

## HTML ET CSS

Avant de faire le cours, un **TP** sur ces deux langages a été fait en salle informatique ([snt.mathemathieu.fr](http://snt.mathemathieu.fr))

Pages 34/35 : docs 1, 3, 4, 5 → rappel de ce qui a été vu en TP

## PROGRAMME

Sur le *Web*, les textes, photos, vidéos, graphiques, sons, programmes sont exprimés et assemblés dans divers formats normalisés par un consortium mondial (W3C : *World Wide Web Consortium*), ce qui permet une circulation standardisée de ces informations.

Les pages *Web* sont écrites dans le **langage de balises HTML** (*HypertextMarkupLanguage*). Leur style graphique est défini dans le **langage CSS** (*Cascading Style Sheets*).

Les pages ont une adresse unique, nommée **URL** (*Uniform Ressource Locator*). Elles sont accessibles via internet en utilisant le protocole HTTP (*Hypertext Transfer Protocol*) ou sa version sécurisée HTTPS qui crypte les échanges. L'affichage des pages est réalisé chez l'utilisateur par un programme appelé navigateur.

Un hypertexte est un texte augmenté de renvois automatiques à des textes, des images ou des sons. Initialement, un hypertexte se restreignait à la mémoire d'un seul ordinateur. Dans une page *Web*, ce renvoi se fait sur n'importe quelle machine du réseau internet, par le truchement de l'adresse de la page *Web* du texte (URL) auquel il fait référence. La toile d'araignée construite par les liens peut être représentée sous la forme d'un graphe qui matérialise la structure du *Web*.

## LANGAGES CÔTÉ CLIENT / CÔTÉ SERVEUR

Il y a quelques années, le web était dit « statique » : le concepteur de site web écrivait son code HTML et ce code était simplement envoyé par le serveur web au client. Les personnes qui consultaient le site avaient toutes le droit à la même page, le web était purement « consultatif ».

Les choses ont ensuite évolué : les serveurs sont aujourd'hui capables de générer eux-mêmes du code HTML. Les résultats qui s'afficheront à l'écran dépendront donc des demandes effectuées par l'utilisateur du site : **le web est devenu dynamique**.

Différents langages de programmation peuvent être utilisés « côté serveur » afin de permettre au serveur de générer lui-même le code HTML à envoyer. Le plus utilisé encore aujourd'hui se nomme PHP.

D'autres langages sont utilisables côté serveur (pour permettre la génération dynamique de code HTML) : Java, Python...

Voici un exemple très simple de code en PHP :

```
<?php
$heure = date("H:i") ;
echo '<h1>Bienvenue sur mon site</h1>
      <p>Il est ', $heure, '</p>';
?>
```

Sans entrer dans les détails, si un client se connecte à un serveur web qui exécute ce code à 18h23, le serveur enverra au client le code HTML ci-dessous :

```
<h1>Bienvenue sur mon site</h1>
<p>Il est 18h23</p>
```

## JavaScript dangereux ?

JavaScript est un langage de programmation exécuté dans le navigateur pour apporter du dynamisme dans les pages web qu'on visite et plein d'autres choses.

Il s'exécute donc le plus souvent dans le navigateur, c'est-à-dire **le client**. On dit alors que JavaScript est un langage de programmation côté client.

Voici quelques exemples de cas où JavaScript est efficace (voire irremplaçable) :

- vérification de saisie dans des formulaires (adresse, email)
- calculs simples suite à des saisies de formulaire (tarifs, calculatrice)
- gestion des cookies (sauvegarde d'information : nombre de visites, caddie virtuel)
- gestion de la navigation (menus dynamique, popup)
- jeux...

JavaScript a été conçu pour limiter (presque) complètement les risques pour le visiteur. En ce sens, on peut dire que **exécuter un JavaScript sur son navigateur ne comporte aucun risque pour les données et le matériel**. JavaScript ne permet pas d'écrire ou de lire un fichier sur le disque dur du visiteur ou sur le serveur. Ainsi, vous ne risquez pas de voir des données confidentielles récupérées à votre insu. De même, il n'est pas possible de lister les logiciels copiés et non enregistrés. Impossible aussi de détruire des fichiers indispensables ou d'introduire un virus sur le poste d'un visiteur.

**MAIS JavaScript permet quand même d'écrire et de lire le disque dur dans la zone réservée aux cookies**. C'est la seule interaction que JavaScript peut avoir avec votre disque dur.

Certains me diront que JavaScript possède des fonctions pour lire des fichiers texte. En fait, **JavaScript permet d'intégrer des applets Java ou ActiveX qui elles sont capables d'agir sur le disque dur**. Il ne s'agit pourtant pas de JavaScript pur : si vous désactivez Java et ActiveX, rien ne se passera.

JavaScript seul ne peut pas non plus récupérer l'adresse IP d'un visiteur. Il est indispensable également de passer par des appels à Java ou à ActiveX, ou plus simplement par un appel Ajax sur le serveur web.

En réalité, **JavaScript est presque sans risque** : il est possible de coder un JavaScript qui ouvre des popups jusqu'à épuiser complètement les ressources système. Dans ce cas, la seule solution est de redémarrer votre ordinateur. On peut aussi imaginer un JavaScript qui tourne en tâche de fond d'une page web, occupant ainsi des ressources et réduisant les performances de votre poste.

Mais les navigateurs sont maintenant bien protégés contre ce genre d'attaque sommaire.

Les performances de JavaScript en termes de sécurité pour les visiteurs font que JavaScript est accepté par (presque) tous. Le prix de cette sécurité et de cette quasi-universalité se traduit par des limites techniques : il ne permet pas de lire et d'écrire sur le disque dur du visiteur (en dehors des cookies) et du serveur.

Il est donc impossible en JavaScript de générer par exemple un fichier de connexions sur le serveur, ou un compteur de visites sur une page donnée.

De même, il ne peut pas s'interfacer avec une base de données : impossible alors de concevoir un forum de discussion, un script de sondage ou de vote, dont les données sont stockées, soit dans un fichier, soit en base sur le serveur. Impossible aussi de lier un catalogue en base sur le serveur à votre boutique en ligne.

Il existe une multitude de technologies actuellement qui permettent de combler les lacunes de JavaScript. On peut les séparer en 2 catégories :

- celles dont le code est exécuté **côté client (Flash, Java...)** : en général, ils nécessitent un plugin et ne sont donc pas compatibles avec tous les navigateurs. La sécurité n'est pas assurée : plantages fréquents, accès au disque, failles de sécurité, etc.
- celles dont le code est exécuté **côté serveur** qui renvoie une page au format HTML (**PHP**, etc.).

Page 34 doc. 2 : **qu'est-ce qu'un CMS ?**

Un système de gestion de contenu ou SGC (*content management system* ou CMS en anglais) est une famille de logiciels destinés à la conception et à la mise à jour dynamique de sites Web ou d'applications multimédia.

Mon site [www.mathemathieu.fr](http://www.mathemathieu.fr) utilise [Joomla !](#).

Principaux logiciels connus :

- Wordpress (blog)
- Joomla (blogs / sites)
- eZ Publish (idem)
- phpBB (forum), etc.

**En classe** : présentation de mon site [www.mathemathieu.fr](http://www.mathemathieu.fr) avec l'interface administrateur.