



L 'évolution du Web : la mobilité

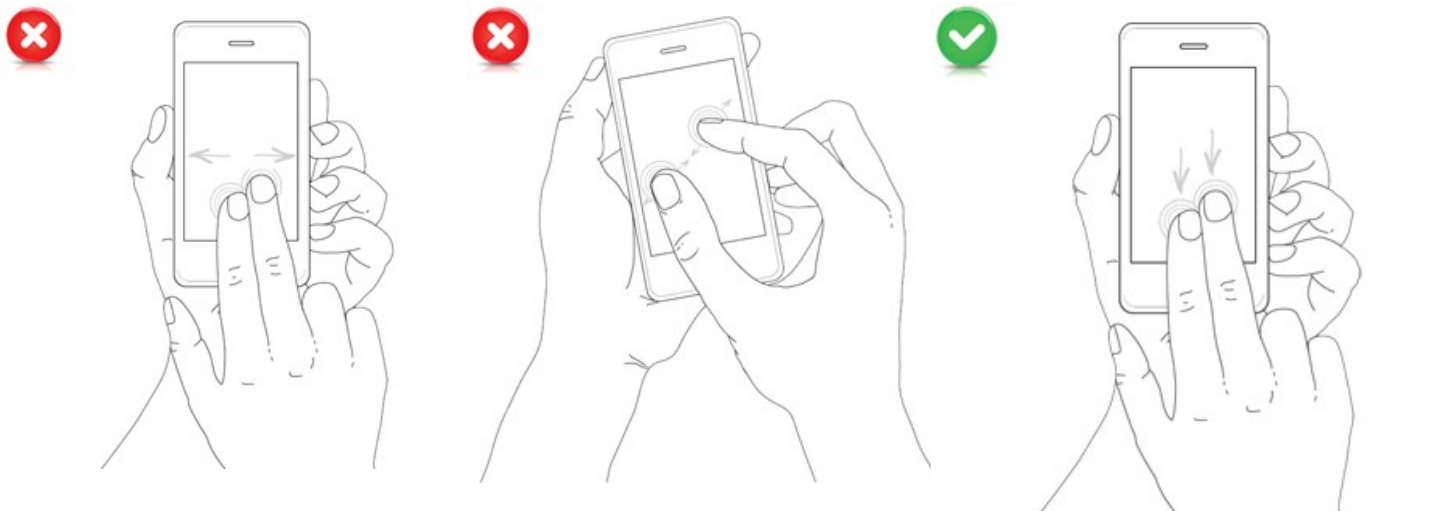
<http://applicook.fr/iut/m4105c/tp1.pdf>

Aujourd'hui on compte autant d'utilisateurs nomades (sur smartphone et tablettes) que traditionnels (PC, portable, Mac).

Problème : Le premier site internet est apparu en 1993, à l'époque les smartphones et tablettes n'existaient pas, ainsi comment afficher un site sur un appareil qui tient dans sa poche alors qu'à la base c'est prévu pour fonctionner sur un écran d'ordinateur ?

Solutions : Internet a beaucoup évolué, la façon de le concevoir et le créer aussi, ces dernières années sont apparus des évolutions majeures : HTML5 et CSS3

Les terminaux nomades sont équipés de navigateurs donc ils sont capable techniquement d'afficher les différents sites. Par contre, les sites qui ne sont pas optimisés ont des gros soucis de navigations, de lecture, de lenteur... ils sont même maintenant pénalisés par Google pour leur référencement. Nous allons au sein de ce module essayer de résoudre cette problématique : comment concevoir et créer un site pour tous les terminaux connectés (montre, tablette, ordinateur, smartphone, tablette, tv ...)



3 Types de site :

- Pas du tout adapté : <http://www.ch-calais.fr>
- Site mobile différent : <https://fr.m.wikipedia.org>
- Même site adapté : <https://eurosport.fr>

TP #1 : HTML5

1. Structure générale

La création d'une page web au format HTML doit respecté des standards spécifiés par le W3C. Nous utiliserons tout au long de ce module HTML5 qui est la dernière révision majeure de HTML dont la dernière version a été finalisé en octobre 2014. Ce langage dispose de nombreux outils et API qui ont permis de plus facilement faire cohabiter le web et la mobilité.

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>M4105C : Un super module</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>

<!-- Le reste du contenu -->

</body>
</html>
```

La structure de base est minimaliste mais avec HTML est apparu un certain nombre de nouvelles balises qui permettent de mieux structurer le contenu, par exemple : <article> <section> <header> ...

- Créer la structure de votre projet et de votre fichier index.html en vous basant sur l'exemple d'avant.
- Organiser votre fichier index de telle sorte qu'on puisse y trouver une en-tête, plusieurs sections et un pied de page.

2. STOP : versioning

Un projet informatique du fait de l'évolution des langages, des utilisateurs, des terminaux... n'est jamais vraiment figé, il est souvent amené à être modifié.

La gestion des versions a plusieurs avantages, elle permet de garder une trace des fichiers modifiés, de mettre facilement en ligne les fichiers qui ont été modifiés, de pouvoir travailler à plusieurs sur un projet, d'avoir des commentaires sur les évolutions, de pouvoir facilement revenir à une version précédente, d'éviter de dire « Monsieur, ma clé USB a pris l'eau j'ai perdu mes fichiers » ...

Il existe plusieurs outils et services qui permettent de faire du contrôle de version, nous utiliserons ici git et <https://github.com/>

L'avantage d'utiliser Github est qu'on disposera d'une URL pour pouvoir tester tout au long du TP nos développements sur nos tablettes / smartphones (bon d'accord il existe d'autres moyens de tester, mais ça sera plus simple ici).

- Créez un compte sur <http://github.com/>
- Suivez **l'étape 1** de cette page <https://guides.github.com/activities/hello-world/>
- Suivez le tutoriel suivant : <https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>
- Pour ceux qui utilisent le réseaux de l'IUT :

```
git config --global http.proxy  
http://nom.prenom:mdpsession@10.10.28.5:3128
```

- Enfin sur votre projet Github, rendez-vous dans la partie « settings », rubrique « Github Pages », dans « Source » sélectionnez « master branch » puis sauvegardez.

À la fin de ces 3 étapes, si tout c'est bien déroulé , vous devriez trouver dans votre dépôt un fichier readme.md et votre fichier index.html

Dans votre navigateur préféré (ne me dites pas que c'est Internet Explorer), entrez l'URL <http://username.github.io/reponame/>

Vous devriez voir apparaître votre magnifique projet.

Modifiez maintenant le fichier index.html en ajoutant du texte par exemple, voici la démarche à suivre pour mettre les modifications en ligne :

```
git commit -a -m "message de votre commit"
git push origin master
```

Retournez sur l'URL de votre projet pour voir si les modifications ont bien été prises en compte :)

N'oubliez pas de « commiter » et « pusher » vos fichiers lorsque vous avez apporté des modifications importantes et surtout pour pouvoir tester sur votre mobile/tablette car il n'y a pas de système simple pour tester en local sur un terminal nomade, on utilisera ainsi cette technique tout au long des Tps.

Si vous êtes amené à ajouter des fichiers à votre projet, il faut au préalable les ajouter à votre répertoire avant de les commiter grâce à la commande suivante :

```
git add .
```

Astuce : Sur Linux, pour sauvegarder votre utilisateur et mot de passe pour Git, exécutez les commandes suivantes :

```
git config --global credential.helper cache
```

```
git config --global credential.helper 'cache --timeout=9800'
```

3. Viewport

- Créez 3 paragraphes de textes.

Astuce : Lors du développement d'un site, comme il est très rare d'avoir le contenu du client, on utilise souvent du faux texte généré aléatoirement qu'on appelle « Lorem Ipsum », voici un exemple de générateur : <http://fr.lipsum.com/>

- Appliquez un fond bleu pour le premier paragraphe, blanc pour le deuxième et rouge pour le dernier.

Astuce : Les couleurs en HTML sont en hexadécimal, Il est toujours compliqué de choisir une couleur vu la multitude de combinaison possible, j'ai l'habitude d'utiliser ce site <http://flatuicolors.com/> qui dispose de couleurs « flat » avec un contraste intéressant

- Dans le dernier paragraphe, insérez l'image présente ici : <http://applicook.fr/iut/images/calais.jpg> (créez un répertoire images/)
- Observez le résultat sur votre smartphone

Le Viewport désigne schématiquement la surface de la fenêtre du navigateur. Je vous invite à lire cet excellent article (profitez c'est en français)

<http://www.alsacreations.com/article/lire/1490-comprendre-le-viewport-dans-le-web-mobile.html> sur le sujet afin de répondre à la question suivante :

- Utilisez la balise meta name="viewport" afin que l'utilisateur puisse zoomer, que le zoom initial soit à 1 et que la largeur de la page corresponde à celle du terminal.
- Observez le résultat

4. Favicon

Un favicon sur un « navigateur traditionnel » est la petite icône que l'on aperçoit au niveau des onglets à côté du titre de la page que vous visitez. Sur mobile, faute de place, celui-ci n'est pas visible, par contre il est visible au niveau des favoris et lorsqu'on ajoute un site en page d'accueil.

- Ajoutez votre site sur l'écran d'accueil et en favoris

Il est possible de modifier l'image générée, grâce notamment à la balise <link> et l'attribut rel ='apple-touch-icon' par exemple.

- Renseignez-vous sur votre smartphone/tablette et changez cette image en effectuant des recherches sur internet

Astuce : lors de vos recherches sur internet, privilégiez les sites ou les développeurs ont l'habitude d'échanger, le plus connu est <http://stackoverflow.com>, en France on trouve de bons conseils sur <http://www.alsacreations.com/> ou <https://openclassrooms.com>

5. Formulaires

Les formulaires existent depuis la nuit des temps en html, mais une des différences principales entre un terminal nomade et un ordinateur est l'absence d'un clavier physique sur le premier cité. Ainsi avec HTML5 de nouveaux types de champs sont apparus afin lorsqu'on sélectionne un champs, d'avoir les bons éléments du clavier qui s'affichent. Par exemple en entrant la ligne ci-après, lorsqu'on sélectionne le champ age, le clavier ne fera apparaître que les chiffres.

```
<input type="number" name="age" placeholder="Age"/>
```


- Créez un formulaire avec les champs suivants : nom, date de naissance, age, page Facebook, email, téléphone en prenant le soin de mettre les bons types au niveau des champs, attention certains types ne sont pas compatibles sur tous les systèmes d'exploitation.

6. Optimisations et performances

Le débit des connexions nomades même si il a beaucoup évolué ces dernières années reste aléatoire et souvent de moins bonne qualité qu'une ligne classique. Il faut donc penser à optimiser son site afin d'éviter de perdre des utilisateurs qui se lasseraient d'attendre trop longtemps le chargement d'une page. Au niveau du HTML, voici quelques conseils à suivre :

- Retailer les images pour qu'elles aient la même taille d'affichage, en effet il est possible de modifier la largeur et la hauteur directement en html, mais l'image chargée par le navigateur reste celle d'origine.
- Spécifier les tailles des images dans la balise pour éviter que le navigateur fasse le calcul.
- Limiter le nombre de requêtes, en définissant par exemple tous les styles dans un seul fichier.
- Penser à minifier (c'est à dire enlever les caractères inutiles) les fichiers CSS et JS une fois votre site développé. De nombreux sites permettent de le faire, je vous laisse chercher vous-même :)
- Lancer certains scripts en fin de page ou avec les attributs defer ou async, pour plus d'informations, lire l'article suivant :
<http://www.alsacreations.com/astuce/lire/1562-script-attribut-async-defer.html>
- Mettez l'image <http://applicook.fr/iut/images/calais.jpg> dans une section
- Utilisez un logiciel de traitement d'image (gimp, painte, photoshop) et retailer l'image pour qu'elle fasse 500px de large et remplacez la précédente.

7. Cache

HTML5 a apporté une fonctionnalité très intéressante pour les mobiles et tablettes : le cache manifest. Les fichiers sont stockés dans **application cache** - une collection de ressources obtenues depuis un fichier **manifest** (* .appcache) incluse dans une application web.

L'utilisation d'un cache applicatif permet les bénéfices suivants :

1. Navigation hors-ligne : les utilisateurs peuvent utiliser un site même s'ils ne sont pas connectés.
 2. Vitesse : les ressources mises en cache sont locales, et se chargent donc plus rapidement.
 3. Réduction de la charge serveur : le navigateur ne télécharge uniquement les ressources qui ont changées sur le serveur.
- Mettez en cache tous les fichiers de votre projet afin de pouvoir consulter votre site en désactivant la connexion à internet.

8. Les standards du web

Le [W3C ou World Wide Web Consortium](#), est un organisme international qui développe des [standards](#) pour le Web afin que les gens puissent communiquer efficacement à travers Internet, autour de formats ouverts garantissant une meilleure inter-opérabilité (c'est-à-dire une meilleure compréhension des systèmes hétérogènes à travers des données et langages standardisés).

Il est très important de respecter ces standards car ça permet entre autre d'apporter une première solution aux problématiques de référencement et d'accessibilité.

En vrac, voici quelques normes à respecter absolument :

- Veiller à bien fermer les balises qui le nécessite
- Les balises doivent toujours contenir un attribut alt qui décrit l'image
- Les liens <a> doivent toujours contenir un attribut title qui informe l'utilisateur sur la cible du lien
- Placer les styles et les scripts dans des fichiers externes aux fichiers html.

Il existe un outil en ligne qui permet de vérifier si son site respecte les normes :
<https://validator.w3.org/>

- Utiliser l'outil précédent afin de rendre votre projet conforme aux standards.