, vous devez demander à Google d’être présent dans son moteur de recherche. C’est une étape primordiale dans le cadre d’un projet-client.

Pour cela, **il faut s’assurer que le site Internet apparaît sur la première page de Google lorsqu'on tape le nom de domaine**. Une astuce consiste à écrire dans la barre de recherche google : “site:nomDeDomaine.extension Sans espace - extension = .fr, .com, …”. Les résultats correspondent à toutes les pages du site qui sont indexées dans Google. Si aucun résultat ne sort, le site n’est alors pas référencé. Si les résultats de la recherche précédente sont mauvais, deux choses à faire :

* **Vérifier la bonne indexation dans les paramètres du compte** lorsque l’on passe par un CMS comme WordPress : panel administration > onglet général > lecture > décocher la case : “demander à Google de ne pas indexer ce site” ;
* **Faire attention aux balises “ne pas indexer” dans le code** : <meta name="robots" content="noindex">.

**Le positionnement**

Pour reprendre notre exemple de supermarché, le positionnement d’un site internet est similaire à celui d’un produit qui apparaît au premier plan dans les rayons ou qui se retrouve caché derrière plusieurs gros paquets. N’oublions pas que presque **9 utilisateurs sur 10 cliquent sur un résultat situé sur la première page de recherche : il faut donc essayer d’optimiser le positionnement de son site web.** Deux méthodes principales :

* **Le référencement naturel** **(SEO, Search Engine Optimization)** :
  + **Inbound SEO**, en se concentrant sur la structure, le contenu et les bonnes pratiques d’un site Internet (Url simple, contenu de qualité qui incite le visiteur à le partager, respect des normes de programmation établie par la [W3C](http://www.w3c.fr/standards/), ajout d’un [robot .txt](http://robots-txt.com/) pour aider les bots d’indexation des moteurs de recherche, etc.). Un important travail d’analyse des mots clés et des contenus qui intéressent l’audience ciblée doit être réalisé afin d’optimiser le référencement de son site et ses pages. Un site bien référencé est un site qui est considéré comme pertinent par Google pour le faire apparaître dans les premiers résultats de recherche de l’internaute. 
  + **Outbound SEO**, en ajoutant des liens qui pointent vers notre site Internet (depuis Wikipédia, les sites partenaires, les réseaux sociaux, etc.). Plus notre site est populaire aux yeux de Google, mieux il sera référencé. On considère un site comme étant “populaire” lorsqu’il reçoit des liens en provenance d’autres sites traitant de la même thématique (ex : un média spécialisé sur la thématique “finance” qui pointe vers le site commercial d’une banque). Plus le lien provient d’un site qui fait référence dans votre thématique, plus le “jus” envoyé vers votre site sera fort et impactera positivement votre référencement.
* **L’utilisation d’Adwords** **(SMO, Social Media Optimization)** qui permet de faire apparaître le site Internet dans les bannières pubs des moteurs de recherche à partir d’un mot-clé acheté. Cette technique n’a que très peu d’impact sur le positionnement “naturel” d’un site. Ce système d’achat publicitaire permet potentiellement d'apparaître dans les premières lignes, au dessus des résultats naturels. Mais attention, il s’agit d’un système d’enchères calculées selon les “mises” de vos concurrents sur les mêmes mots clés. Ce n’est pas parce que vous êtes prêts à payer plus cher que vos concurrents que vous serez forcément premier dans les résultats publicitaires : Google Adwords prend aussi en compte le “Quality Score”, un score de qualité calculé à partir de la pertinence de votre référencement naturel sur une page donnée en réponse à la recherche de l’internaute. **Il faut donc au préalable travailler très sérieusement son SEO pour être pertinent aux yeux de Google, même pour faire de la publicité**, au risque de payer des sommes très élevées pour un résultat médiocre.

Il existe également toute une palette d’outils permettant de tester le référencement de votre site internet, comme [Google Search Console](https://www.google.com/webmasters/tools/home?hl=fr&pli=1) et [WooRank](https://www.woorank.com/fr). Google Search Console propose la possibilité de soumettre une URL qui n’est pas encore référencée dans son moteur via le lien de soumission suivant : <https://www.google.com/webmasters/tools/submit-url?pli=1>

Si vous n’avez que 5 bonnes pratiques à retenir pour être bien référencé, c’est maintenant qu’il faut être attentif :

1. **Définir sa cible d’audience.**
2. **Ne pas penser son site Internet comme un catalogue**, plutôt comme des pages que Google doit trouver pertinentes à afficher aux yeux des internautes dans les résultats du moteur de recherche. Créez du contenu adapté et réfléchi !
3. **Définir les mots clés et groupes de mots clés susceptibles d’être recherchés par les internautes ciblés** pour trouver votre service ou votre produit. Si vous avez des concurrents, analyser leur propre site et les mots clés utilisés, ainsi que dans les résultats Google pour une même recherche.
4. **Évaluer le volume et la concurrence de ces mots clés**, grâce à l’outil keywords planner sur Google Adwords et l’utilisation de Google Trends.
5. **Travailler des mots clés différents et / ou complémentaires** sur les pages de votre site qui correspondront à différents types de recherches des internautes.

Pour aller plus loin :

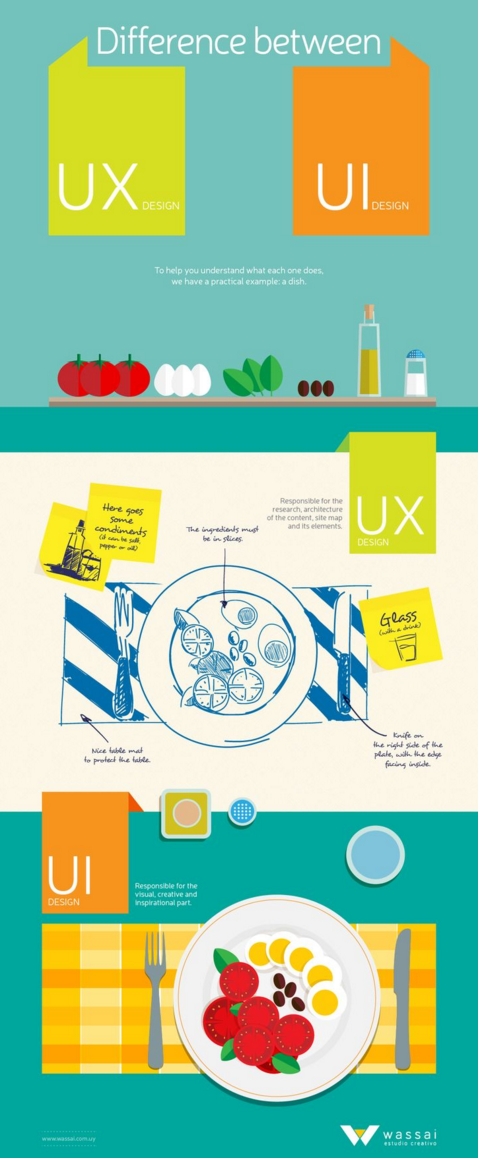
* [Meetupsur le référencement web](https://drive.google.com/open?id=0B2WiP-xVDN3lVE5SaDktLTU0ZGc)
* [Guide de démarrage Google - Optimisation pour les moteurs de recherche](https://static.googleusercontent.com/media/www.google.fr/fr/fr/intl/fr/webmasters/docs/search-engine-optimization-starter-guide-fr.pdf)

1. **Ergonomique, facile à utiliser et intuitif**

On a souvent du mal à comprendre la différence entre ergonomie et design. L’ergonomie, c’est le chemin parcouru sur un site internet ou une application, qui doit être le plus naturel et fluide possible. Le design c’est la peinture rajoutée dessus.

* **UX design** : *User Experience Design*, design de l’expérience utilisateur
* **UI design** : *User Interface Design*, design d’interface utilisateur

Quelques bonnes pratiques pour disposer d’un site ergonomique :

* **Avoir le moins d’interactions et de clics possibles** pour arriver jusqu’à son information.
* **Disposer d’une architecture logique** : des titres plus grands que les sous-titres, eux-mêmes plus grands que les paragraphes, etc.
* **Disposer d’une cohérence des éléments visuels sur le site** : tous les boutons de la même couleur, les titres au même format, etc.
* **Faire tester l’application autour de vous et récolter des témoignages.**

[Trainline](https://www.trainline.fr/) est un exemple de site ergonomique qui propose un service de réservation de billets de train plus intuitif et plus rapide que le site [voyages-sncf.com](http://www.voyages-sncf.com/).

1. **Respecte les règles clés d'un design efficace et responsive**

Des goûts et des couleurs, on ne discute pas. Le design reste quelque chose de très subjectif, difficile donc pour un débutant de s’y retrouver.

**Pour commencer, mieux vaut ne pas chercher l’originalité à tout prix, mais plutôt se concentrer sur la clarté et la simplicité**, en s’appuyant sur les frameworks qui existent : [bootstrap](http://getbootstrap.com/), [foundation](http://foundation.zurb.com/), [semantic ui](http://semantic-ui.com/), [materialize](http://materializecss.com/). Tout est sur étagère, alors n’hésitez pas à vous en servir !

De manière générale, que vous utilisiez des frameworks ou que vous conceviez intégralement votre design, il y a une règle d’or à retenir : **ne jamais déformer le ratio d’une image ou avoir une mauvaise résolution d’image.** Il faut également veiller à ce que le design ne détériore pas la qualité du référencement.

***Responsive design versus Mobile first* ?** Jusqu’à récemment, à la conception d’un site Internet on commençait par imaginer sa version pour ordinateur avant de l’adapter sur mobile en utilisant le *responsive design*, qui permet à un site web de s'adapter à la taille de l’écran du terminal utilisé. L’espace disponible sur un mobile étant plus petit, il fallait alors réduire le contenu du site pour le faire tenir dans le bon format. De plus en plus de sites sont aujourd’hui construits en *mobile first*, en concevant d’abord la version mobile avant de la décliner sur ordinateur. Un très bon exemple de cette stratégie est l’application Instagram, qui est restée pendant deux ans une application exclusivement mobile avant de se doter d’un accès web.

[Plutôt flat ou plutôt material design ?](http://www.wow-digital.ch/web-design/material-design-vs-flat-design/) Si vous ne connaissez pas les grandes tendances design, tenez vous au courant en consultant les sites et blogs spécialisés :

* <http://www.webdesignertrends.com/>
* <http://www.blogduwebdesign.com/>

1. **Veille à l'accessibilité**

**L'accessibilité numérique consiste à permettre à tous, notamment aux personnes souffrant de handicap, d'utiliser des ordinateurs et leurs logiciels, et de consulter ou créer des ressources numériques, sur tout type de support.**

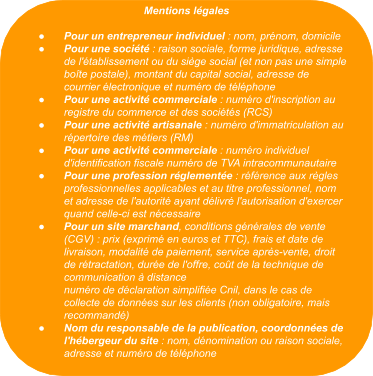
Dès 1997, le W3C s'est intéressé à l'accessibilité et a créé la WAI *(Web Accessibility Initiative)*. La WAI a émis des recommandations permettant de rendre le web plus accessible, les [WCAG 2.0](https://www.w3.org/Translations/WCAG20-fr/#intro) *(Web Content Accessibility Guidelines)*. Si toutes ces règles ne doivent pas forcément être connues par coeur, **les WCAG 2.0 sont incontournables pour le développeur, elles lui permettent de connaître les bonnes pratiques en matière d’accessibilité.** Sur cette base ont également été construits des référentiels nationaux. **En France, les deux référentiels historiques (Accessiweb et RGAA) ont été réunis en 2015, dans le** [**RGAA V3**](http://references.modernisation.gouv.fr/referentiel/)**, norme nationale.**

Quelques exemples de règles importantes en matière d’accessibilité :

* **La page doit être visuellement propre et pas trop chargée**, sans blocs d’images et animations dans tous les sens.
* **L’information doit être bien ordonnée**, le contenu hiérarchisé grâce à l’utilisation de différents niveaux de titre.
* **La création de liens d’évitement** comme “Aller au menu”, “Retour en haut de page”, permet de gagner du temps et d’éviter à l’utilisateur de remonter les pages.
* **Les illustrations informatives doivent contenir une alternative textuelle** pour ceux qui ne peuvent pas les voir correctement : la technique de base en HTML consiste à renseigner l’attribut « alt » du tag « <img> » avec une texte décrivant l’image, en évitant les termes inutiles (« photo », « image », « logo »).

Il est utopique de penser arriver à respecter l’ensemble des règles d’accessibilité, [très peu de géants du web les respectent eux-mêmes](http://wave.webaim.org/report#/google.com)… Il existe cependant des outils à la disposition du développeur pour tester le niveau d’accessibilité de son site :

* <http://wave.webaim.org/> : un vérificateur automatique du bon respect des règles d’accessibilité fondamentales.
* <https://desktop.opquast.com/en/> : un plugin de navigateur.

1. **Respecte les règles, mentions légales, CGU CGV, et mentions de la Wild Code School**

**Les mentions légales ou mentions obligatoires doivent permettre d’apporter une information minimale au consommateur** **sur le créateur et l’hébergeur du site.** Leur indication doit être visible sur chaque page du site, elles varient selon le type de site.

Si ce n’est pas une obligation légale, il est tout de même **conseillé de rédiger également des CGU et CGV** pour permettre au site de limiter certaines de ses responsabilités juridiques :

* **Conditions Générales d’Utilisation (CGU)** : encadrent les modalités, droits et limites d’utilisation d’un site. Elles s’adressent à tous les utilisateurs du site et sont de facto acceptées par l’internaute du seul fait de son accès au site. Elles permettent par exemple de définir ce qu’il est autorisé de dire dans les commentaires.
* **Conditions Générales de Vente (CGV)** : encadrent les modalités, droits et limites liées à toute opération de vente effectuée sur un site. Les CGV sont obligatoires sur les sites de e-commerce à destination des particuliers.

Pour vous aider à rédiger ces CGU et CGV, vous pouvez vous appuyer sur la [liste des principales clauses à insérer](http://www.aecom.org/Vous-informer/Juridique-TIC/Droit-Internet/Commerce-electronique/Sites-internet-mentions-legales-CGU-CGV-et-clauses-particulieres).

Dernière chose, pour valoriser le travail réalisé par les élèves de l’école dans le cadre des projets-clients, ne surtout pas oublier d’ajouter une mention dans le footer du site : **“Réalisé par les élèves de la** [**Wild Code School**](http://www.wildcodeschool.fr/)**”**. Dans certains cas, il est également possible de citer le nom des élèves, avec l’accord du client.

1. **Bonne utilisation des cookies**

Les cookies sont des fichiers que le site internet visité place sur le disque dur de votre ordinateur lors de votre première visite. **Ils sont très utiles aux développeurs puisqu’ils permettent de conserver des données sur les utilisateurs** afin de faciliter leur navigation et de leur proposer un contenu plus adapté à ce qui l’intéresse. Quelques exemples de ce que les cookies permettent de faire : stocker des préférences, des noms d’utilisateurs, enregistrer des produits, etc. Dans le cadre d’un projet dynamique, on utilise le plus souvent des cookies d’authentification pour la connexion.

Les cookies sont néanmoins controversés car ils contiennent parfois des informations personnelles pouvant être exploitées par des tiers. Depuis la directive européenne appelée “paquet télécom” de 2009, **la loi impose aux responsables de sites et aux fournisseurs de solutions d’informer les internautes et de recueillir leur consentement avant l’insertion de cookies sur un site Internet.** Il existe deux types de cookies : le cookie en session de navigateur (dont les données disparaissent lorsque la session de navigation est fermée), et le cookie qui recueille les données dans une base de données (le refus des cookies empêche l’utilisateur d’accéder au site, cas beaucoup plus rare).

Dès qu’un site internet collecte des données personnelles sur ses utilisateurs, [il faut déclarer ce fichier de données à la CNIL](https://www.cnil.fr/fr/declarer-un-fichier). Pour tout savoir sur les cookies qui nécessitent le consentement préalable des utilisateurs et comment recueillir valablement le consentement, le [site de la CNIL](https://www.cnil.fr/fr/site-web-cookies-et-autres-traceurs) est une mine d’information.

1. **Optimisé au niveau du chargement**

**Un utilisateur n’attend en moyenne pas plus de trois secondes devant une page Internet qui charge** **: si ce temps est plus long, il passe à autre chose !** Le temps de chargement compte également dans le référencement naturel d’un site internet. D’où l’importance de disposer d’un site optimisé au niveau du chargement.

Lorsqu’un site Internet ou une application charge, il télécharge tout un ensemble de fichiers CSS, librairie Javascript, etc. Pour ne pas ralentir le temps de chargement des pages, **il faut dans la mesure du possible avoir des fichiers les plus légers possibles.**

Quelques bonnes pratiques pour optimiser le chargement :

* **Retoucher dès le développement les dimensions et la résolution des images** (plutôt le travail des web designer, mais également très utile pour un développeur), pour éviter d’avoir ensuite à redimensionner l’image à chaque chargement.
* **Uglify ou minifier le code**, c’est-à-dire le compresser, lui enlever tous ses espaces jusqu’à le “rendre moche” et illisible, afin de réduire le poids des fichiers CSS et Javascript. Il existe des outils pour automatiser ces tâches, dont des outils de minification : [Gulp](http://gulpjs.com/), [Grunt](http://gruntjs.com/), [Brunch](http://brunch.io/).
* **Tester le chargement de la page en pré-production**, en se connectant en 2G.
* **Réfléchir à l’ordre d’apparition des fichiers “librairie”** quand on les appelle lors du chargement de la page : on ne franchit pas une porte avant de l’avoir ouverte ;)

**Pour mesurer l’efficacité et le temps de chargement d’un site, le développeur peut utiliser la console de son navigateur** (F12), qui lui permettra de réaliser des audits, des tests de chargement, des débugage, etc.

1. **Protégé, qui ne présente pas de failles de sécurité majeures**

Un certificat SSL est un fichier de données qui lie une clé cryptographique aux informations d'une organisation ou d'un individu. **Installé sur un serveur, le certificat active le cadenas et le protocole « https » dans les navigateurs, afin d'assurer une connexion sécurisée entre le serveur web et le navigateur.**

Dans le cadre des projets-clients, **ce qui est développé par nos élèves ne nécessite souvent pas un niveau de sécurité maximum, et donc pas de protocole “https”**, ce qui signifie que la communication entre le client et le serveur est en clair et non chiffré. La protection d’une application devient cependant essentielle à partir du moment où le site permet de réaliser des opérations de paiement ou héberge des informations sensibles.

Même en prenant le plus de précaution possible, aucun système de sécurité n’est parfait, il peut être confronté à des failles non prévues. Il existe quelques bonnes pratiques pour les limiter au maximum :

* Respecter strictement un framework permet déjà de limiter largement les failles de sécurité.
* **Être vigilant à ne pas trop laisser d’éléments publics**, par exemple en [limitant les dossiers upload où l’on peut déposer une image ou un script malveillant](https://www.nbs-system.com/blog/cerberhost-les-failles-de-type-file-upload.html).
* **Etre vigilant aux droits donnés sur les dossiers et les fichiers** au moment de la mise en production, en évitant le chmod 777 sur l’intégralité du projet. Pour les non-initiés, le chmod 777 fait référence à une autorisation en lecture, écriture, exécution.
* Dans la mesure du possible, **éviter de faire du copier-coller de code trouvé sur Internet**, sauf si on le comprend dans son intégralité. Quelqu’un pourrait placer un élément malveillant dans le code qui pourrait nous rendre acteur d’une attaque informatique contre notre gré.
* [Mettre en place un système permettant de hasher (=chiffrer) les mots de passe en base de données](https://openclassrooms.com/courses/securiser-les-mots-de-passe-des-utilisateurs-avec-php), et ne jamais écrire en dur des identifiants dans le code.