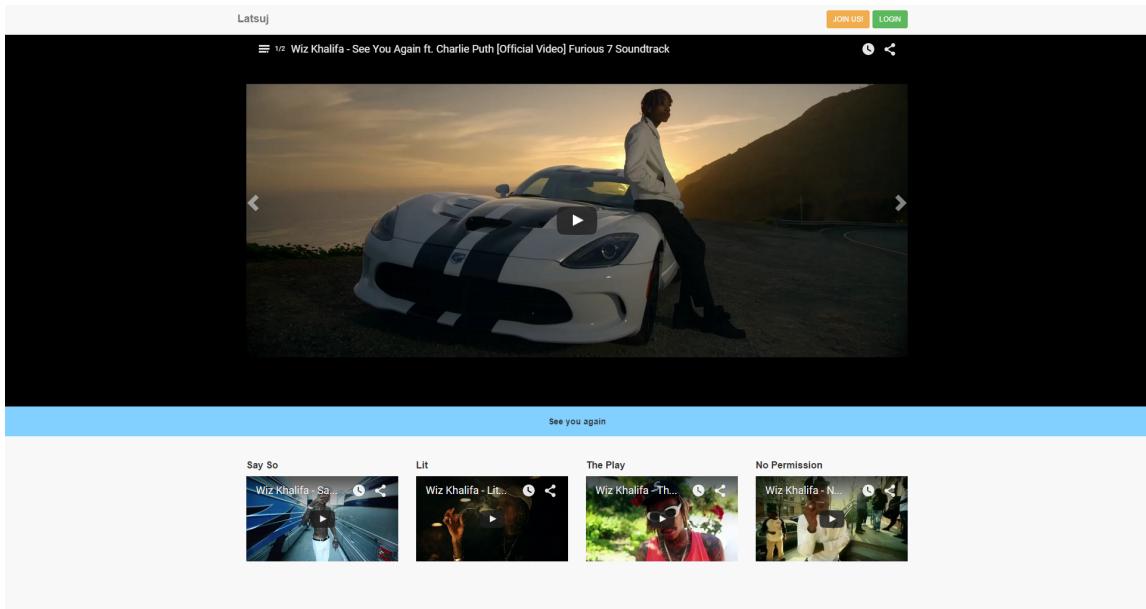


# Bootstrap vs Polymer

DOSSIER D'ANALYSE DES DIFFÉRENCES



**JUSTAL KEVIN**  
2015

Justal Kevin - [justal@polytech.unice.fr](mailto:justal@polytech.unice.fr) - SI5 - IHM

Enseignant :  
Anne Marie Dery - [dery@polytech.unice.fr](mailto:dery@polytech.unice.fr)

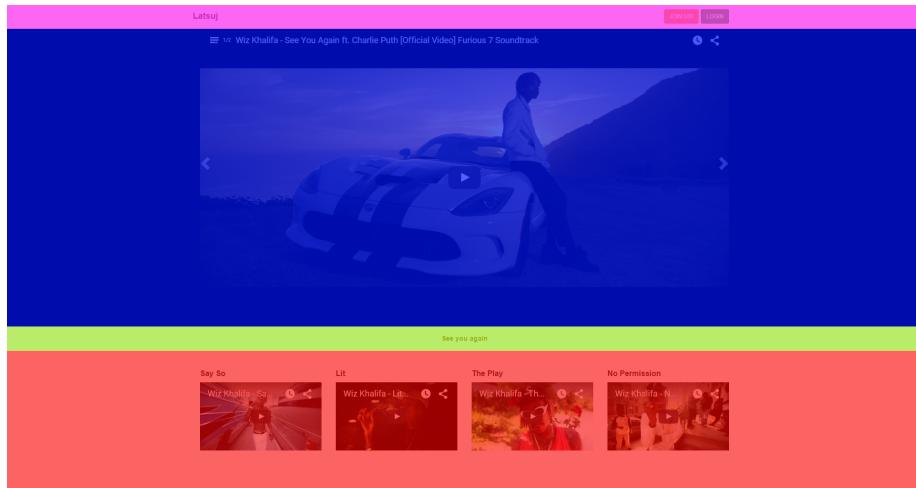
## Table des matières

1	Comment avez vous réalisé l'exemple ?	3
2	Comment teste-t-on les capacités d'adaptations ?	5
3	Boostraps	6
4	Documentations, Outils, liens utiles	6
5	Difficultés rencontrés	6
6	Simple trouvaille d'optimisation	6

## 1 Comment avez vous réalisé l'exemple ?

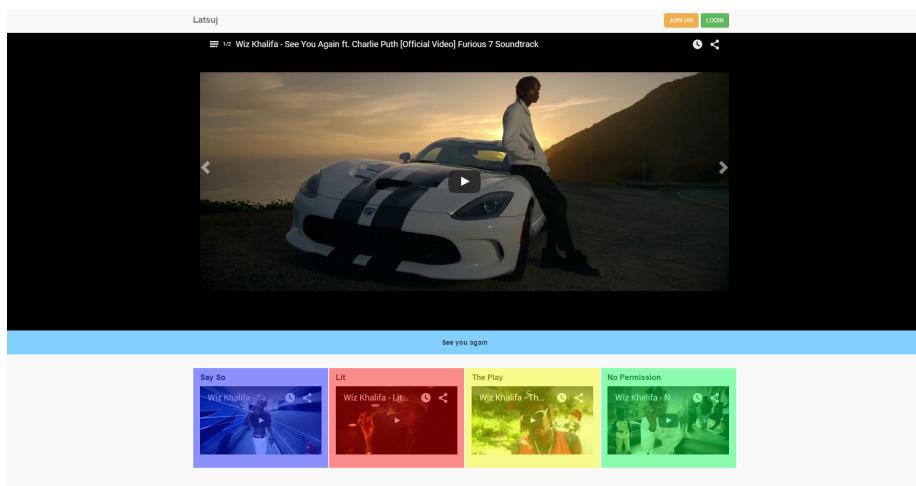
Pour réaliser mon exemple, je me suis dans un premier temps intéressé à la définition même d'un site web adaptatif. Le principe du RWD (*responsive web design*) ou site web adaptatif dans la langue de molière consiste à s'appuyer sur l'usage des *Media queries*, de grilles de positionnement ou encore d'images flexibles pour rendre un site adaptable à son support.

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 totalement. Nous verrons un peu plus loin pourquoi j'ai fait ces choix.

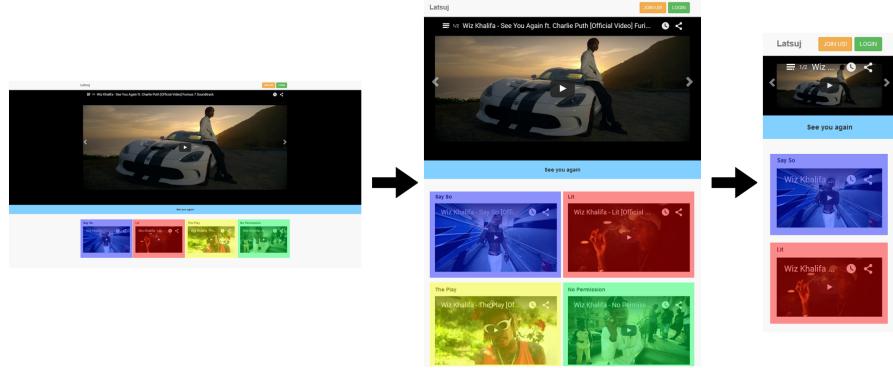


Les deux technologies que j'ai choisie partage la même structure de base. C'est à dire un découpage en quatre grands blocs. En rose sur l'image ci-dessus, on trouve la barre de connexion. En bleu, la zone de visionnage des vidéos. En Jaune, une zone d'information. Enfin, en rouge, une zone pour afficher 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.



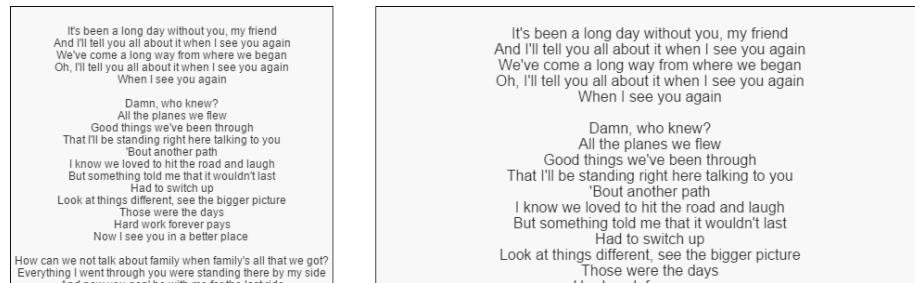
Si nous réduisons la largeur de la fenêtre, le contenu s'adaptera. Dans un premier temps, il n'y aura plus que deux blocs par ligne. Puis, si nous continuons de réduire la fenêtre, il n'y aura plus qu'un seul bloc par ligne et les deux derniers auront été caché.



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, il existe maintenant une innombrable quantité d'appareil informatique 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 seul ne permet pas d'établir ce que l'on peut appeler un site web adaptatif. Le créateur de cette vision, Mr. Ethan Marcotte, a implementé 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 bel et bien se redimensionner mais les images et le texte à très basse résolution deviendront illisible. Il faut donc que les éléments se repositionnent dans la page afin que le contenu soit lisible et agréable à arpenter pour l'utilisateur. C'est pourquoi dans l'exemple ci-dessus, le nombre de bloc par ligne décroît au fur et à mesure que l'on réduit la fenêtre.

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 dimensions de fenêtre, soit le texte serait illisible à une grande résolution, soit le site serait incommodé à basse résolution. Pour résoudre ce problème, une proportions a été spécifier pour l'ensemble des textes suivant la largeur de la fenêtre ou de l'appareil. Sur le site, on retrouve cette particularité sur plusieurs titres et sur les paroles comme nous pouvons le constater ci-dessous. À gauche, on retrouve les paroles des chansons sur téléphone portable tandis que à droite, on retrouve les mêmes paroles écrite avec une plus grande police sur tablette.



Pourquoi ais-je supprimé les deux derniers blocs sur la navigation à basse résolution ? Ce n'est pas une décision anodine. En supprimant ces deux blocs, j'améliore l'expérience de l'utilisateur sur deux aspects.

L'un est purement lié à la technologie, il est rare d'avoir un téléphone branché en Ethernet. Ceci implique que le débit moyen d'un utilisateur sur téléphone est souvent inférieur à celui d'un utilisateur sur ordinateur. J'évite ainsi à l'utilisateur sur téléphone de charger trop d'informations qui n'appartiennent pas au contenu principal de la page. Ce ne sont que des publicités pour les autres musiques du même chanteur. Deuxièmement, et c'est sans doute le point le plus important, suivant **la loi de Fitt**, que je suis sur téléphone ou ordinateur l'indice de difficulté doit rester le même. La loi calcul donne un indice de difficulté par rapport au temps requis pour aller rapidement d'une position de départ à une zone finale de destination. L'utilisateur doit arriver avec la même rapidité et la même facilité aux différentes parties du site. Si l'on regarde l'image sur la droite qui représente le site sur un téléphone BlackBerry, on remarque que le temps pour parvenir à la dernière vidéo en rapport avec le chanteur est aussi rapide sur le téléphone que sur ordinateur. Certes ce n'est pas la même, mais cela reste la dernière vidéo. Si j'avais juste réarrangé les choses, il aurait d'abord fallu descendre pour arriver à la dernière vidéo. Cela ne paraît pas beaucoup plus compliqué mais sans cela, le site se retrouverait complexifié d'après la loi de Fitts.

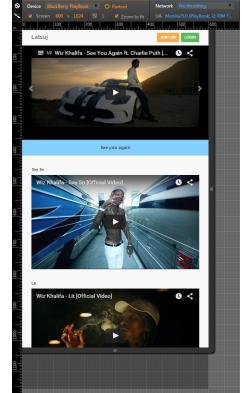


Figure 1: *Affichage du site sous BlackBerry (via Google Chrome).*

## 2 Comment teste-t-on les capacités d'adaptations ?

Pour tester l'affichage et l'adaptation de nos éléments à la fenêtre, il existe plusieurs voies envisageables. J'ai utilisé plusieurs d'entre elle pour effectuer mes tests. La première méthode a été de modifier la taille de la fenêtre sous windows.

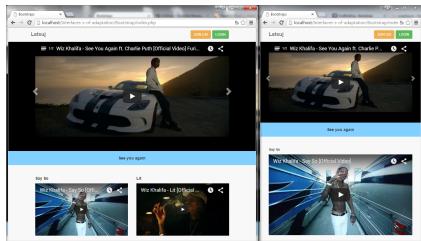


Figure 2: *Adaptation du site à la largeur de la page.*

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor

semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

### 3 Boostraps

## 4 Documentations, Outils, liens utiles

### Wikipédia

Le site d'où j'ai démarré mes recherches, il contient une bonne définition des sites RWD.

[https://fr.wikipedia.org/wiki/Site\\_web\\_adaptatif](https://fr.wikipedia.org/wiki/Site_web_adaptatif)

### What is a responsible web design ?

Les liens suivant sont les articles ou vidéos que j'ai analysé pour écrire ce rapport.

<https://www.youtube.com/watch?t=133&v=iSY38POjLYc>

### Ethan Marcotte

Le site du créateur du RWD qui montre la différence entre un site adaptable et un site responsive.

<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

Un article qui traite du responsible typesetting.

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

### Fitt's law

La description de la loi de Fitt

[https://en.wikipedia.org/wiki/Fitts's\\_law](https://en.wikipedia.org/wiki/Fitts's_law)

<http://webdesign.tutsplus.com/articles/applying-fitts-law-to-mobile-interface-design-webdesign-6919>

## 5 Difficultés rencontrées

Le premier problème rencontré fut lorsque que j'essayai 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 modifié le CSS pour réaliser le remplissage de la page. Pourquoi un tel choix des développeur de bootstrap ?

margin-bottom : Seriously ?

Compatibilité : 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 départ, les enfant en premier.

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

Le carrousel une horreur sur Polymer....

## 6 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 faisait pas moi non plus. Les développeurs de Bootstrap mettent toujours les scripts javascript en fin de page afin d'accélérer le chargement de la page. Cela peut paraître 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 éléments de css pour gagner un caractère de lecture.