

Evaluation - Responsive client technologies

3h - Par groupe de 2

A rendre **impérativement à 12h00** par email à francois@kiliweb.fr

Objectif : reproduire les maquettes fournies dans le dossier « maquettes »

HTML 5

(4 points) - Maquette #1 & 2

- Déclarer le DOCTYPE, le charset UTF-8, le title, 3 feuilles de styles (reset.css, global.css, responsive.css), 1 fichier javascript (app.js)
- Inclure le framework Bootstrap pour la partie CSS
- Inclure au choix la librairie jQuery ou le framework AngularJS (v1) pour la partie Javascript
- Mettre en place la structure HTML correspondante aux maquettes en utilisant les balises HTML5 (main, section, time, etc.)
- Si vous utilisez jQuery, servez-vous du fichier *cities.html*. Utilisez les balises HTML5 lorsque c'est approprié
- Ajouter les icônes « coeur » avec Bootstrap
- Mettre en place le menu déroulant (« Mes favoris »)
- Mettre en places les champs de formulaires HTML5 :
 - Champs de recherche avec placeholder « Rechercher »
 - Budget : slider de 100 à 400€, par tranche de 20€
 - Nombre de nuits minimum (entier entre 1 et 30)
 - Date : champs de sélection de date (décembre 2016 obligatoire)

Responsive web design

(4 points) - Maquette #3

- En utilisant le framework Bootstrap, mettre en place une mise en page fluide, sur toute la largeur de l'écran disponible
- Rendre les images des villes promotionnelles à l'espace disponible : leur taille doit varier en fonction de la taille de la fenêtre
- Afficher alternativement 6 / 4 / 3 / 2 puis 1 ville par ligne, en fonction de l'espace disponible
- Faire une mise en page pour les mobiles correspondant aux maquettes
 - Centrer le logo et les favoris
 - Masquer la barre de recherche
 - S'assurer que la présentation est propre à toutes les résolutions

Vous pourrez coder les règles s'appliquant au mobile dans une feuille de style dédiée (responsive.css). Les « *breakpoints* » recommandés sont 768px et 480px.

CSS3

(4 points) - Maquette #1 & 2

- Coder le CSS correspondant aux maquettes
 - Pour les photos des villes, voir *cities.css*
- Utiliser la police *Luna* pour le titre principal (l'importer en utilisant la spécification CSS3), ainsi que la Google Font *Roboto* pour tous les autres textes
- Faire un dégradé du blanc vers le gris (RGB 240, 240, 240) sur la partie basse des cartes de villes
- Ajouter une légère ombre externe sur les cartes
- Le nom de la capitale comporte une légère ombre externe pour plus de lisibilité
- Faire le thème du champs de recherche
- Au survol du coeur, l'icône devient plus grosse

jQuery ou AngularJS

(8 points) - Maquette #5 et vidéo : <http://cloud.lemonlab.fr/lcOW+>

Vous pouvez utiliser au choix jQuery ou AngularJS pour mettre en place les fonctionnalités suivantes.

- Faire fonctionner le champs de recherche (recherche par nom de ville)
 - La recherche ne doit pas tenir compte de la casse
- Afficher les filtres au clic sur le lien « Filtrer », avec une animation déplié / replié
- Permettre à l'utilisateur de sélectionner ses voyages favoris
 - Au clic sur le coeur, l'icône devient rouge plein
 - Les transitions sur le coeur sont animées
- Les voyages favoris s'affichent dans un menu déroulant, visibles au clic sur le bouton « Mes favoris »
- Le bouton « Mes favoris » doit indiquer le nombre de voyages mis en favoris
- Il est possible de retirer un voyage des favoris en cliquant à nouveau sur le coeur

Bonus (optionnel)

(2 points)

- Faire fonctionner le filtre budget

Annexe

Couleurs :

- Bleu clair : #4990E2
- Bleu foncé : #4481C9
- Rouge : #E74C3C

Remarques :

- Si vous utilisez AngularJS, la liste des destinations est disponible dans le fichier *cities.js*
- Vous aurez probablement besoin de la fonction jQuery \$.inArray, ou de la méthode indexOf pour rechercher dans un tableau.
- Pour supprimer un élément d'un tableau, vous pouvez utiliser la méthode :
array.splice(index, 1);

Livrable :

- Le livrable prend la forme d'un dossier ZIP contenant l'ensemble des fichiers source (.html, .css, .js).
- Indiquez le numéro de votre groupe dans le nom du fichier ZIP