

---

# BOOTSTRAP VS FOUNDATION

Plasticité des Interfaces – Flavien BOSSIAUX (SI5/IHM)

Ces deux frameworks évoluant très rapidement, il est important de prendre en compte que ce document fait état des versions en date de novembre 2014, soit Twitter Bootstrap 3.3.0 et ZURB Foundation 5.4.7

## INTRODUCTION

Bootstrap et Foundation sont tous les deux des framework front-end, mettant à profit des technologies issues du web telles que HTML5, CSS et JavaScript afin de fournir une collection d'outils et modules facile à utiliser, réutiliser et intégrer.

Appelé à l'origine Twitter Blueprint, Bootstrap a été créé courant 2010 par Mark Otto et Jacob Thornton, suivi par beaucoup d'autres développeurs de Twitter grâce au succès de leur première Hack Week.

Il deviendra ensuite le guide de style des outils internes à Twitter avant d'être publié le 19 Août 2011 comme Open Source sur GitHub, où il est encore aujourd'hui le projet le plus populaire.

Développé en interne chez ZURB (une agence de design de produits, souvent web) comme outil de prototypage, et style de base pour tous leurs clients depuis 2008, il fut rendu public sous le nom de Foundation en 2011.

Ils intègrent en particulier :

- Un reset CSS (Normalize.css)
- Une feuille de style complète (écrite grâce à un préprocesseur CSS)
- Un système de grille pour la mise en page et le responsive design, qui sera le plus intéressant pour une approche « Plasticité »
- Des composants statiques
- Des modules JavaScript (écrite grâce à la librairie jQuery)

# BOOTSTRAP VS FOUNDATION

## QUEL CONTEXTE D'USAGE ?

Ici, on entend par contexte d'usage une association Utilisateur – Plateforme – Environnement.

### L'UTILISATEUR

Les deux technologies étant à l'origine interne à des structures plutôt axées nouvelles technologies, on peut imaginer que les premiers utilisateurs étaient des personnes actives, à l'aise avec la technologie et une situation sociale convenable. Mais ce constat est aujourd'hui obsolète.

Il est difficile de catégoriser un utilisateur précis, car il est compliqué de catégoriser les utilisateurs du web. Grâce à ces technologies, on produit des sites web, mais aussi des applications mobiles et ces deux domaines se sont démocratisés au point qu'on ne puisse plus distinguer un utilisateur type.

On ne peut pas catégoriser le sexe, car il y a des sites pour hommes et pour femmes. (l'Équipe, Mademoiselle).

On ne peut non plus cibler une catégorie d'âge, car il y a des sites pour enfants, pour ados, pour adultes et pour séniors. (Site de jeux, de musique, réseaux sociaux, professionnels, de service, de commerce).

Il est ici également compliqué de cibler une catégorie sociale, car de nos jours Internet est accessible pour tout le monde (dispositifs publics, ...) plus ou moins facilement.

On ne peut pas non plus parler de maîtrise de la technologie, car on peut faire des sites simples pour trouver une recette de cuisine, comme un site à but industriel très compliqué et technique.

Pour autant, l'absence de catégorie précise nous permet tout de même de conclure sur une idée globale de l'utilisateur : C'est tout le monde et n'importe qui.

Ces outils ne possèdent aujourd'hui aucun système permettant de détecter et s'adapter à son utilisateur. Elle nous permet cependant, sans s'adapter, de créer des interfaces pour tous types de personnes, mais pas de changer intelligemment entre plusieurs utilisateurs.

### LA PLATEFORME

Bootstrap et Foundation sont tous les deux des solutions web, on abstrait donc ici une certaine notion de plateforme logicielle au profit d'une autre. En effet, les deux outils supportent les systèmes d'exploitation les plus connus (PC, MAC, UNIX, Android, iOS, Windows Phone, Firefox OS, BlackBerry OS), et cette donnée n'est pas aussi pertinente que la suivante en termes de plateforme.

Peu importe le système d'exploitation, ici la réelle plateforme logicielle est le navigateur web et bien évidemment, sa version qui compte vraiment.

Pour ce qui est de la plateforme matérielle, elle est aussi variée que le nombre de dispositifs permettant d'accueillir un navigateur web.

On peut imaginer un petit téléphone peu puissant en calcul, mais également un smartphone performant. On peut continuer l'ascension des tailles d'écrans sur les phablets, de plus en plus populaires et souvent très performantes, pour poursuivre dans le domaine des tablettes qui sont très variées sur leur puissance. On poursuivra naturellement vers des ordinateurs portables, puis des écrans d'ordinateurs de bureau, pour finir sur des écrans de type TV (TV connectée, Navigation sur console de Salon sur la TV). On peut donc balayer toutes les tailles d'écrans, d'un téléphone à un écran de TV.

On peut également trouver de grandes disparités pour ce qui est de la puissance, entre un téléphone, une console de salon, et un ordinateur de bureau.

On ne peut même pas assumer que tous les objets soient connectés au net, en effet, on peut très bien imaginer une application mobile offline.

D'un point de vue plasticité, ces outils ne nous permettent pas d'appréhender les différences des puissances des dispositifs, mais essayent de fournir une solution pour ce qui est de la taille d'écran par le biais d'un système de grille, et de leur densité de pixels. Foundation propose un plugin JavaScript pour essayer de répondre au problème du chargement d'images : Interchange, détaillé plus tard.

## L'ENVIRONNEMENT

Les utilisateurs étant très variés, sur des dispositifs hétérogènes, on peut imaginer énormément d'environnements différents. Une application de recette pourra être utilisée avec des mains sales du bout des doigts, avec beaucoup de bruits environnants, à l'inverse, un intranet professionnel sera utilisé dans le calme d'un open-space sur un grand écran de bureau, alors qu'un site social pourra être utilisé dans son lit avant d'aller dormir dans le silence et l'obscurité, et un site affichant des alertes routières pourra être conçue pour un simple affichage sans interactions.

Ces deux technologies ne permettent pas à ce jour, de réagir à des paramètres comme le bruit, à la luminosité, ou la localisation géographique, ...

## EN BREF

Il est très compliqué de déterminer le contexte d'usage de technologies faites pour créer des solutions webs (mobiles, bureau, ...), car cela dépend trop de la solution en question.

On peut cependant dire que ni Bootstrap, ni Foundation ne permettent pas, seuls, de s'adapter à son utilisateur, ou à son environnement. En revanche, ils fournissent tous les deux des outils similaires pour gérer la diversité des dispositifs d'affichage (taille) bien que Foundation fournisse en plus un composant pour gérer les images en fonction de la taille de l'écran.

## QUEL MOMENT ?

### À LA CONCEPTION

Ces deux frameworks ont évidemment été conçus pour faciliter la vie des développeurs en rassemblant toutes les choses les plus utilisées dans le développement web.

Leur approche ultra-modulaire permet particulièrement de créer des composants indépendants, et réutilisables sur plusieurs sites. Il existe même, pour les deux frameworks, des banques de composants prêts à être utilisés. Les plus populaires seront par la suite intégrés officiellement.

Certains vont plus loin, et proposent directement des templates de site/application (basés sur ces frameworks), qu'il suffira ensuite de remplir. Ceci permet par la suite de diminuer les coups de développements. J'ai pu expérimenter ceci personnellement, en ayant la charge d'un site de location de yachts, qui n'était qu'une simple modification d'un site de vente de t-shirt.

Ces deux frameworks supportent deux modes de développements d'applications qui s'adapteront aux tailles d'écrans : Le Responsive Web Design, et l'Adaptive Web Design. L'Adaptive Web Design est arrivé le premier, et le Responsive Web Design est arrivé un peu plus tard. Leurs composants utilisent un mélange d'Adaptive et de Responsive.

Ils sont également construits sur la base de feuilles de styles écrites grâce à des préprocesseurs, ce qui rend les styles globaux plus plastiques, sachant que toutes les variables de styles sont réunies dans un fichier, il est donc facile de changer les couleurs, le thème et le look and feel général du framework.

Ici la multiplicité des langages n'intervient pas, puisque le web est régi par certains langages standards (HTML, CSS, JavaScript) qui est supporté par tous les matériels visés.

### À L'EXECUTION

La seule plasticité qui aura lieu grâce à ces frameworks est le Responsive WebDesign.

On peut ici trouver une première différence entre Twitter Bootstrap et ZURB Foundation. En effet, Twitter Bootstrap a une empreinte au niveau du style plus importante que ZURB Foundation.

En effet, Bootstrap est à l'origine un produit fait pour créer une solution complète, alors que ZURB Foundation est un outil pour poser des bases, des prototypes. Il garde donc à l'esprit qu'il ne se suffit pas à lui-même, et doit ensuite être stylé, ce qui n'est pas le cas de Bootstrap.

Du point de vue d'un développeur, cela lui permet d'avoir un look and feel déjà intégré, uniforme, et réutilisable partout, en plus de l'aide de l'énorme communauté qu'il y a derrière bootstrap, avec énormément de solutions pré-faites et s'intégrant parfaitement à bootstrap.

Du point de vue d'un designer, il aura beaucoup plus de difficultés à customiser le look and feel de bootstrap.

On peut donc dire que Bootstrap est plus plastique pour les développeurs, pouvant réutiliser des composants déjà stylisés, mais demandera plus d'efforts à un designer pour le customiser. À l'inverse, Foundation demandera aux développeurs de rajouter un style alors qu'il facilitera la tâche pour un designer.

## QUELLE SOLUTION ?

### LES PREPROCESSEURS CSS

Un préprocesseur CSS, est en fait un nouveau langage, une nouvelle syntaxe, inspiré de CSS, auquel on a rajouté de nouvelles fonctionnalités, comme des variables, des fonctions appelées mixins, une gestion des imports, l'imbrication des sélecteurs CSS, des fonctions utilitaires, des opérateurs etc...

Les navigateurs commencent à peine à prendre en charge cette syntaxe, non-standard. On considère qu'il faut écrire nos styles dans la nouvelle syntaxe, qui sera ensuite compilée en une feuille de style CSS classique. Pouvoir éclater les fichiers donne l'avantage de pouvoir mieux organiser son CSS dans le but de faciliter la maintenance et la propreté de ce code. On peut voir émerger certaines méthodes de travail avec ces processeurs (comme SMACSS par exemple).

Une première différence sépare Bootstrap de Foundation ici, car Bootstrap est écrit grâce au préprocesseur LESS, alors que Foundation utilise son principal concurrent : SASS.

On peut parler ici de plasticité, car ces deux préprocesseurs sont très similaires, au point qu'une version SASS de Bootstrap existe, qui n'est qu'une traduction de la version LESS. Ceci est rendu possible car ces deux langages partagent le même modèle, ainsi on peut assez facilement passer de l'un à l'autre.

Leur utilité dans la plasticité de ces deux frameworks est de pouvoir paramétrer le style facilement avec une suite de variable. Par exemple pour changer la palette de couleurs avec Foundation :

```
// $primary-color: #008CBA;  
// $secondary-color: #e7e7e7;  
// $alert-color: #f04124;  
// $success-color: #43AC6A;  
// $warning-color: #f08a24;  
// $info-color: #a0d3e8;
```

Un principe équivalent existe avec Bootstrap. Les versions compilées (donc CSS) des deux frameworks sont donc bien moins plastiques à la conception que leur équivalent avec préprocesseur.

Le modèle de ces syntaxes CSS est juste l'ajout, par rapport à du CSS classique, de concepts de programmation connus, comme des fonctions (mixins), des variables, et certaines facilités syntaxiques.

### LA GRILLE

Une autre solution, et sûrement la principale intégrée par ces deux frameworks et leur grille responsive. Le principe est simple, on divise un container en lignes (row), qui seront-elles-même

divisées en 12 colonnes (par défaut, c'est paramétrable grâce aux préprocesseurs CSS). Ensuite, on pourra définir chaque élément comme étant large de x colonnes.

En plus du rythme horizontal visuel, l'avantage est que l'on peut réorganiser la largeur des éléments et leur ordre en fonction de la taille de l'écran. Deux éléments côte à côte sur un écran large peuvent donc se placer l'un en dessous de l'autre à partir d'une certaine taille. Cela permet de réaliser facilement des sites responsive/adaptive.

Le modèle sous-jacent est assez simple. Notre page est divisé en ligne, large de 12 colonnes, qui seront partagés parmi les éléments de cette ligne.

### GESTION DE LA DENSITE DE PIXELS D'ECRAN

Il n'y a ici aucun équivalent propre à Bootstrap, c'est donc un point fort pour Foundation.

Foundation intègre un composant JavaScript appelé Interchange. Il s'utilise comme ceci :

```
<img data-interchange=["/path/to/default.jpg, (default)], [/path/to/bigger-image.jpg, (large)]">
```

Le principe de ce plugin, est de fournir une image adaptée à une densité, une résolution et une taille d'écran, puis de laisser ce composant charger l'image qui correspond au contexte. Cela évite d'avoir à charger sur un téléphone une image bien trop grosse et destinée à un ordinateur haute résolution, et inversement. On peut ici parler de plasticité puisque les images et leur qualité varieront dynamiquement en fonction des caractéristiques de l'écran.

Ce plugin va un peu plus loin, en ne limitant pas son utilisation à des images, mais à tout type de contenu. Ainsi, on peut afficher une image sur un petit écran, typiquement un téléphone, susceptible d'être sur un réseau téléphonique payant, en lieu et place d'une carte dynamique complète et interactive sur un ordinateur de bureau. On peut donc aller jusqu'à charger un contenu spécifique à certains types d'écrans à l'exécution, ce qui est un plus pour la plasticité.

## CONCLUSION ET AVIS PERSONNEL

Bien que ces deux frameworks, Bootstrap et Foundation soient similaires sur de nombreux points, on peut tout de même les différencier et leur trouver un usage différent qui remonte même à leur origine. En effet, Bootstrap a été un outil afin de créer de la documentation et des outils internes par des développeurs afin de ne pas se focaliser sur le style, mais les fonctionnalités. En revanche, Foundation est issue d'une start-up de product-designer, comme base à leurs projets, et outils de prototypages. On retrouve cette idée dans leur nom, Bootstrap sera donc plus fait pour obtenir un produit fini sans perdre de temps sur le style, alors que Foundation instaure des bases sur lesquelles travailler.

### POURQUOI PREFERER BOOTSTRAP ?

- Sa communauté est bien plus grande, il sera donc rarement utile de programmer un truc soi-même, quelqu'un d'autres l'ayant sûrement déjà fait.
- Si on est un développeur, et que l'on ne veut pas s'intéresser à la partie graphique, Bootstrap fournit un style, aujourd'hui très répandu, de base suffisant.
- Parce que l'on préfère le préprocesseur LESS à SASS.

### POURQUOI PREFERER FOUNDATION ?

- Le style de Foundation est pensé pour être changé, il est donc bien plus facile pour un designer de travailler avec.
- Parce que l'on préfère SASS à LESS (Le portage de SASS de Bootstrap, issue d'une traduction semble contre-intuitif).
- Foundation évolue beaucoup plus vite.

Un exemple, Foundation introduit la grille « moyenne » avec sa version 4.3 le 18 juillet 2013. Bootstrap réagit en ajoutant cette fonctionnalité le 14 août 2013 avec sa release candidate 2 de la version 3.0.0. Foundation intègre Orbit, un composant permettant de faire des sliders depuis septembre 2011, Bootstrap intégrera Carousel le 1<sup>er</sup> Février 2012.

Foundation semble toujours avoir un petit temps d'avance, et la communication à propos des futures versions de Foundation semble le confirmer. Foundation for Email, anciennement ZURB Ink, était une version de Foundation adaptée pour e-mail, qui n'existe pas avec Bootstrap. Les futures versions de Foundation vont intégrer une grille à base de flex-box CSS, ayant un comportement proche de XUL, de Mozilla, permettant d'alléger la lourdeur de la grille.

Foundation prévoit de se séparer en deux branches, une Foundation for the Web, sera dédié à la création de sites web, alors que Foundation for Apps, s'intégrera avec AngularJS de Google, pour faciliter la création d'applications, notamment mobile. Rien de tout cela n'a été communiqué par Bootstrap, bien qu'une version (BootstrapUI d'AngularUI) soit un portage de Bootstrap pour AngularJS, ce n'est développé que par leur communauté, mais pas officiellement par eux.



Pour autant, je recommanderai Bootstrap, pour les personnes ne voulant qu'assembler des briques pour créer leur site web, et se suffisant de thèmes préfaits, ou du thème de base.

J'ai personnellement une grande préférence pour Foundation qui évolue plus vite, innove plus, et laisse beaucoup plus de liberté au niveau du style, qui est donc dans un sens plus plastique, au niveau de son style. Je le préfère également car il utilise SASS, au sommet duquel j'utilise COMPASS, qui est une suite de fonction utile pour le CSS, qui me permet de ne pas gérer les préfixes spécifiques aux utilisateurs notamment, et rends à la conception, encore plus plastique le préprocesseur CSS, en gérant lui-même la compatibilité entre les navigateurs.