



Apprendre le développement WEB au lycée

Avec les langages HTML5, CSS3,
JavaScript, PHP et MySQL

2^e édition

Cyprien Accard

NSI - SNT



CRÉATIONS NUMÉRIQUES

Apprendre le développement Web au lycée

Avec les langages HTML5, CSS3,
JavaScript, PHP et MySQL

NSI - SNT

2^e édition

Cyprien Accard



Collection **CRÉATIONS NUMÉRIQUES**

dirigée par Vincent Maille

<http://creations-numeriques.fr/>

Retrouvez tous les titres de la collection et des extraits
sur www.editions-ellipses.fr



ISBN 9782340-116566

Dépôt légal : juillet 2026

©Ellipses Édition Marketing S.A.

8/10 rue la Quintinie 75015 Paris



Le Code de la propriété intellectuelle et artistique n'autorisant, aux termes des alinéas 2 et 3 de l'article L. 122-5, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1^{er} de l'article L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

www.editions-ellipses.fr

Avant propos

L'informatique est une discipline qui évolue très rapidement. Même si ce livre se veut apporter des bases, une seconde édition était nécessaire.

Depuis la première moûture de ce livre, les versions des logiciels et langages utilisés ont évidemment progressé mais c'est surtout l'apparition et l'utilisation des intelligences artificielles génératives qui paraît changer les choses.

Bien sûr, ces IA sont capables de produire du code plus ou moins compliqués sans - quasiment - d'intervention d'un programmeur. Elles sont aussi capables de « se tromper », même pour des choses simples. Ou alors, il arrive également qu'elles proposent des solutions qui ont l'air parfaites mais qui en réalité ne le sont pas.

Alors, évidemment, quand les demandes ne sont pas trop compliquées, le taux de réussite de ces IA est très élevé. Mais pour des projets plus élaborés, il n'en va pas toujours de même... Toujours est-il que pour être capable de juger de la pertinence de ce qui est proposé ou pour le corriger, il est plus qu'important de connaître soi-même les bases !

L'objectif de ce livre est donc là : Donner les bases de la programmation web, libre au lecteur d'approfondir ou ... de se laisser porter ensuite par des moyens automatisés pour produire ce qu'il souhaite.

En termes d'enseignement, ce livre avait été à l'origine écrit pour pouvoir être utilisé dans le cadre des enseignements ISN (Informatique et Sciences du Numérique) en Terminale S et ICN (Informatique et Création Numérique) en seconde, faisant du développement web un des possibles enseignements du lycée. Cette partie de l'informatique répondait bien aux besoins de ces programmes (réseaux, langages de description, programmation) sans que les établissements scolaires n'aient besoin d'investir dans du matériel.

Aujourd'hui, ces enseignements ont disparu mais le thème du web et du développement web se trouvent dans les programmes de SNT (Sciences du Numérique et Technologie) et surtout de NSI (Numérique et science

informatique). Les élèves sont friands de développement web et celui-ci ouvre de nombreuses perspectives et permet des projets très riches et variés.

Aussi, l'élève ou l'enseignant voulant se former sur le développement web a besoin de bien connaître les bases !

Ce livre se veut permettre cet apprentissage et chaque chapitre comporte de nombreux exemples et de nombreux exercices pour s'approprier les notions étudiées. Sur le site de la collection du livre, vous trouverez des compléments (programmes des corrections, ressources, ...), il vous suffira de taper le numéro de l'exercice ou le **CODE** qui sera présent dans le livre sur fond gris.

<http://www.creations-numeriques.fr/>



Ce livre peut être utilisé seul ou bien en classe avec des élèves. L'objectif n'est en aucun cas de permettre au lecteur de devenir spécialiste des différents langages présentés mais d'initier à la pratique conjointe de ces langages et leurs interactions.

S'agissant de bases, pas mal de « trucs et astuces » liés aux langages présentés ne sont pas mentionnés et il est possible, parfois, de faire plus simple que les solutions présentées. J'ai voulu, de façon délibérée, ne proposer que quelques méthodes de façon à ce que le lecteur puisse mener à bien des projets. Libre à lui, ensuite, de faire mieux.

Chaque langage est présenté avec des exemples et des exercices et des projets finaux mêlant tout ce qui a été présenté dans le livre sont proposés à la fin.

Dans les parties de code présentées, certaines lignes commencent ou se terminent par une flèche. La raison en est simple : les instructions sont parfois plus longues qu'une ligne du livre ! Il ne s'agit donc pas d'essayer de taper ses flèches, mais de comprendre que l'instruction écrite sur plusieurs lignes dans le livre doit être retapée en une seule sur votre logiciel préféré.

De même, parfois, certaines formulations font apparaître des variables pas nécessairement utiles : c'est souvent une volonté soit pédagogique pour mieux comprendre les mécanismes, soit un besoin de pagination, certaines lignes de code (surtout en JavaScript qui est un langage assez « verbeux ») ne pouvant pas tenir sur une seule ligne.

De plus, vous trouverez parfois de petites différences de pagination entre le code présenté dans le livre et celui sur les fichiers du site, pour les mêmes raisons.

Table des matières

A	Historique du web	11
1 -	Débuts d'internet	11
2 -	Création du World Wide Web	12
3 -	CSS	15
4 -	Langages de programmation	16
5 -	Bases de données	18
B	Premiers pas en HTML	21
1 -	Les besoins pour écrire du HTML	21
2 -	Principe de HTML	22
3 -	Structure minimale d'une page HTML	23
4 -	Premières balises	25
5 -	Solutions des exercices	28
C	Structurer ses pages HTML	31
1 -	Des balises pour ajouter des éléments	31
a)	Formater le texte	31
b)	Images	34
c)	Tableaux	37
d)	Formulaires	39
e)	Liens	47
f)	Commentaires	48
2 -	Des balises pour structurer	49
a)	Métadonnées	49
b)	Éléments de la page	50
3 -	Mentions légales	51
4 -	Une utilisation particulière de HTML	52
5 -	Solutions des exercices	52
D	Utiliser CSS	59
1 -	Formater du texte	59
2 -	Modifier l'apparence d'éléments	62
3 -	Modifier la position d'éléments	66
4 -	Héritage	72
5 -	Solutions des exercices	73

E	Externaliser CSS	77
1 -	En début de fichier	77
2 -	Dans un fichier externe	81
3 -	Solutions des exercices	83
F	Quelques rudiments d'algorithmique	89
1 -	Qu'est-ce qu'un algorithme ?	89
2 -	Utiliser des variables	90
3 -	Les structures conditionnelles	93
4 -	Les structures itératives (boucles)	95
5 -	Les fonctions	97
6 -	Solutions des exercices	100
G	Utiliser JavaScript	105
1 -	Fonctionnement de JavaScript	106
2 -	Programmer en JavaScript	107
a)	Variables	107
b)	Structures conditionnelles	111
c)	Structures itératives (Boucles)	115
d)	Fonctions	117
3 -	Réagir à un événement	118
4 -	Interagir avec des éléments de la page	122
5 -	Solutions des exercices	126
H	Installer et utiliser un serveur PHP	141
1 -	Pourquoi utiliser un serveur ?	141
2 -	Choix d'une distribution serveur	142
3 -	Installation d'un serveur	142
4 -	Utiliser le serveur	143
I	Utiliser PHP	145
1 -	Programmer en PHP	145
a)	Premiers programmes	145
b)	Variables	148
c)	Structures conditionnelles	150
d)	Structures itératives (Boucles)	150
e)	Fonctions	151
2 -	Interagir avec une page PHP	152
a)	Méthode GET	152
b)	Méthode POST	155
3 -	Les sessions	157
4 -	Utiliser des fonctionnalités du serveur	160
a)	Variables serveur	160
b)	Fichiers	162
5 -	Solutions des exercices	163

J	Bases de données relationnelles	185
1 -	Qu'est-ce qu'une base de données?	185
2 -	Comment structurer les données?	186
3 -	Solutions des exercices	189
K	Utiliser MySQL	191
1 -	phpMyAdmin	191
2 -	Créer, gérer des bases de données	194
3 -	Le langage SQL	198
a)	Enregistrer des données	198
b)	Lire des données	199
c)	Modifier des données	202
d)	Effacer des données	203
e)	Jointures	203
f)	Création d'une base	205
4 -	Administrer les bases de données	205
5 -	Solutions des exercices	206
L	Interfacer MySQL et PHP	209
1 -	Envoyer des requêtes	209
2 -	Traiter les informations reçues	214
3 -	Sécurité	218
a)	Chiffrer des données dans la base	219
b)	Injection de code	221
4 -	Solutions des exercices	223
M	Utiliser Ajax	237
1 -	Qu'est-ce qu'AjAx?	237
2 -	JQuery et l'objet .ajax	238
3 -	Traitement par la page PHP	239
4 -	Retour et traitement de la requête	240
5 -	Solutions de l'exercice	241
N	Premier projet : le jeu de la vie	247
1 -	Présentation du projet	247
2 -	Cahier des charges	248
3 -	Une solution	248
a)	Le jeu	249
b)	Les comptes et la sauvegarde	249
c)	Toute la solution	249

O Deuxième projet : Une to-do List **265**

- 1 - Présentation du projet 265
- 2 - Cahier des charges 265
- 3 - Une solution 266
 - a) Structure des données 266
 - b) Page d'accueil 267
 - c) La page principale (et ses annexes) 275

Chapitre A

Historique du web

Pour bien comprendre le développement d'un site, il me paraît nécessaire d'avoir une idée de l'historique du développement d'internet et du web, afin d'avoir une vue aussi complète que possible des différentes composantes d'internet et de l'arrivée des divers standards que nous avons l'habitude de côtoyer.

1 - Début d'internet

Internet est l'aboutissement d'un travail entamé aux États-Unis dès la fin des années 50. En 1958, les laboratoires Bell créent le premier modem, pour l'abréviation de modulateur-demodulateur, qui permet d'envoyer des données informatiques sur une ligne téléphonique. La même année, la Defense Advanced Research Projects Agency (DARPA, « Agence pour les projets de recherche avancée de défense ») est créée. Joseph Carl Robnett Licklider, un informaticien, y défend (avec succès) l'idée d'un réseau interconnecté d'ordinateurs.

Quelques années plus tard, Leonard Kleinrock, du MIT, publie une théorie sur la communication de données par paquets.

Mais le point de départ réel de ce qui deviendra Internet est la première conférence ARPANET (Advanced Research Projects Agency Network), en 1967. Cette conférence lance le projet, qui débutera en 1969 et dont la première démonstration effective se fera en 1972.

Dès 1969, les universités américaines de Stanford et l'Université de Californie à Los Angeles (UCLA) sont reliées et le premier mot est transmis sur le réseau : *login*. Pour la petite histoire, la première lettre fut transmise instantanément et un bug fit que les trois suivantes mirent plus d'une

heure à arriver à destination. Ces deux universités seront peu après rejointes par l'université de Californie à Santa Barbara et par celle de l'Utah.

Le réseau est constitué de petits ordinateurs, les Interfaces Messages Processors (IMP), pouvant être considérés comme les ancêtres des routeurs actuels, qui assurent la connexion des ordinateurs hôtes au réseau et permettent le transfert des données par la commutation de paquets. La vitesse de réseau était, au départ, de 50kbits par seconde.

Deux ans plus tard, un ingénieur, Ray Samuel Tomlinson, envoie le premier courriel. 23 ordinateurs sont alors connectés. Le nom d'Internet trouve son origine l'année suivante avec la création de l'InterNetworking Working Group, organisme chargé de la gestion du réseau.

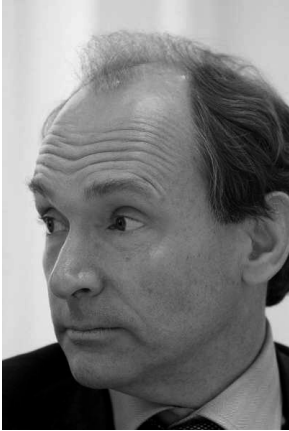
Le réseau s'internationalise en 1973 puisque l'Angleterre et la Norvège y connectent chacun un ordinateur. Le réseau s'agrandissant, le besoin d'un protocole de communication se fait sentir et le protocole TCP/IP (Transmission Control Protocol / Internet Protocol) est défini. Il mettra 10 ans avant d'être adopté.

En 1977, 111 ordinateurs sont connectés et, en 1979, des étudiants américains, créent les NewGroups (forums de discussion Usenet), encore utilisés aujourd'hui.

2 - Création du World Wide Web

Les années 1980 voient l'essor de ce réseau qui deviendra rapidement celui que l'on connaît aujourd'hui. En 1983, le protocole TCP/IP est (enfin) adopté, ainsi que le nom d'Internet. Parallèlement à cela, le premier DNS (Domain Name Server, serveur de noms de domaine) est créé. Environ 1 000 ordinateurs sont connectés au réseau. En 1987, le nombre de connexions passe à 10 000 et, en 1989, 100 000.

Le début des années 1980 est aussi une période faste pour le développement de l'informatique en général, avec le lancement d'ordinateurs à destination du « grand public » comme l'IBM PC (1981), le Sinclair ZX81, ou encore des développements logiciels comme le MS-DOS de Microsoft (tout cela rien que pour l'année 1981).



*Tim Berners-Lee,
par Paul Clarke.
CC BY 2.0*

Si Internet trouve son démarrage aux États-Unis, c'est en Europe, plus précisément au CERN (Centre Européen de Recherche Nucléaire), à la frontière franco-Suisse près de Genève, que le World Wide Web (Toile d'Araignée Mondiale) naît, sous l'impulsion d'un informaticien anglais, Timothy (Tim) Berners-Lee et d'un ingénieur système belge, Robert Cailliau. Nous sommes alors en 1989.

Il s'agissait, au départ, d'un système hypertexte à utiliser sur le réseau du centre afin que les collaborateurs puissent facilement partager des informations. Il s'agit d'un ensemble de documents reliés de façon à ce l'utilisateur puisse consulter directement ce qui l'intéresse.

De façon concomitante, le CERN décide cette année-là d'utiliser le protocole TCP/IP et d'ouvrir sa première connexion à Internet. Le web peut commencer à se développer.

Pour surfer sur le web, un logiciel est indispensable : un navigateur. Le premier s'appelle WorldWideWeb (et sera plus tard rebaptisé Nexus). Il s'agit d'un navigateur en mode texte, très simple de façon à pouvoir fonctionner sur le maximum de types d'ordinateurs possible. Ce navigateur permet d'afficher la première page web, à l'adresse :

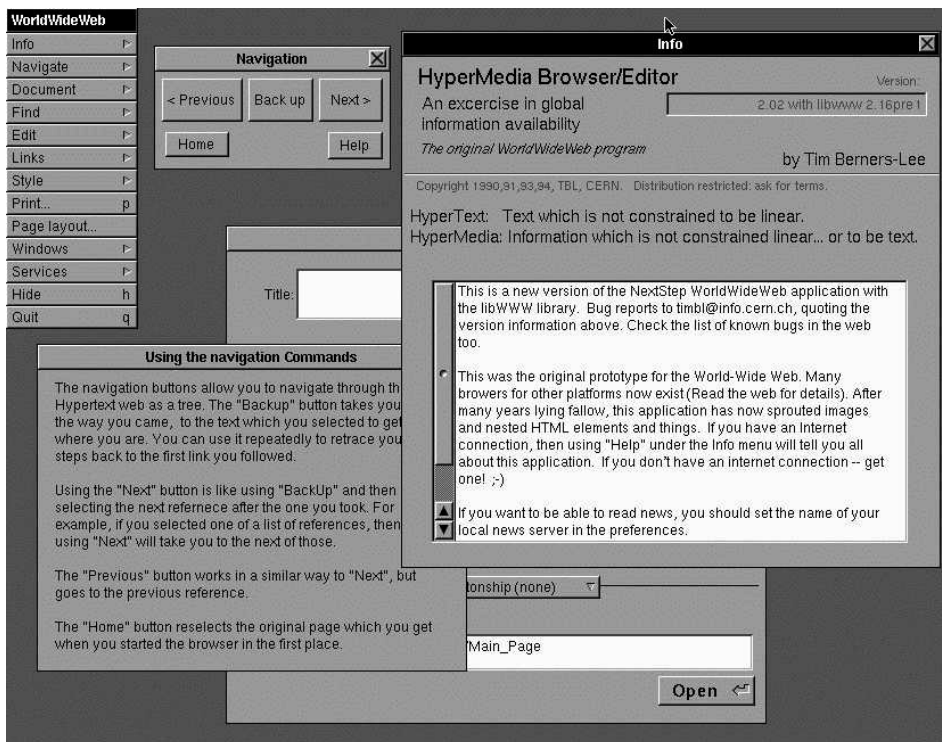
```
http://nxoc01.cern.ch/hypertext/WWW/TheProject.html
```

N'essayez pas de taper cette adresse dans votre navigateur préféré, elle est à présent inopérante ! Mais, le CERN a conservé cette page en ligne a une autre adresse :

```
http://info.cern.ch/hypertext/WWW/TheProject.html
```

La plus ancienne page web conservée date du 13 novembre 1993. Les éléments qui constituent encore le web sont déjà présents : URL (Uniform Resource Locator), que l'on appelle plutôt en français adresse web, HTTP (HyperText Transfer Protocol), le protocole de communication client - serveur en usage sur le web et HTML (HyperText Markup Language) qui est le format de données pour décrire les pages web.

Le premier serveur web tournait sur une station de travail NeXT Cube, sur laquelle un post-it était collé : « This machine is a server, DO NOT POWER IT DOWN!! » Cette machine est un serveur, NE PAS L'ÉTEINDRE !!.



Capture d'écran du navigateur WorldWideWeb, par Tim Berners-Lee.
Image dans le domaine public.

Le 6 août 1991, dans un message publié sur Usenet, Tim Berners-Lee rend le projet public, mais celui-ci ne sera présenté que presque un an plus tard, le 3 novembre 1992. Le web comporte alors 26 sites.

C'est l'année 1993 qui sera la plus importante pour le web puisque le 30 avril, le CERN renonce aux droits d'auteur sur les logiciels du web. Cette année verra aussi l'apparition du premier navigateur graphique, Mosaic, qui permettra dès sa version 0.10 de visualiser des images sur les pages web. Le développement du web ne s'arrêtera plus ! Le nombre de sites va également croître de façon exponentielle avant la fin de la décennie. De 26 sites en 1992, on passe à 623 fin 1993, puis 10 000 fin décembre 1994. Les 100 000 sont franchis en 1996 et le million, en 1997. Le milliard, quant à lui, est plus récent, puis qu'il n'a été franchi qu'en septembre 2014.

En octobre 1994, Tim Berners-Lee fonde le World Wide Web Consor-

tium (W3C), un organisme de standardisation destiné à promouvoir la compatibilité des technologies utilisées sur le web, en vue d'obtenir « Un seul web partout et pour tous ». Le W3C met à disposition un validateur, qui permet de s'assurer que les pages web produites respectent bien les standards du web. il est accessible à l'adresse :

<https://validator.w3.org/>

3 - CSS

Au départ, le HTML n'était qu'un système hypertexte destiné à ce que les collaborateurs du CERN puissent avoir un accès simple aux informations. Il n'était alors nullement question de pages attrayantes, d'images, de vidéo, de son... Mais tout cela s'est doucement développé avec la croissance exponentielle du nombre de sites et l'apparition de navigateurs « populaires » tels Netscape navigator en 1994 ou Internet Explorer, de Microsoft, en 1995.

Les possibilités « esthétiques » de HTML sont assez rudimentaires, mais en détournant de leur fonction principale certains éléments, il est possible de programmer d'assez jolies pages, en utilisant moult images et des tableaux dont la bordure est invisible pour gérer le positionnement des différentes parties de la page. Cette méthode, très en vogue dans les années 1990 est simple à mettre en place pour des petits sites, mais demande d'énormes travaux supplémentaires en cas de modification des pages : Il faut reprendre tout le code et reprogrammer.

C'est pour éviter ces programmations lourdes en cas d'utilisation de plusieurs pages et pour éviter de devoir tout reprogrammer à chaque changement sur la page qu'est née l'idée de séparer « la forme et le fond », ou, plus exactement, la structure du document et la partie esthétique des pages.

En 1996, la norme CSS1, pour Cascading Stylesheets, ou, en français, feuilles de style en cascade, qui permet de gérer à part et donc de modifier simplement le style d'un élément d'une page web ou de toute une page, voire de tout un site.

L'utilisation du langage CSS permet de jouer sur le « style », c'est-à-dire des éléments comme la position sur la page, la couleur, la taille des caractères. Il est possible, au choix, de définir ce style soit pour chaque élément, soit une fois pour toute sur la page, soit, si le site comporte plusieurs pages, dans un fichier séparé commun à toutes les pages.

De son côté, HTML gère la structure du document, c'est-à-dire la nature de chaque élément (texte, paragraphe, lien, image, etc...).

Depuis sa création en 1996, CSS s'est amélioré à de nombreuses reprises profitant de la puissance accrue des ordinateurs et la complexification des navigateurs. Le W3C a commencé à rédiger la version 2 en 1998; la version 3 commence dès 1999. À l'heure actuelle, la plupart des sites utilisent du CSS3 bien que la version 4 soit déjà en cours de développement depuis 2010 mais la version majoritairement utilisée reste la 3.

4 - Langages de programmation

Très vite, le besoin de pouvoir réagir aux interventions de l'utilisateur s'est fait sentir. HTML, même complété avec du CSS, permet juste un affichage mais aucun retour programmé vers l'utilisateur. HTML et CSS ne permettent, seuls, que la programmation de sites statiques.

Pour passer au web dynamique, il y a nécessité d'un langage de programmation serveur, d'abord. En effet, les premières interactions dont ont eu besoin les concepteurs de sites étaient des interactions du type client-serveur.

PHP



*Rasmus Lerdorf,
par Ed.
Australie, 2007
CC BY SA 2.0*

C'est ainsi qu'en 1994, un programmeur canadien né au Groenland, Rasmus Lerdorf, a créé la première version du langage de programmation PHP, pour Personal Home Page (Page personnelle en français), devenu depuis PHP : Hypertext Preprocessor (acronyme récursif).

Lerdorf souhaitait conserver une trace des visiteurs qui venaient consulter son CV sur sa page personnelle. Il a donc développé une bibliothèque logicielle en C qu'il a enrichi au fur et à mesure avant de publier son code sous licence libre en 1995.

La version actuelle de PHP est la 8.4.8 (version 8 sortie en 2020). À noter que la version 6 a été annulée et que la version 5 a longtemps été utilisée. En 2013, plus de 244 millions de sites utilisent PHP. Néanmoins, il semble que l'utilisation de PHP ait passé un pic : ce langage était utilisé par

80% des sites dans le monde en 2016 (à son apogée), en 2022, l'utilisation « chute » à 78%. Ce langage reste donc très utilisé.

PHP est un langage orienté objet, impératif, fonctionnel, procédural et interprété.

JavaScript

Mais, si un langage serveur est indispensable pour la sauvegarde d'informations, il n'est pas possible de l'utiliser pour effectuer des actions sur le poste client. Il faut un autre interpréteur de code, fonctionnant côté client, soit par l'intermédiaire d'applets, soit directement interprété par le navigateur. C'est le cas de JavaScript.



*Brendan Eich,
par Darcy Padlila.
CC BY SA 3.0*

JavaScript, qui ne partage avec Java que le début de son nom, est un langage interprété par les navigateurs, créé par Brendan Eich en 1995. On raconte que le langage, créé pour Netscape Communications Corporation, a été développé en 10 jours seulement.

Eich s'est inspiré de nombreux langages pour développer JavaScript, notamment de Java mais en simplifiant la syntaxe. Évidemment, le premier navigateur embarquant un moteur pour ce langage fut un navigateur de Netscape, Navigator 2.0, sorti en mars 1996.

Un des soucis rencontré par JavaScript est que tous les navigateurs ne reconnaissent pas de la même façon les instructions, certaines étant spécifiques à certains navigateurs. Aussi, en 2006, John Resig lance la bibliothèque JQuery pour faciliter l'écriture des scripts JavaScript et uniformiser les différentes fonctions à utiliser suivant les navigateurs.

Il est possible de « mélanger » JavaScript et PHP, ou plus exactement d'appeler des programmes PHP depuis JavaScript par la méthode Ajax, pour Asynchronous JavaScript and XML.

La version actuelle de Javascript est la 13. Javascript est un langage orienté objet, impératif, fonctionnel et interprété (par les navigateurs).

5 - Bases de données

Pour terminer ce tour d'horizon historique et technique du web, maintenant que nous avons connaissance de la gestion de la structure d'une page (HTML), de son apparence (CSS), d'un langage serveur (PHP) et d'un langage client (JavaScript), il nous faut un moyen de sauvegarder facilement des données afin de les exploiter.

Il est bien sûr possible de déposer ses fichiers sur un serveur web, comme on le fait pour les images ou les vidéos que l'on peut proposer sur son site, ou alors de créer via PHP des fichiers de données qu'on viendra lire, mais cela n'est pas pratique et demande un investissement important en programmation. La programmation de sites web est adaptée à l'utilisation de structures de gestion de données : les SGBDR (Système de gestion de bases de données relationnelles).

Nous utiliserons ici un des logiciels créé par un finlandais, Michael Widenius. Widenius a créé en 1995 MySQL (prononcer [maj.es.ky.ɛl]), du nom de sa première fille, My (se prononce [my]). Depuis, MySQL a été racheté par Sun Microsystems en 2008, entreprise elle-même rachetée par Oracle Corporation en 2009.

En 2009, Michael Widenius a créé un autre SGBDR, MariaDB (du nom de sa deuxième fille, Maria) pour continuer le développement en tant que projet Open Source, c'est-à-dire respectant les principes de libre redistribution, d'accès au code source et de création de travaux dérivés.

Les deux outils sont assez proches pour l'utilisateur et tous deux basés sur le langage SQL, pour Structured Query Language. Ce langage, très commun dans le monde des bases de données, permet de définir les données, de les manipuler, de les exploiter avec une syntaxe proche de phrases en anglais.

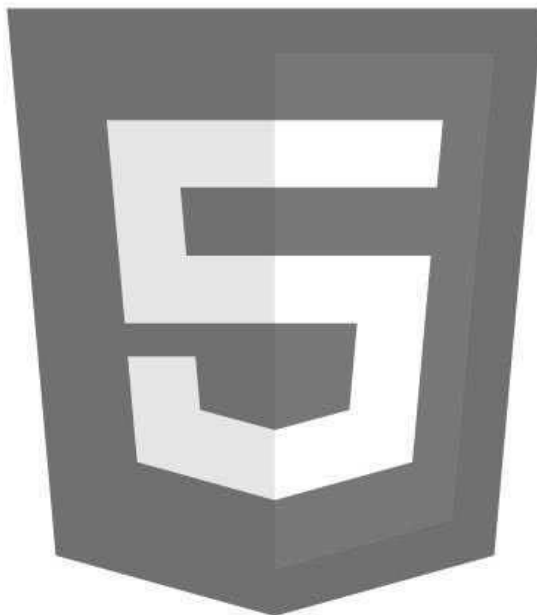
HTML

Pour développer un site web, la première chose à faire est de maîtriser le langage structurant les pages, le HTML (HyperText Markup Language).

Ce langage, très simple à utiliser, permet néanmoins de faire de grandes choses. Le développement HTML ne nécessite rien d'autre qu'un éditeur de texte. Un navigateur est bien entendu utile pour visualiser les pages créées...

Une fois les principes de base acquis, la connaissance de toutes les balises est fastidieuse et inutile. Aussi, il ne s'agira pas ici de créer un dictionnaire exhaustif de HTML 5, mais plutôt de poser les bases permettant de se poser les bonnes questions pour utiliser ce langage de description.

HTML



Chapitre B

Premiers pas en HTML

À ses débuts, le langage HTML était l'unique façon de structurer la forme et le fond. Pour nos premiers pas, nous allons donc suivre cette philosophie.

Dans la mesure où, à l'heure actuelle, HTML n'est plus utilisé pour l'embellissement, nous n'approfondirons pas ces méthodes qui commencent à dater, mais celles-ci nous permettront néanmoins d'écrire nos premières pages dont l'aspect esthétique sera loin des standards actuels mais dont le code sera relativement simple.

1 - Les besoins pour écrire du HTML

Nous allons pouvoir nous lancer dans la programmation de pages web avec HTML. Mais de quoi avons nous besoin pour cela? À priori, pour mettre en ligne et proposer un site web, il faut pour cela un serveur web qui peut être compliqué à installer. Certes certains hébergeurs permettent d'obtenir un peu de place gratuitement sur leurs serveurs, ou alors il est possible de louer un espace chez un hébergeur spécialisé, mais pour nos essais de pages HTML, nous n'avons pas besoin de tout cela pour commencer.

Deux logiciels très simples suffisent pour commencer : un navigateur, tel que Mozilla Firefox, Google Chrome (ou son équivalent libre, chromium), Microsoft Edge, Opéra... Ou tout autre navigateur, puisque le travail d'un navigateur est d'interpréter le code HTML et de le transformer en page à afficher.

Le navigateur permet l’affichage et donc la visualisation de notre travail. Mais, ce travail, encore faut-il le programmer et pour cela, il faudra un autre type de logiciel, très simple, un éditeur de textes. Sous Windows, il y a certes le bloc-notes, mais celui-ci n’est pas du tout pratique à l’usage.

L’éditeur Notepad++, un éditeur de textes très agréable, utilisant la coloration syntaxique et publié sous licence GPL. Il n’est à priori disponible que sous linux. Néanmoins, avantage du logiciel libre, une version multiplateforme a été écrite : il s’agit de NotepadNext. À titre personnel, développant sur des machines Linux, j’utilise plutôt Geany, un autre éditeur de textes sous licence GNU-GPL.

Vous pourrez trouver le premier à cette adresse :

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



Le second :

<https://github.com/dail8859/NotepadNext>



Et le troisième :

<http://www.geany.org/>



2 - Principe de HTML

Le HTML dérive du langage SGML, ou Standardized Generalised Markup Language, qui est un langage de description par balises, une référence en milieu professionnel. Pour coder en HTML, il suffira d’ouvrir un éditeur de textes et de saisir des balises. Seulement, il faut respecter un certain formalisme pour que le navigateur puisse comprendre ce que l’on souhaite afficher.

Les balises HTML sont des mots clés insérés entre les symboles < et >. Par exemple, tout fichier HTML contient la balise <html>. Il existe

deux types de balises, celles qui vont **par paire** (comme `<html>`, qui est « fermée par » `</html>` et celles qui sont **orphelines**, comme la balise `<meta>` (Ces balises seront détaillées un peu plus loin).

Ces balises, imbriquées les unes dans les autres, construisent ainsi un document arborescent.

Les premières normes HTML stipulaient l'écriture des balises en majuscules. Avec HTML5, la convention est de les écrire en minuscules.

Autre convention, au moment d'enregistrer le fichier (dont l'extension sera, au choix, `.htm` ou `.html`), on retiendra que, comme la majeure partie des serveurs web fonctionnent sous Linux, un système d'exploitation qui prend en compte la casse (c'est-à-dire qui fait la différence entre minuscules et majuscules) et que, dans une adresse web, les espaces et caractères accentués ou « exotiques » tels les signes de ponctuation par exemple sont mal venus, on prendra la précaution d'écrire le nom du fichier en minuscules et seulement avec des lettres, des chiffres et le tiret haut (-) et le tiret bas (_).

3 - Structure minimale d'une page HTML

Nous pouvons à présent entamer notre première page web, qui ne contiendra ... RIEN! On va ici détailler la structure de la page, la version minimale que celle-ci doit contenir.

```
<!doctype html>
<html>
    <head>
        <title>
            Titre de la page
        </title>
        <meta charset="utf-8">
    </head>
    <body>
    </body>
</html>
```

Vous pouvez retrouver ce morceau de code qui, bien que très vide, nous sera très utile, sur le site en tapant le code `HTML0`.

La première balise, sur la première ligne, `<!doctype html>` n'est pas à proprement parler une balise HTML, mais elle permet au navigateur de savoir que le fichier est un fichier HTML suivant la norme HTML5.

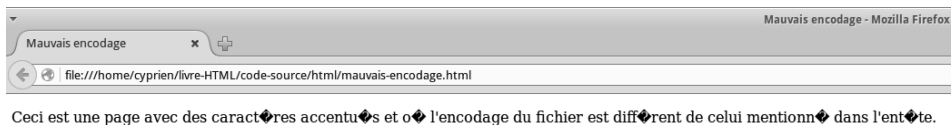
La structure du fichier html se détaille en-dessous. Les balises HTML à proprement parler sont comprises entre la balise `<html>` et la balise fermante `</html>`.

Entre ces deux balises qui structurent le fichier, se trouvent deux zones, l'entête HTML, délimité par les balises `<head>` et `</head>` et le corps de la page, délimité par `<body>` et `</body>`.

L'entête du fichier contient des données et métadonnées qui ne seront pas affichées sur la page. Ici, nous avons limité les informations présentes dans l'entête à deux : Le titre, délimité par les deux balises `<title>` et `</title>`, qui est affiché uniquement dans la barre de fenêtre ou dans l'onglet du navigateur et l'encodage de la page, permettant un affichage correct, défini dans la balise `<meta>`.

L'encodage est une métadonnée très importante qui permet au navigateur de bien associer la façon dont a été enregistré le fichier avec ce qui est prévu pour l'affichage et la lecture. Il est donc primordial d'utiliser le même encodage pour l'enregistrement du fichier que celui déclaré via la balise `<meta>`. De plus, l'encodage doit être déclaré dans les 1024 premiers caractères du fichier.

Les pages web qui ne déclarent pas correctement leur encodage et utilisent des caractères accentués ont un rendu assez étrange : les caractères accentués sont « remplacés » par des signes « cabalistiques ». L'encodage le plus courant sur le web est l'UTF8.



Exemple de page web avec un mauvais encodage. Le titre, « Mauvais encodage » entre les balises `<title>` et `</title>` est présent dans l'onglet.

Revenons un instant sur la balise `<meta>` présente dans le code. Celle-ci présente un **attribut**, `charset`, qui spécifie l'encodage. Cette balise peut en contenir d'autres, pour d'autres métadonnées dont nous parlerons plus tard. Tout attribut qui prend une valeur est suivi du signe égal et de la valeur entre guillemets. Ici, la valeur attribuée à l'encodage est `utf-8`.

Attention cependant, il ne suffit pas de spécifier l'encodage dans le fichier html. Il est nécessaire que le fichier soit enregistré avec le même encodage. Cela se règle assez facilement avec les éditeurs de texte mentionnés ci-avant. Dans Notepad++, le menu « encodage » permet de choisir celui-ci. Il est paramétré de base sur l'UTF-8. Dans Geany, il faut aller dans le menu « Document » puis choisir « Définir l'encodage ». Seul NotepadNext ne le permet pas (encore... un menu prévient que ce n'est pas encore implémenté, mais je n'ai pas de doute sur le fait que ça le sera prochainement), mais là aussi, c'est bien l'UTF-8 qui semble être sélectionné de base.

Il est évidemment possible de choisir un autre type d'encodage, comme l'ISO-8859-1 ou ISO-8859-15, encodages régulièrement utilisés sous Windows. L'important est qu'il y ait bien adéquation entre l'encodage utilisé et celui spécifié dans le fichier.

Je vous conseille de télécharger sur le site (ou de recopier) ce morceau de code et de le sauvegarder. Il contient les informations et la structure minimale d'une page web et nous servira systématiquement de point de départ pour les autres pages.

4 - Premières balises

Pour afficher quelque-chose sur la page, il faut écrire du code entre les balises `<body>` et `</body>`. Et cela s'écrit également avec des balises ! Nous allons, dans cette section, voir quelques balises et leur utilisation.

Bien qu'il soit possible d'écrire du texte sans aucune balise, la norme HTML5 tente de supprimer cette possibilité. Bien entendu, cela fonctionnera tout de même car les navigateurs interprètent le HTML5 mais aussi toutes les versions précédentes ! Mais comme nous essayons de prendre de bonnes habitudes, nous allons nous conformer aux prescriptions : nous écrirons donc tout élément « dans » des balises de façon à ce que chaque élément soit bien inclus dans l'arborescence du document, le **DOM**, pour **Document Object Model** : Il s'agit une représentation structurée du document sous forme d'un arbre, qui n'est évidemment pas affichée, mais permet la manipulation de la page, notamment de façon dynamique (nous y reviendrons plus loin).

Il y a plusieurs façons d'écrire du texte, nous allons commencer par voir la balise `<p>`, comme paragraphe, qui permet d'écrire un paragraphe.