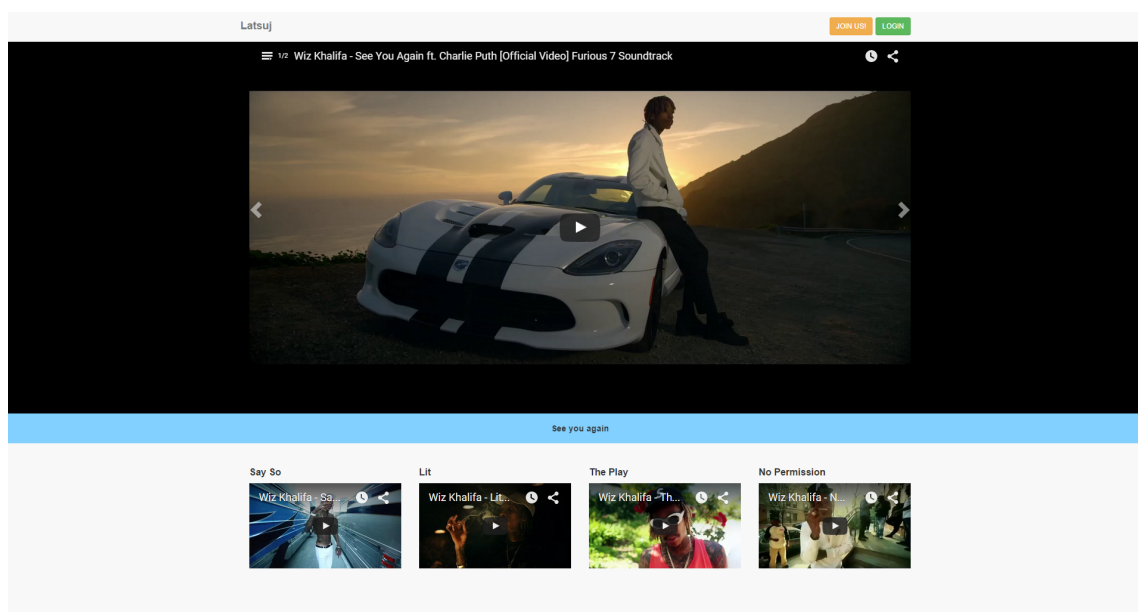


Bootstrap contre Polymer

DOSSIER D'ANALYSE DES DIFFERENCES



JUSTAL KEVIN

2015

Justal Kevin - justal@polytech.unice.fr - SI5 - IHM

Enseignant :
Anne Marie Dery - dery@polytech.unice.fr

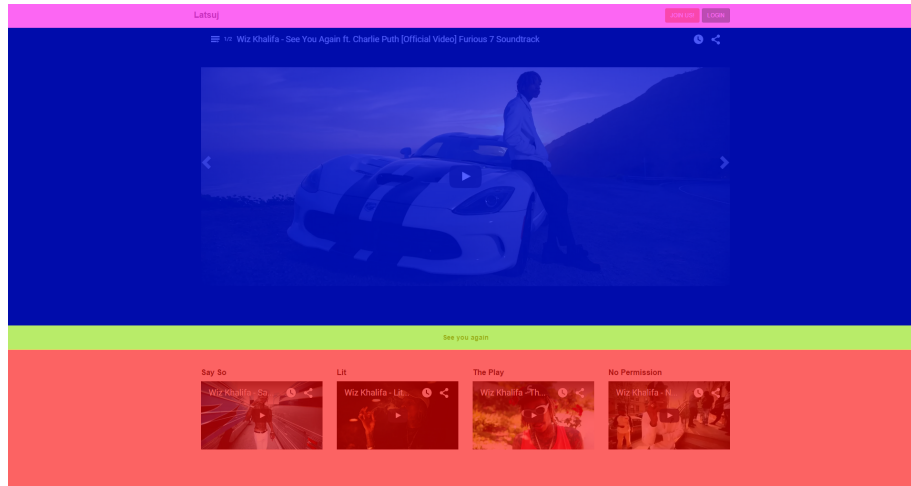
Table des matières

| | | |
|----------|---|----------|
| 1 | Site Web Adaptatif | 3 |
| 2 | Boostraps | 4 |
| 3 | Documentations, Outils, liens utiles | 4 |
| 4 | Difficultés rencontrés | 5 |
| 5 | Simple trouvaille d'optimisation | 5 |

1 Site Web Adaptatif

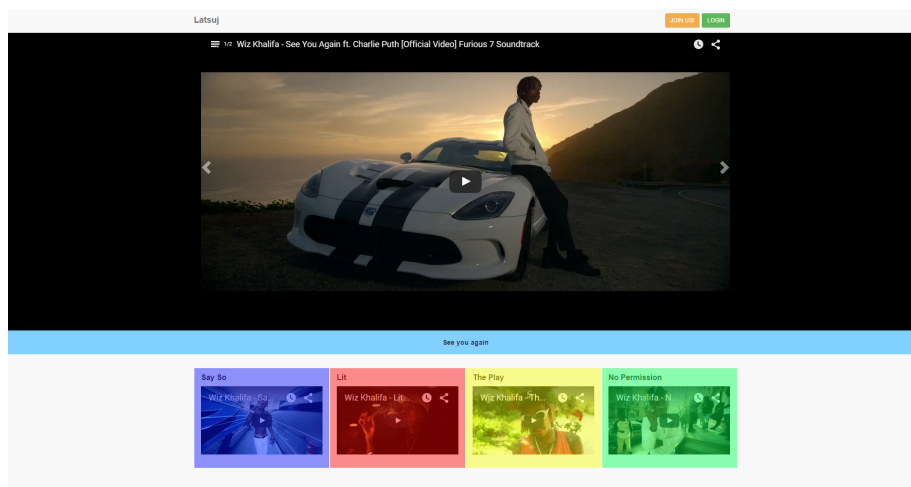
Le site a été pensé sur le principe du RWD (*responsive web design*) ou site web adaptatif dans la langue de molière. Ce concept s'appuie sur l'usage des *Media queries*, des grille ou encore des images flexibles. Au fur et à mesure que nous creuserons les deux frameworks, nous verrons que Polymer s'appuie principalement sur les media queries tandis que Bootstrap lui, possède en plus un système de grille très utile.

Pour développer, je suis parti de la version ordinateur, puis j'ai remis en forme les éléments à mesure que la largeur de l'écran diminuait voire je les supprimais. Nous verrons un peu plus loin pourquoi j'ai choisi de supprimer des éléments.

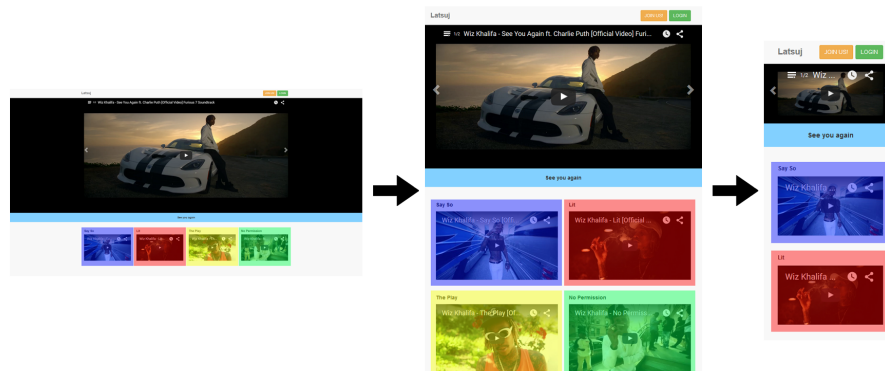


Le site a un découpage en quatre grands blocs. En rose sur l'image ci-dessus, on trouve la barre de connexion. En bleu, la zone de visionnage des vidéos. En Jaune, une zone d'information. Enfin, en rouge, une zone pour afficher les vidéos où le chanteur est le même que la vidéo dans la zone bleu.

Pour observer précisément le principe de RWD, nous allons nous intéresser particulièrement à la zone rouge en bas du site. Cette dernière illustre à la perfection tous les aspects que l'on attend d'un site adaptatif. Colorions chaque divisions de cette partie du site d'une couleur unique.



Si nous réduisons la largeur de la fenêtre, le contenu s'adaptera. Dans un premier temps, il n'y aura plus que deux blocs par ligne. Puis, si nous continuons de réduire la fenêtre, il n'y aura plus qu'un seul bloc par ligne et les deux derniers auront été caché.



Le site s'adapte donc aux dimensions de notre appareil ou fenêtre. À quoi cela peut-il bien servir ? Il serait plus adapté de se demander quels sont les problèmes que cela résout-il ? Comme on le voit autour de nous, les ordinateurs ne sont plus les seuls éléments ou gadgets nous entourant, ils existent maintenant une innombrable quantité d'appareils informatiques de tous types et de toutes dimensions. Il est important qu'un site internet ne laisse aucun utilisateur sur le bas côté. Comme il est impensable de concevoir une application ou un site internet pour chaque appareil, il faut donc faire un site qui puisse s'adapter suivant les dimensions de l'appareil.

Mais ce n'est pas tout, l'adaptation seule ne permet pas d'établir ce que l'on peut appeler un site web adaptatif. Le créateur de cette vision, Mr. Ethan Marcotte, a implémenté un site (<http://alistapart.com/d/responsive-web-design/ex/ex-site-flexible.html>) qui s'adapte à la largeur de l'écran. Cependant, il pointe du doigt certains détails. Par exemple, lorsque l'on redimensionne la page, les éléments vont bien se redimensionner mais les images et le texte à très basse résolution deviendront illisibles. Il faut donc que les éléments se repositionnent dans la page afin que le contenu soit lisible et agréable à parcourir pour l'utilisateur.

Le site est aussi *responsible typesetting*. La taille en pixel du texte varie suivant la taille de la fenêtre. Pour l'utilisateur, il est sans aucun doute plus agréable de pouvoir lire les paroles d'une chanson phrase par phrase. Or, si la taille du texte restait la même pour toutes les dimensions de fenêtre, soit le texte serait illisible à une grande résolution, soit le site serait inconfortable à basse résolution. Pour résoudre ce problème, une proportion a été spécifiée pour l'ensemble des textes suivant la largeur de la fenêtre ou de l'appareil.

Fitt's law

2 Bootstraps

3 Documentations, Outils, liens utiles

Ethan Marcotte

<http://alistapart.com/d/responsive-web-design/ex/ex-site-flexible.html>

<http://alistapart.com/d/responsive-web-design/ex/ex-site-linearize.html>

Responsible typesetting <http://blog.line0.eu/responsible-typesetting/>

4 Difficultés rencontrés

Le premier problème rencontré fut lorsque que j'essaya de coder une balise div de telle manière que celle-ci remplisse entièrement l'espace de l'application. Cette chose extrêmement simple n'est pourtant pas implémenté dans Bootstrap 3.0 et les versions supérieur alors que cela se trouvait dans les versions antérieur avec la class span. Après de longue recherches, il apparait donc impossible en pur Bootstrap de remplir un div à cent pour cent de la balise parent. De ce fait, j'ai du modifier le CSS pour réaliser le remplissage de la page. Pourquoi un tel choix des développeur de bootstrap ?

margin-bottom : Seriously ?

Compatibilite : WTF polymer !

min-height ? WTF do not work ! OK parce que tous ces putain d'elements sont en inline et non en block... Ok l'erreur

Suivre un ordre pour appeler les modules au depart, les enfant en premier.

encapsulation des elements ? content :X Merci la doc....pourrie.

Le carousel une horreur sur Polymer....

5 Simple trouvaille d'optimisation

En farfouillant sur les documentations de Bootstrap, je suis tombé sur une optimisation qui a retenu mon attention. Une chose simple et pourtant efficace que je ne faisais pas moi non plus. Les developpeurs de Bootstrap mettent toujours les scripts javascript en fin de page afin d'accelerer le chargement de la page. Cela peut paraitre stupide comme remarque mais je tiens à m'en souvenir, j'en fait donc par dans mon document.

Petite astuce, enlever les ; sur le dernier elements de css pour gagner un caractere de lecture.