

## Visite d'une page

La visite d'une page Web fait intervenir trois personnages :

- Le visiteur, qui regarde la page sur son ordinateur
- Le Webmaster du site visité, en fait plusieurs personnes et plusieurs fonctions :
  - Les auteurs qui contribuent à la création de la page : auteurs des textes, metteurs en page, graphistes qui définissent le style du site, programmeurs qui gèrent les coulisses du site etc. Ils sont sous la direction du Webmaster proprement dit ;
  - Le propriétaire du site, le plus souvent une entreprise ou une association. Il joue le rôle juridique important de responsable du contenu du site. Si l'on voulait comparer avec un journal, on dirait que le Webmaster est le rédacteur en chef tandis que le propriétaire est le directeur de la publication.
- Il y a en plus un intermédiaire indispensable, l'hébergeur. C'est une entreprise qui possède de très puissants ordinateurs reliés au réseau mondial. Elle met à disposition de ses clients propriétaires de sites un quota de place sur disque renfermant les informations constituant chaque site. L'hébergement est normalement payant, en fonction de la place occupée et des services annexes demandés. Il y a aussi, pour les amateurs qui veulent avoir un site personnel, des hébergeurs gratuits à condition d'accepter quelques publicités intercalées dans le site. En France, les hébergeurs supportent une part de la responsabilité juridique du site. Quelquefois, certains propriétaires sont leur propre hébergeur : c'est le cas de certaines grosses entreprises ou certaines institutions officielles (universités etc.).

La visite de sites sur Internet se fait grâce à un logiciel spécial appelé navigateur (en anglais *browser*, terme qui met en évidence le fait que les visiteurs passent souvent d'un site à un autre d'une façon qui semble aléatoire ; on a proposé la traduction « butineur »). Tout site est formé d'une ou plusieurs pages (on dit pages Web pour ne pas confondre avec la notion de page écran). La page Web est l'unité de visite : lorsqu'on est en train de visualiser une page Web, on ne la voit pas forcément entièrement pour des raisons de taille écran, mais on peut rendre visibles les parties invisibles par simple défilement de l'écran sans que le navigateur ait à émettre une requête sur le réseau.

La visite se fait en deux étapes :

- téléchargement : lorsqu'une page Web est demandée, le navigateur émet une requête sur le réseau. En réponse, il reçoit un fichier qu'il copie sur le disque dur local (du visiteur) dans une zone appelée cache internet.
- visualisation : ensuite, le navigateur interprète ce fichier pour produire la visualisation de la page sur écran.

Cette dualité a deux avantages :

- elle permet au navigateur d'interpréter la page et de gérer la visualisation sans être gêné par d'éventuels problèmes de transmission : ceux-ci seront apparus pendant la première phase, éventuellement terminée par un constat d'impossibilité d'obtenir la page.
- si l'on demande une page qu'on a visitée il y a peu de temps, il y a des chances que son fichier soit encore dans le cache et on peut donc la visualiser sans nouvelle transmission, ce qui gagne du temps.

Mais ce qui est surtout à retenir de ceci est que **une page Web est un fichier** et, de plus, un fichier texte, donc lisible et constructible avec un simple logiciel de traitement de textes rudimentaire comme le Bloc-notes.

Élément qui va encore plus loin, il y a parfaite équivalence pour le navigateur entre le fichier page Web qu'il obtient depuis le réseau et un fichier identique qui serait sur un disque local. Pour la bonne raison que le fichier venant du réseau est d'abord copié dans le cache local ; c'est donc un fichier local que le navigateur interprète. Il en résulte que si, au lieu de taper dans la zone dite adresse du navigateur, la désignation Internet d'une page Web (cela s'appelle une adresse URL, nous verrons plus loin comment elles sont construites), l'utilisateur tapait une désignation de fichier local (sous la forme `disque:répertoire\nom`), il obtiendrait tout autant la visualisation de la page Web, à condition, bien sûr, que le fichier ait bien la structure convenable. Le principal but de ce livre est d'apprendre cette structure.

Une autre façon d'obtenir la visualisation d'un tel fichier est de faire glisser vers la fenêtre du navigateur, l'icône de ce fichier telle qu'on la trouve dans la fenêtre de visualisation du répertoire qui le contient fournie par n'importe quel logiciel explorateur de disques.

Une conséquence fondamentale de cette propriété est que vous pouvez vérifier le fonctionnement des pages Web que vous construisez en local chez vous avant d'envoyer les fichiers chez votre hébergeur (on dit « mettre en ligne les fichiers »). La plupart des Webmasters évitent de mettre en ligne des fichiers sans une telle vérification : il faut assurer l'exactitude des informations fournies et le fonctionnement de la page et ne pas oublier qu'une faute de frappe affectant même un seul caractère peut mettre une page sens-dessus-dessous, ce qui ne manquera pas de faire tordre de rire à vos dépens les alligators de Floride et les kangourous d'Australie (si, si : ils ont accès à Internet !) si vous la mettez en ligne sans vérifier.

### ***Désignation des fichiers***

Si votre site a une certaine complexité, vous allez organiser les différents fichiers qui le constituent en plusieurs répertoires et sous-répertoires. Ainsi un de ces fichiers aura chez vous la désignation : `disque:répertoire_principal_de_votre_site\sous-répertoire\nom_fichier.extension` (nous avons utilisé le séparateur \ valable pour DOS et Windows ; sous Unix/Linux vous auriez / et : sur Mac). Chez votre hébergeur, ce sera :

*disque:arborescence.../.../répertoire\_principal\_attribué\_à\_votre\_site/sous-répertoire/nom\_fichier.extension* (nous avons ici le séparateur / car les hébergeurs sont souvent sous Unix).

Le navigateur utilise une autre désignation de votre fichier Web qu'on appelle une URL (Universal Ressource Locator), de la forme : *protocole\_site/sous-répertoire/nom.extension* où *protocole-site* est `http://` pour un site Web normal (`http=HyperText Transfer Protocol`). Le Web utilise d'autres protocoles ou pseudo-protocoles :

<code>https://</code>	http sécurisé
<code>ftp://</code>	transferts de fichiers (File Transfer Protocol)
<code>news:</code>	connexion à des forums
<code>telnet:</code>	votre ordinateur simule un terminal sur une machine éloignée
<code>mailto:</code>	envoi d'un courrier électronique
<code>javascript:</code>	exécution d'une instruction JavaScript (nous y reviendrons beaucoup).
<code>file://</code>	fichier sur le disque local. En fait, il suffit de donner la désignation du fichier, le navigateur comprendra qu'il est local.

La désignation du site implique de déterminer le serveur (l'ordinateur de l'hébergeur), et le site voulu parmi les sites qui sont chez cet hébergeur. Le serveur est théoriquement défini par ce qu'on appelle son adresse IP, groupe de chiffres du genre 194.214.26.91. Il est évidemment plus facile pour les visiteurs d'utiliser un nom symbolique qui rappelle le contenu du site, par exemple `www.louvre.fr`. Alors, comment trouve-t-on le site ? Il y a sur le réseau des serveurs spéciaux, connus des navigateurs, qui fournissent la correspondance nom du site – localisation. Ces serveurs s'appellent des DNS (Domain Name Server).

Donc, le navigateur ayant besoin d'un fichier, il émet d'abord une requête vers un DNS pour localiser l'hébergeur. Il lui envoie alors sa requête de fichier. Le serveur de l'hébergeur reconnaît le site voulu parmi ses clients et il se positionne sur le répertoire principal attribué à ce site. Ensuite, s'il y a des sous-répertoires, il n'y a plus qu'à descendre l'arborescence et fournir le fichier. Au passage, les \ ou autres séparateurs sont transformés en /.

### ***Types de fichiers et extensions***

Le type de fichier le plus souvent demandé est une page Web normale (que nous allons apprendre à construire). Un tel fichier a, au choix, l'extension `.htm` ou `.html`. Le Webmaster est entièrement maître de décider de son choix, mais le visiteur doit s'y conformer : s'il demande `toto.html` alors que le fichier implanté par le Webmaster est `toto.htm`, il aura le message « Impossible d'afficher la page ». Le Webmaster doit lui aussi se conformer à ce choix lorsqu'il implante un lien, y compris pour les liens aux autres pages de son propre site. Notre conseil est le suivant : **choisissez une fois pour toutes `.htm`** : cela fait un caractère de moins à taper et vous n'aurez

plus d'hésitation au moment d'implanter un lien vers une page créée par vous ou par quelqu'un de votre équipe. La question reste pour les pages « étrangères ».

Des pages Web peuvent se référer à d'autres types de fichiers, d'extensions :

- .gif, .jpg : fichiers images ; normalement, ces deux types sont les seuls fichiers images visualisables sans logiciels de compléments par les navigateurs courants. Il peut y avoir aussi les fichiers .bmp, moins usités car plus encombrants. Un nouveau type, d'extension .png est en train de se développer.
- .css : fichiers de styles.
- .js : fichiers de scripts JavaScript externes.
- .pdf : fichiers documents pdf . Il faut qu'un logiciel de lecture de pdf soit installé sur l'ordinateur du visiteur. Même problème pour d'autres documents ; si un logiciel convenable n'est pas installé, la plupart des navigateurs proposent de télécharger le document sans le visualiser.
- .asp, .php et autres : un lien peut demander l'exécution d'un programme soit en local, soit sur le serveur ; le fonctionnement est tributaire de l'existence du logiciel voulu chez le visiteur ou l'hébergeur. Beaucoup d'hébergeurs (surtout gratuits) n'autorisent pas de telles possibilités.

### ***Majuscules et minuscules***

Beaucoup de serveurs fonctionnent sous Unix / Linux. Dans ces systèmes, les majuscules et minuscules ne sont pas équivalentes dans une désignation de fichier : si vous demandez toto.htm alors que le fichier est Toto.htm, la recherche ne réussira pas et vous aurez « Impossible d'afficher la page ». D'où notre conseil : **n'utilisez que des minuscules** dans les noms de répertoires, les noms de fichiers, les extensions. Bien sûr, ce conseil n'évite les hésitations que pour votre propre site ; pour les sites étrangers, vous devez vous conformer à ce qui existe.

### ***Et si on ne fournit pas de nom de fichier ?***

En fait, on peut appeler un site sans préciser de nom de fichier ; cela revient à demander la page d'accueil du site grâce au mécanisme suivant : ne pas préciser de nom de fichier revient à appeler le répertoire principal du site et si, sur ce répertoire, il y a un fichier de nom `index.htm`, `index.html`, `default.htm` etc. (la liste des fichiers possibles forme une option du logiciel de gestion du serveur et elle est spécifiée par l'administrateur de l'hébergeur), alors c'est ce fichier qui est fourni. Moralité, fournissez votre page d'accueil sous le nom `index.htm`.

Si aucun fichier du genre n'est fourni, alors le navigateur affiche la liste des fichiers présents, ce qui est très gênant, car cela facilite la vie des pirates. Une option peut l'empêcher, mais il n'est pas sûr que l'hébergeur l'active. Par conséquent, il est, à notre avis, **impératif, que vous fournissiez un fichier `index.htm`** (éventuellement page vide) dans tous les répertoires et sous-répertoires de votre site.

# Connaissances et outils nécessaires

## **Connaissances**

Nous faisons abstraction dans ce qui suit des connaissances qui concourent à décider de ce qu'on met dans le site Web en construction. Ces connaissances sont l'apanage des stylistes, des graphistes et des stratèges en marketing. Nous donnerons quelques brefs conseils qui empiètent sur ce domaine, mais ce livre est centré sur les aspects techniques et informatiques.

### *Langage HTML*

Pour développer un site Web, il faut au moins connaître un minimum du langage HTML. Bien qu'il existe certains logiciels qui sont censés en dispenser, cette connaissance est en fait incontournable si vous voulez développer un site un tant soit peu élaboré : il faut pouvoir juger et modifier ce qu'un tel logiciel aura généré.

### *Langages de programmation*

Des éléments de programmation interviennent à plusieurs niveaux :

- On incorpore au fichier HTML des « scripts », c'est-à-dire des petits programmes pour créer des animations, gérer les formulaires etc. Il y a plusieurs langages possibles pour cela ; le plus répandu est JavaScript, traité dans ce livre. Son concurrent est VBScript.
- Certains liens peuvent faire exécuter un programme sur le serveur : on les appelle des scripts CGI. Un langage souvent employé est C si le serveur est sous Unix. Mais on a introduit de nouveaux langages comme Pearl ou PHP. PHP associé à MySQL est maintenant très utilisé pour manipuler des bases de données et faire de la mise à jour en ligne du site. Un système récemment développé, appelé SPIP, facilite la construction et la mise à jour d'un site en ligne : il est basé sur PHP/MySQL.
- On peut aussi faire exécuter certains programmes sur le serveur à l'aide d'ASP, mais tous les serveurs ne le permettent pas.

### *Hébergement*

Une fois préparé votre site, il faut le mettre en ligne, donc choisir un hébergeur. Nous présenterons quelques brefs éléments sur ce point en fin d'ouvrage.

## **Outils**

Les outils à mettre en œuvre sont les suivants :

### *Pour HTML*

Le seul outil à mettre en œuvre est un simple éditeur de texte, ou traitement de texte simple. Le Bloc-notes livré avec Windows (si vous utilisez un PC, sur MAC, il y

a BBEdit) peut suffire, mais il n'est pas très ergonomique. Il y a beaucoup de logiciels de traitements de textes très simples qu'on peut télécharger gratuitement ; un choix est proposé sur [www.winsite.com](http://www.winsite.com) ou [www.anshare.com](http://www.anshare.com). Pour notre part, nous utilisons PFE (Programmer's File Editor, proposé sur <http://telecharger.caloga.com>) : un de ses avantages est de toujours conserver la précédente version de chaque fichier, ce qui peut être prudent.

En fait, pour trouver où télécharger un logiciel, il suffit de chercher "nom\_du\_logiciel telecharger" sur un moteur de recherche comme Google.

#### *Pour JavaScript*

Pas besoin d'outil : les navigateurs ont normalement leur interpréteur JavaScript.

#### *Pour les autres langages*

Pour les autres langages, il faut, pour une exécution chez le visiteur être sûr que les interpréteurs ou compilateurs soient installés ; pour une exécution sur le serveur, vous avez une meilleure maîtrise de la situation, puisqu'il n'y a à connaître que ce qu'il en est chez votre hébergeur : la vraie question est de savoir si celui-ci autorise des exécutions sur le serveur.

Pour des sites à mettre à jour en ligne, des systèmes ont été développés récemment notamment SPIP. Il est basé sur le couple PHP / MySQL mais il dispense de connaître à fond ces langages au prix d'une restriction sur la forme du site qui doit être structuré en rubriques et articles.

#### *Génération de HTML sans connaissances*

Il existe différents logiciels. Nous parlerons un peu de FrontPage, ou plutôt de sa version simplifiée FrontPage Express. Les navigateurs offrent souvent un logiciel de création de pages Web assez simples : c'est le cas de Netscape (Composer) ; Mozilla aussi. Star Office (concurrent Open Source de Microsoft Office) en a un aussi.

Word offre le moyen de créer des pages Web sans connaître HTML puisque, ayant créé un texte, vous pouvez demander sa sauvegarde en tant que fichier .htm et la présentation que vous aurez définie dans votre document se retrouvera dans la page Web créée. Le problème est que le fichier ainsi généré est très gros et très complexe, à un point tel que cette solution perd beaucoup de ses attraits.

De même, Excel permet de sauvegarder un classeur en tant que fichier .htm ; là encore, on obtient un résultat complexe et encombrant, mais moins qu'avec Word. Pour créer facilement un tableau, vous le créez sous Excel, vous sauvegardez en HTML et, dans le fichier créé, vous copiez ce qui va de <table..> à </table> et le collez dans votre page.

DreamWeaver de MacroMedia est très élaboré et perfectionné : on peut voir simultanément sur deux fenêtres le fichier HTML et l'effet produit par le navigateur... En outre, il gère la transmission des fichiers vers le serveur de façon immédiate.

Mais, comme nous l'avons déjà dit, il faut tout de même connaître un minimum de HTML (et donc lire ce livre).

### *Mise en ligne du site*

La mise en ligne du site consiste simplement à envoyer chez l'hébergeur les fichiers que vous aurez préparés. Il faut pour cela un logiciel de FTP, car c'est ce protocole qui est utilisé. Votre hébergeur vous aura attribué un numéro de compte et un mot de passe que vous spécifierez au logiciel FTP.

Rappelons à ce propos deux définitions. On appelle **Downloading** (téléchargement) le transfert d'un fichier du serveur vers le visiteur et **Uploading** (remontée) le transfert de données du visiteur vers le serveur. Le downloading nécessite des procédures de sécurité : il faut empêcher tout un chacun de lire certaines données stratégiques, mais il en faut encore plus pour l'uploading, car il serait tragique qu'un visiteur mal intentionné fasse entrer un virus dans le serveur.

C'est pourquoi le protocole HTTP permet le downloading : le visiteur peut lire les pages Web (certaines pages nécessitent un mot de passe), mais pas le uploading (seules très peu de données comme les réponses à un questionnaire peuvent remonter). Le protocole FTP permet les échanges de fichiers dans les deux sens puisqu'il faut fournir un mot de passe. Quelques sites (dits FTP anonymes) dispensent du compte et du mot de passe, mais ils ne permettent que le downloading.

Normalement, un logiciel de FTP est incorporé au système. Sinon, l'hébergeur vous en fournira un, et il y en a beaucoup en téléchargement gratuit, comme Lite WSFTP etc.

### ***Intranet***

Un site Intranet fonctionne exactement comme un site Internet. Il est formé de fichiers qui ont exactement la même forme et les mêmes propriétés, donc ce livre vous permettra aussi de créer un site Intranet pour votre entreprise. La seule différence est que les fichiers doivent

être implantés sur un serveur, et les visiteurs doivent utiliser des ordinateurs, qui appartiennent **physiquement** à un certain réseau (le réseau d'une entreprise ou même un sous-réseau) ; aucune visite n'est possible depuis un ordinateur extérieur à ce réseau.

Quelquefois, on simule un Intranet en créant un site visitable de partout sur le Web, mais en fournissant un mot de passe. Bien sûr, c'est une sécurité bien inférieure à la sécurité procurée par l'impossibilité physique qui caractérise un véritable Intranet.

### ***Les différents navigateurs***

Un problème épineux qui se pose aux Webmasters est qu'il existe sur le marché différents navigateurs, sous différentes versions (tout le monde n'a pas la dernière

version), fonctionnant sur différentes plateformes (Windows en majorité, mais il y a Unix / Linux et Mac) et ces différents navigateurs ne produisent pas forcément le même effet à partir d'un fichier Web donné.

Bien sûr, pour les pages les plus élémentaires, il y a normalement compatibilité. Les différences commencent à apparaître dès que l'on passe à des pages un peu élaborées, notamment si l'on cherche à produire des effets d'animation, ou si l'on fait appel à des programmes JavaScript : le JavaScript des différents navigateurs ne gère pas forcément les mêmes objets ou les mêmes événements, ou, pour un même objet, cela peut être sous des noms différents.

Évidemment, il serait très fastidieux de devoir écrire une version de chaque page pour chaque navigateur, sans compter que, pour un site multilingue, le nombre de pages à créer est encore multiplié ; de quoi refroidir les vocations de Webmasters !

Le problème ne se pose en fait pas dans les mêmes termes selon le type de site à créer : un site de vente en ligne ne peut se permettre de négliger aucun client potentiel, donc doit fonctionner sous toutes plateformes, tous navigateurs et toutes versions pas trop anciennes, ce qui oblige généralement à faire exécuter un maximum de choses sur le serveur, car on en a le contrôle.

D'autres sites ont moins d'impératifs et fonctionnent sur un nombre restreint de navigateurs. Certains sites affichent en entrée un message du genre « Ce site est optimisé pour le navigateur x, version y », ce qui peut être frustrant pour certains visiteurs, un peu comme si, arrivant en 2 CV sur une autoroute, un panneau vous disait que vous en profiteriez mieux si vous aviez une Rolls. Sauf que les conséquences financières sont moindres vu que les particuliers peuvent toujours télécharger gratuitement la dernière version de tous les navigateurs ; encore faut-il que l'ordinateur soit assez récent pour accepter cette version.

De fait, avec la progression exponentielle du marché, la majorité des gens ont une version récente. Par ailleurs, le marché est largement dominé par Microsoft Internet Explorer. Les suivants sont Mozilla (successeur de Netscape) et Safari (sur Mac). D'autre part, les incompatibilités que nous déplorons ont tendance à diminuer avec les nouvelles versions.

FireFox est une amélioration de Mozilla qui se répand beaucoup. FireFox, Mozilla et Netscape 7 ayant exactement le même comportement, lorsque dans la suite nous en citons un, la remarque concerne aussi les autres.

C'est pourquoi, dans ce livre, nous nous efforcerons de donner des solutions et des exemples autant que possible **passé-partout**, c'est-à-dire fonctionnant au moins sur Internet Explorer et Mozilla. Nous signalerons, là où ce sera nécessaire, les différences avec Mozilla et Safari. C'est d'ailleurs surtout dans les programmes JavaScript que ces différences sont irréductibles.

Ce problème montre en outre combien il est indispensable de tester les pages Web avant leur mise en ligne et les Webmasters consciencieux effectuent ces tests sur plusieurs navigateurs, et même plusieurs versions.