

# TP2 - Mise en forme d'un quiz avec CSS

## 22 septembre 2016

### 1 Objectifs

Le but de ce travail pratique, réalisé en équipe de deux personnes exactement, est de vous familiariser avec CSS3. Vous commencerez aussi à vous familiariser à l'environnement du côté serveur en utilisant Pug (anciennement Jade) pour éliminer la répétition de code.

### 2 Introduction

Tout au long du trimestre, vous implémenterez un site web qui génère des questionnaires (quiz) pour les personnes qui désirent tester leurs connaissances sur les notions du Web. Nous nous attarderons principalement sur des questions sur le domaine du HTML5, CSS3 et Javascript, mais nous pouvons étendre ces questions à d'autres technologies du Web telles que HTTP, NodeJS, AngularJS, etc.

### 3 Principes de fonctionnement

Nous présentons dans cette section les principes de fonctionnement du questionnaire, afin de vous fournir une vision globale de ce que vous devrez implémenter tout au long du trimestre. Mais toute la logique et les calculs derrière ces principes ne seront pas implémentés avant le TP3.

La page centrale du site correspond au tableau de bord de l'utilisateur. Le tableau de bord donne la possibilité à l'utilisateur d'effectuer des tests rapides ou un examen officiel. Il contient également diverses informations sur les résultats des tests de l'utilisateur tels que le nombre de tests rapides réussis sur le nombre de tests rapides effectués, et la moyenne des examens.

Les tests rapides permettent à l'utilisateur d'améliorer ses connaissances sur le Web. Le choix des questions du test se fera de façon aléatoire selon une base de données. Cette base de données va contenir l'ensemble des questions possibles sur chaque domaine.

Les examens officiels permettent de tester officiellement les connaissances de l'utilisateur. À côté du bouton qui débute l'examen, l'utilisateur devra rentrer les paramètres de l'examen. Ces paramètres peuvent correspondre au domaine dans lequel il désire se faire évaluer (HTML, CSS ou Javascript) et le nombre de questions que devra contenir l'examen.

Les pages de tests rapides et d'examens sont très similaires.

La page de test rapide contient la question, ainsi qu'un choix multiple de réponses à la question (des boutons radio ou des cases à cocher). L'utilisateur a le choix de retourner au menu (sans avoir besoin de répondre à la question), ou de répondre à la question et de cliquer sur le bouton **suivant**. S'il clique sur le bouton **suivant**, une nouvelle question à laquelle l'utilisateur doit répondre s'affiche, et ainsi de suite.

La page d'examen ressemble énormément à la page de test rapide. Nous retrouvons une question avec les réponses possibles. Ensuite, l'utilisateur a le choix d'abandonner l'examen (en conséquence, il aurait une note de 0 pour cet examen), ou de répondre à la question et d'aller à la question suivante.

## 4 Travail à réaliser

Pour ce TP, vous pouvez partir du travail que vous aviez réalisé pour le TP1.

### 4.1 Installation d'un projet Express avec Pug

Avant de pouvoir utiliser Pug, vous allez devoir installer Node.js (déjà installé sur les postes de Poly). Le lien de téléchargement se trouve sur <https://nodejs.org/>.

Ceci vous permettra d'utiliser le Node Package Manager en ligne de commande (npm). Les commandes sont invoquées en utilisant *npm* suivi du nom de la commande. Nous utiliserons principalement les commandes *install*, *uninstall*, *start*, *stop*.

Lorsqu'on installe un paquet avec *npm install packageName*, on a quelques options. L'option *-g* installe le paquet globalement. D'habitude, ce type d'installation est à éviter, sauf pour les cas où l'on veut appeler l'application installée à partir de la ligne de commande (comme on va faire dans l'étape suivante avec express generator) et que l'installation n'est pas spécifique au projet. Sans cette option, le paquet sera installé pour le projet seulement, dans le répertoire *node\_modules*. De plus, l'option *--save* permet d'inclure le paquet dans la liste des dépendances. Ainsi, si vous copiez un nouveau projet, vous pouvez simplement lancer la commande *npm install* sans arguments afin d'installer tous les paquets dans la liste des dépendances.

Pour débiter votre projet avec un squelette Express, commencez par installer express generator avec la commande *npm install express-generator -g*.

Ensuite, utilisez la commande *express myAppName* pour générer votre squelette, puis changer de répertoire vers celui que vous venez de créer. La console vous donnera les commandes qu'il faudra utiliser plus tard pour démarrer votre serveur.

Par contre, il faudra effectuer une petite modification avant d'installer les dépendances : comme Pug/Jade a changé de nom, nous devons désinstaller Jade et installer Pug. Donc, commencez par enlever la dépendance avec *npm uninstall jade --save*, puis ajoutez la dépendance à Pug avec *npm install pug --save*. Ensuite, ouvrez le fichier *app.js* et modifiez le view engine de jade à pug.

Vous pouvez maintenant installer le reste des dépendances avec *npm install*.

Dernière étape avant de tenter de lancer l'application : allez dans le répertoire *views* et renommez tous les fichiers *.jade* à *.pug*.

Finalement, vous pourrez utiliser la commande que le système vous a fourni à l'installation du squelette (`DEBUG=myapp:* npm start` sur mac/Linux et `set DEBUG=myapp:* & npm start` sur Windows, ou `myapp` correspond au nom de votre application) pour lancer votre serveur. Par défaut, le serveur utilise le port 3000 du localhost.

## 4.2 Utilisation des templates avec Pug

Au premier TP, vous avez remarqué qu'une portion importante de votre code HTML était répétée sur chaque page et que ce n'était que la section principale qui changeait. Avec Pug, vous pourrez éliminer ce problème en utilisant des templates.

Dans le répertoire *routes*, vous aurez un fichier `index.js`. Ce fichier permettra de retourner une réponse en particulier en fonction de la requête. Vous ajouterez des routes dans ce fichier pour afficher les pages de votre site.

Les fichiers `.pug` par défaut se retrouvent dans le répertoire *views*. Vous remarquerez que le fichier `index.pug` implémente des parties du template `layout.pug`. Utilisez le même type de structure pour la partie fixe de vos pages.

Tout au long de ce processus, vous transformerez aussi votre code HTML en code Pug. Vous pourrez vous inspirer de certains outils en ligne qui peuvent aider à faire la conversion initiale. Consultez aussi le site de Pug (<https://pugjs.org/api/getting-started.html>) pour la référence du langage.

Pour ce TP, vous n'êtes pas obligé d'utiliser de fonctionnalités plus complexes que la syntaxe de base et les templates. Cependant, vous êtes libres de le faire pour faciliter certaines de vos tâches. Par exemple, le fait que Node puisse passer des variables à Pug, peut être utile si vous voulez être conscient de la page sur laquelle vous vous trouvez.

Pour plus d'exemples concrets avec Pug, visitez <http://cssdeck.com/labs/learning-the-jade-templating-engine-syntax> et <http://cssdeck.com/labs/jade-templating-tutorial-codecast-part-2>.

## 4.3 Ajout de styles avec CSS

Dans le TP1, vous avez uniquement élaboré la structure de vos pages. Vous n'avez pas encore ajouté une mise en forme pour vos différents éléments. Au cours du TP2, vous allez effectuer la mise en forme de votre site web à l'aide de CSS3.

Selon certaines directives, vous serez obligés de modifier votre structure initiale afin de répondre aux exigences.

Voici les directives de ce travail pratique :

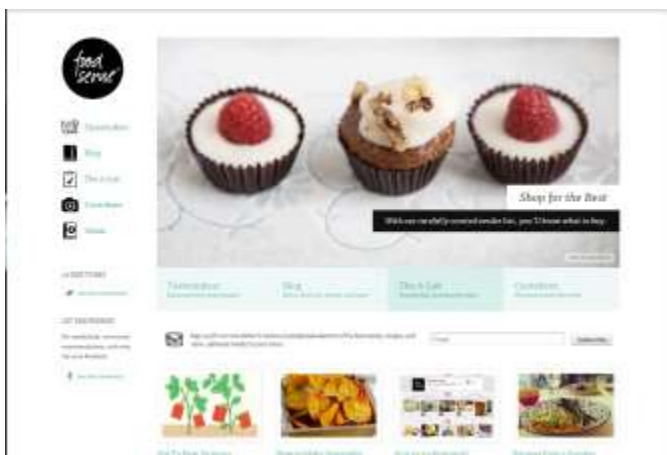
1. Donnez un style à votre menu de navigation. La page actuelle doit clairement être indiquée dans le menu de navigation

Par exemple,



2. Dans la page du tableau de bord et des pages de question, vous devez créer une boîte contenant les statistiques de la personne. Cette boîte devra être placée à droite des pages.
3. Donnez un style aux éléments du site web. Par exemple, structurez le formulaire d'initialisation de l'examen, donnez du style aux boutons/liens, séparez les différentes sections de la page, etc. Bref, sortez votre côté artistique.
4. Dans la page d'instruction, insérez une image flottante à côté du texte.
5. Dans la page du tableau de bord, vous devez créer un "bouton" nommé «Détails». Lorsqu'on clique sur ce "bouton", une boîte «modale» apparaîtra par-dessus la page actuelle. Cette boîte contiendra l'ensemble des notes obtenues par la personne sur les examens. Autour de cette boîte, l'arrière-plan est d'une couleur noire avec une opacité inférieur à 1. Si on clique sur cet arrière-plan, alors la boîte se ferme et nous sommes de retour à la page originale. Ceci doit se faire sans Javascript.
6. Finalement, vous avez également la responsabilité de rendre votre site web adaptatif. Voir le lien : <http://fr.openclassrooms.com/informatique/cours/apprenez-a-creeer-votre-site-web-avec-html5-et-css3/mise-en-page-adaptative-avec-les-media-queries>

Exemple d'un site web adaptatif → Simplement agrandir ou réduire la fenêtre sur votre fureteur pour voir le résultat : <http://foodsense.is/>



À la base, votre mise en forme doit s'adapter aux écrans de l'école. Pour une mise en forme adaptative, nous allons ajouter une mise en forme pour les tablettes 768px et 1024px seulement. Pas besoin d'aller plus bas que 768px.

Le CSS et son intégration constitueront la partie principale évaluée. La structure du HTML doit aussi être respectée. Enfin, le code Jade doit permettre d'éviter la répétition de code.

## 5 Évaluation

Globalement, vous serez évalués sur la qualité de votre mise en forme CSS. Plus précisément, le barème d'évaluation est le suivant :

Exigences Points	Points
Utilisation convenable de Pug	20
Menu de navigation	15
Statistiques de la personne	15
Détails des examens	15
Image flottante dans la page d'instruction	5
Site web adaptatif	15
Originalité et qualité globale	15

L'évaluation sera faite à partir de la page d'accueil. À partir de cette page, je devrais être capable de consulter tous les pages de votre site web.

Le fureteur **Firefox** sera utilisé pour tester votre site web.

## 6 Direction de la remise

1. Pour la remise, on demande de placer votre code HTML et CSS dans un dossier compressé nommé : TP2\_matricule1\_matricule2. Vous avez le choix du type de compression.  
Ne pas inclure le répertoire node\_modules. Si vous avez bien géré vos dépendances, on devrait pouvoir le recréer.
2. Vous devrez également créer un fichier nommé temps.txt à l'intérieur du dossier de votre projet. Vous indiquerez le temps passé **au total** pour ce TP.
3. Le travail pratique doit être remis **avant** 23h55 le dimanche 2 octobre 2016 sur Moodle.

## 7 Pénalités

1. La politique pour les travaux pratiques remis en retard : -20% pour chaque jour de retard.
2. Les directives 1,2 et 3 de la remise du TP : -5% si ces directives ne sont pas respectées.

Bonne chance!