

# HTML5



**Travail de Bachelor réalisé en vue de l'obtention du Bachelor HES**

par :

**Nemanja SUBOTIĆ**

Conseiller au travail de Bachelor :

**Peter DAEHNE, professeur HES**

**Genève, 12 septembre 2011**

**Haute École de Gestion de Genève (HEG-GE)**

**Filière Informatique de Gestion**

## Déclaration

Ce travail de Bachelor est réalisé dans le cadre de l'examen final de la Haute école de gestion de Genève, en vue de l'obtention du titre de Bachelor en Informatique de Gestion. L'étudiant accepte, le cas échéant, la clause de confidentialité. L'utilisation des conclusions et recommandations formulées dans le travail de Bachelor, sans préjuger de leur valeur, n'engage ni la responsabilité de l'auteur, ni celle du conseiller au travail de Bachelor, du juré et de la HEG.

« J'atteste avoir réalisé seul le présent travail, sans avoir utilisé des sources autres que celles citées dans la bibliographie. »

Fait à Genève, le 12.09.2011

Nemanja Subotić

## Remerciements

Tout d'abord, je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide, qui m'ont soutenu tout en long de la réalisation et qui ont contribué à l'élaboration de ce mémoire.

Un sincère remerciement à mon professeur de la HEG, Monsieur DAEHNE, qui s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire. Je le remercie aussi pour l'inspiration, l'aide et le temps qu'il m'a consacré.

En outre, je tiens à adresser mes remerciements à ma compagne RICHARD Katia pour sa générosité, son soutien ainsi que les précieux conseils qu'elle a su m'offrir pendant les 8 semaines de la réalisation de ce travail.

Enfin, je tiens à exprimer un grand merci à PULFER Cynthia, PASQUIER Karine et ROHR Salomé qui ont eu la gentillesse de lire et corriger ce travail.

## Résumé

Le but principal de ce travail de Bachelor est de découvrir HTML5 sous plusieurs formes. En commençant par l'histoire et l'évolution du Web, puis en passant par les points forts ainsi que les cas pratiques, nous obtenons une brève synthèse d'un thème de grande dimension.

Dans ce document, plusieurs aspects primordiaux sont abordés. Il s'agit d'une description des sujets qui ont été choisis pour mieux comprendre HTML5. En effet, il existe un essor important de cette technologie à travers la nouvelle version de balisage. Donc, en premier lieu ce travail décrit un rappel des bases d'HTML, puis de ses caractéristiques. Ensuite nous exposons les problèmes actuellement existants. Puis, pour finir, nous abordons une vue des technologies concurrentes telles que Flex<sup>TM</sup> et Flash.

De plus, de nombreuses nouveautés du monde du Web sont, ici, abordées, sous une forme théorique, puis d'un point de vue plus pratique. Ce qui permet, tout au long de ce document, de présenter des cas pratiques, puis de démontrer visuellement une application du code.

# Table des matières

Déclaration.....	2
Remerciements .....	3
Résumé .....	4
Table des matières.....	5
Liste des Tableaux .....	7
Liste des Figures.....	7
Histoire.....	9
1.1 Histoire du WEB.....	9
1.2 Histoire de HTML .....	9
1.2.1 HTML1.0.....	10
1.2.2 HTML2.0.....	10
1.2.3 HTML3.0 & HTML3.2.....	10
1.2.4 HTML4.0.....	11
1.3 HTML5 de demain .....	11
2. HTML5 et ses prédécesseurs .....	12
2.1 Balises .....	12
2.2 Formulaires et input .....	15
2.2.1 Nouvelles balises de form .....	15
2.2.2 Nouveaux attributs de form .....	15
2.2.3 Nouveaux attributs d'input.....	16
3. Détection des caractéristiques HTML5 .....	19
3.1 HTML5 Doctype.....	19
3.2 Canvas .....	20
3.2.1 Dessiner sur le canvas .....	21
3.2.1.1 Dessiner avec Paths .....	22
3.3 Multimédia .....	24
3.3.1 Audio .....	24
3.3.2 Vidéo .....	26
3.4 Géolocalisation .....	28
3.4.1 Les fonctions .....	29
3.5 Le stockage local.....	30
3.5.1 Histoire .....	30
3.5.2 Fonctionnement de Web Storage.....	30
3.6 Offline Web applications .....	33
3.6.1 Le manifeste de cache .....	34
3.6.2 Fonctionnement d'Offline Web .....	34
3.6.3 Conserver du cache .....	35
4. Appareils mobiles.....	36
5. Problèmes actuels du HTML5 .....	38

5.1	Compatibilité avec les navigateurs .....	38
5.1.1	Les points sensibles des navigateurs.....	39
5.2	Mouture de HTML5.....	40
6.	HTML5 contre Flex & Flash .....	41
6.1	Adobe Flex™ .....	41
6.1.1	HTML5 vs Flex™ .....	43
6.2	Adobe Flash .....	43
6.2.1	HTML5 vs Flash .....	44
6.3	HTML5 est-il le vainqueur? .....	44
	Conclusion.....	46
	Webographie .....	47
	Bibliographie .....	50
	Annexe 1 Installation plateforme Web .....	51
	Annexe 2 Géolocalisation (3.4) .....	52
	Annexe 3 Les éléments non supportés par les navigateurs.....	54

## Liste des Tableaux

Tableau 1	Les nouvelles balises de HTML5 .....	12
Tableau 2	Les attributs de l'élément audio .....	25
Tableau 3	Les attributs de l'élément vidéo.....	27
Tableau 4	Navigateurs et HTML5 .....	38
Tableau 5	Navigateurs et certains éléments HTML5 .....	39

## Liste des Figures

Figure 1	Chronologie de HTML .....	9
Figure 2	Forme d'un site Web.....	14
Figure 3	Quirks mode du Web.....	19
Figure 4	Dessin de rectangle avec canvas.....	21
Figure 4.1	Code de rectangle avec canvas.....	21
Figure 5	Dessin HEG.....	22
Figure 5.1	Explication du dessin HEG .....	23
Figure 6	Les codecs vidéo.....	26
Figure 7	Géolocalisation de HTML5.....	28
Figure 8	Tableau de Web Storage.....	31
Figure 9	Outils de développement des navigateurs sans les données.....	31
Figure 10	Outils de développement des navigateurs avec les données.....	32
Figure 11	Affichage du cache hors connexion dans Firefox .....	33
Figure 12	Inputs des appareils mobiles.....	36
Figure 13	Application vs plateforme Web.....	37
Figure 14	Processus de compilation Flex .....	42
Figure 15	Adobe Edge prototype .....	45

# Introduction

Depuis plus de 20 ans, les utilisateurs d'Internet cherchent des informations via une application Web. Cette recherche débute en général via un site Web qui propose un moteur de recherche. Le but étant d'aboutir sur une page Web qui contient les informations recherchées.

Durant les deux dernières décennies, le comportement, les coutumes et la façon de vivre des entreprises ainsi que le rythme de vie des individus, se sont complètement métamorphosés. Cette évolution a suivi le progrès d'Internet et par conséquent celui du World Wide Web. Ce changement attire un grand nombre d'utilisateurs et par conséquent les rend attentifs à Internet.

La toile mondiale a aussi permis une expansion des autres outils et applications, tels que, des animations ainsi que des outils de travail. Ces derniers offrent aujourd'hui une énorme diversité qui est disponible sur les pages Web.

Le World Wide Web est un ensemble de ressources hypertextes réalisé grâce aux documents HTML (Hypertext Markup Language). En résumé, un document HTML contient principalement du texte consulté, du texte en langage HTML et aussi d'autres langages. De ce fait, la base de la toile mondiale repose sur le langage HTML et les ressources supplémentaires, comme JavaScript etc. Ces dernières améliorent le dynamisme d'une page web.

C'est pourquoi, se pose la question : que va-t-il arriver à la toile mondiale et aux ressources supplémentaires avec une nouvelle version de HTML ? Quelles seront les nouvelles fonctionnalités de HTML5 ? Est-ce qu'il s'agit vraiment d'une révolution ou tout simplement d'une amélioration mineure du Web ?

L'objectif premier de ce travail de Bachelor est non seulement d'étudier les fonctionnalités de HTML5, mais également de les comparer avec ses anciennes versions ainsi que d'autres langages.

Pour comprendre l'évolution de HTML, il est important de connaître l'histoire du Web et l'évolution de ce langage informatique à travers le temps. C'est pourquoi nous commencerons par l'étude de l'histoire du Web.



*« Les concepts formulés ici en théorie, sont développés en pratique sur un site Web. Le site peut être trouvé sur le CD, qui est joint à la version papier. La plateforme peut être utilisée en localhost et en annexes vous trouverez les instructions pour installer le WAMP ou MAMP pour une meilleure utilisation. »*



# Histoire

## 1.1 Histoire du WEB

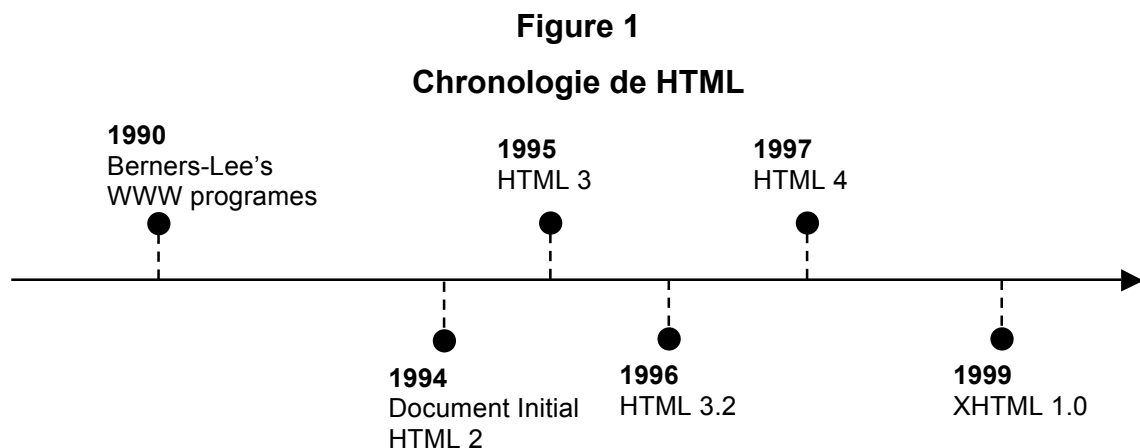
Physicien britannique, informaticien et professeur au MIT (Massachusetts Institute of Technology) Timothy John Berners-Lee est considéré comme l'inventeur du World Wide Web. En mars 1989, âgé de 34 ans, Monsieur Berners-Lee travaillait au CERN (Suisse). Durant son travail, il effectue une première proposition d'un système hypertexte pour accéder aux nombreuses formes de documentation et de rapports produits par le CERN. En fait, il s'agissait d'une amélioration d'un simple programme hypertexte, « Enquire », que le CERN utilisait depuis 1980. Le but principal de « Enquire » était de suivre les divers projets du CERN et de les partager avec d'autres machines connectées sur le réseau existant du CERN.

Suite à un échec en 1989, Monsieur Berners-Lee s'associe, une année plus tard, avec Robert Cailliau et, ensemble, ils réalisent une révision de la première proposition du projet. Ainsi, à la fin de l'année 1990, ils obtiennent le World Wide Web grâce au premier HTTP, navigateur WEB et à HTML. Ce projet a été conçu au sein du CERN. Le premier site web fut justement créé au CERN et a été mis en ligne en août 1991.

En 1994, Berners-Lee fonda le World Wide Web Consortium et fit en sorte que le Web soit disponible librement et sans brevet.

## 1.2 Histoire de HTML

HTML est un langage de balisage qui a évolué tout au long de son existence. La *figure 1* illustre les changements que ce langage a subis. En effet, de nouvelles normes et spécifications sont apparues presque chaque année. Le but étant d'offrir une plus grande palette de méthodes, permettant de réaliser des sites plus efficaces.



### **1.2.1 HTML1.0**

HTML1.0 étant la première version de HTML, offrait des possibilités réduites sur l'Internet. Elle a été réalisée pour le CERN et elle était très peu connue à l'époque. Cet outil disposait de 20 éléments et le but principal était d'afficher du texte sur le Web.

### **1.2.2 HTML2.0**

HTML2 était une petite amélioration par rapport à son prédécesseur. Il s'agissait de la première version avec des normes officiellement déclarées. Ces normes représentaient la base selon laquelle tous les navigateurs ont été mesurés. De plus, elles étaient appliquées jusqu'à la sortie de HTML3.2. En effet, le but de HTML2 était d'être supporté par tous les navigateurs de l'époque. De plus, des fonctions supplémentaires ont été apportées, telles que le titre, la liste, les images, les accents, etc.

### **1.2.3 HTML3.0 & HTML3.2**

HTML3 est compatible avec ses anciennes versions. De plus, il a été créé pour supporter tout type de données, afin de lier un ensemble d'informations. Cette version propose des nouvelles fonctionnalités comme les tableaux, car à l'époque, la création des tableaux n'était pas encore réalisable, même s'ils étaient très demandés par les mathématiciens ou les économistes. Cela a permis une meilleure représentation des chiffres. Des éléments additionnels ont été également inclus pour afficher des bannières statiques pour des logos d'entreprises, ou encore des éléments `link` pour créer des menus de navigation. De plus, les formulaires ont été étendus et de nouveaux types de champs ont été insérés pour une meilleure manipulation.

HTML3 était une énorme amélioration par rapport à la version 2. Malheureusement, le projet de HTML3 n'a jamais connu la gloire. Principalement à cause d'une énorme différence avec ses prédécesseurs et le fait que HTML3 n'a pas été supporté sur les navigateurs de l'époque. De plus, Internet Explorer et Netscape ont créés leurs propres éléments qui fonctionnaient seulement sur leurs propres browsers d'où la non intégration de la version HTML3.

Par contre, plusieurs des concepts les plus stables, comme les flux de tableaux de texte, les images, l'alignement page ainsi que d'autres éléments caractéristiques, ont été inclus dans HTML3.2 ou encore HTML4.

Il a fallu attendre jusqu'en 1996, à la sortie de HTML3.2 pour voir une évolution de HTML. Cette version a été présentée et développée selon les normes du World Wide Web Consortium. Ce changement a permis une homogénéité des balises qui n'était pas présente avant. Etant donné que certaines balises ont été des extensions, réalisées par Netscape et Microsoft, elles fonctionnaient seulement sur leurs propres navigateurs. Ces extensions permettaient le fonctionnement de Style Sheets, Frames, Scripting, Math et Internationalization. Avec la sortie de HTML3.2, les navigateurs n'avaient plus besoins de ces extensions.

#### **1.2.4 HTML4.0**

HTML4 était une énorme évolution en comparaison de toutes les versions précédentes de HTML, car les extensions existantes dans les navigateurs n'ont jamais été incluses dans HTML4. Cela a permis une cohérence des pages Web avec les navigateurs. En outre, le fait que la version 4 soit compatible avec tous les navigateurs existants a suscité un grand intérêt de la part des développeurs. Ce qui a amené à une augmentation des sites Internet et permis au grand public l'utilisation de cet outil. De plus, le Cascading Style Sheet a été ajouté afin d'améliorer le design des pages Web, ce qui a donné une nouvelle possibilité de réalisation ainsi qu'un nouveau style d'affichage.

### **1.3 HTML5 de demain**

La cinquième révision majeure du langage de base du World Wide Web, est encore en projet. Même si la réalisation de certaines fonctionnalités de la dernière version est possible, le contenu de HTML5 et ses documents sont encore en discussion.

Cependant avec l'introduction d'un certain nombre d'API, qui permettent de créer des applications Web en ligne et hors ligne ou encore d'afficher la vidéo et écouter l'audio, il est évident que la différence entre HTML4 et HTML5 est énorme.

Le but principal de HTML5 est de simplifier la création des sites Web dynamique en ajoutant les nouveautés qui n'ont pas pu être réalisées sans les langages tiers. Ainsi l'effort d'un développeur pourrait être concentré sur le langage de balisage et ses nouveaux éléments.

## 2. HTML5 et ses prédécesseurs

Etant donné que HTML5 est une évolution des versions précédentes de HTML, il doit forcément reprendre la structure de ses prédécesseurs. Prévu pour 2012, avec une recommandation finale pour 2022, HTML5 a sagement gardé la compatibilité avec toutes les réalisations Web effectuées jusqu'à ce jour.

L'objectif premier de HTML5 est de simplifier le code et d'éviter toutes sortes de complications. Ainsi, un langage révolutionnaire, en terme de publications sur le Web, est obtenu, en fusionnant HTML4, DOM2HTML<sup>1</sup> et XHTML1. De plus, ce langage offre de nouvelles fonctionnalités et possibilités.

Par de nouvelles balises, telles que `audio`, `video`, `progress` ou encore `canvas`, HTML5 devient peu à peu un outil de haute compétence. Par la montée en puissance des nouvelles feuilles de style (CSS3) et du nouvel élément `canvas`, il est possible d'estimer et comprendre les capacités, que la nouvelle série HTML peut offrir aux internautes.

### 2.1 Balises

Même si ce langage de balises est toujours en développement, il est déjà connu que certaines balises, ont été supprimées et d'autres ajoutées. Par contre, les balises ont hérité des mêmes caractéristiques que précédemment. Ainsi, elles ne sont toujours pas sensibles à la casse, elles doivent être correctement imbriquées et toute balise ouverte doit être fermée.

**Tableau 1**  
**Les nouvelles balises de HTML5**

<b>section</b>	définit les sections dans un document, telles que les chapitres, en-têtes, pieds de page, ou toutes autres sections du document.
<b>article</b>	partie indépendante du site, comme un commentaire.
<b>aside</b>	associe la balise qui le précède.
<b>header</b>	introduit un groupe d'éléments de navigation pour le document.

---

<sup>1</sup> Document Object Model (DOM) Level 2 HTML Specification

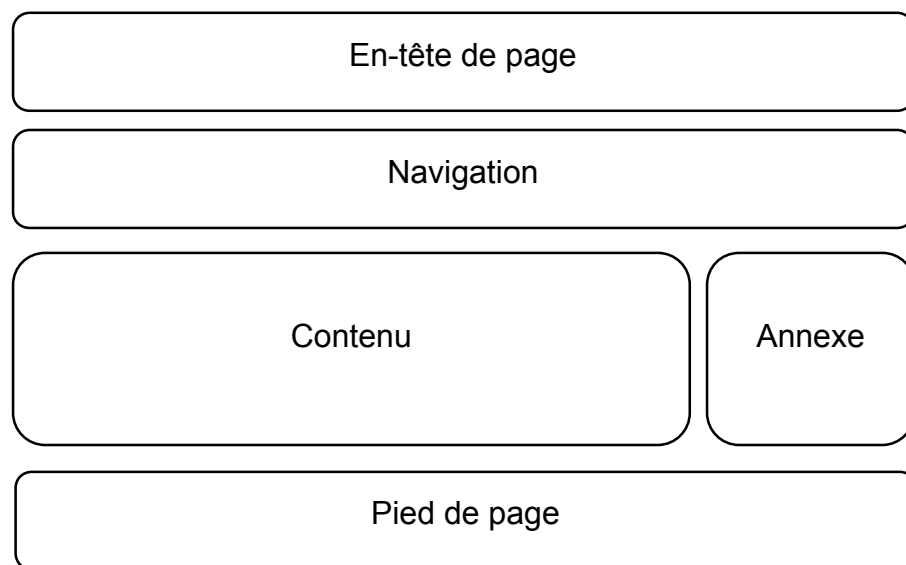
<b>footer</b>	définit le pied de page d'un article ou un document. Contient généralement le nom de l'auteur, la date du document, le jour où il a été écrit et / ou les coordonnées de l'auteur.
<b>nav</b>	définit une section dans la navigation.
<b>figure</b>	définit des images, des diagrammes, des photos, du code etc...
<b>figcaption</b>	légende pour la balise <figure>.
<b>audio</b>	pour définir un son, comme la musique ou les autres flux audio (streaming).
<b>video</b>	insérer un contenu vidéo en streaming.
<b>track</b>	insérer un sous-titre (au format WebVTT) à une vidéo affichée avec la balise vidéo.
<b>embed</b>	définit un contenu incorporé, comme un plug-in.
<b>mark</b>	définit un texte marqué.
<b>meter</b>	permet d'utiliser les mesures avec un minimum et maximum connus, pour afficher une jauge.
<b>progress</b>	définit une barre de progression sur le travail en cours d'exécution.
<b>canvas</b>	permet d'afficher des éléments graphiques en utilisant un script pour l'animer.
<b>time</b>	définit une date ou une heure, ou les deux.
<b>command</b>	définit un bouton.
<b>details</b>	précise les détails supplémentaires qui peuvent être masqués ou affichés sur demande.
<b>keygen</b>	permet de générer une clé (sécurisé).
<b>output</b>	représente le résultat d'un calcul (comme celui effectué par un script).
<b>ruby</b>	spécifie une annotation ruby (pour la typographie asiatique).

Source : S.n. HTML5. In *Wikipedia : l'encyclopédie libre* [en ligne]. Dernière modification le 06 septembre 2011 à 15h56. <http://fr.wikipedia.org/wiki/HTML5> (consulté le 18.07.2011)

Certaines de ces balises seront très utiles pour la conception des pages Web. Par exemple, les balises nommées « balises d'organisation » vont permettre aux développeurs de simplifier leur code.

En visualisant une page Web, il est possible d'apercevoir que, la plupart du temps, elle est constituée d'un en-tête, d'un menu permettant la navigation, d'un contenu, d'une annexe et d'un pied de page, comme la figure 2 ci-dessous l'illustre :

**Figure 2**  
**Forme d'un site Web**



Source : VAN LANCKER, Luc. HTML5 et CSS3 : Maîtrisez les standards des applications Web. 2011. p. 143



Voici le premier exemple, qui permet d'illustrer d'un point de vue pratique ce qui est expliqué ci-dessus en théorie. Pour cela, veuillez-vous référer au site suivant :

<http://localhost/HTML/balise.html>

Ces divisions, c'est-à-dire les `div` sont remplacées par les balises<sup>2</sup> de HTML5. Le code de la page est plus clair et plus rapidement consulté. Cela apporte aux développeurs une meilleure manipulation des balises. Ainsi il est plus aisé d'appliquer les règles du CSS.

---

<sup>2</sup> Ces balises peuvent être retrouvées dans le tableau en couleur rouge

## 2.2 Formulaires et input

Le formulaire étant l'élément le plus utilisé, il occupe une place très importante dans la création d'une plateforme Web. En fait, la balise `form` est l'unique élément de HTML permettant d'envoyer des informations depuis le client Web vers le serveur. Elle permet de récupérer les données saisies par un utilisateur final.

Les formulaires n'ont pas subi de modifications depuis quinze ans. Par contre, leur aspect graphique ainsi que leur utilisation ont progressé avec les Cascading Style Sheets et l'implémentation de JavaScript.

Avec l'arrivée de HTML5, les développeurs comme les utilisateurs vont découvrir une différence, ainsi qu'une nouvelle sensation d'utilisation.



« Pour une meilleure présentation des démonstrations veuillez utiliser les navigateurs convenus pour cet effet. Vous trouverez des propositions des meilleurs navigateurs à côté du lien ainsi que sur la page elle-même. »

### 2.2.1 Nouvelles balises de form

- **datalist**
  - Il s'agit d'un hybride entre les éléments `input` et `select`. Cela permet aux utilisateurs de sélectionner une option proposée ou d'entrer une nouvelle valeur.
- **keygen**
  - L'utilisation principale de `keygen` est d'authentifier l'utilisateur dans des paramètres de sécurité. `Keygen` génère une paire de clés. L'une est privée et stockée sur le client tandis que l'autre est publique et est envoyée au serveur.
- **output**
  - Il représente le résultat d'un calcul. Ainsi le résultat peut être obtenu sans rafraîchir la page.

### 2.2.2 Nouveaux attributs de form

- **autocomplete**
  - `Autocomplete` est un attribut de `form` et d'`input` qui permet aux navigateurs de prédire l'entrée pour un champ. La prédiction s'effectue en fonction des valeurs enregistrées précédemment pour le champ.

```
<form action="demo_form.asp" method="get" autocomplete="on">  
  First name: <input type="text" name="fname" /><br />  
  Last name: <input type="text" name="lname" /><br />
```

```
<input type="submit" />
</form>
```

- **novalidate**
  - L'attribut `novalidate` permet de soumettre un formulaire sans réaliser la validation des champs. Par exemple, cela permet d'éviter de vérifier des champs d'e-mail.

```
<form action="demo_form.asp" novalidate="novalidate">
  E-mail: <input type="email" name="user_email" />
  <input type="submit" />
</form>
```

### 2.2.3 Nouveaux attributs d'input

- **autocomplete**
  - Il fonctionne comme l'attribut de `form`.

```
<input type="email" name="email" autocomplete="off" />
```

- **autofocus**
  - Il s'agit d'un attribut qui permet d'affecter le focus à un champ précis au moment du chargement de la page.

```
Nom : <input type="text" name="nom" autofocus="autofocus" />
```

- **form *tampering***
  - L'attribut `form` d'input spécifie un champ qui appartient à un formulaire, mais qui n'est pas inclus entre les balises du formulaire.

```
<form action="form.php" method="get" id="formuser">
  First name: <input type="text" name="firstname" />
  <input type="submit" />
</form>
Last name: <input type="text" name="lastname" form="formuser" />
```



- **form overrides** (formaction, formenctype, formmethod, formnovalidate, formtarget)
  - Les attributs de **form overrides** permettent de remplacer certains des attributs définis dans l'élément de formulaire `<form>`.

```
<form action="form.php" method="get" id="form">
  E-mail: <input type="email" name="usermail" />
  <!--Soumettre validation normal-->
  <input type="submit" value="Submit" />
  <!-- Soumettre autant qu'admin -->
  <input type="submit" formaction="adm.php" value="Admin" />
  <!-- Soumettre sans validation -->
  <input type="submit" formnovalidate="true" value="Non"/>
</form>
```

- **height and width**
  - Ces attributs représentent la taille (hauteur et largeur) d'une image utilisée pour l'élément `input`.

```
<input type="image" src="smile.gif" width="24" height="24" />
```

- **list**
  - List fait le lien entre un élément `input` et une `datalist`. Ainsi l'élément `input` peut être utilisé comme une sorte de liste déroulante.

```
<input list="voiture" />
<datalist id="voiture">
  <option value="BMW">
  <option value="VW">
</datalist>
```

- **min, max and step**
  - La plupart du temps ces attributs sont utilisés avec le type `number`. Ils permettent de spécifier la valeur minimum et maximum acceptées (`min` et `max`). Tandis que l'attribut `step`, qui peut être combiné avec `min` et `max`, spécifie l'incrément de la valeur.

```
<input type="number" name="points" min="0" max="10" step="3" />
```

- multiple
  - Cet attribut spécifie que plusieurs valeurs peuvent être sélectionnées. Il est utilisé pour les champs de saisie tels que l'e-mail ou les fichiers.

```
<input type="file" name="img" multiple="multiple" />
```

- pattern (regexp)
  - L'attribut pattern est utilisé avec les inputs comme recherche, text, url, téléphone, email et mot de passe. En fait, il s'agit d'une expression régulière qui est utilisée pour valider un champ de saisie. Ainsi, le type de valeur attendu peut être spécifié précisément.

```
<input type="text" pattern="[0 - 9] {5}">
```

- placeholder
  - Il permet de montrer visuellement dans le champ, les types des données attendues. De cette manière, des utilisateurs ont un indice qui décrit la valeur souhaitée.

```
<input type="text" name="nom" placeholder="Nom" >
```

- required
  - L'attribut required indique que le champ de saisie doit être rempli avant d'effectuer une validation. Si ce n'est pas le cas, un message d'erreur sera affiché pour avertir l'utilisateur.

```
<input type="text" name="nom" required>
```



Pour s'apercevoir de la différence de formulaire entre HTML4 et HTML5, référez-vous à la page suivante, en utilisant **Google Chrome** :

<http://localhost/HTML/form.html>



Pour effectuer le test sur les nouveaux types de formulaire et input, référez-vous à la page suivante, en utilisant le navigateur **Opera** :

<http://localhost/HTML/forms.html>

### 3. Détection des caractéristiques HTML5

HTML5 représente clairement un énorme pas en avant dans le monde des applications Web. Cela se manifeste par de nouvelles caractéristiques qui facilitent le développement. Pour éviter une réorganisation des pages Web existantes, la dernière version de HTML est conçue pour être compatible avec le HTML4.

En fait, HTML5 n'est pas un simple successeur de HTML4 qui a hérité seulement de ses normes, mais il est aussi un descendant de XHTML1.0 et DOM2. De ce fait, la dernière version d'HTML possède de plus grandes possibilités et des nouvelles fonctionnalités inconnues jusqu'alors. Ces améliorations n'ont pas complexifié son utilisation, mais au contraire l'ont simplifié.

#### 3.1 HTML5 Doctype

Un des changements majeurs réalisés en HTML5 est le Doctype. En absence de ce dernier, le navigateur n'aura pas le support, c'est-à-dire les règles, selon lesquelles il affichera le code source de la page HTML ou XHTML. Donc, cela peut entraîner des différences importantes selon navigateurs.

HTML4 admet de multiples variantes de Doctype, que les pages Web utilisent actuellement, tandis que HTML5 propose un Doctype unique et simplifié.

**Figure 3**  
**Quirks mode du Web**

#### **HTML4 :**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

#### **HTML5 :**

```
<!DOCTYPE html>
```

Ce mode de compatibilité ou le respect des standards utilisé par les navigateurs Web est aussi appelé le « quirks mode ».

## 3.2 Canvas

Le navigateur Mosaic, créé en 1992 au Centre national de Supercomputing Applications (NCSA), était le premier navigateur ayant la possibilité d'afficher des images en ligne avec du texte. Depuis sa mise sur le marché en 1993, les images sont restées statiques. En fait, il est possible de manipuler les images se trouvant sur le serveur, mais une fois l'image chargée chez le client, celle-ci ne peut plus être transformée.

Le but de la balise `canvas` est de donner un certain dynamisme aux pages Web. Cet élément représente un espace de forme rectangulaire sur la page qui peut être utilisé pour des créations diverses.

Le principe du canvas a été inventé par Apple™, dans l'application Dashboard pour le système d'exploitation Mac OS X Tiger. Cette application est utilisée pour héberger des Widgets, c'est-à-dire des mini-applications choisies par l'utilisateur, dans un espace prévu à cet effet. Par la suite, les développeurs d'Apple™ ont intégré cette fonctionnalité dans le navigateur Safari.

Des graphismes de jeux, du texte ou encore des images, animations, vidéos et audios peuvent être traités dans l'élément canvas, pour obtenir un site dynamique.

L'élément canvas possède deux attributs. Il s'agit de `width` et `height` (largeur et hauteur). Ces deux attributs optionnels permettent de dessiner l'espace rectangulaire dans lequel s'affichera l'objet.

```
< canvas id="myCanvas" width="400" height="200"> </canvas>
```

Dans le cas où les attributs ne sont pas spécifiés, le canvas sera toujours initialisé aux valeurs par défaut (300px \* 150px).

Pour dessiner dans le `canvas`, il faut utiliser JavaScript. En tout premier, il faut déclarer le canvas en appelant la méthode `getElementById`. Cela permet de se référer au canvas au moyen d'un identifiant (id) spécifié. Ensuite, il faut initialiser le contexte souhaité en invoquant la méthode: `canvas.getContext`. Pour le moment, il existe un seul contexte de dessin et il s'agit du contexte « 2D ». Cependant, les développeurs envisagent d'intégrer le contexte « 3D » dans les années à venir.

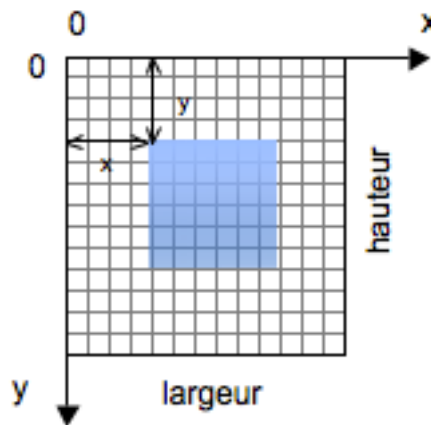


« Dans le cas où vous souhaitez tester le canvas et effectuer des nouvelles réalisations, je vous invite à télécharger [modernizr.js](#). Il s'agit d'une librairie JavaScript qui apporte une vérification supplémentaire. »

### 3.2.1 Dessiner sur le canvas

Pour réaliser un dessin, comme dans la plupart des langages qui se basent sur le canvas pour dessiner, il est essentiel de se référer aux pixels. L'origine, définie par les coordonnées  $<0, 0>$  se trouve dans le coin en haut à gauche.

**Figure 4**  
**Dessin de rectangle avec canvas**



La figure 4 représente une grille sur laquelle est dessiné un carré. La plupart du temps, une unité correspond à un pixel. Ainsi la position du carré bleu est calculée selon la hauteur et la largeur  $<x, y>$  depuis les points initiaux.

**Figure 4.1**  
**Code de rectangle avec canvas**

```
<!DOCTYPE HTML>
<html>
  <body>
    <canvas id="myCanvas" width="100" height="100" >
      Your browser does not support the canvas element.
    </canvas>
    <script type="text/javascript">
      var canvas=document.getElementById("myCanvas");
      var context=canvas.getContext("2d");
      context.fillStyle="#0000FF";
      context.fillRect(4,4,6,6);
    </script>
  </body>
</html>
```

Pour réaliser le rectangle de la figure 4 il suffit d'utiliser deux fonctions. Comme démontré dans la figure 4.1, la fonction **fillStyle()** permet de définir la couleur avec

laquelle l'objet sera rempli, alors que la fonction **fillRect()** permet de dessiner un rectangle plein.

```
canvas.fillRect( x, y, largeur, hauteur);
```

En se référant à la figure 4 et en utilisant la fonction **fillRect()**, il est facile de réaliser le dessin.

Toutefois l'élément canvas contient plusieurs fonctions qui permettent de dessiner des objets géométriques. Ainsi, en toute simplicité, il est possible de dessiner des lignes, des cercles, des caractères, de rajouter des images et tout autre objet en combinant les méthodes existantes.

### 3.2.1.1 Dessiner avec Paths

La méthode permet de dessiner n'importe quelle forme sur le canvas. Ce moyen mérite d'être mentionné car il s'agit d'un procédé simple à réaliser. Pourtant il est nécessaire de bien comprendre le concept.

La méthode **path** permet de créer une liste des points, qui seront reliés par des lignes. Les méthodes comme **lineTo()**, **arcTo()**, **quadraticCurveTo()**, **ebezierCurveTo()** peuvent être utilisées pour réaliser des lignes droites ou courbées.

Pour initialiser la méthode Path il faut appeler la fonction **beginPath()** et pour la fermer il faut utiliser la fonction **closePath()**. Si le point final est connecté avec le point initial, la méthode **path** est considérée comme fermée.

**Figure 5**  
**Dessin HEG**



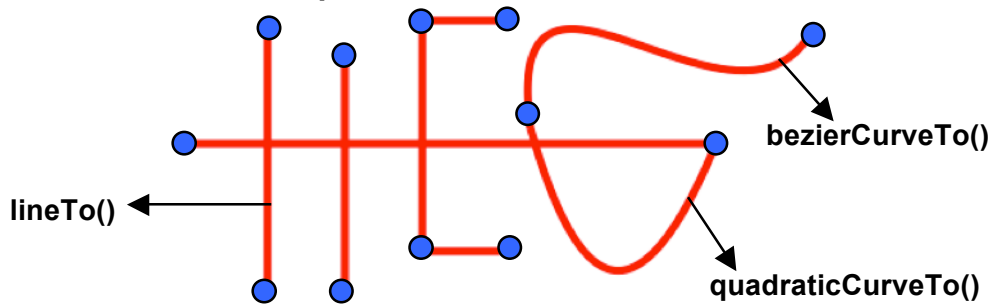
La figure 5 représente un dessin qu'il est possible de réaliser en employant le concept de **path**.



L'exemple de la figure 5 peut être trouvé sur site Web en se référant à la page suivante : <http://localhost/HTML/canvaspath.html>

Figure 5.1

### Explication du dessin HEG



Le dessin de la figure 5.1 a été créé avec 6 lignes droites **lineTo()**, une courbe quadratique **quadraticCurveTo()** et une courbe de Bézier **bezierCurveTo()**. Dans cet exemple, les points initiaux et finaux de chaque ligne sont illustrés par un point bleu. Pour définir le point initial d'un nouveau chemin à un endroit précis, il faut utiliser la fonction **moveTo()**. De cette façon, le nouveau point débutant d'un chemin est spécifié. Si l'utilisation de **moveTo()** n'était pas employé dans cet exemple, chaque nouveau point déclaré par **lineTo()** serait relié au précédent par une ligne droite. Pour dessiner les lignes, il suffit d'invoquer la méthode **stroke()** qui permet de tracer effectivement les lignes entre les points.

```
function canvasApp() {  
    var theCanvas = document.getElementById("canvasOne");  
    var context = theCanvas.getContext("2d");  
  
    context.beginPath();  
    context.moveTo(50, 100); // positionnement(x, y)  
    context.lineTo(392, 100); // line 1 (middle)  
    context.moveTo(100, 20);  
    context.lineTo(100, 200); // line 2 (H - left)  
    context.moveTo(150, 40);  
    context.lineTo(150, 200); // line 3 (H - right)  
    context.moveTo(200, 20);  
    context.lineTo(200, 170); // line 4 (E - back)  
    context.moveTo(197, 20);  
    context.lineTo(250, 20); // line 5 (E - top)  
    context.moveTo(197, 170);  
    context.lineTo(250, 170); // line 6 (E - bottom)  
    context.moveTo(270, 90);  
    context.quadraticCurveTo(320, 270, 390, 100); // quadratic curve  
    context.moveTo(270, 90);  
    context.bezierCurveTo(250, -60, 400, 110, 450, 30); // bezier curve  
    context.lineWidth = 5;  
    context.strokeStyle = "#ff0000";  
    context.stroke();  
    context.closePath();  
}
```

### **3.3 Multimédia**

Le multimédia est un outil vidéo qui permet grâce à l'image de diffuser beaucoup d'informations. La communication actuelle se fait principalement par le multimédia. L'évolution du journal télévisé, des vidéos en ligne, des tutoriels, le démontre. C'est pourquoi, cet outil est devenu une source essentielle d'informations. En fait, que ce soit des images, des vidéos, de l'audio ou tout simplement des animations, le multimédia représente un support de diffusion sur le Web. Conjointement à l'augmentation de la bande passante, le multimédia sur Internet devient de plus en plus répandu.

De nos jours, pour obtenir des informations visuelles provenant d'Internet, les utilisateurs doivent disposer d'un plug-in ou d'un media player. Ce sont des logiciels qui communiquent avec le navigateur et qui sont capables d'afficher des fichiers multimédias.

La manière de visionner les vidéos sur le Web est aussi appelée « streaming ». Les sites Web les plus connus basés sur le streaming sont **YouTube**, **Dailymotion** et **Megavideo**. Le streaming permet aux fournisseurs de contenu de partager les vidéos en les diffusant sur le Web. De cette manière, les utilisateurs peuvent éviter le téléchargement du contenu, comme c'est le cas avec le podcast.

Par contre, pour lire ces vidéos, les consommateurs sont obligés de disposer d'un outil de lecture. D'où le problème posé par quelques smartphones qui n'intègrent pas certains outils tiers pour afficher le multimédia.

Pour cette raison un des éléments le plus populaire et très puissant dans le HTML5 est l'arrivée de la vidéo et de l'audio. En fait, dès l'intégration complète de HTML5 sur le Web, les sites ne seront plus obligés de s'appuyer sur des outils tiers comme Flash ou Silverlight pour lire la vidéo et l'audio.

#### **3.3.1 Audio**

Jusqu'à présent HTML n'était pas capable de diffuser de l'audio sans la présence d'un plug-in comme Adobe Flash. Aujourd'hui avec l'implémentation d'un nouvel élément, il est tout à fait possible de créer, en toute simplicité, une page Web qui diffuse du son.

```
< audio src= "music.mp3"> </audio >
```

Selon l'exemple ci-dessus, il est clair que l'implémentation d'audio est très facile. Par contre HTML5 définit 5 attributs pour cet élément. Il s'agit d'attributs élémentaires qui permettent une manipulation basique d'un élément audio.



**Tableau 2**  
**Les attributs de l'élément audio**

<b>src</b>	définit la source du fichier.
<b>autoplay</b>	un attribut booléen qui permet de spécifier si le son doit s'activer automatiquement dès que c'est possible.
<b>loop</b>	un attribut booléen indiquant si le son doit fonctionner en boucle.
<b>controls</b>	permet d'afficher les contrôles de média par défaut.
<b>preload</b>	un attribut permettant de définir avec « none », « auto » ou « metadata » si le navigateur a l'autorisation de précharger le son.

Le problème principal de l'élément `audio` vient des formats existants. En fait, il existe plusieurs formats, tels que MP3, AAC et Vorbis (plus connu sous la forme d'un fichier « .ogg »). La différence entre ces trois formats est le brevet, car le format MP3 est breveté. Cela signifie que les entreprises doivent payer les droits pour pouvoir le décoder. Par exemple, Vorbis, est principalement utilisé par les petites entreprises ou les groupes open source tels que Firefox, tandis qu'Apple<sup>TM</sup>, étant une grande compagnie, oriente ses produits vers l'extension AAC via son prédécesseur MP3.

Par contre, il existe un moyen pour s'assurer que tous les navigateurs peuvent lire un contenu audio. Il faut tout d'abord créer le même contenu sous deux formats différents. Ensuite, il faut définir la balise `audio` de telle manière qu'elle reconnaisse les deux formats.

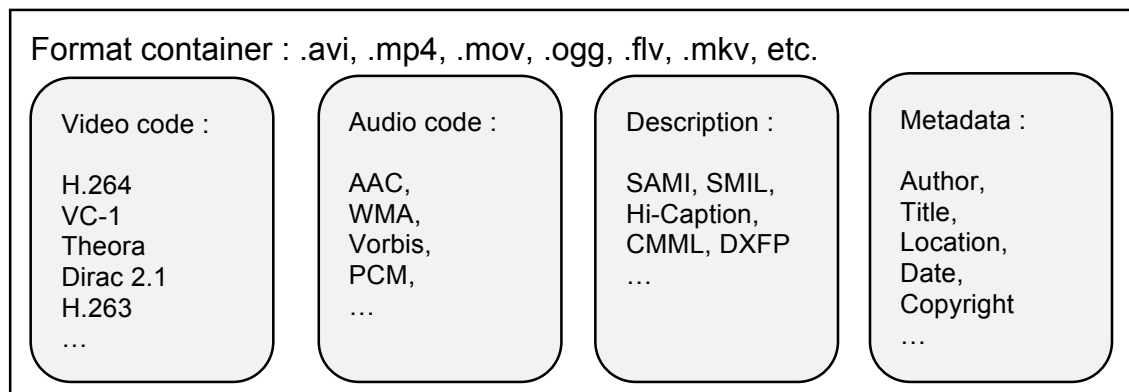
```
< audio >
  < source src= "music.mp3">
  < source src= "music.ogg">
</audio >
```

Ainsi, il est certain que tous les navigateurs pourront lire le contenu audio. En effet, un navigateur pouvant lire le fichier Vorbis, sautera le premier élément source pour lire le format qu'il connaît.

### 3.3.2 Vidéo

La video, étant le nouvel élément de HTML, fonctionne comme l'élément audio. Donc comme pour l'audio, HTML5 essaye d'homogénéiser les méthodes permettant de diffuser la vidéo. Par contre, la vidéo, en tant que diffuseur d'image et de son, est beaucoup plus complexe à gérer.

**Figure 6**  
**Les codecs vidéo**



Source : S.n. Codecs and formats. In : W3C [en ligne]. Modifié en 2008.  
[http://www.w3.org/2008/01/video-on-the-web.html#\(10\)](http://www.w3.org/2008/01/video-on-the-web.html#(10)) (Consulté le 20.08.2011)

Comme présenté dans la figure 6, il existe plusieurs formats de vidéo. Chaque format vidéo contient différents codecs de vidéo et de son. Mp4 est une des principales préoccupations pour les navigateurs, car son codec (H.264) est breveté et les licences doivent être payées par les entreprises qui l'exploitent.

Par contre, la solution pour les codecs est pratiquement la même comme dans le cas de l'audio. Il suffit d'encoder la vidéo en plusieurs formats, pour s'assurer qu'elle fonctionnera sur tous les navigateurs. Pour effectuer l'encodage il suffit d'utiliser le « Miro Video Converter ». Il s'agit d'un logiciel gratuit qui fonctionne sur Mac OS X comme sur Windows<sup>TM</sup>. Il est très facile à utiliser et il ne perd pas la qualité de la vidéo.

En raison d'absence d'un format vidéo commun, l'utilisation de plusieurs d'éléments source, permettra aux navigateurs de choisir le format automatiquement selon ses capacités.

```
<video poster="star.png" autoplay loop controls>
  <source src="movie.webm" type="video/webm; codecs="vp8, vorbis" />
  <source src="movie.ogv" type="video/ogg; codecs="theora, vorbis" />
</video>
```

L'élément `video` contient neuf attributs par défaut, lui permettant de modifier le fonctionnement du lecteur :

**Tableau 3**  
**Les attributs de l'élément vidéo**

<b>muted</b>	indique l'état par défaut de l'audio.
<b>autoplay</b>	un attribut booléen qui permet de spécifier si la vidéo doit s'activer automatiquement dès que c'est possible.
<b>controls</b>	permet d'afficher les contrôles de média par défaut.
<b>height</b>	définit la hauteur du lecteur vidéo.
<b>loop</b>	un attribut booléen indiquant si la vidéo doit lire en boucle.
<b>poster</b>	indique l'url d'une image représentant la vidéo.
<b>preload</b>	un attribut permettant de définir avec « none », « auto » ou « metadata » si le navigateur a l'autorisation de précharger la vidéo.
<b>src</b>	définit la source du fichier vidéo.
<b>width</b>	définit la largeur du lecteur vidéo.

La présence de la balise `video`, donne le meilleur environnement pour intégrer la vidéo dans une page Web. De plus, avec la coopération entre la `video` et le `canvas`, il est très probable d'obtenir des étapes plus évoluées. Cela nous permettra de faire toutes sortes de transformations et d'être d'avantage créatif sur la toile.



L'exemple de l'audio peut être trouvé sur la page suivante :

<http://localhost/HTML/multiaudio.html>



L'exemple de la vidéo peut être visionné sur la page suivante :

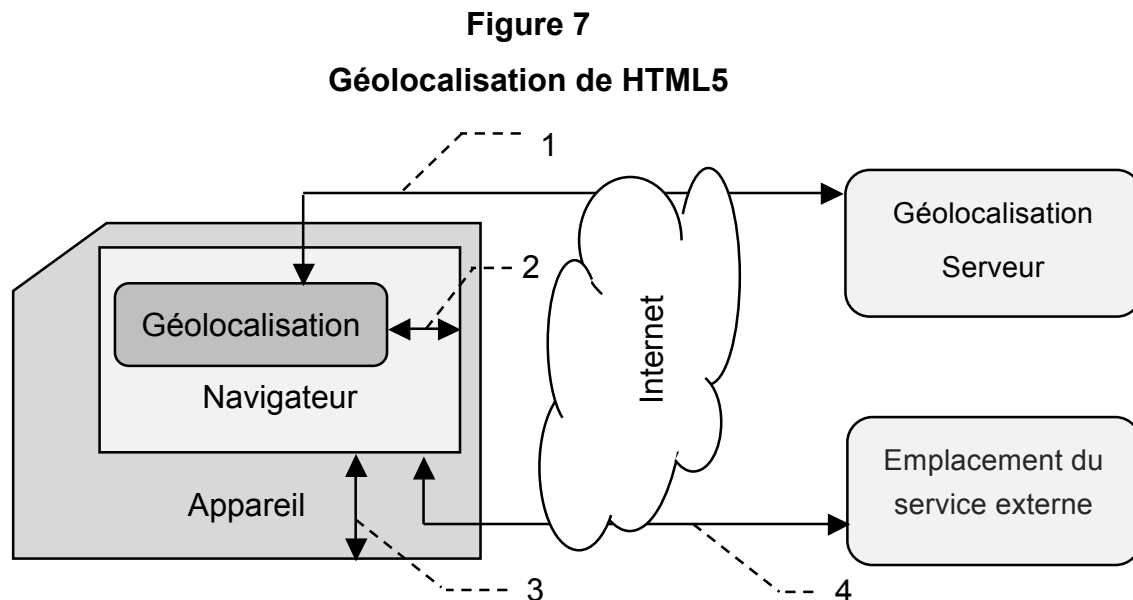
<http://localhost/HTML/multivideo.html>

### 3.4 Géolocalisation

La géolocalisation permet de repérer un objet sur une carte selon ses coordonnées géographiques. La plupart du temps, avec les nouvelles technologies, la localisation est effectuée grâce à un système de positionnement par satellite (GPS). Mais ces dernières années, les technologies de géolocalisation se sont diversifiées, particulièrement avec l'arrivée des Smartphones et l'accroissement des réseaux informatiques. De ce fait, sur le marché d'aujourd'hui, il existe plusieurs méthodes de géolocalisation, telles que :

- **GPS** (satellite)
- **adresse IP** (correspondance avec bases de données)
- **triangulation GSM/3G Cell IDs** (mobiles / tablettes)
- **WiFi** (mobiles et bases de données adresses MAC)
- **puce RFID**

La géolocalisation de HTML5 est soutenue dans les nombreux navigateurs. En combinant les capacités de HTML5 avec la géolocalisation API, cela permettra aux développeurs de créer des applications pour les navigateurs qui rivaliseront avec les applications natives.



Source: LUBBERS, Peter, ALBERS, Brian, SALIM, Frank. *Pro HTML5 Programming : powerful APIs for Richer Internet Application Development*. New York : Apress, 2010. P. 93

Comme montré dans la figure 7, l'application de géolocalisation n'a pas un accès direct à un appareil. La géolocalisation s'effectue via un navigateur qui avertira l'utilisateur en lui demandant une permission pour effectuer la localisation. Cela permet une sorte de sécurité pour les utilisateurs.

Les étapes suivantes sont représentées dans la figure 7 ci-dessus:

1. Un utilisateur accède à une application de géolocalisation dans le navigateur.
2. En chargeant l'application web, cette dernière demande les coordonnées à partir du navigateur. Pour cela, il fait appel aux fonctions de la géolocalisation. Le navigateur intercepte les requêtes et la permission de l'utilisateur.
3. Si l'utilisateur a accordé la permission, le navigateur récupère les informations de coordonnées de l'appareil. Par exemple, l'adresse IP, Wi-Fi, ou les coordonnées GPS.
4. Le navigateur envoie ces coordonnées vers un service de localisation externes en toute sécurité. Puis retourne un emplacement détaillé qui désormais peut être envoyé à l'hôte de l'application de géolocalisation HTML5.

### 3.4.1 Les fonctions

La géolocalisation contient trois fonctions :

- `getCurrentPosition`
  - la fonction qui permet d'obtenir l'emplacement actuel
- `watchPosition`
  - la fonction asynchrone permettant d'avertir de tout changement de position. Donc elle garde la position actuelle en surveillant les changements possibles de l'emplacement
- `clearWatch`
  - la fonction *clearWatch*, permet d'arrêter le processus en effaçant les données de la position

```
navigator.geolocation.getCurrentPosition(  
    ...  
);
```

```
var watchId = navigator.geolocation.watchPosition(  
    successfunction, errorfunction, {  
        maximumAge:5000, timeout:2000  
    }  
);  
function cancelButton(){  
    clearWatch(watchId);  
}
```

Pour le moment la géolocalisation est récente et les services offerts sont plutôt rares. Mais avec le temps cette prestation pourra changer le fonctionnement de certains sites Web.

### **3.5 Le stockage local**

Le stockage local, aussi appelé Web Storage ou DOM Storage, est intégré aux spécifications HTML5. Il permet aux pages Web de stocker des paires clé / valeur au niveau local. Cela permet aux navigateurs de garder les informations, même après la fermeture ou le redémarrage du navigateur.

#### **3.5.1 Histoire**

Le HTTP cookie est la première technique, aujourd'hui encore utilisée, permettant un stockage de données local. Cookie, étant un simple fichier de stockage, se base sur la paires clé/ valeur. Il peut être utilisé pour l'authentification ou l'identification de la session. De plus, il est très souvent utilisé pour conserver les données relatives de l'utilisateur durant la navigation.

Un des problèmes principaux des cookies est le piratage. Les cookies peuvent être interceptés et lus par une personne tierce. Ainsi cette personne obtiendra l'intégralité du contenu des cookies. Le deuxième problème est la taille et le nombre des cookies. En effet, le cookie dispose d'une taille limitée à 4 ko. De plus un client Web ne peut pas avoir plus de 300 cookies sur son disque dur.

Mais il existe d'autres façons de stocker des informations en local. Ces fonctionnalités ont été développées durant les dernières années. Par exemple Adobe™ a introduit dans Flash 6 des « Flash Cookies ». Ainsi il pouvait gérer jusqu'à 100ko d'objets Flash de données. Par la suite, les versions ont été améliorées grâce à l'avancement de Flash.

Google™ a aussi développé un plugin navigateur open source, en 2007, permettant de stocker des quantités illimitées de données dans les tables de base SQL. En effet, ce plugin, nommé Google Gears, fournit un API pour une base de données SQL basée sur SQLite.

Par contre certains de ces stockages ne fonctionnaient pas sur tous les navigateurs. C'est la raison pour laquelle le HTTP cookie est resté la façon de stocker la plus répandue. Le HTML5 permet de résoudre les problèmes existants.

#### **3.5.2 Fonctionnement de Web Storage**

HTML5 Web Storage permet aux applications de faciliter le stockage et d'écrire sur des fichiers en JavaScript. Désormais, il est donc possible de stocker de grandes quantités de données sur le système natif du client. En fait, ce système de stockage permet aux

applications d'améliorer considérablement la vitesse de navigation. De plus, le stockage local se veut plus persistant car il n'a pas de date d'expiration contrairement au cookie. Un énorme avantage de Web Storage par rapport aux cookies se trouve dans la répercussion des données. Chaque fois, lorsqu'une nouvelle donnée est stockée, elle est répercutée sur toutes les fenêtres et les onglets du navigateur. Ces données sont stockées dans un tableau associatif au dictionnaire.

**Figure 8**

**Tableau de Web Storage**

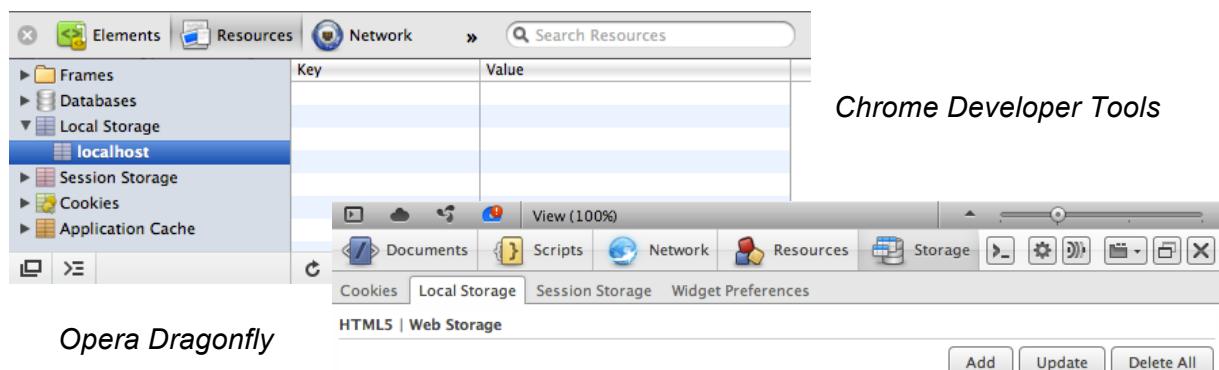
Clé	Valeur
"prenom"	"Titi Toto"
"articles"	456   678   321
"date"	"2011/08/17 16:04:25"

La Figure 8 représente un tableau contenant l'identifiant d'un client, les numéros d'articles qui ont été achetés, ainsi que la date d'achat. Comme on peut le voir, une valeur correspond toujours à une clé et une clé peut contenir plusieurs valeurs.

Ce même tableau peut être trouvé dans les navigateurs :

**Figure 9**

**Outils de développement des navigateurs sans les données**



La figure 9 illustre des présentations du Web storage de deux navigateurs différents. La ressemblance entre la figure 8 et 9 est évidente, car il s'agit du même tableau qui contient des clés et des valeurs identiques. Cependant il existe une différence entre ces deux navigateurs, car dans l'Opera Dragonfly il y a la possibilité de manipuler ses

clés et ses valeurs manuellement, cela en utilisant les trois boutons « Add », « Upload » et « Delete All » situés à droite du plug-in.

Pour ajouter des données à un objet de stockage local, il suffit d'utiliser la méthode **setItem**. De ce fait, deux arguments sont rajoutés sur chaque ligne. Le premier argument est la clé et le deuxième est la valeur de cette clé.

```
localStorage.setItem('prenom', 'Titi Toto');
```

En faisant cette manipulation, il faut être attentif à ne pas écraser une clé existante, car, il est impossible d'avoir deux identifiants avec le même nom.

Pour récupérer les données stockées dans l'objet localStorage, il faut utiliser la méthode **getItem**.

```
var prenom = localStorage.getItem("prenom");
```

La méthode **getItem** prend la clé comme argument et retourne la valeur associée à cette clé. Si la clé passée en argument n'existe pas, la valeur « *undefined* » est retournée.



En se référant à l'exemple sur <http://localhost/HTML/localstorage.html>, les résultats suivant peuvent être obtenus, comme dans la figure 10.

**Figure 10**

### Outils de développement des navigateurs avec les données

The image shows two browser developer tools side-by-side. The top tool is Chrome Developer Tools, with the 'Local Storage' tab selected, displaying a table with keys 'date', 'prenom', and 'articles' and their respective values. The bottom tool is Opera Dragonfly, also with the 'Local Storage' tab selected, showing a similar table with the same data. Both tools have 'Add', 'Update', and 'Delete All' buttons at the bottom right of the storage view.

Key	Value
date	Thu Aug 18 2011 16:15:16 GMT+0200 (CEST)
prenom	Titi Toto
articles	456 678 321



« Pour accéder à **Chrome Developer Tools** ou **Opera Dragonfly** sur Windows utilisez les touches **Control - Shift - I** ou sur Mac **⌘ - ⌥ - I** (**Command - Option - I**). »



### 3.6 Offline Web applications

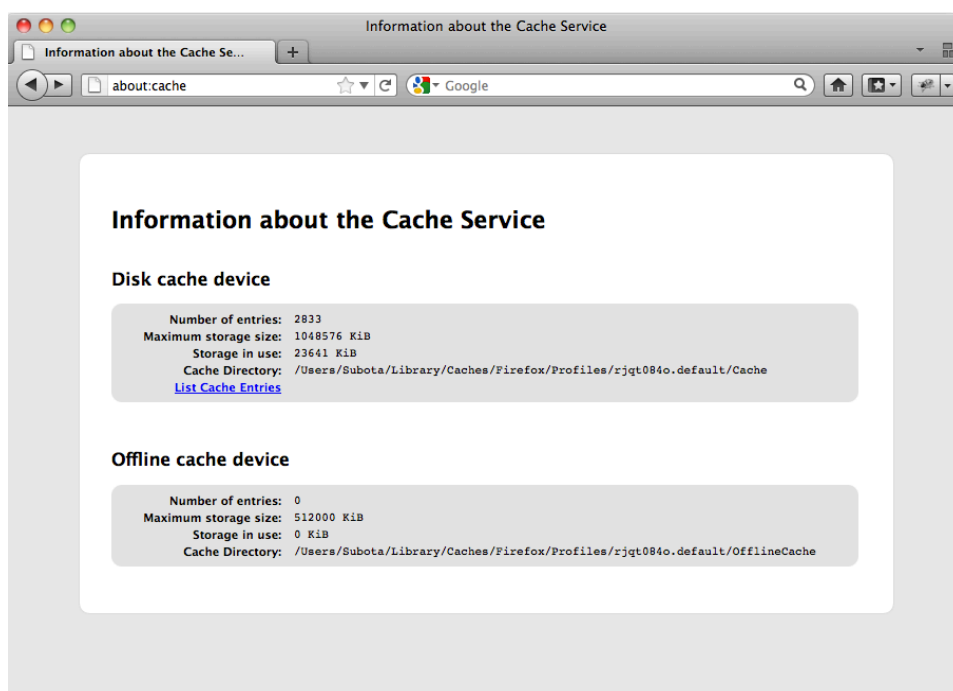
En utilisant Internet, il arrive qu'une coupure de connexion réseau se manifeste, ce qui peut provoquer des pertes de données et de fichiers. Afin de permettre une utilisation continue sans les interruptions, HTML5 donne un moyen de gérer ces cas particuliers. Une nouvelle spécification nommée « Offline Web », permet de conserver une copie des fichiers utiles pour la plateforme. Grâce à cette copie, le navigateur pourra afficher le contenu de la page Web hors ligne. Cette application peut être utilisée pour lire et rédiger les e-mails, modifier et afficher les présentations, etc.

Pour que les plateformes puissent fonctionner sans le réseau, il suffit d'indiquer au navigateur quelle partie du site Web doit être stockée dans son cache local. L'utilisation de stockage hors ligne permet d'éviter les demandes du réseau. Pour que cela puisse fonctionner, le manifeste du cache doit être mis à jour. Ainsi, le navigateur est au courant que toutes les ressources sont également mises à jour. Cela permet d'économiser de la bande passante, ce qui peut être très utile pour les appareils mobiles.

La plupart des navigateurs sont capables d'afficher les données de cache. En tapant dans la barre URL « *about:cache* » le navigateur affichera des détails sur le cache de l'application. Entre autre, Firefox offre la possibilité d'afficher les fichiers individuels dans le cache.

**Figure 11**

#### **Affichage du cache hors connexion dans Firefox**



### 3.6.1 Le manifeste de cache

Les applications Web se composent d'une liste d'adresses, qui est constituée des fichiers, HTML, JavaScript, CSS et des images. Ces adresses ou URL sont référencées par le fichier de manifeste. En fait, lors de la connexion à un réseau pour la première fois, le navigateur télécharge et stocke localement les ressources listées dans ce fichier. De cette façon, le navigateur qui n'est plus connecté à un réseau Web se référera aux copies des ressources existantes localement.

### 3.6.2 Fonctionnement d'Offline Web

Pour que le manifeste de cache puisse fonctionner, il faut indiquer le nom de fichier dans la balise html « `manifest="/offline.manifest"` ». De cette manière, le navigateur se réfère à ce fichier en question. Cette opération doit être répétée sur toutes les pages existantes du site Web, afin que ce dernier puisse fonctionner entièrement.

```
<!DOCTYPE html>
<html lang="en" manifest="/offline.manifest">
...
</html>
```

Le contenu d'un fichier manifeste peut se présenter de la manière suivante :

```
CACHE MANIFEST
example.html
example.js
example.css
example.gif
```

La liste de manifeste doit contenir tous les fichiers CSS, chaque image ainsi que tous les fichiers JavaScript pour que la page Web puisse fonctionner complètement. Si un des éléments n'est pas appelé dans la liste, il ne s'affichera pas au moment de la coupure de connexion réseau.



« Pour que la liste de manifeste puisse être atteignable hors connexion, il faut indiquer au serveur que le fichier manifeste doit être lu comme `text/cache-manifest`. En définissant la directive « `AddType text/cache-manifest .manifest` » dans le « `.htaccess`, ». Cela permet au serveur de prendre en considération le fichier pour effectuer le cache selon la liste définie. »

Il existe aussi moyen de rediriger l'utilisateur vers un fichier particulier, si une ressource n'est pas disponible hors connexion. Pour effectuer cela, il faut utiliser une section nommée FALLBACK. Cette section permet d'indiquer au navigateur d'afficher une page précise, si l'utilisateur tente d'accéder à une ressource non mise en cache.

```
FALLBACK:  
/exemple.html /offline.html
```

La section FALLBACK dispose de deux URL. La première URL représente la page demandée par l'utilisateur qui s'affichera seulement si la connexion réseau est activée. Si l'utilisateur tente d'attendre la page `exemple.html` hors connexion, il sera redirigé vers la page `offline.html`.

De plus, cette section peut mettre en cache chaque page que l'utilisateur a visité en ligne, afin qu'il puisse avoir accès à ces pages en mode hors connexion. De cette manière, la déclaration explicite de chaque fichier peut être évitée, dans la liste de manifeste.

Finalement, il existe la section NETWORK, utilisée pour dire au navigateur quelles ressources sont disponibles uniquement en ligne. Grâce à cette section, il est possible de préciser que les pages, de type login ou inscription, ne devront pas s'afficher en mode hors connexion.

```
NETWORK:  
login.php  
inscription.php
```

### 3.6.3 Conserver du cache

Une fois le contenu du cache téléchargé, il sera conservé dans le navigateur. Pour effacer le contenu, il existe deux méthodes : soit l'utilisateur effacera le cache par ses propres moyens, soit le développeur déclenchera une mise à jour du cache. Pour effectuer une mise à jour automatique, il suffit de déclarer la version du cache. Cela se fait grâce aux commentaires et chaque incrémentation de la version, déclenchera une mise à jour du cache.

```
CACHE MANIFEST  
# Version 9  
  
CACHE:  
/css/style.css
```



L'exemple du cache manifest se trouve sur la page suivante :

<http://localhost/HTML/offline.html>

## 4. Appareils mobiles

Avec l'arrivée des smartphones et des tablettes, l'utilisation des appareils mobiles, capables d'aller sur Internet a augmenté durant ces dernières années. De plus, de nouveaux modèles et systèmes d'exploitation sont apparus sur le marché, ce qui a rendu difficile le travail des développeurs, car la conception et le développement des plateformes Web ont dû être hautement personnalisés pour les appareils spécifiques. En conséquence, toutes les plateformes ayant pour but l'adaptation à cette technologie, étaient obligées de se baser sur des langages tiers, comme JavaScript, jQuery et autres, pour obtenir un affichage similaire à celui d'une application mobile.

Les nouvelles fonctionnalités de HTML5 permettront aux navigateurs Web mobile, capable de supporter le langage de balisage, d'afficher plus facilement le contenu. Par exemple, selon les valeurs d'attribut type, le navigateur mobile affichera un clavier différent sur l'écran.

```
<input id="email" name="email" type="email">  
<input id="website" name="website" type="url">  
<input id="phone" name="phone" type="tel">
```

**Figure 12**  
**Inputs des appareils mobiles**



Source : KEITH, Jeremy. HTML5 pour les web designers. Paris : Eyrolles, 2010. P. 48

Selon le type attribué au champ de saisie, le navigateur va s'adapter. Il affichera le clavier prévu à cette utilisation, comme illustré dans la figure 12.

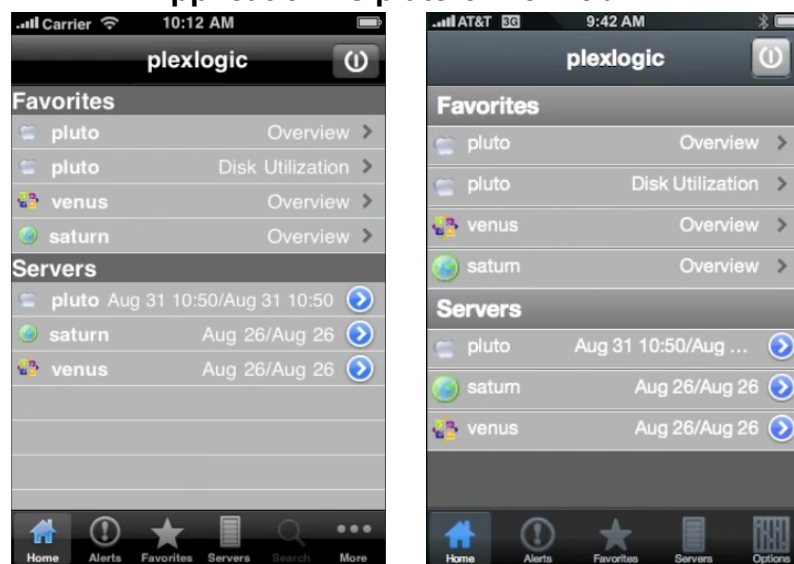
Les fonctionnalités qui existent avec HTML5 permettront aux développeurs de se faciliter la tâche et de réaliser d'autres créations. Le fait de pouvoir utiliser certaines plateformes hors ligne, pourra offrir aux utilisateurs une nouvelle manière d'utiliser les appareils mobiles. De plus, cela diminuera la bande passante et encouragera les utilisateurs à se connecter à ce type de plateforme, tout en diminuant le coût de connexion.

Cependant, il existe des outils d'interface spécialement conçus pour le Web mobile qui profitent du HTML5, CSS3 et JavaScript, notamment, la dernière version du Framework Sencha Touch. Celle-ci est basée sur les standards du Web pour développer l'interface utilisateur pour des applications Web mobiles. La bibliothèque de cet outil utilise les capacités de HTML5 et de style CSS3 pour enrichir les plateformes Web. Ces dernières offrent un riche usage à tel point qu'elles peuvent être comparées aux applications mobiles.

L'avantage de cet outil, permettant de créer une plateforme Web, est de fonctionner sur plusieurs systèmes mobiles tels que iOS et Android. De plus, la différence entre les applications natives et les plateformes Web, créées avec Sencha Touch est quasiment invisible.

**Figure 13**

### Application vs plateforme Web



Source : touchNOC[pseudonyme]. HTML5 Sencha App VS, native iPhone App side by side (HD). In YouTube[en ligne]. Modifié le 3 septembre 2010 [http://www.youtube.com/watch?v=61h8UGsi\\_M](http://www.youtube.com/watch?v=61h8UGsi_M) (consulté le 30.08.2011)

Comme montré dans la figure 13, l'écran gauche représente l'application native d'un iPhone, tandis que l'écran à droite est une plateforme réalisée avec Sencha Touch.

## 5. Problèmes actuels du HTML5






HTML5 devient de plus en plus populaire. Il est utilisé, d'ailleurs, par les grandes entreprises et plateformes telles que : **Facebook**, **YouTube**, et bientôt **Microsoft**. Mais comme toute nouvelle technologie, la possibilité qu'elle contienne des erreurs est très élevée. Cela implique de nouvelles difficultés qui peuvent engendrer beaucoup de problèmes.

### 5.1 Compatibilité avec les navigateurs

Dans le domaine des pages Web et des applications, HTML5 est omniprésent. Malgré cela, la question est de déterminer si tous les navigateurs peuvent, à ce jour, supporter les nouvelles caractéristiques de HTML5?

Pour répondre à cette question, il est nécessaire d'effectuer des tests sur ces navigateurs. Le site d'évaluation <http://beta.html5test.com/> est utilisé pour obtenir une note moyenne de chaque navigateur. Le site Web affiche les résultats pour les navigateurs qui supportent les spécifications HTML5. Ainsi, il est possible de savoir quelle partie de HTML5 est prise en charge.

**Tableau 4**  
**Navigateurs et HTML5**

Navigateur	Version	Point	Point Bonus
 Google Chrome	12.0.742	327 / 450	13
 Opera	11.50	296 / 450	7
 Mozilla Firefox	5.0	286 / 450	9
 Apple Safari	5.0.5	253 / 450	7
 Microsoft Internet Explorer	9	141 / 450	5

Source : The HTML5 test : how well does your browser support HTML5 ? [en ligne]. Modifié le 22 juin 2011. <http://beta.html5test.com/results.html> (consulté le 25.07.211)



« Les navigateurs sont exécutés avec les paramètres par défaut. Il est possible que vos résultats puissent varier légèrement en raison des paramètres, du système d'exploitation et d'autres facteurs. »

Source : ibid.






Le tableau 4 représente une comparaison de plusieurs navigateurs. Leurs fonctionnements sont enregistrés grâce à un algorithme de JavaScript. Cet algorithme permet, de tester chaque composant de HTML5 afin de déterminer les points obtenus pour chaque browser. Ensuite, si ces composants ont une meilleure compatibilité que ce qui est attendu, des bonus seront ajoutés. Par exemple, si le navigateur prend en charge plusieurs codecs, alors deux points supplémentaires lui seront accordés, par codec.

Le tableau 4 montre aussi que Google Chrome est actuellement le meilleur navigateur, suivi par Opera et Firefox. Mais comment comprendre ce résultat de la part du navigateur Google par rapport à ses rivaux ? Il faut savoir que Google Chrome est l'une des versions les plus récentes du marché. Il est plus abouti que les autres et utilise le moteur de rendu HTML WebKit. En fait, depuis le lancement de Google Chrome OS beta, fondé sur le navigateur Web Chrome et un noyau Linux, Google™ a investi toutes ses forces pour améliorer ses applications Web et notamment le navigateur. Subséquemment, les créateurs du moteur de recherche Google sortaient des nouvelles versions du navigateur presque chaque mois depuis début 2011.

### 5.1.1 Les points sensibles des navigateurs

La différence entre les navigateurs est évidente et majeure. Les points sensibles pour les navigateurs sont les « Parsing rules » ou les règles d'analyse, **Forms**, **Microdata**, **WebGL**, **Files**, **LocalMultimedia** et **Notifications**.<sup>3</sup>

**Tableau 5**  
**Navigateurs et certains éléments HTML5**

Point	Parsing rules	Form	Microdata	WebGL	Files	LM	Notific.
 327	11/11 + 2	59 /98	0 /15	23 /25	10 /20	0 /20	0 /10
 296	1 /11	90 /98	0 /15	0 /25	10 /20	0 /20	0 /10
 286	11/11 + 2	51 /98	0 /15	23 /25	10 /20	0 /20	0 /10
 253	1 /11	38 /98	0 /15	0 /25	0 /20	0 /20	0 /10
 141	1 /11 + 1	7 /98	0 /15	0 /25	0 /20	0 /20	0 /10

Source : The HTML5 test : how well does your browser support HTML5 ? [en ligne]. Modifié le 22 juin 2011. <http://beta.html5test.com> (consulté le 25.07.211)

<sup>3</sup> Les explications de Microdata, WebGL et autres peuvent être trouvés dans l'annexe 3.

Le tableau 5 représente les notes les moins hautes selon le site [html5test.com](http://html5test.com). Puisque HTML5 est toujours en version beta, il est tout à fait normal que les navigateurs ne suivent pas encore le développement dans tous les domaines. Par contre la majorité des aspects ont déjà été réalisés et fonctionnent relativement bien.

Entre autre, il est possible de constater que les navigateurs Opera et Mozilla Firefox, d'après leurs notes, sont des concurrents très sérieux et proches l'un de l'autre. Selon les points les plus critiques, Mozilla Firefox a dispersé ses efforts sur plusieurs points, tandis que Opera s'est concentré majoritairement sur les nouveautés de *form* de HTML5.

## **5.2 Mouture de HTML5**

Dans le courant du mois d'août 2011, les chercheurs d'ENISA (The European Network Information Security) ont publié un document en expliquant des problèmes qui peuvent s'avérer graves dans la nouvelle version d'HTML.

ENISA a été créé en 2004 et son siège se trouve à Heraklion, en Grèce. Cette agence est supervisée par un conseil d'administration et elle a pour but d'améliorer la sécurité des réseaux et des informations dans l'Union européenne.

Dans son document de 61 pages, l'Agence considère que les normes HTML5 devront être révisées par W3C. En analysant la sécurité de HTML5, ils ont identifiés 51 problèmes et menaces potentielles. De nombreuses nouvelles fonctionnalités pourront représenter des menaces pour les utilisateurs.

Précisément, en appliquant, la spécification appelée *form tampering*, dans une plateforme Web, la vulnérabilité de cette dernière peut s'avérer élevée. En fait, cette spécification permet de placer le bouton d'envoi d'un formulaire n'importe où sur la page. En injectant un autre code HTML, il est possible de dissimuler le bouton en dirigeant les informations vers l'attaquant plutôt que vers le site web légitime.

```
<form action="form.php" method="get" id="formnom">
  Nom: <input type="text" name="firstname" />
</form>
<input type="submit" form="formnom" />
```

Le stockage Web semble présenter aussi des risques de sécurité. En fait, même s'il est possible de récupérer les données de cookies, certains développeurs considèrent que stockage Web peut être beaucoup plus vulnérable que les cookies. Il est tout à fait



possible qu'un attaquant vole des cookies pour obtenir le mot de passe d'un site Web. Avec le stockage Web, il suffit d'entrer dans le navigateur de l'utilisateur pour s'emparer des données, qui peuvent être constituées de diverses informations. D'autre part, il se peut qu'un attaquant puisse avoir la possibilité d'insérer des données dans le stockage Web et ainsi les manipuler avec certains sites.

À partir du moment où l'attaquant s'introduit dans le navigateur, il aura une plus grande palette de possibilités pour contrôler les données. Que ce soit avec la géolocalisation pour obtenir les coordonnées de sa victime, ou par d'autres bugs comme l'injection HTML et autres, il est clair que HTML5 n'est pas encore au point pour assurer une sécurité maximale pour les utilisateurs.

Il est donc important de garder la perspective sur la vie privée des utilisateurs. D'ailleurs cette vision est soutenue par tout le monde, ce qui force les développeurs de HTML5 à repousser la première sortie des normes et d'aborder les questions sur la confidentialité et la sécurité des utilisateurs.

## 6. HTML5 contre Flex & Flash

Grâce aux nouveautés proposées par HTML5, beaucoup de gens se demandent ce qu'il adviendra des outils existants sur le Net. Principalement pour les applications Internet riches, qui semblent être remplaçables par HTML5.

Les applications Internet riches (AIR) permettent d'effectuer de nombreux traitements côté client. De ce fait, un certain nombre de calculs sont effectués chez le client au lieu d'être envoyé au serveur. Cela permet une économie de temps, ainsi le nombre de rafraîchissements du navigateur. De plus, tous les ordinateurs équipés du Flash Player sont capables de lancer les applications Internet riches

Il existe deux manières de mettre en place des AIR :

- en utilisant les technologies Web, telles que le DOM, le XHTML et l'ECMAScript tout en suivant la méthode AJAX
- en utilisant un framework de développement conçu pour les AIR, tel que Flex d'Adobe, JavaFx

### 6.1 Adobe Flex™

Développé par Macromedia, le framework Flex™ a été rachetée par Adobe™ en 2005. Adobe Flex™ est un kit de développement logiciel (SDK), qui permet de développer des plateformes d'applications Internet riches (AIR). Les applications Flex™ peuvent

être écrites en utilisant Adobe Flash Builder et elles ont des caractéristiques de programmes lourds, sauf qu'elles sont livrées via le Web.

Les applications développées en Flash offrent une plateforme robuste pour l'utilisateur. De plus, la productivité et l'interactivité de ce dernier sont renforcées par la facilité d'utilisation et d'exécution. Tous les changements effectués sur l'application, se font via la couche client, sans déposer une demande au serveur. Ainsi, toutes les mises à jour se font automatiquement, ce qui représente une économie de temps et de charge pour les serveurs, la base de données et le réseau.

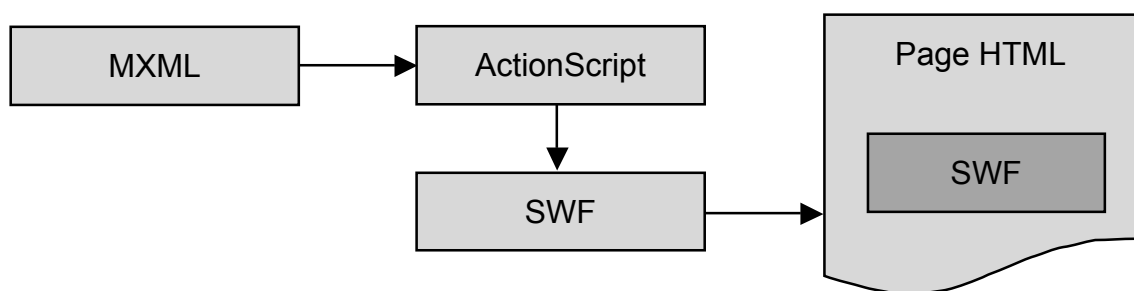
Pour développer en Flex™, il faut utiliser deux langages qui permettront la création d'une application Web :

- l'ActionScript
- le MXML

L'**ActionScript** est un langage de script, orienté objet, basé sur ECMAScript. Il est utilisé dans les applications Adobe Flex ou Flash et apporte de l'interactivité aux animations et aux différents objets multimédias. Donc, il permet de réaliser des applications complexes.

Le **MXML** est un langage basé sur XML permettant de définir l'interface utilisateur et les comportements d'applications Internet. MXML peut contenir des bouts de code d'ActionScript et il est souvent utilisé avec Flex Serveur, qui va par la suite compiler dynamiquement les fichiers SWF.

**Figure 14**  
**Processus de compilation Flex**



Source : BARON, Vianney ... [et al.] *Flex and Air*. Paris : MicroApplication, 2008. P.45

Pour mieux comprendre le fonctionnement de la figure 14, il est possible d'examiner le code à la page suivante pour comprendre l'affichage de « Hello world ! » sur le navigateur.

```
<?xml version="1.0" encoding="utf-8"?>
<mx: Application xmlns:mx=http://www.adobe.com/2006/mxml
    layout="absolute" creationComplete="printHello()">
    <mx:Script>
    <![CDATA[
        public function giveHelloWorld():String { return : "Hello world !" ; }
        public function printHello():void {
            textAreaId.text = giveHelloWorld() ;
        }
    ]]>
    </mx:Script>
    <mx: TextArea id = "textAreaId" éditable="false" />
</mx:Application>
```

### 6.1.1 HTML5 vs Flex™

HTML5 est loin d'être terminé, mais devient une menace pour Adobe Flex™. Même si Flex™ permet de créer des formulaires, des graphiques et que ces applications peuvent paraître professionnelles, il reste un outil contenant des erreurs connues par les développeurs. Malgré le support cohérent des navigateurs, Flex™ est tout de même mal supporté sous Linux, Mac et certains appareils mobiles.

D'autre part, certaines applications peuvent se montrer lourdes au moment du chargement, ce qui peut ralentir la navigation. De plus, si le client dispose d'une ancienne machine, le temps d'exécution peut s'avérer très long.

Le Flex™ offre la réutilisation des contrôles et des classes, ce qui permet aux développeurs expérimentés de gagner du temps. Par contre, le développeur est obligé de réaliser, par ses propres moyens, de simples fonctionnalités, qui peuvent s'avérer très longues à coder. Tandis que certaines de ces fonctionnalités existent par défaut en HTML, ce qui permet, au contraire de Flex™, une simple implémentation.

## 6.2 Adobe Flash

Adobe Flash est une plateforme permettant d'ajouter sur une page Web de l'animation, de la vidéo et de l'interactivité. Il est très souvent utilisé pour implémenter des jeux, des publicités et des animations flash.

A l'origine Flash a débuté avec le SmartSketch application, permettant d'utiliser les stylos pour les ordinateurs. Sans grand succès, il a été racheté par de nombreuses entreprises et modifié pour satisfaire leurs besoins. En 1995, SmartSketch fût modifié

pour fonctionner avec l'animation (image par image) et fût délivré comme FutureSplash Animator. Une année plus tard, FutureSplash a été acquis par Macromedia et publié sous le nom de Flash.

Les fichiers Flash se basent sur l'extension .swf et sont inclus dans les pages Web pour être lus par le plugin Flash ou le lecteur Flash Player. Donc, pour qu'un navigateur puisse lire de la vidéo ou l'audio, ce premier se basera sur le Flash pour lire les codecs et ensuite les afficher sur la page Web.

### **6.2.1 HTML5 vs Flash**

Une grande partie de Flash est utilisée dans la publicité, la vidéo et l'audio. Cette utilisation de Flash est représentée par 70% des plateformes Web. Cette façon de créer des pages Web pourra bientôt être remplacée par les nouveaux éléments de HTML5. Même si cette menace inquiète beaucoup les entreprises, Flash reste toujours le leader dans le domaine. De plus, il est connu pour sa rapidité et sa sécurité, ce qui explique pourquoi il est utilisé sur 98% des PC connectés à Internet. D'ailleurs c'est la raison pour laquelle il a réussi à s'intégrer sur le marché.

En offrant toutes sortes d'animations et d'interactions 2D comme 3D, Flash permet une meilleure gestion des jeux online, des vidéos et de l'audio. Par contre avec l'arrivée des smartphones et des tablettes, l'utilisation de Flash a diminué. Car Flash, étant une AIR, demande beaucoup de ressources, ce qui affaiblit très vite la batterie sur les appareils portables, tout en augmentant l'utilisation du CPU. De plus, Flash ne supporte pas du tout les interfaces tactiles et fait très souvent bloquer des systèmes d'exploitation mobiles. Ce sont des raisons lesquelles Apple<sup>TM</sup> n'a pas intégré cet outil dans ses appareils mobiles.

En effet, une partie de Flash, se trouvant sur Internet, pourra déjà être remplacée par HTML5, car HTML5 est plus rapide au moment du chargement. De plus, les fonctionnalités de HTML5 sont présentes par défaut dans les nouveaux navigateurs, tandis que Flash doit être ajouté manuellement.

## **6.3 HTML5 est-il le vainqueur?**

En défiant toute concurrence, Ajax possède une plus grande popularité qu'Adobe. C'est principalement à cause du prix élevé que les entreprises doivent payer à Adobe. En conséquence c'est exclusivement les grandes compagnies qui peuvent budgéter cet outil extraordinaire. De plus, le coût de ce logiciel est l'argument le plus souvent cité, surtout au moment de comparaison avec HTML5. En effet, il est possible

Même si les performances d'Adobe sont souvent comparées à celles d'HTML5, Adobe se montre, une fois de plus, ouvert aux nouvelles technologies du Web. En effet, la société a annoncé le lancement d'un nouvel outil beta, baptisé Adobe Edge (<http://labs.adobe.com/>). Cet outil permet de créer des animations riches, non pas en Flash, mais au format HTML, CSS et JavaScript.

Adobe Edge est, donc, un logiciel permettant de créer des plateformes Web dynamiques en HTML5 sans forcément toucher au code. En outre, l'ergonomie du logiciel ressemble à celle utilisées pour créer des animations Flash et Flex™.

HTML5  
Nemanja SUBOTIĆ

## Conclusion

A travers ce travail de Bachelor, j'ai voulu vous présenter ce qui me semblait l'aspect principal de HTML5. Cependant cela n'est qu'un bref aperçu, car HTML5 est toujours en évolution et donc certains domaines sont encore limités et inexplorables. A la base, HTML5 est conçu pour effectuer de nombreuses fonctionnalités. Toutefois, elles ne sont pas encore supportées par tous les navigateurs et en conséquence inapplicables.

Pour ne citer que quelques pistes, dans un futur proche, il sera possible d'effectuer un drag et drop, d'afficher des animations en 3D, de créer des jeux vidéo et d'utiliser de nombreuses nouvelles fonctionnalités.

Entre autre, le web et HTML5, sont en constante évolution et présents dans beaucoup de domaines, même s'ils ne sont pas encore au point. Pour ne citer qu'un exemple, prenons : RERODOC (<http://doc.rero.ch/>). Il s'agit d'une banque de données dans laquelle il est possible d'effectuer des recherches documentaires. L'Université de Genève, ainsi que l'Infothèque de la HEG utilisent cet outil. Une fois la recherche effectuée, il est possible de consulter en ligne le texte intégral en format PDF. L'outil utilisé pour effectuer cette tâche se nomme « Multivio ». Il est conçu avec HTML5 et permet d'afficher le contenu du document recherché.

A titre professionnel, HTML5 est une technologie de grande envergure qui dépasse largement les anciennes versions. D'un point de vue pratique, il est évident que les développeurs ont une plus grande palette de choix et de possibilités. Cela signifie également, un changement prochain de l'utilisation du Web pour les internautes. Quant à l'apport de ces variations, il est nécessaire de signaler que le design des sites Web se fait avec la nouvelle version CSS. Le CSS3 permet d'obtenir les effets 3D, tout en jouant avec les ombres dessinées sur les éléments. Ainsi il gère également la transparence ou encore la rotation des éléments. De ce fait, HTML5 représente le noyau dur du site qui est affiné via le design grâce à CSS3.

En effet, au cours de la rédaction de ce travail de Bachelor, je réalise qu'il ne s'agit que d'une « photo » prise sur une technologie qui avance à pas de géant. Dans quelque temps, ce que j'ai pu constater, écrire et explorer sera déjà en utilisation. Néanmoins, je reste optimiste sur ce que j'ai appris et j'espère pouvoir bientôt utiliser cette base acquise comme tremplin pour l'avenir.

## Webographie

- AHMED, Tariq. What is adobe flex ? In : *CFLEX : community Flex* [en ligne]. Modifié le 17 juin 2007. [http://www.cflex.net/about\\_adobe\\_flex.cfm](http://www.cflex.net/about_adobe_flex.cfm) (consulté le 19.08.2011)
- CHINNATHAMBI, Kirupa. HTML5 Local storage. In : *Kirupa.com* [en ligne]. Modifié le 05 août 2011. [http://www.kirupa.com/html5/html5\\_local\\_storage.htm](http://www.kirupa.com/html5/html5_local_storage.htm) (consulté le 19.08.2011)
- DELGADO, Ernest. HTML5 video. In : *HTML5 rocks* [en ligne]. Modifié le 3 août 2010. <http://www.html5rocks.com/en/tutorials/video/basics/> (consulté le 19.08.2011)
- FLEETING, James. Working with HTML5 localStorage(). In : *HTML 5 Tutorial : tutorials, tips and news for the designer* [en ligne]. Modifié en mai 2010. <http://html5tutorial.net/tutorials/working-with-html5-localstorage.html> (consulté le 19.08.2011)
- FOX, Pamela. *Flex vs. HTML5 for RIAs* [en ligne]. Ignite Sydney, 2009. <http://www.slideshare.net/wuzziwug/flex-vs-html5-for-rias-presentation> (consulté le 19.08.2011)
- FOX, Pamela. HTML5 vs Flex for rich internet applications. In : *Ignite : enlighten us, but make it quick* [en ligne]. 2011. <http://igniteshow.com/videos/html5-vs-flex-rich-internet-applications> (consulté le 19.08.2011)
- HTML5 video. In : *video JS : HTML5 video player* [en ligne]. 2010. <http://videojs.com/html5-video/> (consulté le 19.08.2011)
- HTML5CODE. *HTML4 vs HTML5* [en ligne]. 2011. <http://www.html5code.com/tutorials/html4-vs-html5/> (consulté le 27.07.2011)
- KAPPERT, Lars. Introduction to HTML5 web storage. In : *Six revisions : useful information for web developers & designers* [en ligne]. Modifié le 15 août 2011. <http://sixrevisions.com/html/introduction-web-storage/> (consulté le 19.08.2011)
- KER, Oliver. *HTML 5 doctor : helping you implement HTML 5 today* [en ligne]. 2011. <http://html5doctor.com/> (consulté le 03.08.2011)
- LAWSON, Bruce. HTML5 : hollow demos and forgetting the basics. In : *Archive for the HTML5 category* [en ligne]. Modifié le 5 août 2011. <http://www.brucelawson.co.uk/category/html5/> (consulté le 03.08.2011)
- LOOP, Infinite. Le stockage local HTML5 remplacera-t-il les cookies ?. In : *Code 18 : manuel du savoir-faire à l'usage des geeks et des curieux* [en ligne]. <http://code18.blogspot.com/2010/10/le-stockage-local-html5-remplacera-t-il.html> (consulté le 19.08.2011)
- MOREIRA, Miguel. Multivio : solution HTML5 pour visionner des documents. In : *epfl.ch* [en ligne]. Modifié le 31 août 2010. <http://ditwww.epfl.ch/SIC/SA/SPIP/Publications/spip.php?article2122> (consulté le 04.12.2011)
- MOW [pseudo]. Flash vs HTML5 : la fin du Flash ?. In : *Clubic.com : vous facilite le hi-tech* [en ligne]. Modifié le 18 juillet 2009. <http://www.clubic.com/forum/internet-general/flash-vs-html5-la-fin-du-flash-id645724-page1.html> (consulté le 19.08.2011)

PETER, Ian. History of the world wide web. In : *Nethistory* [en ligne]. Modifié en 2004. <http://www.nethistory.info/History%20of%20the%20Internet/web.html> (consulté le 18.07.2011)

ROWELL, Eric. 1.5.4 HTML5 Canvas circle tutorial. In : *HTML5CanvasTutorials* [en ligne]. Modifié le 16 décembre 2010. <http://www.html5canvastutorials.com/tutorials/html5-canvas-circles/> (consulté le 08.08.2011)

ROWELL, Eric. 1.8.1 HTML5 Canvas text tutorial. In : *HTML5CanvasTutorials* [en ligne]. Modifié le 2 décembre 2010. <http://www.html5canvastutorials.com/tutorials/html5-canvas-text/> (Consulté le 04.08.2011)

S.n. Canva. In : *Mozilla developer network* [en ligne]. 2011. <https://developer.mozilla.org/fr/HTML/Canvas> (consulté le 04.08.2011)

S.n. Enquire. In : *Wikipedia : the free encyclopedia* [en ligne]. Dernière modification le 25 août 2011 à 14h44. <http://en.wikipedia.org/wiki/ENQUIRE> (consulté le 18.07.2011)

S.n. Google Chrome OS. In : *Wikipedia : l'encyclopédie libre* [en ligne]. Dernière modification le 27 août 2011 à 02h03. [http://fr.wikipedia.org/wiki/Google\\_Chrome\\_OS](http://fr.wikipedia.org/wiki/Google_Chrome_OS) (consulté le 25.07.2011)

S.n. Google Chrome. In : *Wikipedia : l'encyclopédie libre* [en ligne]. Dernière modification le 08 septembre 2011 à 13h56. [http://fr.wikipedia.org/wiki/Google\\_Chrome](http://fr.wikipedia.org/wiki/Google_Chrome) (consulté le 25.07.2011)

S.n. History of the world wide web. In : *Wikipedia : the free encyclopedia* [en ligne]. Dernière modification le 26 août 2011 à 10h05. [http://en.wikipedia.org/wiki/History\\_of\\_the\\_World\\_Wide\\_Web](http://en.wikipedia.org/wiki/History_of_the_World_Wide_Web) (consulté le 18.07.2011)

S.n. HTML5 Audio. In : *the expressive web : beta* [en ligne]. 2011. <http://beta.theexpressiveweb.com/#!/html5-audio> (consulté le 03.08.2011)

S.n. HTML5 Canvas. In : *W3schools.com* [en ligne]. 1999-2011. [http://www.w3schools.com/html5/html5\\_canvas.asp](http://www.w3schools.com/html5/html5_canvas.asp) (consulté le 08.08.2011)

S.n. HTML5 Web Storage. In : *W3schools.com* [en ligne]. 1999-2011. [http://www.w3schools.com/html5/html5\\_webstorage.asp](http://www.w3schools.com/html5/html5_webstorage.asp) (consulté le 19.08.2011)

S.n. HTML5. In : *W3C Working Draft* [en ligne]. 2011. <http://www.w3.org/TR/html5/forms.html> (consulté le 27.07.2011)

S.n. Langage de balisage. In : *Wikipedia : l'encyclopédie libre* [en ligne]. Dernière modification le 25 août 2011 à 15h14. [http://fr.wikipedia.org/wiki/Langage\\_de\\_balisage](http://fr.wikipedia.org/wiki/Langage_de_balisage) (consulté le 19.07.2011)

S.n. Le stockage local de données en HTML5. In. *Mistra formation : formations en informatique* [en ligne]. Modifié le 24 août 2011. <http://www.mistra.fr/tutoriel-html5-stockage-local-key-storage.html> (consulté le 19.08.2011)

S.n. Microdata (HTML5). In : *Wikipedia : the free encyclopedia* [en ligne]. Dernière modification le 24 août 2011 à 08h 34. [http://en.wikipedia.org/wiki/Microdata\\_%28HTML5%29](http://en.wikipedia.org/wiki/Microdata_%28HTML5%29) (consulté le 25.07.2011)

S.n. Mosaic (web browser). In : *Wikipedia : the free encyclopedia* [en ligne]. Dernière modification le 29 juillet 2011 à 20h21. [http://en.wikipedia.org/wiki/Mosaic\\_%28web\\_browser%29](http://en.wikipedia.org/wiki/Mosaic_%28web_browser%29) (consulté le 03.08.2011)



S.n. Multiple Attribute. In : *Leakbali : w3 tutorial, web tutorial* [en ligne]. 2006-2011. <http://www.leakbali.com/T-html5/form-attributes/83/407/multiple-attribute/> (consulté le 03.08.2011)

S.n. Quirks mode. In : *Wikipedia : the free encyclopedia* [en ligne]. Dernière modification le 10 avril 2011 à 06h07. [http://en.wikipedia.org/wiki/Quirks\\_mode](http://en.wikipedia.org/wiki/Quirks_mode) (consulté le 27.07.2011)

S.n. Robert Cailliau. In *Wikipedia : the free encyclopedia* [en ligne]. Dernière modification le 09.08.2011 à 19h14. [http://en.wikipedia.org/wiki/Robert\\_Cailliau](http://en.wikipedia.org/wiki/Robert_Cailliau) (consulté le 18.07.2011)

S.n. Tim Berners-Lee. In : *Wikipedia : the free encyclopedia* [en ligne]. Dernière modification le 08 septembre 2011 à 16h51. [http://en.wikipedia.org/wiki/Tim\\_Berners-Lee](http://en.wikipedia.org/wiki/Tim_Berners-Lee) (consulté le 18.07.2011)

S.n. Webkit. In : *Wikipedia : l'encyclopédie libre* [en ligne]. Dernière modification le 08 septembre 2011 à 14h14. <http://fr.wikipedia.org/wiki/WebKit> (consulté le 25.07.2011)

S.n. *When can I use... : compatibility tables for support of HTML5, CSS3, SVG and more in desktop and mobile browsers* [en ligne]. Modifié le 29 août 2011. <http://caniuse.com/> (consulté le 03.08.2011)

S.n. World wide web. In : *Wikipedia : l'encyclopédie libre* [en ligne]. Dernière modification le 23 août 2011 à 21h42. [http://fr.wikipedia.org/wiki/World\\_Wide\\_Web](http://fr.wikipedia.org/wiki/World_Wide_Web) (consulté le 18.07.2011)

S.n. World wide web. In : *Wikipedia : the free encyclopedia* [en ligne]. Dernière modification le 5 septembre 2011 à 16h20. [http://en.wikipedia.org/wiki/World\\_Wide\\_Web](http://en.wikipedia.org/wiki/World_Wide_Web) (consulté le 18.07.2011)

SHOANNON, Ross. The history of HTML. In : *HTML source : HTML tutorials* [en ligne]. Modifié le 01 août 2010 <http://www.yourhtmlsource.com/starthere/historyofhtml.html> (consulté le 19.07.2011)

SIGHT. *The HTML 5 Test : how well does your browser support HTML 5 ?* [en ligne]. 21 août 2011. <http://beta.html5test.com/index.html> (consulté le 25.07.2011)

SUCAN, Mihai. HTML5 canvas : the basics. In : *Dev.Opera : follow the standards, break the rules* [en ligne]. Modifié le 08 janvier 2009. <http://dev.opera.com/articles/view/html-5-canvas-the-basics/> (consulté le 04.08.2011)

VALDES, Ray. HTML5 and the future of Adobe Flash. In : *Gartner* [en ligne]. Modifié le 10 février 2010. [http://blogs.gartner.com/ray\\_valdes/2010/02/10/html5-and-flash/](http://blogs.gartner.com/ray_valdes/2010/02/10/html5-and-flash/) (consulté le 19.08.2011)

WILSON, Brian. HTML Overview : Intex DOT HTML/CSS. In : *Blooberry.com* [en ligne]. [Sans date]. <http://blooberry.com/indexdot/history/html.htm> (consulté le 19.07.2011)

YE, Wayne. HTML5 web storage in essence. In : *The code project : your development resource* [en ligne]. Modifié le 28 février 2011. <http://www.codeproject.com/KB/HTML/Web-Storage-In-Essence.aspx> (consulté le 19.08.2011)

## Bibliographie

FULTON, Steve, FULTON, Jeff. *HTML5 canvas : native interactivity and animation for the web*. Beijing : O'Reilly, 2011. 628 p.

GASSNER, David. *Flash Builder 4 and Flex 4 bible*. Indianapolis, Ind. : Wiley Publ., 2010. 1028 p.

KEITH, Jeremy. *HTML5 pour les web designers*. Paris : Eyrolles, 2010. 87 p. (A book apart ; no 1).

LAWSON, Bruc, SHARP, Remy. *Introducing HTML5*. Eighth : New Riders, 2011. 223 p.

LUBBERS, Peter, ALBERS, Brian, SALIM, Frank. *Pro HTML5 Programming : powerful APIs for Richer Internet Application Development*. New York : Apress, 2010. 284 p.

MARK, Pilgrim. *HTML5 : up and running*. Sebastopol, Calif. : O'Reilly, 2010. 205 p.

MEYER, Jeanine. *The essential guide to HTML5 : using games to learn HTML5 and JavaScript*. [S.l.] : Friends of, 2010. 359 p.

MICHEL, Martin. *HTML5 et CSS3 : [l'essentiel des pratiques actuelles]*. Paris : Pearson Education, 2011. 369 p. (Le guide de survie).

S.n. *Flex & [and] Air : [utilisez la technologie Flash pour développer vos applications Internet]*. Paris : Micro Application, 2008. 415 p. (Webmasters).

VAN LANCKER, Luc. *HTML5 et CSS3 : maîtrisez les standards des applications Web*. Nantes : ENI, 2011. 417 p. (Ressources informatiques. Informatique technique).

VINCENT, Jean-Pierre, VERRECCHIA, Jonathan. *HTML5 : de la page web à l'application web*. Paris : Dunod, 2011. 238 p. (InfoPro. Etudes, développement, intégration).

# Annexe 1

## Installation plateforme Web

### Installation plateforme Web

#### Installation WAMP (PC)

WampServer est une plateforme de développement Web sous Windows. Il vous permet de développer des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL.

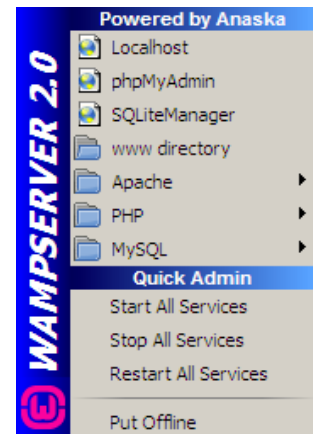


Faite un double clic sur le fichier d'installation et laissez-vous guider. Tout est géré par l'installateur de WampServer.

Une fois installé et lancé le WampServer, l'icône de wamp apparaîtra dans les outils de taskbar menu. En faisant un clic droit sur l'icône dans les outils, vous faites l'appel à la fenêtre du WampServer.

En suite il faut cliquer sur « *www directory* » pour accéder à l'emplacement où il faut placer le fichier contenant le site Web.

Pour utiliser la plateforme il suffit d'indiquer dans la barre URL du navigateur « <http://localhost/HTML> ». De cette manière, vous utilisez la plateforme comme si elle se trouvait sur un serveur.



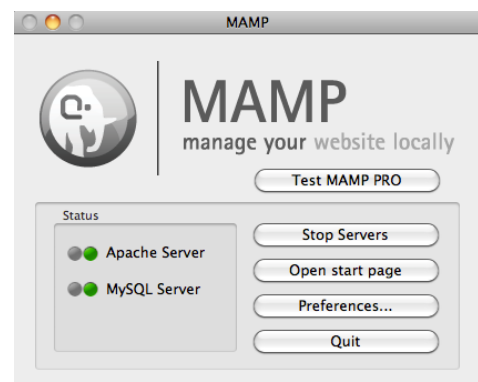
#### Installation MAMP (MAC)



Pour installer le MAMP sur un Mac, il suffit de faire glisser le logiciel MAMP fourni sur le cd dans le répertoire « Applications » se trouvant sur votre ordinateur.

Ensuite il faut se diriger jusqu'au dossier « *htdocs* », qui se trouve dans l'application MAMP et il faut copier le fichier contenant le site Web.

Par la suite vous pouvez démarrer le logiciel MAMP. Une fois le MAMP démarré il suffit d'indiquer dans la barre URL du navigateur « <http://localhost/HTML> ».



## **Annexe 2**

### **Géolocalisation (3.4)**

#### **Global Positioning System (GPS)**

Le GPS fonctionne grâce aux satellites. Le but principal des satellites est de transmettre les informations en permanence, pour que le récepteur puisse les analyser. Une fois le message reçu, le récepteur calcule sa propre position en chronométrant les signaux envoyés par l'un des satellites. En déterminant le temps de chaque envoi de message, le récepteur obtient la distance de chaque satellite et de leurs positions dans l'orbite. Selon la disposition des satellites le récepteur obtient son emplacement. Pour effectuer sa localisation, le récepteur peut communiquer avec trois satellites, selon la méthode mathématique de trilatération. Mais la plupart du temps, le récepteur utilise plus de quatre satellites pour calculer sa position, car l'envoi d'un message entre trois satellites peut créer des erreurs de positionnement, à cause du temps que le message prend pour parcourir la distance entre le satellite et le récepteur.

#### **Adresse IP**

Une adresse IP est un numéro unique attribué à toutes les machines connectées sur un réseau informatique utilisant l'Internet Protocol. Ce numéro est composé de 12 chiffres et il peut être statique ou dynamique. Chaque adresse IP appartient à un bloc relié à une région. Ces blocs sont contrôlés par une institution du registre local. Pour cette raison, il est facile d'identifier une adresse IP selon son emplacement. De plus, avec l'avancement de la technologie, la situation géographique d'un appareil, ayant une adresse IP, peut être calculé plus précisément.

#### **GSM/3G Cell IDs**

Global System for Mobile Communications est la plus ancienne des technologies de téléphone mobile. GSM est une technologie 2G qui est disponible dans la plupart des pays. Tout dernièrement cette technologie a commencé à migrer vers les services 3G et 4G. En fait, il s'agit d'un réseau cellulaire, qui permet aux téléphones mobiles de communiquer entre eux. Chaque appareil mobile dispose d'un numéro unique qui représente l'identifiant de cet appareil. Contrairement au GPS, cette technologie ne se base que sur la triangulation. En mesurant la distance entre les émetteurs, dont les positions sont connues, et le récepteur, il est donc facile de déterminer la position de

l'appareil. C'est la raison pour laquelle cette technologie fonctionne mieux dans les environnements urbains.

### **WiFi**

La technologie WiFi fonctionne comme les adresses IP, sauf que les appareils contenant les périphériques WiFi peuvent se déplacer. Pour cette raison la technologie WiFi est basée sur l'adresse MAC. Le Media Access Control (MAC) est un identifiant physique stocké dans une interface réseau d'appareil sans fil. En utilisant cette adresse de la même manière qu'une adresse IP, un lieu physique, selon une latitude et de longitude, peut être obtenu.

### **Puce RFID**

Les puces Radio Frequency Identification sont des petits objets électroniques munies d'une antenne qui peuvent être collés ou incorporés dans un objet, un produit ou un organisme vivant. Les émetteurs de radiofréquence activent les puces à courte distance, en leur fournissant l'énergie dont ceux-ci ont besoin. Une fois la puce activée, elle transmet des informations diverses. De cette manière l'emplacement d'une puce peut être localisée, à une courte distance de 200 m.

## Annexe 3

### Les éléments non supportés par les navigateurs

#### WebGL

WebGL est une API pour les graphismes 3D sur le Web. En fait, il s'agit d'une bibliothèque logicielle qui étend la capacité du JavaScript afin de lui permettre de générer des graphiques 3D.

Donc comme expliqué dans le chapitre de « Canvas », il est possible d'obtenir les dessins 2D en appelant **getContext("2D")** sur l'élément `canvas`. Le but de HTML5 est d'utiliser également l'élément `canvas`, mais à travers un context 3D. Pour y arriver, `canvas` passera par WebGL qui gère les textures et les « *vertex buffers* », au lieu d'évoquer les méthodes à dessiner des lignes et des formes de remplissage.

#### Notifications

Notifications API permet l'affichage des notifications à l'utilisateur pour des événements donnés. Appelé autrement « desktop notification », il a donc la possibilité de notifier l'utilisateur, quel que soit l'onglet du navigateur sur lequel est le focus, d'un nouveau mail, tweets ou un événement du calendrier.

#### Microdata

Microdata est un moyen de spécifier le contenu. Le contenu sert à décrire un type d'information. Chaque type d'information représente un élément spécifique.

Autrement dit, microdata est une soutenance pour le vocabulaire qui aide à décrire un élément dans une page Web. Le fait d'intégrer le microdata dans une page Web aide le moteur de recherche à mieux comprendre quelles informations la page contient. De cette manière, il est plus facile d'annoter les éléments HTML avec des balises lisibles par machine que d'utiliser les approches similaires comme RDFa ou les microformats.

#### File

File est un moyen pour interagir avec des fichiers locaux, via la spécification File API. Le but est d'obtenir les informations que le fichier contient sans avoir l'obligation de l'envoyer au serveur. Donc le contenu sera traité localement, ce qui diminue la bande passante et la charge de serveur.