

Du site dynamique au document imprimable

Guillaume Ayoub – Kozea – PyConFr 17 octobre 2015



Votre gentil présentateur

- Guillaume Ayoub, a.k.a. gayoub, liZe, Yabz
- Cogérant de Kozea – Plateformes digitales pour professionnels de santé, logiciel libre et baby-foot
- Utilisateur de Python depuis 10 ans
 - Radicale – Serveur de calendriers et contacts
 - CairoSVG – SVG → PNG
 - Pyphen – Cou-pu-re de mots
 - WeasyPrint – HTML+CSS → PDF



Mensonge & dogme

Ne croyez pas tout ce que je dis.

Faites semblant un instant.



Web ?

Présenter des données structurées mais hétérogènes sous forme de documents liés et interactifs



Rendu du navigateur

Des **pages** dynamiques liées **et** des **outils** **et** un **navigateur**



Pourquoi rendre les documents imprimables ?

- Archivage
- Affichage
- Présentation
- Lecture
- ...

Et je ne vous parle que de ce qui est automatisable, dont le rendu fonctionne quel que soit le contenu.



Rendu imprimable

Des **pages** statiques ordonnées **et** des **outils** **et** un **support**



Bonne idée n°1 : mise en page manuelle

On va créer des fonctions pour dessiner les paragraphes, les tableaux, les graphes, les images, les images à côté des paragraphes dans les tableaux, et le reste.

- Mise en page = code
- Style = code

On a vraiment du boulot, mais ça va bien se passer.



Bonne idée n°2 : format automatiquement paginé

On va générer du RTF, du RTFD, du LaTeX, du OpenDocument, du OfficeOpenXML.

- Mise en page = outil de rendu, ou presque
- Style = règles de style, ou presque

On a du boulot, mais ça va forcément bien se passer.



Bonne idée n°3 : HTML

On va faire une application pour générer des documents en HTML pour l'impression et faire des feuilles de style en CSS (oui, on peut), puis on va les transformer en PDF (oui, on peut).

- Mise en page = moteur de rendu CSS
- Style = feuille de style CSS

On a un peu de boulot, mais ça va très bien se passer.



Très bonne idée

On n'a qu'à utiliser le HTML et le CSS de l'application !

C'est génial !



Python là dedans ?

Une bonne pile d'outils :

- Flask
- Pygal
- WeasyPrint
- Flask-Weasyprint
- CairoSVG

Beaucoup d'autres outils sont disponibles et (presque) aussi performants sur ces problématiques.



Comment faire son HTML

- On fait une structuration sémantique du HTML
- On n'essaie pas de faire du style en HTML
- On utilise des technologies qui se dégradent bien en affichage statique



Comment faire son CSS

- On sépare le commun, le *screen* et le *print*
- On se forme sur le CSS
- On se forme sur *Paged Media*
- On se forme sur la mise en page et la typographie
- On croise les doigts



Mensonge & dogme

Au fond de vous, vous savez déjà ce qui ne fonctionne pas.

Demandez-moi !



Contact & liens

kozea.fr

github.com/Kozea

guillaume.ayoub@kozea.fr

