Conclusion du stage

Responsive Web Design

1. La présentation de RWD
2. Définition du RWD :

Un site web adaptatif est une notion de conception de sites web qui regroupe différents principes et technologies dans laquelle un site est conçu pour offrir au visiteur une expérience de consultation optimale facilitant la lecture et la navigation. L’utilisateur peut ainsi consulter le même site web à travers une large gamme d’appareils (ordinateurs, smartphones, tablettes, etc.) avec le même confort visuel et sans avoir recours au défilement horizontal ou au zoom avant/arrière sur les appareils tactiles notamment, manipulations qui dégradent considérablement l’expérience utilisateur.

1. Techniques de RWD :

Un site web sur RWD utilise la technologie CSS3 media queries pour adapter mieux aux dispositifs avec la résolution différent.

a. Media queries permet à la webpage d'utiliser des règles CSS différentes en fonction des résolutions différentes des dispositifs. On peut modifier la mise en page par les points de rupture.

b. Le concept de grille fluide est réalisé en un dimensionnement relatif.(Comme les pourcentages ou em)

c. Les images ont un besoin de flexibilité et ils sont également redimensionnées en unité relative.

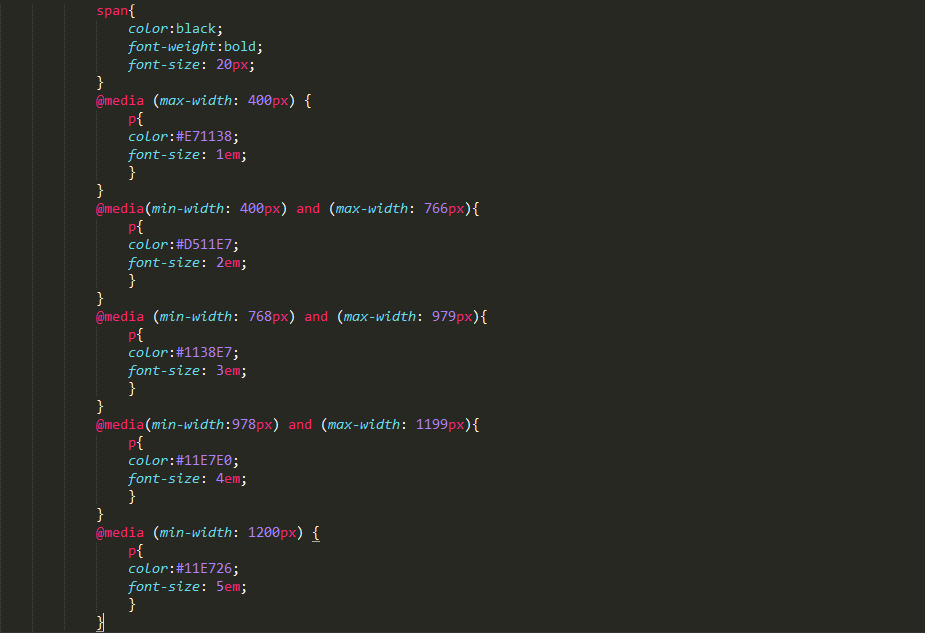
1. Analyse / réflexion théorique du RWD :

L’idée de base du RWD est que le même site web est affichée bien sur les différents dispositifs avec la résolution différente. Pour la résolution différente, la taille des caractères, la mise en page et la résolution des images sur un site web etc. ont besoins d’être modifiés.

Au début, pour un même site web, on fait plusieurs versions à distribuer les dispositifs différents. Mais ce n’est pas pratique, si on a besoins de modifier le texte du site web, il faut faire des modifications sur les versions différents. Et après, on fait une intégration pour simplifier ces opérations, et on fait un fichier .html pour montrer le texte et plusieurs fichiers .css pour préciser les stylesheets différents sur les dispositifs différents. Mais comme ça, on trouve qu’il y a beaucoup de fichiers .css et ce n’est pas pratique pour la gestion des fichiers.

Donc, RWD apparaît. Il intégre les stylesheets différents dans un même fichier .css par @media queries. Pour la même partie de stylesheet des stylesheets différents, on la écrit en dehors de @media, et les différents parties, on les écrit dans les @media différents.

La suivante est un petit exemple simple pour montrer comment utiliser @media queries (les caractères avec la taille différente et la couleur différente pour la résolution différente):



*min-width* et *max-width* sont le minimum et le maximum largeur où le stylesheet suivant est utilisé.

1. RWD résout principalement la meilleur montre avec les suivants changements :

a. Changement de résolution.

b. L’orientation des dispositifs(Horizontal ou vertical).

c. Interaction entre les utilisateurs et les dispositifs(Souris ou tactile).

Mais ici, j’ai analysé les deux premières questions, changements de résolution et l’orientation des dispositifs. Le noyau de ces deux problèmes est à adapter aux résolutions différents.

1. Techniques générales :
2. Changement de mise en page
3. Supprimer/cacher des parties (image)
4. Changement de parties (navigation)
5. Déplacement de parties (images ou form)
6. Taille de parties (taille de l’image ou taille du texte)
7. Frameworks utiles :

Actuellement, il y a plusieurs Frameworks supers sur le développement du RWD, c’est pratique pour satisfaire nos demandes de développer un site web adaptatif. Ici, je discute principalement 4 Frameworks gratuits le plus populaire : Bootstrap, Foundation, Skeleton et Html5boilerplate.

L’exemple suivant est une même webpage utilisant séparément 4 Frameworks :



Pour les frameworks ci-dessus, ils réalisent principalement un site web adaptatif par la grille fluide . Ils divisent un ligne en 12 petites unités, et l’emplacement de chaque unité dans la webpage n’est pas fixe. L’emplacement de chaque unité est modifié selon la résolution d’écran. Donc la phénomène ‘overflow’ n’apparaît pas et on peut réaliser un site web adaptatif.

Pour la rédacteur d’un site web, il ne met que les différents parties de son site web qui sont divisé selon les différents fonctions dans les différents unités.

1. Plusieurs modules sur RWD
2. Texte :
3. Déplacement d’une colonne

Exemple : <http://codepen.io/bradfrost/full/pwmHf>

1. Taille du texte  
   Exemple :



Noter : les utilisation des em, rem et %.



1. Navigation :

On a besoin de résoudre principalement :

----Il n’a pas de ‘overflow’ pour la navigation avec la diminution de la résolution du site web.

----Lorsque la résolution est assez petite, la navigation peut être modifié à une icône.

1. Ajouter d’un menu déroulant

Exemple : <http://codepen.io/bradfrost/full/sHvaz>

Un meilleure exemple (animation) :

Exemple : <http://www.adtile.me/fixed-nav/>

1. Un menu sélectionnant

Exemple : <http://codepen.io/bradfrost/full/hkuzA>

1. Un menu cache dans la gauche de la webpage

Exemple : <http://codepen.io/bradfrost/full/IEBrz>

1. Un menu collant (animation)

Exemple : <http://ethercycle.com/stickymenu/>

1. Une série des petits icons circulaires dans la droit de la webpage

Exemple : <http://dujuan.in/fixed-middle-right-nav/>

1. Un menu icône

Exemple : <http://tympanus.net/Development/ResponsiveMultiLevelMenu/index4.html>

Noter : Supposons que vous voulez mettre le navigation dans le haut de la disposition toujours. Même si l'utilisateur roule la webpage au bas de la page, la navigation est fixé au haut de la page, vous pouvez faire comme la suivante:

*<div class= “contain-to-grid sticky”>*

*<nav class=”top-bar”>*

*<!—nav content-->*

*</nav>*

*</div>*

1. Tables de données :

On a besoin d’assurer à éviter:

---- La table est ‘overflow’ dans la webpage.

----Le contenu est ‘overflow ’ dans la table.

1. Transformation de la table (un ligne = un unité)

Exemple : : <https://css-tricks.com/examples/ResponsiveTables/responsive.php>

1. Transformation de la table en liste

Exemple : <http://codepen.io/aarongustafson/full/ucJGv>

1. Les colonnes prioritaires

Exemple : : <https://www.filamentgroup.com/examples/rwd-table-patterns/>

1. Echanger les colonnes et les lignes

Exemple : <http://dbushell.com/demos/tables/rt_05-01-12.html>

1. Cacher la table

Exemple : <http://output.jsbin.com/apane6/14>

Par les exemples ci-dessus, on peut trouve :

Pour la table, une façon plus commun, RWD peut cacher les colonnes moins importantes quand la résolution d’écran diminue.

Ici , on peut distribuer les différentes noms de classes aux étiquettes (les colonnes d’une table) selon leur différents niveaus importants, ensuite on peut utiliser ‘display :none ;’ dans @media pour les classes spéciales selon les différents resolutions.

1. Calendrier :
2. Vue en mois devient vue en semaine

Il y a un exemple utilisant Jquery :



1. Réaliser plusieurs fonctions sur calendrier utilisant AngularJS, Bootstrap et AngularUI :



1. Graphiques :
2. Un exemple le plus simple est : lorsque la résolution d’un écran est modifiée, la taille et la mise en page des images est modifié aussi.

Exemple : <http://codepen.io/bradfrost/full/xkcBn>

1. Pour l’histogramme, il est également développé sous la forme adaptatif. Par exemple, on peut diminuer le contenu de l’histogramme et ne montrer que le contenu principal.



Noter : la théorie principale est pareille avec la table.

1. Form :

---- La taille de la zone de saisie est modifiés automatiquement.

---- La position relative entre la titre et la zone de saisie est modifiée automatiquement.

Exemple : <http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-forms.php>

Plusieurs exemples utilisant 4 frameworks (Bootstrap, Foundation, Skeleton et Html5boilerplate):



1. Forum :

Exemple : <http://www.pluxml.org/>

1. AngularJS (<https://angularjs.org/>)
2. Présentation :

AngularJS est un framework JavaScript libre et open-source développé par Google.

AngularJS est fondé sur l’extension du langage HTML par des nouvelles balises (tags) et attributs pour aboutir à une définition déclarative des pages web, par opposition à l’utilisation systématique de l’élément div​ et à la définition des éléments de présentation en javascript. Le code HTML étendu représente alors la partie « vue » du patron de conception MVC (Modèle-vue-contrôleur) auquel AngularJS adhère, avec des modèles appelé « scopes » et des contrôleurs permettant de définir des actions en code javascript impératif. AngularJS utilise une boucle de dirty-checking (qui consiste à surveiller et à détecter des modifications sur un objet JavaScript) pour réaliser un data-binding bidirectionnel permettant la synchronisation automatique des modèles et des vues.

AngularJS embarque un sous-ensemble de la bibliothèque open source jQuery appelé jQLite, mais peut aussi utiliser jQuery si elle est chargée.

1. Objectifs du framework :
2. Découpler les manipulations du DOM de la logique métier. Cela améliore la testabilité du code.
3. Considérer le test d'une application aussi important que l'écriture de l'application elle-même. La difficulté de la phase de test est considérablement impactée par la façon dont le code est structuré.
4. Découpler les côtés client et serveur d'une application. Cela permet au développement logiciel des côtés client et serveur de progresser en parallèle, et permet la réutilisabilité de chacun des côtés.
5. Guider les développeurs pendant toute la durée du périple de la construction d'une application : de la conception de l'interface utilisateur, en passant par l'écriture de la logique métier, jusqu'au test de l'application.
6. Rendre les tâches faciles évidentes et les tâches difficiles possibles.

La modèle d’AngularJS est réalisé par $scope.

L’AngularJS utilise la modèle MVC pour atteindre le but finale :Modulaire et Multiplexage.

1. Analyse / réflexion théorique d’AngularJS :

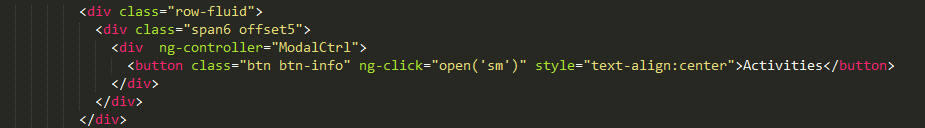
Le framework AngularJS utilise la modèle MVC pour réaliser Modulaire et Multiplexage de la programmation. On déclare un contrôleur pour une vue dans le fichier .html, et ensuite développe un contrôleur dans le fichier .js pour ajouter les intégration des utilisateurs (les fonctions). De cette façon, on peut séparer bien Modèle, Vue et Contrôleur, cela facilite plus la maintenance et la modification des codes au futur.

1. L’exemple simple d’AngularJS :

Au début, on fait une déclaration dans l’étiquette <html> pour informer le compilateur que la programmation est contrôlé par AngularJS depuis ici.



Et plus, on a besoins d’ajouter un contrôleur pour une partie de vue dans le fichier .html utilisant ‘ng-controller’.



Ensuite, on fait un module dans le fichier .js.

Enfin, on fait un contrôleur réel, et on peut ajouter plusieurs fonctions dans ce contrôleur.



Plus d'exemples sur AngularJS est la suivante :



1. Bilan

Grâce à ce stage, j’ai étudié principalement à utiliser les deux frameworks Bootstrap et AngularJS pour développer rapidement un site web adaptatif.

En général, si nous voulons développer un site web adaptatif, il faut suivre les étapes suivantes :

1. le contenu que le site web montre, les fonctions que le site web réalise et la mise en page sur les différents dispositifs.
2. Réaliser le fichier .html, et ajouter tout le texte dans le fichier .html.
3. Mettre les différentes parties de html dans les unités différents (e.g. Bootstrap), et modifier les @media différents dans le fichier .css selon vos plans de la mise en page des résolutions différentes.
4. Détecter l’adaptation de votre site web pour les résolutions différentes. (<http://www.alsacreations.com/article/lire/1634-comment-tester-un-site-responsive-partie-1.html>)
5. Réaliser les intégrations avec les utilisateurs dans votre site web.
6. Détecter tout le site web (e.g. adaptatif , intégration etc.) et enfin finir.

Bootstrap est un stylesheet, donc ce n’est pas difficile pour comprendre et utiliser. Mais pour le framework AngularJS, parce qu’il résout principalement l’intégration avec les utilisateurs, il y a beaucoup de fonctions et nous les comprenons difficilement. En même temps, ce framework utilise la modèle MVC, et c’est différent avec les autres programmations sur le fil de la pensée et la façon de résoudre les problèmes, donc il faut dépenser plus de temps pour utiliser bien.

Yishuo LYU

18/08/2015