Mettre le web en page(s)

Lucie Anglade et Guillaume Ayoub – 16 décembre 2022

Nous sommes Lucie et Guillaume, et comme beaucoup de gens nous travaillons tous les jours avec les technologies du web.

Comme beaucoup de gens, nous aimons créer de belles architectures en HTML pour y insérer notre contenu. Comme beaucoup de gens, nous adorons peindre de belles couleurs et construire de belles structures avec nos fichiers CSS.

Mais notre travail est un peu différent de celui de beaucoup de gens : nous ne créons pas de sites internet. Notre HTML et notre CSS ne sont pas affichés dans des navigateurs. Ils sont transformés en fichiers PDF pour être imprimés, lus sur des téléphones et des ordinateurs, ou tout simplement archivés.

En d'autres termes : nous mettons littéralement le web en pages.

Comment ? Pourquoi ? Nous allons voir plus en détail que cette idée n'est pas aussi étrange qu'elle en a l'air…

Un peu d’histoire

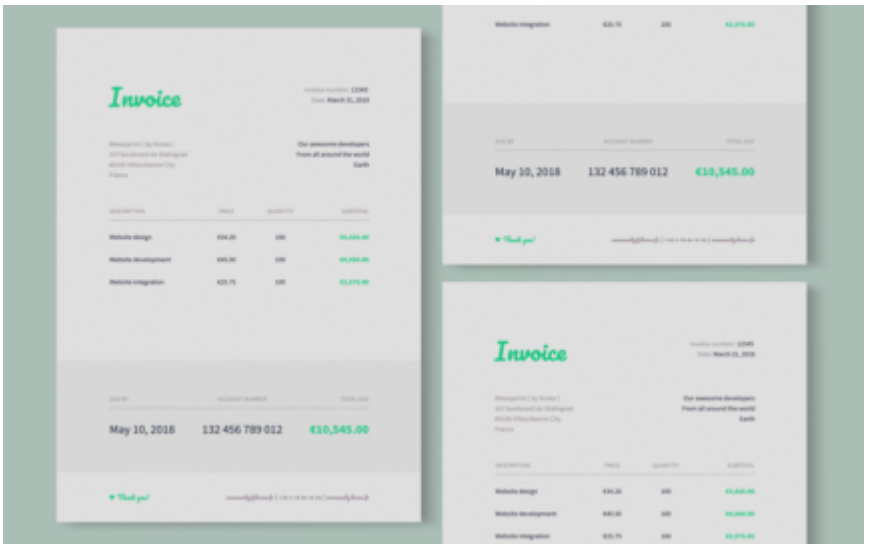
Les faiblesses des outils classiques

Soyons francs : utiliser du HTML et du CSS pour créer des fichiers PDF, ce n’était pas notre première idée. Il existe déjà une tonne d’outils pour faire de beaux documents, de Microsoft Word à LaTeX, de LibreOffice Writer à Google Docs, en passant par Adobe InDesign… Ces logiciels sont généralement amplement suffisants pour répondre à une large liste de besoins, qui vont de la simple lettre écrite en deux minutes au magazine prêt à partir à l’impression.

Pourtant, il existe un cas auquel tous ces outils sont assez peu adaptés : la génération automatique de documents.

Imaginons que vous ayez un site de vente en ligne. Vous adorez créer de superbes bols en céramique 🥣 que vous présentez au monde entier sur votre boutique. Les clients adorent ce que vous faites, et vous vendez rapidement des dizaines et des dizaines de bols. Bravo ! Les utilisateurs remplissent leur panier, ils entrent les chiffres de leur carte bleue pour payer, et… ils aimeraient bien avoir une facture.

Hum… Vous n’allez tout de même pas écrire toutes vos factures à la main ! La mise en page est toujours la même, mais le contenu varie un peu : désignation, prix, adresse… Comment faire ?



Mais dites donc, vos factures sont aussi jolies que vos bols !

La même question se pose si vous développez un site de création de cartes de visite en ligne, où vous aimeriez que vos clients choisissent parmi de nombreux styles différents, dans lesquels ils pourraient modifier les couleurs, les polices de caractères ou le logo. Pareil si vous souhaitez générer des affichettes promotionnelles ou des étiquettes électroniques pour votre magasin. Encore pareil si vous voulez imprimer des diplômes, des bulletins ou des emplois du temps dans votre école.

Bref… Vous voyez l'idée 😁.

C'est là que la possibilité d'utiliser les technologies du web pour générer des documents imprimables commence à prendre forme.

Après tout, dans chacun de ces cas, il faut générer à la volée un document dont la structure et la mise en page sont bien définies. Pour afficher du contenu venant d'une base de données, HTML n'a plus à faire ses preuves. Vous avez à disposition une pléthore d'outils, de frameworks, de bibliothèques et autres langages pour créer toutes les pages que vous souhaitez. Et pour mettre en page ce HTML, quoi de mieux que du CSS ? Vous pourrez créer une ou plusieurs feuilles de style selon vos envies, que vous pourrez même personnaliser avec des variables ou un préprocesseur (comme Sass) si cela vous chante.

D'accord, on voit bien comment tout cela marche pour afficher des sites avec un navigateur. Mais pour un document paginé, très franchement, ce ne serait pas un peu tiré par les cheveux ?

Les forces des technologies du web

À vrai dire, et aussi étonnant que cela puisse paraître, ça ne l'est pas du tout, et pas uniquement pour générer quelques factures pour vos jolis bols. Il y a même de bonnes chances que vous ayez dans la bibliothèque de votre salon, sans le savoir, un livre fait en HTML/CSS ! Les éditeurs restent souvent assez discrets sur les technologies qu'ils utilisent, mais certains comme Hachette (Anglais) en parlent publiquement. De vrais livres faits en HTML/CSS, vendus à de vrais clients, sans que personne ne s'en rende compte !

Alors, si de grands éditeurs mondiaux utilisent ces technos « pour de vrai », il est possible que cela marche pour d'autres, non ?

Ce n'est d'ailleurs pas le fruit du hasard… Il est temps de faire un peu d'archéologie, revenons un moment en 1996. Le monde de l'informatique ne jure que par Windows 95, le nouveau système d'exploitation de Microsoft, qui n'a même pas de navigateur internet installé par défaut. 90% des internautes (qui ne sont pas bien nombreux) utilisent Netscape, qui va cette année-là intégrer un langage créé à la hâte en 10 jours : JavaScript.

C'est également en cette année de grâce 1996 qu'est publiée la première version de la spécification CSS 2 (Anglais). Dans l'idée de construire un standard ouvert et interopérable, le W3C rédige un document qui sert encore de référence plus de 25 ans après. On trouve dedans une définition claire de la syntaxe, des sélecteurs, et un bon nombre des propriétés de base que l'on utilise encore aujourd'hui. Mais ce n'est pas tout…

On y trouve également un chapitre entier sur la génération de documents paginés (Anglais).

Les créateurs de CSS voient loin et adorent penser en dehors des clous. Le format n'a pas été seulement pensé pour être affiché sur les tubes cathodiques des PC et des Mac de 1996, ni à être uniquement manipulé avec une souris et un clavier. On pense déjà à afficher des sites sur une télévision, sur des plage braille, sur des terminaux portables (le premier iPhone sortira plus de 10 ans après), à lire ce contenu avec des synthétiseurs vocaux… ou, bien sûr, à imprimer des documents !

Si l'idée est louable (et incroyablement visionnaire), elle pose cependant de nombreux problèmes concrets. Découper automatiquement un document web sur plusieurs pages dont la taille est fixe comporte des problématiques spécifiques, et CSS 2 en couvre une partie : forcer ou éviter les coupures de pages, ajouter des marges et y insérer des numéros de pages, définir un format pour l'impression. C'est un bon début, mais ce n'est pas suffisant. Sans que l'on ne s'en rende toujours compte, les tonnes de documents imprimés qui nous entourent chaque jour fourmillent de petits détails dont il faut prendre soin pour atteindre une mise en page de qualité.

La pratique

Alors à quoi ça ressemble de mettre le web en pages ?

Lorsque l'on crée un document, que ce soit une lettre, une affiche, un livre ou autre chose, de nombreux choix s'offrent à nous. Par exemple, le format : mon document va-t-il être rendu sur une page A4, une page A6, en 15cm x 24cm, en portrait, en paysage ? C'est quelque chose qui est facilement définissable en CSS. Choisissons le format A4 :

```
@page {
          size: A4;
          }
```

C'était plutôt facile !

Avez-vous déjà remarqué que, sur un document de plusieurs pages, les marges ne sont souvent pas les mêmes selon si l'on est sur une page de gauche ou de droite, et que les numéros de pages ne sont pas toujours du même côté ? Ça aussi, c'est facilement définissable en CSS :

```
@page :left {
          margin: 20mm 10mm 20mm 15mm;
          @bottom-left {
          content: counter(page)
          }
          }
          @page :right {
          margin: 20mm 15mm 20mm 10mm;
          @bottom-right {
          content: counter(page)
          }
          }
```

Vous avez saisi l'idée, **CSS permet de répondre à des problématiques aussi bien pour du web que pour des documents paginés.** Essayons avec d'autres problématiques encore plus spécifiques aux documents 🤯 !

C'est beau un livre

Vous voyez ce que sont les lignes de conduite ? Mais si, ce sont ces petits points que l'on trouve dans les tables des matières, qui relient le titre d'un chapitre à son numéro de page. Bon, OK, ce ne sont pas tout le temps des points, mais vous avez saisi le concept. Eh bien, ça aussi c'est prévu en CSS :

```
#table-of-content a::after {
          content: leader('.') ' ' target-counter(attr(href), page);
          }
```

Ça peut paraître un peu barbare comme ça, regardons d'un peu plus près.

`target-counter()` permet de récupérer le numéro de la page à laquelle commence le chapitre. Et `leader()`, c'est notre star de la table des matières, c'est cette fonction (oui, il y a des fonctions en CSS) qui va permettre de mettre des petits points entre le nom du chapitre et son numéro de page. Chaque ligne aura le bon nombre de points selon l'espace disponible, et les points de toutes les lignes seront parfaitement alignés : `leader()` fait les choses bien.

Nous n'allons pas faire un tour exhaustif de tout ce qu'il est possible de faire, mais CSS regorge de fonctionnalités et propriétés qui vont nous aider à mitonner un document aux petits oignons. Que vous vouliez définir si l'on peut couper ou non à l'intérieur d'un bloc, que vous souhaitiez inté-

grer des notes de bas de page, du texte sur de multiples colonnes, des en-têtes reprenant le titre du chapitre en cours, des références croisées (avec automatiquement le nom du chapitre dont on parle et le numéro de page où le trouver), des pages de garde, du texte en filigrane de la page, ou que sais-je encore, vous trouverez souvent une manière d'arriver à vos fins.

En créant des documents avec HTML/CSS, vous bénéficiez de toutes ces choses spécifiques aux documents paginés déjà prévues par CSS, mais vous avez aussi à votre disposition toute la puissance du CSS que vous utilisez déjà pour vos sites !

Quitte à mettre des exemples de CSS dans cet article, nous nous en sommes servis pour générer un simple PDF à partir de cet article 😄. Si vous voulez voir le code plus en détails, le CSS est disponible dans ce dépôt.

L'article mis en pages

Les outils pour faire vos documents

Alors c'est bien beau, on a notre contenu dans du HTML et un style magnifique dans du CSS, mais comment générer un PDF avec ça ?

Il existe de nombreux outils différents qui permettent de transformer du HTML/CSS en PDF. Chacun de ces outils a ses forces et ses faiblesses, mais tous restent un minimum interopérables car utilisant ces mêmes standards HTML et CSS.

Parmi ces outils, le premier que vous pouvez essayer facilement n'est rien d'autre que… votre navigateur.

Imprimer votre page HTML vers un document PDF va, sans grande surprise, vous donner un PDF. Cette solution est très pratique et ne nécessite pas d'installer quoi que ce soit sur votre ordinateur.

La génération de fichiers PDF avec votre navigateur préféré a ses limites. La principale est que votre navigateur est conçu pour vous permettre de naviguer sur le web. C'est sa fonctionnalité principale (vous le saviez déjà) et c'est ce qui fait que les navigateurs sont loin d'intégrer toutes les propriétés et fonctions CSS spécifiques à l'impression, comme les notes de bas de page.

C'est pour cela que des outils spécifiques ont été développés. Contrairement aux navigateurs, ces outils sont spécifiquement développés pour générer des documents paginés, et sont plus regardants concernant les fonctionnalités spécifiques à ce médium.

Mais, quels sont ces outils ? On peut citer par exemple PagedJS, Vivliostyle, PrinceXML, PDFreactor, Antenna House ou encore WeasyPrint.

Parmi ces outils, certains sont libres et facilement accessibles :

- PagedJS est une bibliothèque JavaScript qui, en plus de transformer votre HTML/CSS en PDF, vous permet de voir le rendu de votre document en direct dans le navigateur.

- Vivliostyle est lui en TypeScript, utilisable en ligne de commande directement depuis votre terminal et vous permet de prévisualiser également votre PDF. Mention spéciale pour Vivliostyle qui gère particulièrement bien l'écriture de droite à gauche et de haut en bas (et oui, il y a aussi ça à prendre en compte quand on fait des documents).
- Quant à WeasyPrint, il s'agit d'une bibliothèque Python qui s'utilise directement en ligne de commande ou que vous pouvez appeler depuis votre application Python. Et c'est un outil qui vaut particulièrement le coup d'être essayé, puisque c'est nous qui le développons 😉.

Si vous voulez voir ce que sont capables de générer ces outils, n'hésitez pas à jeter un œil aux:

- livres créés avec PagedJS (Anglais) ;
- livres et magasines créés avec Vivliostyle (Anglais) ;
- livres, rapport, lettres et autres créés avec WeasyPrint (Anglais).

Ça donne un peu envie d'essayer, non ?

La diversité du web

La création de documents PDF avec les technologies du web est une niche, certes, mais elle fourmille d'idées, d'outils, de solutions. Les personnes de ce microcosme partagent souvent des valeurs et des envies, et confrontent parfois avec véhémence leurs points de vue antagonistes. Tout le monde n'est pas d'accord, bien évidemment 😁, mais au moins les choses avancent grâce à l'implication des personnes derrière les standards, des développeurs d'outils et de leurs utilisateurs.

Ce monde qui bouillonne, c'est le monde du web tel que nous le connaissons, tout du moins à présent. Les plus anciennes et les plus anciens d'entre vous se rappellent sans aucun doute les heures sombres du web du début des années 2000, avec Internet Explorer 6 et ses 90% de parts de marché, qui n'avait que faire de l'interopérabilité et de l'innovation comme tout ogre omnipotent qui se respecte.

La position actuelle de Chrome (Anglais) se rapproche chaque jour un peu plus d'une situation quasi monopolistique. À court ou moyen terme, cela pourrait replonger un écosystème particulièrement innovant dans la torpeur et les contraintes imposées par un despote. Google prend de moins en moins de pincettes pour pousser dans son navigateur des outils de surveillance et pour favoriser la publicité en ligne. L'entreprise ne le fait pas par pure méchanceté, elle le fait pour une raison simple : accroître ses revenus. Alors qu'un relatif équilibre avait été trouvé avec les autres implémentations, les tendances actuelles (désamour pour Firefox, utilisation du moteur de rendu de Chrome par Edge et Opera) laissent à penser que le rapport de force n'existe plus vraiment pour contrer l'agenda assumé du moteur de recherche.

Face à cela, il nous semble important de continuer à jouer le jeu de l'interopérabilité. Le web, et internet en général, fonctionnent grâce à de nombreux outils qui reposent tous sur les mêmes protocoles, langages et formats, comme TCP/IP, HTTP, HTML, CSS, JavaScript… Une des forces de ces technologies réside dans leurs multiples implémentations : que vous fassiez un site statique, une API REST, un scraper, une web app ou un service de streaming, vous allez pouvoir choisir les langages que vous aimez et les outils que vous préférez. Tout cela est possible parce que ces technologies sont le fruit de consensus au sein de consortiums, qui tentent de privilégier au mieux l'intérêt des usagers sur l'intérêt d'un acteur privé

ou public particulier. **Et au final, pour nous, participer aux consensus en développant cet usage alternatif est une manière modeste mais sincère de tenter de faire vivre ce web ouvert.**

Nous sommes Lucie et Guillaume, et comme beaucoup de gens nous travaillons tous les jours avec les technologies du web. Nous aimerions que cela continue longtemps, nous aimerions pouvoir encore inventer et mettre en pages nos contenus, indépendamment des volontés de gros acteurs dont nous ne partageons pas forcément les objectifs. Alors, à notre toute petite échelle, avec nos jolis documents débordant de belles lettres et de couleurs harmonieuses, avec nos outils libres et nos standards ouverts, avec notre passion et notre bonne humeur, nous tentons de faire vivre cette diversité du web 🌱.

À propos de Lucie Anglade et Guillaume Ayoub

Professionnels de la mise en pages du web à CourtBouillon, nous sommes également tous deux développeurs de logiciels libres, amateurs de cuisine et toujours partants pour discuter autour d'un verre ! Nous organisons aussi les Meetup Python de Lyon.