

Chapitre I : D'Internet à la page Web

L'informatique est un élément omniprésent dans la vie quotidienne. Consulter ses mails ou visiter un site sont devenus des gestes ordinaires. Pourtant, cette simplicité d'usage dissimule une architecture technologique complexe qu'il est nécessaire de comprendre avant de pouvoir appréhender ce que sont les applications Web et d'en développer. L'objectif de ce chapitre est de présenter comment fonctionne le Web avant de découvrir les bases du langage qui permet de créer des pages Web : le HTML.

1.1 Un monde en réseau

1.1.1 Réseau et protocole

La structure fondamentale de l'informatique moderne est le réseau. Si l'on reprend la définition du dictionnaire Larousse, un réseau est un « ensemble formé d'éléments qui communiquent ou s'entrecroisent ». En informatique, un réseau peut être décrit comme étant un ensemble d'appareils électroniques qui échangent des informations entre eux.

Le premier réseau informatique a vu le jour à la fin des années soixante, sous l'impulsion d'une agence de recherche, DARPA, rattachée au Ministère de la Défense américain. Son objectif était de concevoir un mode de communication décentralisé, qui portait le nom d'*ARPAnet*, reliant initialement les ordinateurs de quatre centres de recherche.

Pour ce faire, les scientifiques du DARPA ont mis au point un *protocole* commun à tous les éléments du réseau. Un protocole est un ensemble de règles et de conventions qui permettent, à l'instar d'une langue, d'échanger des informations. Avec le développement de l'informatique au cours des années quatre-vingt, *ARPAnet* a considérablement évolué. D'abord réservé au seul territoire américain, le réseau est devenu international à partir de 1985. Les chercheurs et les entreprises ont ensuite progressivement étendu les capacités et les usages de ce réseau en mettant au point de nouveaux protocoles, à l'instar de la messagerie électronique.

ARPAnet, réseau limité et monovalent, est alors devenu ce que nous désignons aujourd'hui par le nom d'Internet. Ce terme issu de la contraction des mots *interconnected* et *networks* désigne l'essence même

de ce qu'est Internet, c'est-à-dire un ensemble de réseaux accessibles partout dans le monde.

1.1.2 Le Web, un réseau ordinaire

Parmi les réseaux qui forment Internet, le World Wide Web occupe une place particulière. Il s'agit d'un système mis au point par deux scientifiques du CERN, Robert Cailliau et Tim Berners-Lee, afin de mettre en forme et de présenter des documents. Leur objectif était de créer une manière de partager et de diffuser des travaux scientifiques. Ces ressources prenaient la forme de pages Web, textes reliés les uns aux autres par des liens. Ces ressources étaient identifiées par une adresse unique, également appelée URL, afin de pouvoir être retrouvées rapidement pour des consultations ultérieures.

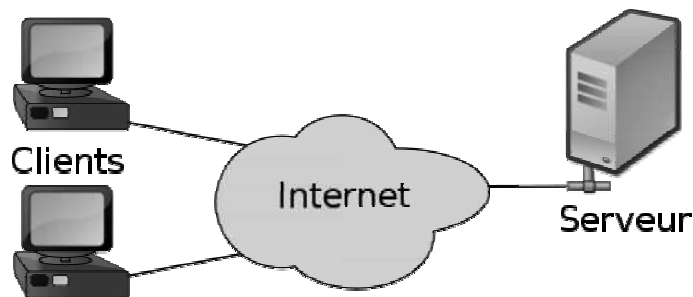


Figure 1 : Représentation schématique de l'architecture du Web
(source : David Vignoni, Gnome-fs-client.svg)

Si Web et Internet sont par abus de langage souvent confondus, il est important de noter que le Web n'est qu'un sous-ensemble d'Internet. Le Web a été créé dans les années quatre-vingt en Europe alors qu'Internet a vu le jour durant les années soixante, aux États-Unis. Le socle technologique du Web et d'Internet est également très différent. Ce dernier repose sur deux protocoles fondamentaux, *TCP* et *IP*, qui servent de socle à tous les autres.

Le Web est un réseau qui regroupe deux catégories d'acteurs : des *clients* et des *serveurs*. Nous approfondirons ces concepts au cours du chapitre III. Le Web fonctionne à l'aide de l'*Hypertext Transfer Protocol* dont le sigle est HTTP. Comme son nom l'indique, HTTP permet à un serveur et un client d'échanger des *hypertextes*, terme désuet pour désigner une page Web.

Un site Web se définit comme un ensemble cohérent de pages Web reliées entre elles. Le tout premier site Web est toujours en ligne et est accessible

sur le site du CERN¹. Avec le développement de l'informatique, les usages du Web ont évolué au-delà de son rôle initial de publication et de consultation de documents statiques.

Le but du présent ouvrage est d'apprendre à concevoir des applications Web. Le concept d'application est protéiforme et finalement assez difficile à définir précisément.

Une première approche consisterait à dire qu'à la différence d'un site Web, une application Web effectue un traitement d'information et est alors capable d'interagir avec l'utilisateur. Les moteurs de recherche, les jeux en ligne, les réseaux sociaux ou les plateformes d'e-commerce sont ainsi quelques exemples d'applications Web couramment utilisées.

1.1.3 Le HTML, langue du Web

Le HTML est un langage de programmation qui permet de construire des pages Web. Le HTML est dit *déclaratif* : son rôle est de *décrire* le contenu d'une page à l'aide de *balises*. Depuis sa création en 1989, le langage HTML a toujours évolué simultanément au Web. Les possibilités offertes par le langage ont beaucoup changé. Ces évolutions s'expliquent par les besoins nouveaux des développeurs, à l'instar de l'essor du Web mobile. De nombreuses balises et attributs ont ainsi été ajoutés et modifiés au fil du temps. Les évolutions du langage sont formalisées par une association, le World Wide Web Consortium² (W3C). Son rôle est de permettre au langage HTML d'évoluer dans le temps et de permettre au Web de continuer de croître. Le W3C a été fondé et reste influencé par Tim Berners-Lee. L'organisation est financée par les dons des plus grandes entreprises informatiques du monde.

Il est parfois utile de connaître les grandes lignes de l'histoire du langage car il n'est pas rare de trouver des ressources qui contiennent des informations obsolètes ou désuètes. Dans la pratique, l'usage d'éléments dépassés n'est pas nécessairement un problème car les navigateurs sont assez souples dans leur manière d'interpréter des pages HTML. C'est notamment le cas de l'attribut `align` qui est encore utilisé pour centrer un élément dans une page Web, même si son usage est déconseillé depuis

1. Il est accessible sur le site de l'organisation
<http://info.cern.ch/hypertext/WWW/TheProject.html/>

2. Le site français de l'organisation permet de suivre le développement du Web et les travaux en cours
<http://www.w3c.fr/>

2007. Les manières d'écrire le langage ont également évolué. Il était ainsi d'usage d'écrire en majuscule les balises jusqu'au début des années 2000. La version la plus récente du HTML est le HTML5, qui succède au HTML 4.01. Ce dernier a été formalisé par le World Wide Web Consortium en 2007. HTML5 se caractérise par l'ajout de nombreuses balises qui permettent d'ajouter des fonctionnalités multimédias au langage, à l'instar de la balise `< video >` ou `< canvas >`. Il permet également de faciliter la conception d'applications plus riches, en permettant par exemple de stocker des informations dans le navigateur. Toutes les nouvelles balises et attributs ne sont toutefois pas totalement intégrés aux navigateurs et quelques problèmes de compatibilité peuvent apparaître. Il existe de nombreux sites qui permettent de savoir quelles sont les balises supportées par les navigateurs³.

Entre 2000 et 2006, une version stricte du HTML a été développée : le XHTML. Son objectif était de rapprocher le HTML du XML, un langage qui sert à structurer des informations. Le XHTML exigeait ainsi que toutes les balises soient correctement fermées (``, `<input >`, etc.). Le développement du XHTML a été définitivement arrêté au profit du HTML5 depuis le 2 juillet 2009⁴.

Il est d'usage de commencer chaque page Web par une déclaration de *type de document* ou *doctype*. Cette dernière est une balise qui indique quelle version du HTML est utilisée par le développeur. Le *doctype* du HTML5 est `< ! DOCTYPE html >` et est placé à la première ligne des pages Web. Ce livre contient principalement du HTML5 et ne fait pas usage d'éléments obsolètes. La déclaration de type de document est toutefois omise pour gagner en concision et lisibilité.

1.2 Premier pas en programmation

1.2.1 Les outils du programmeur Web

Nous apprendrons à construire des applications Web à l'aide de deux principaux langages de programmation, le HTML et JavaScript. Pour être efficace, le développeur Web doit avoir à sa disposition les bons outils.

3. Can I Use fournit une liste exhaustive des incompatibilités
<http://caniuse.com/>

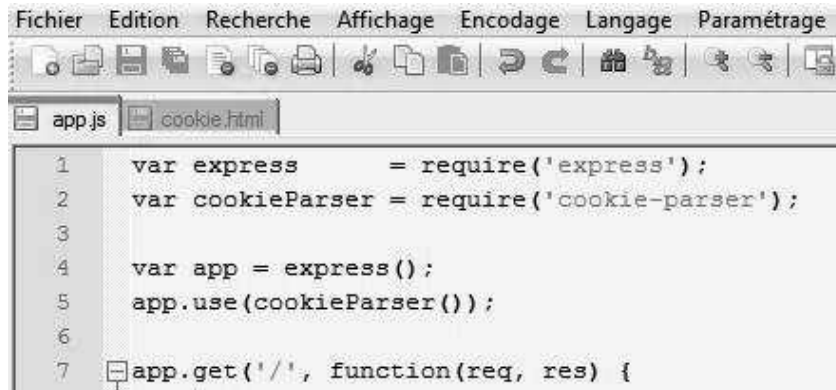
4. Ce communiqué se trouve sur le site du W3C, dans la FAQ du XHTML
<https://www.w3.org/2009/06/xhtml1-faq.html/>

Tous les logiciels que nous utiliserons ici pour programmer sont gratuits et la majorité d'entre eux sont Open Source.

La programmation se ramène toujours à l'écriture de *lignes de code*, c'est-à-dire des instructions rédigées par un programmeur et qui seront interprétées par l'ordinateur. Dans notre cas, nous écrivons des pages Web et des scripts.

Dans le cadre d'un apprentissage de la programmation, un *éditeur de texte* est largement suffisant. Ce type de logiciel permet de créer des fichiers qui contiennent du texte, sans considération de style ou de mise en page, à la différence des logiciels plus élaborés de *traitement de texte*. Notepad (Windows), TextEdit (Mac OS) et Vim (Linux) sont trois éditeurs de texte basiques, présents par défaut sur tous les ordinateurs.

S'ils suffisent pour découvrir le codage, il est préférable d'installer rapidement un éditeur de texte conçu pour programmer, à l'instar de Sublime Text⁵, Notepad ++⁶ (Windows) ou encore TextWrangler⁷ (Mac OS). Ces derniers offrent des fonctionnalités qui permettent aux développeurs de gagner en rapidité, comme la coloration syntaxique ou l'autocomplétion des lignes de code.

The image shows a screenshot of a code editor window. The title bar contains menu items: 'Fichier', 'Edition', 'Recherche', 'Affichage', 'Encodage', 'Langage', and 'Paramétrage'. Below the title bar is a toolbar with various icons for file operations and editing. The editor has two tabs: 'app.js' and 'cookie.html'. The 'app.js' tab is active, showing the following JavaScript code with line numbers 1 through 7 on the left margin:

```
1  var express      = require('express');
2  var cookieParser = require('cookie-parser');
3
4  var app = express();
5  app.use(cookieParser());
6
7  app.get('/', function(req, res) {
```

Figure 2 : un éditeur de texte conçu pour la programmation contient de nombreuses fonctionnalités pour aider à coder, comme la numérotation des lignes (ci-dessus, Notepad ++)

5. Site officiel de Sublime Text

<http://www.sublimetext.com/>

6. Site officiel de Notepad++

<https://notepad-plus-plus.org/fr/>

7. Page de présentation de TextWrangler

<http://www.barebones.com/products/textwrangler/>

La programmation Web sollicite fortement le navigateur Web, qui sera ici le client de référence. Tout navigateur permet également d'accéder au code d'une page pour en comprendre son fonctionnement. Il est recommandé d'installer un navigateur Web moderne et de le mettre régulièrement à jour. Mozilla Firefox, Opera ou Google Chrome sont quelques choix possibles de navigateur.

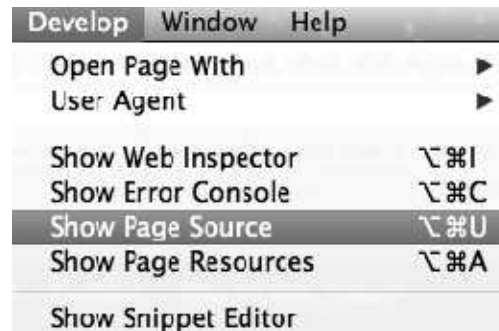


Figure 3 : accès au code d'une page avec Safari (raccourci CTRL + U pour les PC)

Nous utiliserons ici notre navigateur pour visualiser les pages que nous avons créées et pour tester en profondeur les scripts. Il est conseillé de l'utiliser en même temps qu'un *logiciel d'inspection*. Ce dernier permet notamment de révéler et de comprendre les erreurs de programmation ou encore de visualiser les échanges de requêtes et de réponses (chapitre III et suivant). Si ces fonctions ne sont pas indispensables pour l'internaute lambda, elles feront gagner beaucoup de temps au programmeur. Firebug⁸ est l'un des inspecteurs les plus répandus. Il convient de noter que la plupart des navigateurs comportent par défaut des outils d'inspection. Ces derniers sont souvent accessibles via les réglages du logiciel ou les menus contextuels.

8. Site de présentation de Firebug
<http://getfirebug.com/>

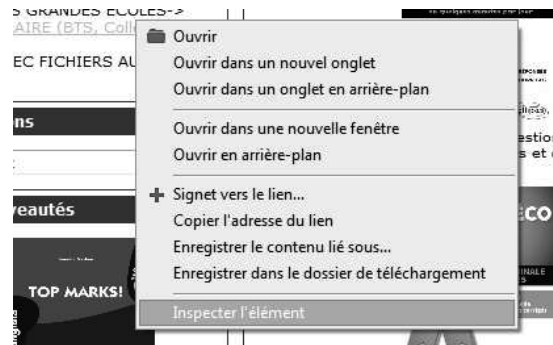


Figure 4 : l'inspecteur Web d'Opera se trouve dans le menu qui apparaît après un clic droit

Il existe enfin de très nombreux logiciels spécialisés qui permettent d'optimiser le code écrit. S'il est impossible de tous les citer, il peut être utile d'en avoir au moins un à disposition. Un outil simple d'accès est Dirty Markup⁹. Il permet de rendre plus facilement lisible le code d'une page Web et d'en corriger les erreurs courantes.

Il peut être également utile de mentionner l'existence des *environnements de développement intégrés* ou IDE. Un IDE regroupe en un seul endroit tous les outils utiles pour les développeurs. Ceux-ci sont assez complexes à appréhender et peuvent déconcerter les néophytes. Aptana, Webstorm et Eclipse sont quelques IDE courants pour le développement Web.

L'éditeur et l'inspecteur nous suffiront pour les tâches quotidiennes du programmeur. Nous aurons l'occasion d'utiliser et d'installer au cours de ce livre d'autres outils plus spécifiques, Node.js et Cloud Foundry. Ils feront l'objet d'une présentation plus détaillée, respectivement au cours du troisième et quatrième chapitre.

1.2.2 Les bases du HTML

La première brique technologique pour créer une application Web est la page Web. C'est elle qui est « envoyée » par le serveur aux clients. Une page Web est en fait un texte mis en forme par le langage de programmation présentée précédemment : l'*Hypertext Markup Language* (HTML en abrégé).

HTML est un langage de programmation qui sert à structurer une page Web. La brique élémentaire du HTML est la *balise*. Les balises sont des mots-clefs qui servent à décrire le rôle de chaque élément d'une page

⁹. Dirty Markup est un outil de mise en forme de HTML qui fonctionne en ligne <http://www.dirtymarkup.com/>

Web. HTML comporte aujourd'hui plusieurs dizaines de balises. Nous présenterons ici le fonctionnement des principales balises.

Le HTML est une technologie standard du fait de son âge. Sa manipulation, aussi bien en lecture qu'en écriture se fait à l'aide de logiciels déjà présents sur tous les ordinateurs. Les pages Web sont ainsi lues grâce à un navigateur. Pour les créer, n'importe lequel des éditeurs de texte présentés plus haut suffira. Il convient de noter que, si les logiciels de traitement de texte comme Word permettent d'enregistrer des documents en tant que page Web, cette démarche est fortement déconseillée dans le cadre de l'apprentissage du HTML. Les pages générées contiennent en effet des balises complexes qui rendent difficile la compréhension et la modification du code HTML produit.

Créer une page Web revient à écrire un texte. Il devient une page Web lorsqu'il est lu, ou interprété, par un navigateur. Ce dernier le lit ligne à ligne et analyse les balises pour produire un rendu visuel, compréhensible par un humain.

Les balises constituent ainsi l'ossature de la page et indiquent ce qu'elle contient : des images, du texte, des liens externes et ainsi de suite. Une balise est un mot qui est placé entre deux chevrons, à l'instar de `<html>` ou `<body >`. Les balises fonctionnent par paire : il faut une balise de début et, dans la plupart des cas, une balise de fin. Le rôle des balises est de définir la nature du contenu qu'elles délimitent.

La première balise d'une paire est une balise *ouvrante*. Pour créer une balise de fin, il suffit d'ajouter un `/` après le premier chevron. Ainsi, `<html>` et `<body>` sont deux balises ouvrantes, dont les balises de fin sont respectivement `</html >` et `</body>`

Pour créer une page Web, il faut écrire et enregistrer des balises dans un fichier¹⁰. Un fichier qui contient une page doit se terminer par une extension en .html. Par défaut, les éditeurs de texte créent des fichiers dont l'extension est .txt. Ils ne peuvent pas être lus par un navigateur. Pour y remédier, il suffit de changer l'extension du fichier en question.

10. Il existe également des éditeurs HTML en ligne, qui permettent de visualiser et de tester rapidement une page HTML ou un script. JSFiddle est l'un d'entre eux
<http://jsfiddle.net/>