

Cours De Développement Web



Formateur : Cayrol ABOURI

COURS DE DÉVELOPPEMENT WEB I

Objectifs pédagogiques :

À la fin de ce cours, vous saurez...

- Les concepts fondamentaux du web
- Maîtriser les langages HTML et CSS
- Créer des pages web en HTML et CSS
- Structurer une page web
- Transformer une maquette en page web
- Envoyer un site en ligne

I. Les concepts fondamentaux du web

1. Le réseau informatique :

Le vocable réseau est antérieur à l'informatique et il peut être compris comme un ensemble d'éléments reliés entre eux par un ensemble de chemins. C'est ainsi qu'on parle du réseau routier, du réseau maritime, du réseau ferroviaire... Le terme a beaucoup évolué dans le temps pour désigner également tout groupe d'individus réunis dans le cadre d'une activité réalisée en commun. On parle alors de réseaux de terroristes, de réseaux de documentalistes, de réseaux de francs-maçons etc. Appliqué à l'informatique, le réseau désigne un ensemble d'ordinateurs et parfois de terminaux¹ reliés entre eux de manière à échanger des informations et des programmes ou de partager des ressources (imprimante, scanner).

Les réseaux informatiques sont généralement différenciés à travers l'étendue géographique de leur couverture. C'est ainsi qu'on parle de réseau étendu ou international lorsque les ordinateurs sont distants de plusieurs centaines ou de milliers de kilomètres. Ces types de réseaux sont appelés WAN (Wide Area Network). Le plus connu d'entre eux est l'Internet. Il existe un autre type de réseau dont les ordinateurs ne sont distants que de quelques mètres à quelques kilomètres. Ce sont les réseaux locaux, en anglais LAN (Local Area Network).

Entre ces deux réseaux il peut exister des réseaux intermédiaires tels que les réseaux métropolitains dont les ordinateurs se trouvent dans la même ville.

2. Le protocole :

Le protocole est un ensemble de règles (codes) à respecter pour établir un échange d'informations entre ordinateurs. Ces règles régissent la communication entre les systèmes informatiques. Le plus utilisé par l'Internet est le protocole TCP/ IP (Transmission Control Protocol / Internet Protocol). Le protocole IP permet l'envoi des données par petits paquets. Il existe d'autres types de protocoles tels que le FTP (File Transfert Protocol) et le protocole UUCP (Unix to Unix Copy Program).

3. L'Internet :

Vocabulaire fort prisé aujourd'hui mais dont le sens reste encore peu connu d'un grand public.

Nous avons alors choisi de l'expliquer après les termes réseau et protocole pour en faciliter la Compréhension.

L'Internet est avant tout un réseau physique, c'est à dire un ensemble de réseaux de communication interconnectés dans le monde. Il peut alors être compris au sens de réseau des réseaux en ce sens qu'il relie des millions d'ordinateurs de par le monde et qui n'appartiennent pas forcément à un pays, ne fonctionnent pas forcément avec les mêmes systèmes et ne contiennent pas souvent les mêmes types d'information.

Face à la multiplicité des réseaux d'ordinateurs et de protocoles et pour répondre au besoin d'interconnecter lesdits réseaux, l'Internet se propose de définir un ensemble de protocoles communs à plusieurs réseaux et met en oeuvre une méthode d'interconnexion à la fois physique et logique pour donner lieu à un réseau mondial. Il conviendrait dès lors de définir l'Internet comme un réseau informatique mondial, constitué d'un ensemble de réseaux, reliant des millions d'ordinateurs de types différents mais pouvant échanger des informations. Cette connexion se fait le plus souvent au moyen de lignes téléphoniques publiques et privées.

L'histoire d'Internet remonte à la guerre froide en 1969 avec le programme ARPANET¹ du Pentagone. Aujourd'hui, les réseaux qui constituent l'Internet sont des propriétés d'Instituts universitaires, des agences gouvernementales et internationales, des associations à but non lucratif, des sociétés publiques et privées etc., qui autorisent l'accès à leurs informations.

L'Internet n'est donc la propriété de personne et nul ne peut en réclamer le monopole, même l'armée américaine qui en est à l'origine.

L'Internet regorge de fonctionnalités dont les plus connues sont la messagerie électronique qui permet l'envoi et la réception de courrier, l'accès au forum de discussion, l'échange et le téléchargement de fichiers (FTP), le pilotage d'ordinateur à distance (Telnet), la conversion en temps réel (IRC), le World Wild Web (WWW) etc.

L'Internet est donc défini tantôt comme un magasin de connaissances, tantôt comme une grande poste, une banque de son (discothèque), une banque d'images (vidéothèque)

bibliothèque...l'Internet est pour ainsi dire une « X thèque » puisque contenant toute sorte de données et remplissant à la fois divers rôles sociaux que la science laissera à l'humanité.

4. Le web

Web veut dire en anglais toile d'araignée, c'est également l'abréviation courante de World Wide Web. Cette dernière expression est également abrégée par WWW ou 3W ou encore W3 et les Canadiens l'appellent tout simplement la toile. Cette métaphore renvoie à une toile d'araignée couvrant le monde entier.

Créé en 1989 par le professeur Tim Berners-Lee du Centre Européen de recherche Nucléaire (CERN de Genève), pour améliorer la convivialité de l'exploitation de l'Internet, le Web se définit de par son contenu comme un grand réservoir d'informations exploitables sur l'Internet. C'est donc la partie multimédia de l'Internet permettant à la fois la diffusion de textes, de sons, d'images etc.

Le web est composé de documents hypertextes, (c'est à dire des textes qui contiennent des liens vers d'autres documents textes) et hypermédia (document qui contient des liens vers du texte, du son, des images, de la vidéo etc.) connectés entre eux et circulant à travers l'Internet.

Les liens hypertextes sont des renvois qui, d'un simple clic, nous permettent de circuler d'un endroit à un autre dans une même page web, d'une page à une autre dans le même site ou d'une page à un autre site web. Ils peuvent également nous permettre, s'ils sont liés à une adresse e-mail, de contacter directement la personne ou l'institut auquel ils renvoient.

5. La page Web et le site Web

La page web est le document qui apparaît sur un écran d'ordinateur et qui se présente sous forme de texte, de son d'images etc. Chaque page web est écrite dans un langage particulier appelé le HTML. Ce langage est basé sur des balises (ou tags), qui contiennent des instructions, lesquelles sont interprétées par les différents navigateurs et apparaissent sur votre écran. Il s'agit donc d'un langage de mise en forme du texte. Le code HTML d'une page est appelé le "code source". Les codes sources de chaque page web sont normalement en libre accès, vous pouvez les voir en faisant un clic droit sur la souris, puis cliqué sur : "Affichage le code source" pour Google chrome, puis "Afficher la Source" pour Internet explorer. Le

HTML est le langage de base des pages web, cependant il existe de nombreux autres langages qui interviennent dans la programmation d'une page web (PHP, JavaScript, python etc...).

Le site web quant à lui est constitué de l'ensemble des pages créées par une structure ou un individu reliées entre elles par des liens hypertextes, stockées sur un serveur et accessibles via l'Internet. Chaque site web possède un identificateur appelé adresse URL, permettant de la retrouver sur le Web et d'identifier le type de protocole d'accès qui lui est associé. Le

protocole étant l'ensemble des règles (codes) à respecter pour établir un échange d'informations entre ordinateurs.

Jusqu'en 1996, la plupart des sites hors des États-Unis utilisaient un nom de domaine géographique à deux lettres, par exemple : .sn pour le Sénégal ; .fr pour la France etc. Depuis quelques années plusieurs organisations préfèrent dissocier leur image de celle de leur pays.

Elles choisissent pour cela des noms de domaine liés à leur activité par exemple .com

(commercial) ; .org (organisation) ; .net (Internet). Les noms de domaine sont gérés actuellement par la société américaine Network Solution Inc. (NSI). Les adresses IP qui identifient les ordinateurs sont attribuées par des registres américains ARIN et européens

RIPE.

Depuis le deuxième Forum Africain pour le Développement (ADF) tenu par la CEA (Consumer Electronics Association) à Addis-Abeba en 1997, les professionnels de l'Internet africain réclament une participation de l'Afrique à l'ICANN qui est l'Organisation pour l'Assignation des noms de domaine. Il a été ainsi créé une institution dénommée Africa Network Information Center (AFRINIC) dont le but est « d'allouer et d'enregistrer les ressources Internet dans la région africaine ; assister la communauté africaine dans le développement des procédures, des mécanismes, des normes pour répartir efficacement les ressources Internet et développer les politiques publiques sur l'Internet en Afrique ».

II. La typologie des sites web

La typologie des sites web oppose souvent les sites statiques à ceux dits dynamiques. Il convient juste de présenter ici les propriétés et le mode de fonctionnement de chacune de ces deux architectures afin de permettre à tout un chacun d'opérer un choix en fonction de ses objectifs et ses moyens.

1. Le site statique

Le site statique est constitué de pages HTML prédéfinies, créées une fois pour toutes à l'aide d'un éditeur HTML. Le concepteur du site réalise avec un éditeur HTML des pages web stockées sur le serveur web. Celui-ci renvoie ces pages à la demande au visiteur. Par la suite ces pages ne pourront être modifiées que via un éditeur HTML, par le concepteur. Le contenu de ces pages est fixe comme un fichier Word, et n'est pas modifié par le serveur. Le site est donc dit "statique" car son contenu ne change que par une intervention humaine et non pas par des fonctions automatiques opérées par le serveur.

Le site statique ne présente pas souvent de séparation entre le fond et la forme de présentation du contenu mais sa mise en place nécessite un minimum de compétence en conception web

(Programmation et webdesign). Le coût de réalisation d'un site web statique est beaucoup moins élevé que celui d'un site dynamique et son hébergement ne pose pas de contraintes particulières.

2. Le site dynamique

Le site dynamique est un site Web dont les pages HTML se construisent lors de sa consultation par un internaute en sollicitant des bases de données filtrées par des outils logiciels de mise en forme.

La mise à jour des sites dynamiques tant sur le fond que sur la forme est facilitée. Le site dynamique permet de plus d'intégrer des fonctions de personnalisation. Cette architecture peut en revanche se révéler coûteuse à mettre en place, délicate à piloter notamment au niveau de l'hébergement.

Le gestionnaire du site et son visiteur utilisent le même outil : le navigateur web. Cependant les pages avec lesquelles travaille l'administrateur ne sont pas les mêmes que celles que le

visiteur voit : il encode dans des pages web sous la forme de "formulaires", qui alimentent une base de données dans laquelle sont stockées des informations. De son côté, le visiteur visualise des pages qui font appel au contenu de la base de données. C'est le serveur web qui s'occupe de récupérer le contenu des formulaires que remplit l'administrateur, et de renvoyer cette information dans les pages vues par le visiteur.

Le site web dynamique est fortement recommandé pour les portails à forte volumétrie ou intégrant des flux d'origines diverses ce qui est souvent le cas des portails documentaires, sites marchand, Forum de discussion etc.

La mise à jour des sites dynamiques est facilitée grâce aux outils d'administration. Elle (mise à jour) ne nécessite pas une compétence technique particulière.

Les bases de HTML5

Vous n'avez jamais entendu parler du HTML, ou alors seulement de façon très vague ? Pas de panique, les explications arrivent dès le premier chapitre... et la pratique suit juste après ! ;)

Nous commencerons par présenter comment les sites web fonctionnent, puis nous téléchargerons tous les programmes (gratuits) nécessaires pour bien travailler. À la fin de cette partie, vous saurez déjà insérer du texte, des liens et des images !

Comment fait-on pour créer des sites web ?

Visionner la vidéo du Chapitre 1 de la Partie 1 sur Vimeo

Bonjour et bienvenue à toutes et à tous !

Voici donc le premier chapitre de ce cours pour débutants, qui va vous apprendre à créer votre site web !

Nous allons passer un certain temps ensemble, tout dépendra de la vitesse à laquelle vous apprendrez. Si vous lisez ce cours régulièrement et à une bonne vitesse, vous l'aurez terminé en une à deux semaines. Mais si vous avez besoin d'un peu plus de temps, ne vous inquiétez pas : le principal est que vous y alliez à votre rythme, de préférence en prenant du bon temps.

Je vous propose de commencer par la question la plus simple mais aussi la plus importante : **comment fonctionnent les sites web ?**

Le fonctionnement des sites web

Comment fonctionnent les sites web ?

Non, n'ayez pas peur de poser des questions même si vous pensez qu'elles sont « bêtes ». Il est très important que nous en parlions un peu avant de nous lancer à fond dans la création de sites !

Je suis certain que vous consultez des sites web tous les jours. Pour cela, vous lancez un programme appelé le navigateur web en cliquant sur l'une des icônes représentées à la figure suivante.



Les icônes des navigateurs web les plus répandus

Avec le navigateur, vous pouvez consulter n'importe quel site web. Voici par exemple un navigateur affichant le célèbre site web Wikipédia :



Le site web Wikipédia

Je suis sûr que vous avez l'habitude d'utiliser un navigateur web ! Aujourd'hui, tout le monde sait aller sur le Web... mais qui sait vraiment comment le Web fonctionne ? Comment créer des sites web comme celui-ci ?

J'ai entendu parler de HTML, de CSS, est-ce que cela a un rapport avec le fonctionnement des sites web ?

Tout à fait ! Il s'agit de **langages informatiques** qui permettent de créer des sites web. Tous les sites web sont basés sur ces langages, ils sont incontournables et universels aujourd'hui. Ils sont à la base même du Web. Le langage HTML a été inventé par un certain Tim Berners-Lee en 1991...

Tim Berners-Lee suit encore aujourd'hui avec attention l'évolution du Web. Il a créé le *World Wide Web Consortium* (**W3C**) qui définit les nouvelles versions des langages liés au Web. Il a par ailleurs créé plus récemment la *World Wide Web Foundation* qui analyse et suit l'évolution du Web.

De nombreuses personnes confondent (à tort) Internet et le Web. Il faut savoir que le Web *fait partie* d'Internet.

Internet est un grand ensemble qui comprend, entre autres : le Web, les e-mails, la messagerie instantanée, etc.

Tim Berners-Lee n'est donc pas l'inventeur d'Internet, c'est « seulement » l'inventeur du Web.

Les langages HTML et CSS sont à la base du fonctionnement de tous les sites web. Quand vous consultez un site avec votre navigateur, il faut savoir que, en coulisses, des rouages s'activent pour permettre au site web de s'afficher. L'ordinateur se base sur ce qu'on lui a expliqué en HTML et CSS pour savoir ce qu'il doit afficher, comme le montre la figure suivante.

[illegible]

Wikipedia:fr:encyclopédie

fr.wikipedia.org/wiki/Wikipedia:Accueil_principal

Accueil · Discussion · Lire · Voir les outils · Afficher l'historique · Recherche

Wiki Loves Monuments. Photographiez un monument historique, aidez Wikipedia et gagnez !

Bienvenue sur Wikipedia
Le projet d'encyclopédie libre que tout doit connaître

1 152 912 articles en français
dernier jour: 2009-06-06

Culture — Géographie — Histoire — Sciences — Société — Technologies
Liste des portails thématiques

Contribuer · Prendre avis · Aide · Communauté · Wikifications · Références · Faire un don · Imprimer / exporter · S'abonner · Aide · Aide à l'écriture · Aide à l'écriture

Lumière sur

La bataille de Tewkesham a lieu au cours de la guerre des Deux-Roses le 28 mars 1471 au sud-ouest d'York, entre les villages de Tewkes et Stratton. C'est la plus grande et la plus sanglante bataille à avoir jamais été livrée sur le sol anglais avant que la journée la plus meurtrière de notre histoire ne s'achève, dans les comarques médianes, plus de 60 000 soldats des Mairies d'York et de Lancastre se sont combattus, ce dimanche du Vendredi pendant plusieurs heures et dans des conditions météorologiques déplorables, et une proclamation diffusée une semaine après la bataille rapporte que 28 000 hommes ont péri sur le champ de bataille. Cet engagement provoqua un changement momentané en Angleterre, Édouard IV remporta sur Henri VI sur le trône et montra les préoccupations des Lancastre à l'extérieur.

Henri VI a un caractère faible et ne dispose pas de toutes ses facultés mentales.

W Présentation

Wikipedia est un projet d'encyclopédie collective établie sur Internet, universel, multilingue et fonctionnant sur le principe du wiki. Wikipedia a pour objectif d'offrir un contenu constamment mis à jour, copié et vérifié, que chacun peut modifier et améliorer.

Le texte du projet est défini par des principes fondateurs. Son contenu est sous licence Creative Commons by-sa et peut être copié et réutilisé sous la même licence – même à des fins commerciales – sans aucune d'œ

9

HTML et CSS sont deux « langues » qu'il faut savoir parler pour créer des sites web. C'est le navigateur web qui fera la traduction entre ces langages informatiques et ce que vous verrez s'afficher à l'écran. Vous vous demandez sûrement pourquoi il faut connaître deux langages pour créer des sites web ? Je vous réponds sans plus tarder !

HTML et CSS : deux langages pour créer un site web

Pour créer un site web, on doit donner des instructions à l'ordinateur. Il ne suffit pas simplement de taper le texte qui devra figurer dans le site (comme on le ferait dans un traitement de texte Word, par exemple), il faut aussi indiquer où placer ce texte, insérer des images, faire des liens entre les pages, etc.

Les rôles de HTML et CSS

Pour expliquer à l'ordinateur ce que vous voulez faire, il va falloir utiliser un langage qu'il comprend. Et c'est là que les choses se corsent, parce qu'il va falloir apprendre deux langages !

Pourquoi avoir créé deux langages ? Un seul aurait suffi, non ?

Vous devez vous dire que manipuler deux langages va être deux fois plus complexe et deux fois plus long à apprendre... mais ce n'est pas le cas ! Je vous rassure, s'il y a deux langages c'est, au contraire, pour faciliter les choses. Nous allons avoir affaire à deux langages qui se complètent car ils ont des rôles différents :

- **HTML** (*HyperText Markup Language*) : il a fait son apparition dès 1991 lors du lancement du Web. Son rôle est de gérer et organiser le contenu. C'est donc en HTML que vous écrirez ce qui doit être affiché sur la page : du texte, des liens, des images... Vous direz par exemple : « Ceci est mon titre, ceci est mon menu, voici le texte principal de la page, voici une image à afficher, etc. ».
- **CSS** (*Cascading Style Sheets*, aussi appelées *Feuilles de style*) : le rôle du CSS est de gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte...). Ce langage est venu compléter le HTML en 1996.

Vous avez peut-être aussi entendu parler du langage XHTML. Il s'agit d'une variante du HTML qui se veut plus rigoureuse et qui est donc un peu plus délicate à manipuler. Elle n'est plus vraiment utilisée aujourd'hui.

Dans ce cours, nous allons travailler sur la dernière version de HTML (HTML5) qui est aujourd'hui le langage d'avenir que tout le monde est incité à utiliser.

Vous pouvez très bien créer un site web uniquement en HTML, mais celui-ci ne sera pas très beau : l'information apparaîtra « brute ». C'est pour cela que le langage CSS vient toujours le compléter.

Pour vous donner une idée, la figure suivante montre ce que donne la même page sans CSS puis avec le CSS.

HTML
(pas de CSS)



HTML + CSS



Avec et sans CSS

Le HTML définit le contenu (comme vous pouvez le voir, c'est brut de décoffrage !). Le CSS permet, lui, d'arranger le contenu et de définir la présentation : couleurs, image de fond, marges, taille du texte...

Comme vous vous en doutez, le CSS a besoin d'une page HTML pour fonctionner. C'est pour cela que nous allons d'abord apprendre les bases du HTML avant de nous occuper de la décoration en CSS. Vos premières pages ne seront donc pas les plus esthétiques, mais qu'importe ! Cela ne durera pas longtemps.

Les différentes versions de HTML et CSS

Au fil du temps, les langages HTML et CSS ont beaucoup évolué. Dans la toute première version de HTML (HTML 1.0) il n'était même pas possible d'afficher des images !

Voici un très bref historique de ces langages pour votre culture générale.

Les versions de HTML

- **HTML 1** : c'est la toute première version créée par Tim Berners-Lee en 1991.
- **HTML 2** : la deuxième version du HTML apparaît en 1994 et prend fin en 1996 avec l'apparition du HTML 3.0. C'est cette version qui posera en fait les bases des versions suivantes du HTML. Les règles et le fonctionnement de cette version sont donnés par le W3C (tandis que la première version avait été créée par un seul homme).
- **HTML 3** : apparue en 1996, cette nouvelle version du HTML rajoute de nombreuses possibilités au langage comme les tableaux, les applets, les scripts, le positionnement du texte autour des images, etc.
- **HTML 4** : cette version aura été utilisée un long moment durant les années 2000. Elle apparaît pour la première fois en 1998 et propose l'utilisation de frames (qui découpent une page web en plusieurs parties), des tableaux plus complexes, des améliorations sur les formulaires, etc. Mais surtout, cette version permet pour la première fois d'exploiter des feuilles de style, notre fameux CSS !
- **HTML 5** : c'est la dernière version. De plus en plus répandue, elle fait beaucoup parler d'elle car elle apporte de nombreuses améliorations comme la possibilité d'inclure facilement des vidéos, un meilleur agencement du contenu, de nouvelles fonctionnalités pour les formulaires, etc. C'est cette version que nous allons découvrir ensemble.

Les versions de CSS

- **CSS 1** : dès 1996, on dispose de la première version du CSS. Elle pose les bases de ce langage qui permet de présenter sa page web, comme les couleurs, les marges, les polices de caractères, etc.
- **CSS 2** : apparue en 1999 puis complétée par CSS 2.1, cette nouvelle version de CSS rajoute de nombreuses options. On peut désormais utiliser des techniques de positionnement très précises, qui nous permettent d'afficher des éléments où on le souhaite sur la page.
- **CSS 3** : c'est la dernière version, qui apporte des fonctionnalités particulièrement attendues comme les bordures arrondies, les dégradés, les ombres, etc.

Notez que HTML5 et CSS3 ne sont pas encore des versions "officiellement" finalisées par le W3C. Cependant, même s'il peut y avoir des changements mineurs dans ces langages, je vous recommande chaudement de commencer dès aujourd'hui avec ces nouvelles versions. Leurs apports sont nombreux et valent vraiment le coup. La plupart des sites web professionnels se construisent aujourd'hui sur ces dernières versions.

L'éditeur de texte

De quel logiciel vais-je avoir besoin pour créer mon site web ? Vais-je devoir casser ma tirelire pour acheter un logiciel très complexe que je vais mettre des mois à comprendre ?

Il existe effectivement de nombreux logiciels dédiés à la création de sites web. Mais, je vous rassure, vous n'aurez pas à débours un seul centime. Pourquoi aller chercher un logiciel payant et compliqué, alors que vous avez déjà tout ce qu'il faut chez vous ?

Eh oui, accrochez-vous bien parce qu'il suffit de... Bloc-Notes !



Le logiciel Bloc-notes de Windows

Incroyable mais vrai : on peut tout à fait créer un site web uniquement avec Bloc-Notes, le logiciel d'édition de texte intégré par défaut à Windows. D'ailleurs, j'avoue, c'est comme cela que j'ai commencé moi-même il y a quelques années.

Il y a cependant des logiciels plus puissants aujourd'hui et personne n'utilise vraiment Bloc-Notes. On peut classer ces logiciels de **création de site web** en deux catégories :

- Les **WYSIWYG** (*What You See Is What You Get* - Ce Que Vous Voyez Est Ce Que Vous Obtenez) : ce sont des programmes qui se veulent très faciles d'emploi, ils permettent de créer des sites web sans apprendre de langage particulier. Parmi les plus connus d'entre eux : Mozilla Composer, Microsoft Expression Web, Dreamweaver... et même Word ! Leur principal défaut est la qualité souvent assez mauvaise du code HTML et CSS qui est automatiquement généré par ces outils. Un bon créateur de site web doit tôt ou tard connaître HTML et CSS, c'est pourquoi je ne recommande pas l'usage de ces outils.

- Les **éditeurs de texte** : ce sont des programmes dédiés à l'écriture de code. On peut en général les utiliser pour de multiples langages, pas seulement HTML et CSS. Ils se révèlent être de puissants alliés pour les créateurs de sites web !

Vous l'aurez compris, je vais vous inviter à utiliser un éditeur de texte dans ce cours.

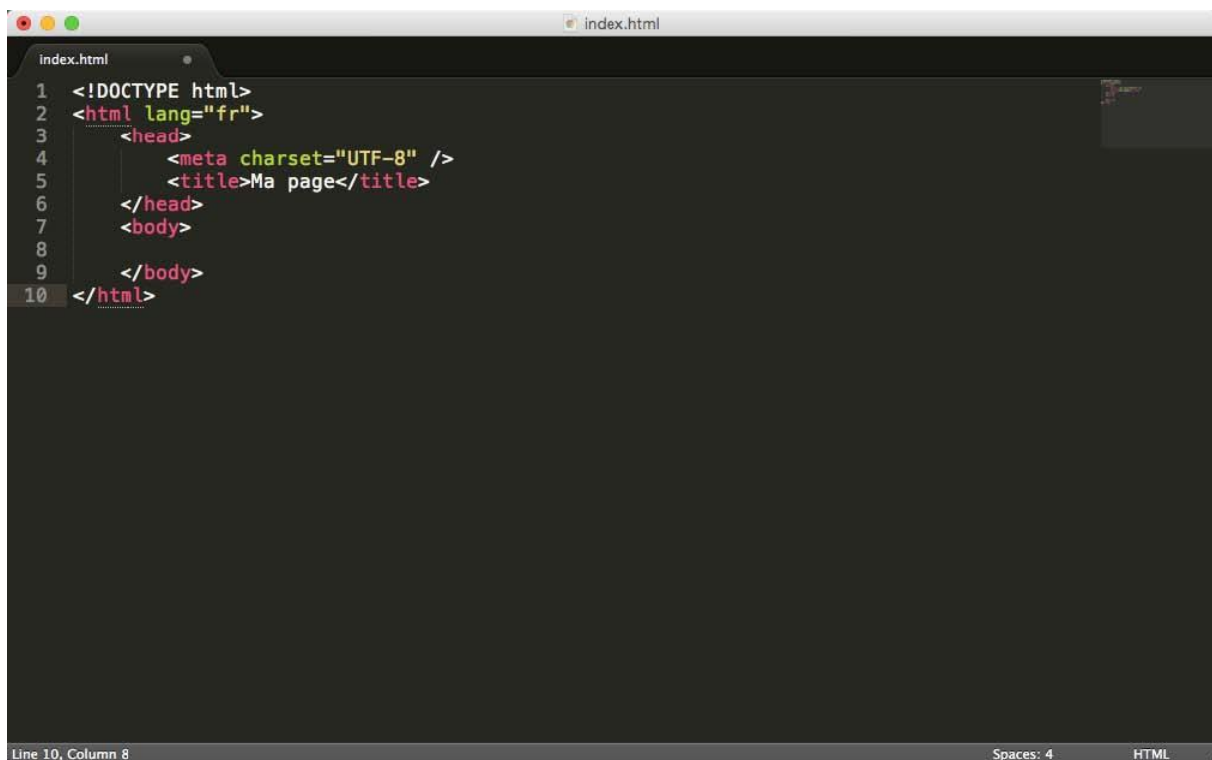
De nombreux éditeurs de texte fonctionnent que vous soyez sous Windows, Mac OS X ou Linux (ils sont disponibles partout). Je vais vous en proposer plusieurs pour que vous ayez le choix, mais, comme je sais que vous allez me demander celui que j'utilise, je vais commencer par vous présenter... Sublime Text !

Sublime Text : mon éditeur

Sublime Text est un éditeur de texte devenu très populaire parmi les développeurs. On l'utilise aussi bien pour développer en HTML et CSS que dans d'autres langages (Python, Ruby, etc.). Il fonctionne sur Windows, MacOS X et Linux.

Site web de Sublime Text

Il a l'avantage d'être simple, épuré et facile à lire dès le départ. Pas de centaines de boutons dont on ne comprend pas à quoi ils servent.



L'éditeur Sublime Text : c'est beau, c'est propre, c'est pur

Malgré les apparences, il ne faut pas croire qu'il est limité. Au contraire : il est possible de l'étendre avec tout un système de plugins. Ca devient un peu plus compliqué et on ne rentrera pas là-dedans, mais il faut savoir que certains personnalisent énormément leur Sublime Text pour gagner du temps !

En somme, **Sublime Text est à la fois simple et puissant**. Même pour l'usage basique que nous allons avoir, il s'avèrera très pratique.

Sublime Text peut tout à fait être utilisé gratuitement, mais son auteur demande à payer au bout d'un certain temps d'usage. Vous pourrez toujours continuer à l'utiliser gratuitement mais de temps en temps un écran vous rappellera que ce serait bien de payer pour le logiciel. ;)

Personnellement, je considère qu'il en vaut vraiment le coup et je l'ai acheté. Je vous laisse choisir et vous faire votre propre idée à ce sujet.

Sous Windows

Voici quelques logiciels que vous pouvez essayer sous Windows si vous voulez en tester plusieurs :

- Sublime Text (j'insiste ;o) ;
- Notepad++ ;
- Brackets ;
- jEdit ;
- PSPad ;
- ConTEXT ;
- ... et bien d'autres si vous recherchez « Éditeur de texte » sur le Web.

Sous Mac OS X

Je recommande la plupart des mêmes logiciels car ils sont multi-plateformes. Voici une petite sélection :

- Sublime Text ;
- Brackets ;
- jEdit ;
- Smultron ;
- TextWrangler.

Sous Linux

Les éditeurs de texte sont légion sous Linux. Certains d'entre eux sont installés par défaut, d'autres peuvent être téléchargés facilement *via* le centre de téléchargement (sous Ubuntu notamment) ou au moyen de commandes comme `apt-get` et `aptitude`. Voici quelques logiciels que vous pouvez tester :

- Sublime Text ;
- Brackets ;
- gEdit ;
- Kate ;
- vim ;
- Emacs ;
- jEdit.

Les navigateurs




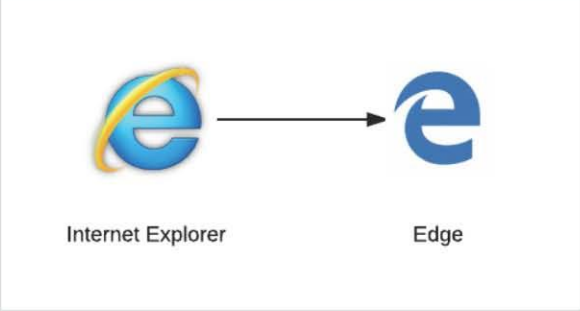


Pourquoi le navigateur est important

Le navigateur est le programme qui nous permet de voir les sites web. Comme je vous l'ai expliqué plus tôt, le travail du navigateur est de lire le code HTML et CSS pour afficher un résultat visuel à l'écran. Si votre code CSS dit « Les titres sont en rouge », alors le navigateur affichera les titres en rouge. Le rôle du navigateur est donc essentiel !

On ne dirait pas, mais un navigateur est un programme extrêmement complexe. En effet, comprendre le code HTML et CSS n'est pas une mince affaire. Le principal problème, vous vous en rendrez vite compte, c'est que *les différents navigateurs n'affichent pas le même site exactement de la même façon* ! Il faudra vous y faire et prendre l'habitude de vérifier régulièrement que votre site fonctionne correctement sur la plupart des navigateurs.

Les navigateurs sur ordinateur Télécharger les navigateurs

Il existe de nombreux navigateurs différents. Voici les principaux à connaître :

Navigateur	OS	Téléchargement	Commentaires
Google Chrome 	Windows Mac Linux	Téléchargement	Le navigateur de Google, simple d'emploi et très rapide. C'est le navigateur que j'utilise au quotidien.
Mozilla Firefox 	Windows Mac Linux	Téléchargement	Le navigateur de la fondation Mozilla, célèbre et réputé. Je l'utilise fréquemment pour tester mes sites web.
Internet Explorer 	Windows	(Déjà installé sur Windows)	Le navigateur de Microsoft, qui équipe tous les PC Windows jusqu'à Windows 10.
Edge 	Windows	(Déjà installé sur Windows 10)	<p>Le nouveau navigateur de Microsoft, qui équipe tous les PC à partir de Windows 10. Il ressemble à Internet Explorer (les logos sont proches !) mais c'est une toute nouvelle version bien plus à jour. Edge est le remplaçant d'Internet Explorer.</p> <div data-bbox="794 1131 1382 1440">  </div> <p>Internet Explorer est remplacé par Edge</p>
Safari 	Windows Mac	Téléchargement (Déjà installé sur Mac OS X)	Le navigateur d'Apple, qui équipe tous les Mac.
Opera 	Windows Mac Linux	Téléchargement	L'éternel <i>outsider</i> . Il est moins utilisé mais propose de nombreuses fonctionnalités.

Il est conseillé d'installer plusieurs navigateurs sur son ordinateur pour s'assurer que son site fonctionne

correctement sur chacun d'eux. De manière générale, je conseille de tester son site web régulièrement au moins sur Google Chrome, Mozilla Firefox et Internet Explorer/Edge. Notez que Safari et Google Chrome affichent les sites web quasiment de la même façon. Il n'est pas forcément nécessaire de tester son site sur Safari et Google Chrome, même si c'est toujours plus sûr.

La figure suivante vous montre un aperçu du résultat produit par quelques-uns de ces principaux navigateurs sur la page d'accueil de Google.



Aperçu de quelques navigateurs

Comprendre les différences entre navigateurs

A vue de nez, ces navigateurs se ressemblent beaucoup. Mais comme je vous le disais plus tôt, les navigateurs n'affichent pas toujours un même site web *exactement* de la même façon. Pourquoi ? Cela est dû au fait que les navigateurs ne connaissent pas toujours les dernières fonctionnalités de HTML et CSS. Par exemple, Internet Explorer a longtemps été en retard sur certaines fonctionnalités CSS (et paradoxalement, il a aussi été en avance sur quelques autres).

Pour compliquer les choses, **plusieurs versions des navigateurs co-existent**. Aujourd'hui, un navigateur comme Chrome sort une nouvelle version presque tous les mois. Les mises à jour sont (heureusement) de plus en plus fréquentes.

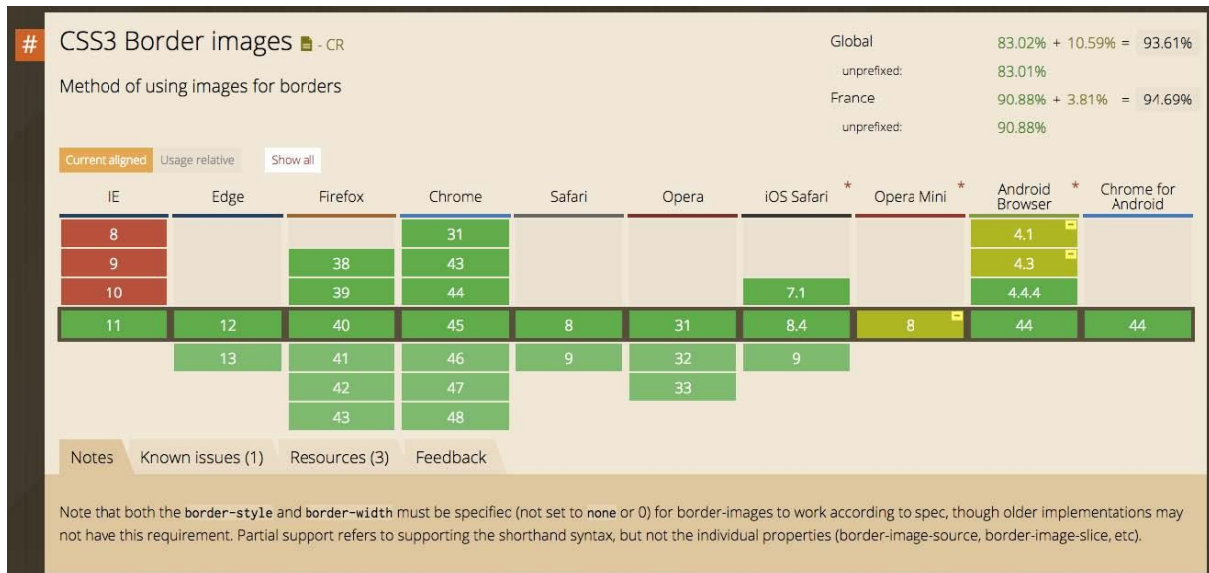
Chaque version prend en charge de nouvelles fonctionnalités mais, si les utilisateurs ne mettent pas à jour leur(s) navigateur(s), cela devient un problème pour les **webmasters** comme vous qui créez des sites web.

Chrome a résolu en grande partie le problème en mettant en place des mises à jour automatiques, sans intervention de l'utilisateur. Firefox a semble-t-il décidé de suivre le rythme lui aussi. Internet Explorer est de plus en plus à jour, et son remplaçant Edge n'a pas à rougir des autres navigateurs.

Bref, j'ai un peu le sentiment de parler comme un ancêtre du web en disant ça, mais on a beaucoup de chance aujourd'hui : les navigateurs supportent un grand nombre de

fonctionnalités. La compatibilité reste toujours un problème malgré tout, mais ce n'est pas aussi grave qu'à une époque.

Le célèbre site caniuse.com tient à jour une liste des fonctionnalités prises en charge par les différentes versions de chaque navigateur (figure suivante).



caniuse.com vous permet de connaître la compatibilité d'une fonctionnalité

Ca peut paraître un peu compliqué, mais comme je vous le disais les navigateurs supportent aujourd'hui très bien un grand nombre de fonctionnalités. Les problèmes viennent le plus souvent d'anciennes versions d'Internet Explorer (IE7, IE8...) mais celles-ci sont si peu utilisées que je vous recommande de les ignorer.

Il est possible de tester son site sous le navigateur Internet Explorer à l'aide d'une machine virtuelle comme [VirtualBox](https://www.virtualbox.org/) (gratuit). Le site modern.ie de Microsoft offre des "images disque" qui vous permettent de faire tourner sur votre ordinateur n'importe quelle version de Windows avec Internet Explorer ou Edge. Attention cependant : ces images sont grosses et consomment de la mémoire.

Les navigateurs sur mobile

En plus des navigateurs que je vous ai présentés, il faut savoir qu'il existe des variantes de ces navigateurs conçues pour les téléphones portables, en particulier pour les **smartphones**. De plus en plus de personnes consultent aujourd'hui des sites web sur leur portable, il faut donc connaître un minimum le fonctionnement des navigateurs des téléphones.

En fait, vous n'allez pas être dépayés : la plupart des navigateurs sur smartphones sont les mêmes que sur ordinateur, dans une version plus légère adaptée aux mobiles. Tout dépend du type de téléphone.

- **iPhone** : sur l'iPhone d'Apple, le navigateur utilisé est Safari Mobile. Il s'agit d'une version *light* et néanmoins très complète de Safari pour ordinateur.

- **Android** : les portables sous Android bénéficient du navigateur Chrome Mobile. Là encore, il s'agit d'une version adaptée aux mobiles.
- **Windows Phone** : sous Windows Phone, on retrouve... Internet Explorer/Edge Mobile ! Le principe est le même que pour les précédents navigateurs : il s'agit d'une version dédiée aux mobiles.
- **Blackberry** : les Blackberry font exception car ils ont leur propre navigateur (il n'existe pas d'équivalent sur ordinateur). Néanmoins, les versions les plus récentes de ce navigateur se basent sur un noyau commun à Safari et Chrome (il s'agit du moteur de rendu Webkit). Par conséquent, l'affichage est en général proche de celui proposé par Safari et Chrome. Enfin, il faut reconnaître que les Blackberry sont de moins en moins utilisés.

Les navigateurs pour mobiles prennent en charge la plupart des dernières fonctionnalités de HTML et CSS. De plus, le système de mise à jour automatisé des mobiles nous garantit que les utilisateurs auront le plus souvent les dernières versions.

Sachez néanmoins que des différences existent entre ces différents navigateurs mobiles et qu'il est conseillé de tester son site sur ces appareils aussi ! En particulier, l'écran étant beaucoup moins large, il faudra vérifier que votre site s'affiche correctement.

Les tablettes tactiles sont équipées des mêmes navigateurs, l'écran est simplement plus large. Ainsi, l'iPad

est fourni avec Safari Mobile.

En résumé

- Le Web a été inventé par Tim Berners-Lee au début des années 1990.
- Pour créer des sites web, on utilise deux langages informatiques :
 - o HTML : permet d'écrire et organiser le contenu de la page (paragraphe, titres...);
 - o CSS : permet de mettre en forme la page (couleur, taille...).
- Il y a eu plusieurs versions des langages HTML et CSS. Les dernières versions sont HTML5 et CSS3.
- Le navigateur web est un programme qui permet d'afficher des sites web. Il lit les langages HTML et CSS pour savoir ce qu'il doit afficher.
- Il existe de nombreux navigateurs web différents : Google Chrome, Mozilla Firefox, Internet Explorer, Safari, Opera... Chacun affiche un site web de manière légèrement différente des autres navigateurs.

- Dans ce cours, nous allons apprendre à utiliser les langages HTML et CSS. Nous travaillerons dans un programme appelé « éditeur de texte » (Sublime Text, Notepad++, jEdit, vim...).

Votre première page web en HTML

Visionner la vidéo du Chapitre 2 de la Partie 1 sur Vimeo

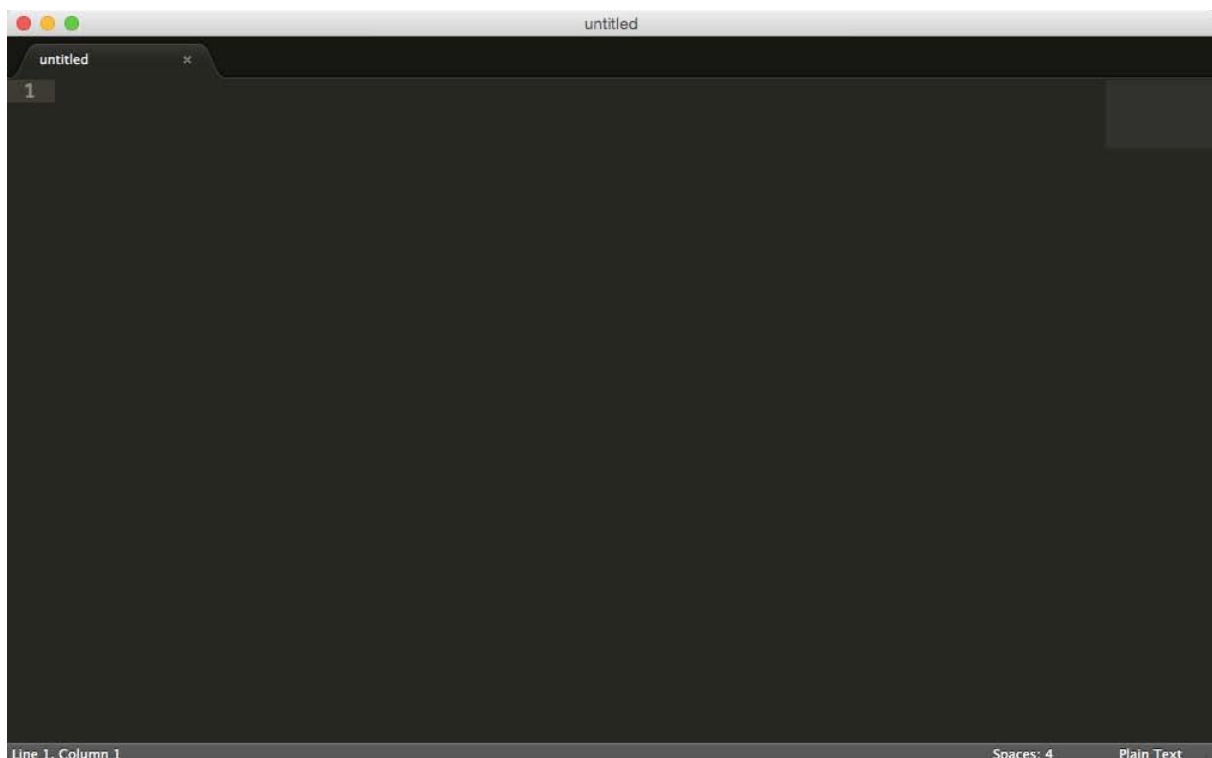
Ça y est, vous avez installé tous les logiciels ? Vous devriez maintenant avoir un éditeur de texte pour *créer votre site* (comme Sublime Text) et plusieurs navigateurs pour le *tester* (Mozilla Firefox, Google Chrome...).

Dans ce chapitre, nous allons commencer à pratiquer ! Nous allons découvrir les bases du langage HTML et enregistrer notre toute première page web ! Alors oui, bien sûr, ne vous attendez pas encore à réaliser une page web exceptionnelle dès ce second chapitre, mais patience... cela viendra !

Créer une page web avec l'éditeur

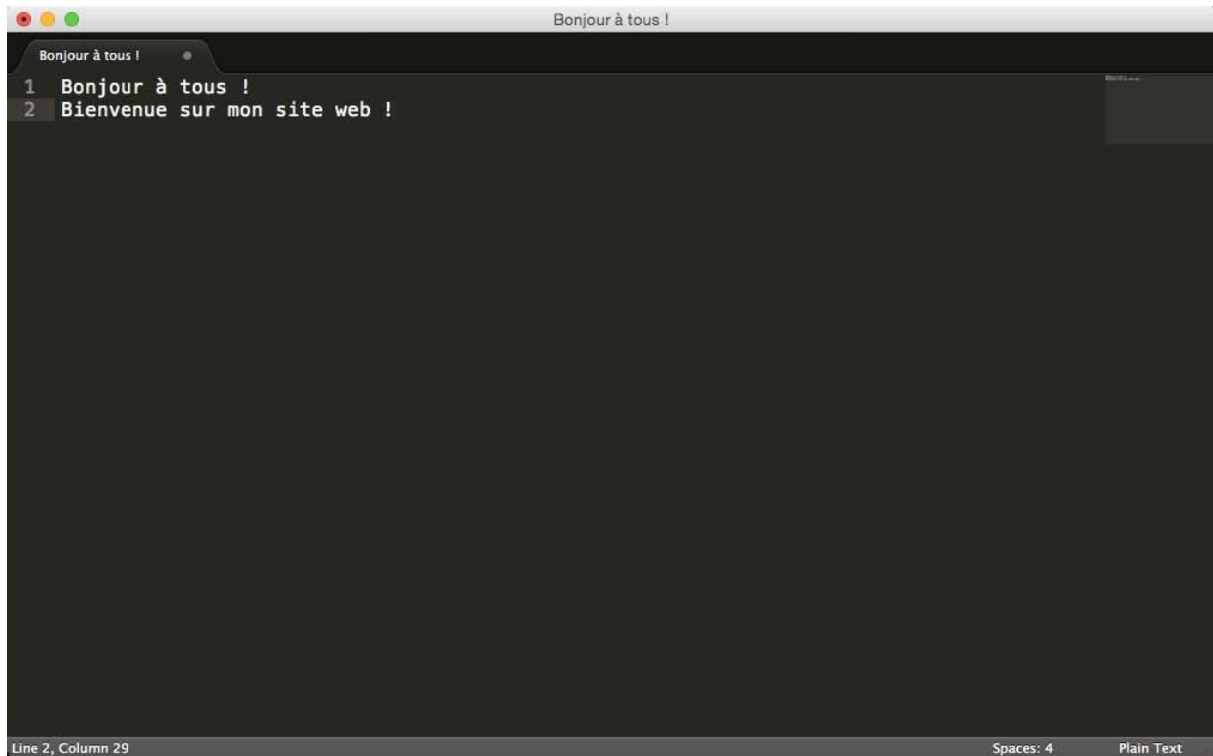
Allez, mettons-nous en situation ! Comme je vous l'ai dit, nous allons créer notre site dans un éditeur de texte. Vous avez dû en installer un suite à mes conseils dans le premier chapitre : qu'il s'appelle Sublime Text, Notepad++, Brackets, jEdit, vim, TextWrangler... peu importe. Ces logiciels ont un but très simple : vous permettre d'écrire du texte !

Dans la suite de ce cours, je travaillerai avec Sublime Text. Je vais donc l'ouvrir :



Ouverture de Sublime Text

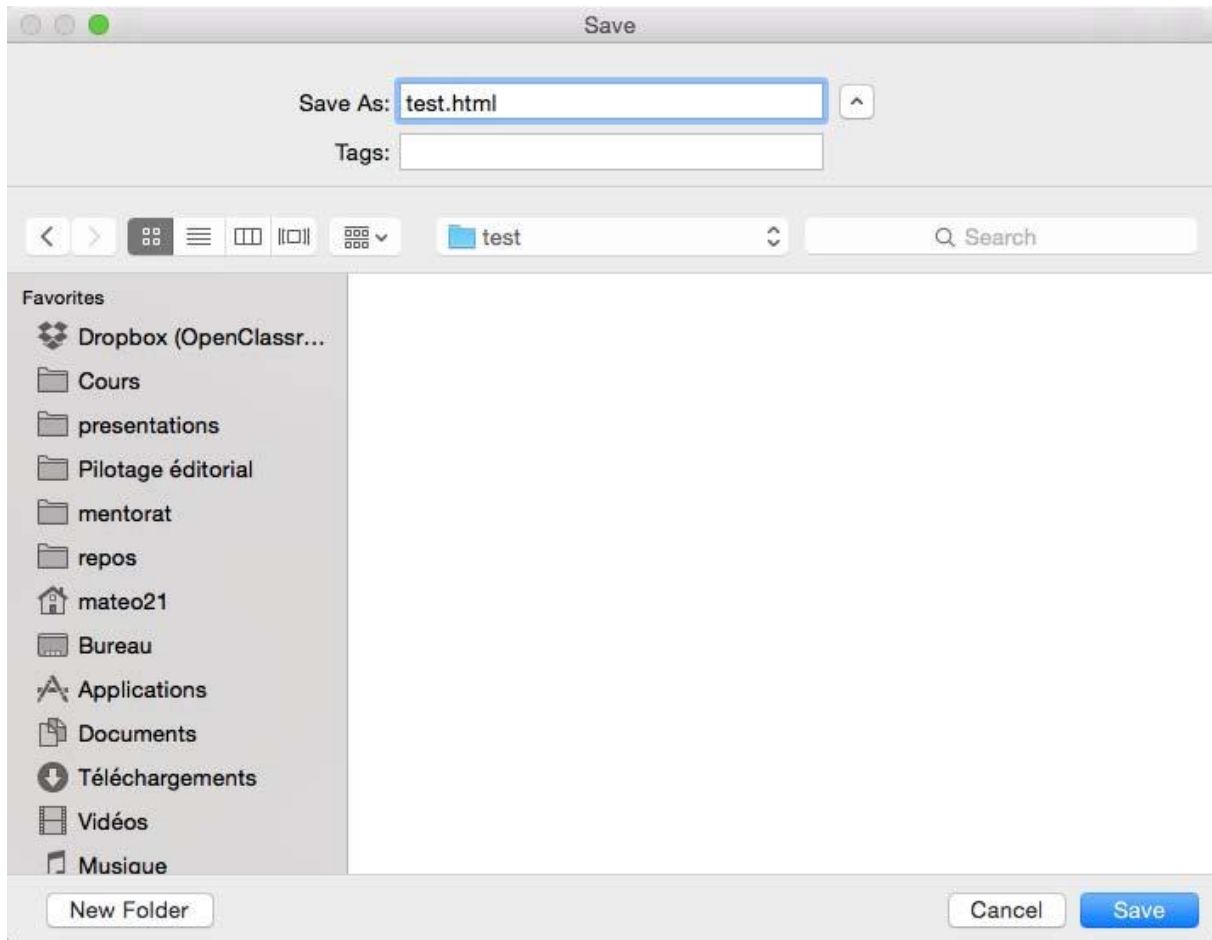
Bon, qu'est-ce qu'on fait maintenant ? Qu'est-ce qu'on écrit sur cette feuille blanche (euh... noire) ? On va faire un petit essai. Je vous invite à écrire ce qui vous passe par la tête, comme moi à la figure suivante.



Du texte dans Sublime Text

Vous pouvez écrire les mêmes phrases que moi ou ce que vous voulez ; le but est d'écrire quelque chose.

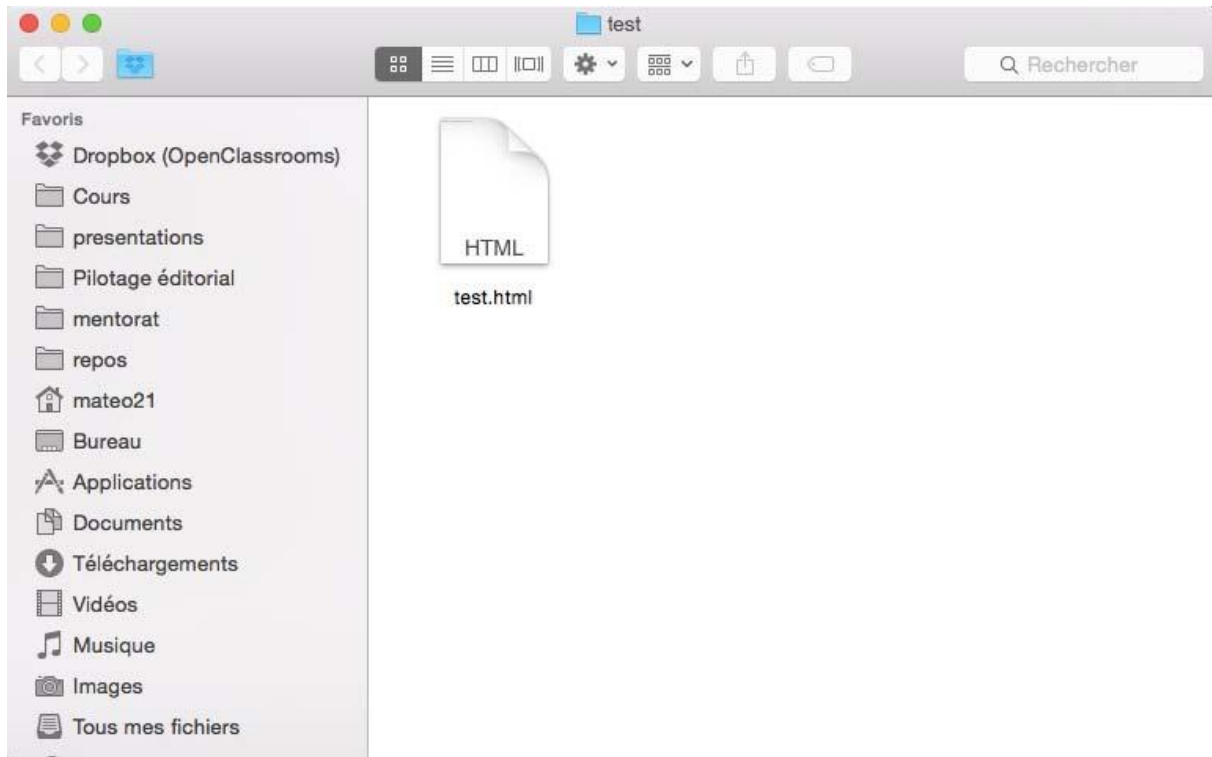
Maintenant, enregistrons ce fichier. Pour cela, c'est très simple : comme dans tous les programmes, vous avez un menu Fichier > Enregistrer (ou File > Save en anglais). Une boîte de dialogue vous demande où enregistrer le fichier et sous quel nom. Enregistrez-le où vous voulez. Donnez au fichier le nom que vous voulez, en terminant par .html, par exemple test.html, comme indiqué à la figure suivante.



Enregistrement d'un fichier sous Sublime Text

Je vous recommande de créer un nouveau dossier dans vos documents qui contiendra les fichiers de votre site. Pour ma part j'ai créé un dossier `test` dans lequel j'ai mis mon fichier `test.html`.

Ouvrez maintenant l'explorateur de fichiers dans le dossier où vous avez enregistré votre page (l'Explorateur sous Windows ou le Finder sous Mac). Vous y verrez le fichier que vous venez de créer :



Le fichier dans l'explorateur

L'icône qui représente le fichier dépend de votre navigateur web par défaut en général. Vous pouvez y voir une icône de Firefox, de Chrome... ou un aperçu comme ici. N'y prêtez pas attention. Faites simplement un double-clic sur ce fichier et... votre navigateur s'ouvre et affiche le texte que vous avez écrit.



La page web affichée

Cela ne marche pas bien, on dirait ! Tout le texte s'affiche sur la même ligne alors qu'on avait écrit deux lignes de texte différentes !?

En effet, bien vu ! Le texte s'affiche sur la même ligne alors qu'on avait demandé à l'écrire sur deux lignes différentes. Que se passe-t-il ?

En fait, pour créer une page web il ne suffit pas de taper simplement du texte comme on vient de le faire. En plus de ce texte, il faut aussi écrire ce qu'on appelle des **balises**, qui vont donner des instructions à l'ordinateur comme « aller à la ligne », « afficher une image », etc.

Les balises et leurs attributs

Bon, tout cela était trop facile. Évidemment, il a fallu que ces satanés informaticiens s'en mêlent et compliquent les choses. Il ne suffit pas d'écrire « simplement » du texte dans l'éditeur, il faut aussi donner des instructions à l'ordinateur. En HTML, on utilise pour cela des **balises**.

Les balises

Les pages HTML sont remplies de ce qu'on appelle des balises. Celles-ci sont invisibles à l'écran pour vos visiteurs, mais elles permettent à l'ordinateur de comprendre ce qu'il doit afficher. Les balises se repèrent facilement. Elles sont entourées de « chevrons », c'est-à-dire des symboles `<et >`, comme ceci : `<balise>`

À quoi est-ce qu'elles servent ? Elles indiquent la nature du texte qu'elles encadrent. Elles veulent dire par exemple : « Ceci est le titre de la page », « Ceci est une image », « Ceci est un paragraphe de texte », etc.

On distingue deux types de balises : les balises en paires et les balises orphelines.

Les balises en paires

Elles s'ouvrent, contiennent du texte, et se ferment plus loin. Voici à quoi elles ressemblent :

```
<titre>Ceci est un titre</titre>
```

On distingue une balise ouvrante (`<titre>`) et une balise fermante (`</titre>`) qui indique que le titre se termine. Cela signifie pour l'ordinateur que tout ce qui n'est *pas* entre ces deux balises... n'est pas un titre.

```
Ceci n'est pas un titre <titre>Ceci est un titre</titre> Ceci n'est pas un titre
```

Les balises orphelines

Ce sont des balises qui servent le plus souvent à insérer un élément à un endroit précis (par exemple une image). Il n'est pas nécessaire de délimiter le début et la fin de l'image, on veut juste dire à l'ordinateur « Insère une image ici ».

Une balise orpheline s'écrit comme ceci :

```
<image />
```

Notez que le / de fin n'est pas obligatoire. On pourrait écrire seulement `<image>`. Néanmoins, pour ne pas les confondre avec le premier type de balise, les webmasters recommandent de rajouter ce / (*slash*) à la fin des balises orphelines. Vous me verrez donc mettre un / aux balises orphelines et je vous recommande de faire de même, c'est une bonne pratique.

Les attributs

Les attributs sont un peu les options des balises. Ils viennent les compléter pour donner des informations supplémentaires. L'attribut se place après le nom de la balise ouvrante et a le plus souvent une valeur, comme ceci :

```
<balise attribut="valeur">
```

À quoi cela sert-il ? Prenons la balise `<image />` que nous venons de voir. Seule, elle ne sert pas à grand chose. On pourrait rajouter un attribut qui indique le nom de l'image à afficher :

```
<image nom="photo.jpg" />
```

L'ordinateur comprend alors qu'il doit afficher l'image contenue dans le fichier photo.jpg.

Dans le cas d'une balise fonctionnant « par paire », on ne met les attributs que dans la balise ouvrante et pas dans la balise fermante. Par exemple, ce code indique que la citation est de Neil Armstrong et qu'elle date du 21 Juillet 1969 :

```
<citation auteur="Neil Armstrong" date="21/07/1969">  
C'est un petit pas pour l'homme, mais un bond de géant pour l'humanité.  
</citation>
```

Toutes les balises que nous venons de voir sont fictives. Les vraies balises ont des noms en anglais (eh oui !), nous allons les découvrir dans la suite de ce cours.

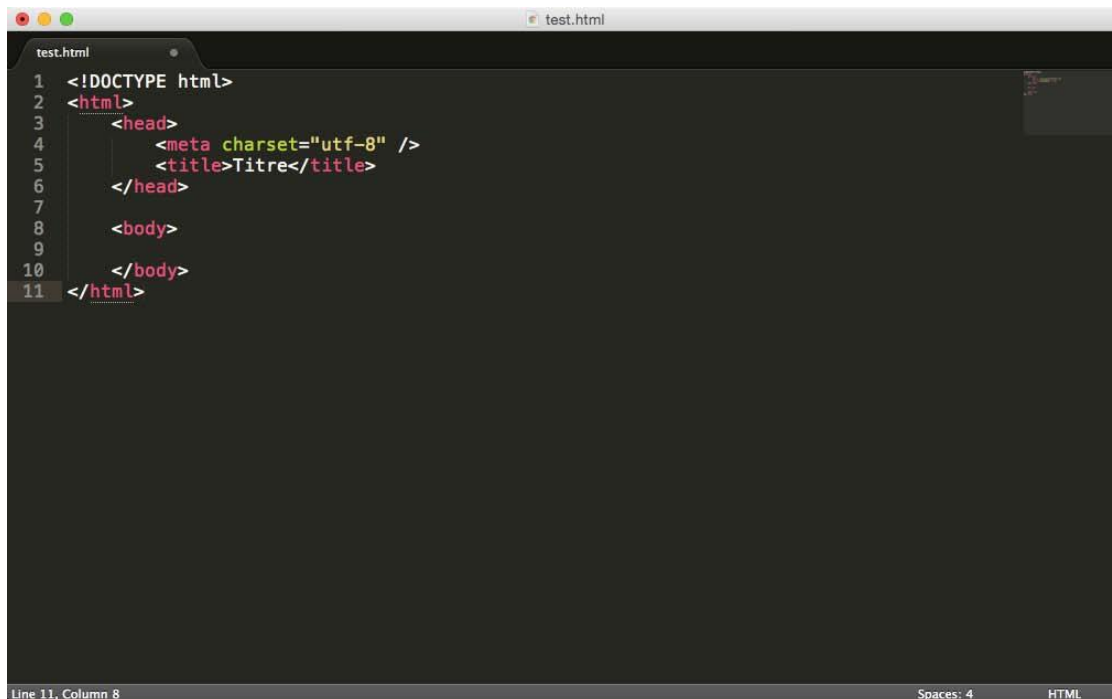
Structure de base d'une page HTML5

Reprenons notre éditeur de texte (dans mon cas Sublime Text). Je vous invite à écrire ou à copier-coller le code source ci-dessous dans Notepad++. Ce code correspond à la base d'une page web en HTML5 :

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8" />  
    <title>Titre</title>  
  </head>  
  
  <body>  
  
  </body>  
</html>
```

J'ai mis des espaces au début de certaines lignes pour « décaler » les balises. Ce n'est pas obligatoire et cela n'a aucun impact sur l'affichage de la page, mais cela rend le code source plus lisible. On appelle cela l'**indentation**. Dans votre éditeur, il suffit d'appuyer sur la touche **Tab** pour avoir le même résultat.

Copié dans Sublime Text, vous devriez voir:



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Titre</title>
6   </head>
7
8   <body>
9
10  </body>
11 </html>
```

Code HTML5 minimal dans Sublime Text

Vous noterez que les balises s'ouvrent et se ferment dans un ordre précis. Par exemple, la balise `<html>` est la première que l'on ouvre et c'est aussi la dernière que l'on ferme (tout à la fin du code, avec `</html>`). *Les balises doivent être fermées dans le sens inverse de leur ouverture.* Un exemple :

- `<html><body></body></html>`: **correct**. Une balise qui est ouverte à l'intérieur d'une autre doit aussi être fermée à l'intérieur.
- `<html><body></html></body>`: **incorrect**, les balises s'entremêlent.

Euh, on pourrait avoir des explications sur toutes les balises que l'on vient de copier dans l'éditeur, m'sieur ?

Bien sûr, c'est demandé si gentiment. :) Ne prenez pas peur en voyant toutes ces balises d'un coup, je vais vous expliquer leur rôle !

Le doctype

```
<!DOCTYPE html>
```

La toute première ligne s'appelle le **doctype**. Elle est indispensable car c'est elle qui indique qu'il s'agit bien d'une page web HTML. Ce n'est pas vraiment une balise comme les autres

(elle commence par un point d'exclamation). Vous pouvez considérer que c'est un peu l'exception qui confirme la règle.

Cette ligne du doctype était autrefois incroyablement complexe. Il était impossible de la retenir de tête. Pour XHTML 1.0, il fallait écrire :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">. Dans le cadre de HTML5, il a été décidé de la simplifier, pour le plus grand bonheur des webmasters. Quand vous voyez une balise doctype courte (<!DOCTYPE html>), cela signifie que la page est écrite en HTML5.

La balise </html>

```
<html>  
</html>
```

C'est la balise principale du code. Elle englobe tout le contenu de votre page. Comme vous pouvez le voir, la balise fermante </html> se trouve tout à la fin du code !

L'en-tête <head> et le corps <body>

Une page web est constituée de deux parties :

- L'en-tête <head>: cette section donne quelques informations générales sur la page comme son titre, l'encodage (pour la gestion des caractères spéciaux), etc. Cette section est généralement assez courte. Les informations que contient l'en-tête ne sont pas affichées sur la page, ce sont simplement des informations générales à destination de l'ordinateur. Elles sont cependant très importantes !
- Le corps <body>: c'est là que se trouve la partie principale de la page. Tout ce que nous écrirons ici sera affiché à l'écran. C'est à l'intérieur du corps que nous écrirons la majeure partie de notre code.

Pour le moment, le corps est vide (nous y reviendrons plus loin). Intéressons-nous par contre aux deux balises contenues dans l'en-tête...

L'encodage (charset)

```
<meta charset="utf-8" />
```

Cette balise indique l'encodage utilisé dans votre fichier .html.

Sans rentrer dans les détails, car cela pourrait vite devenir compliqué, l'encodage indique la façon dont le fichier est enregistré. C'est lui qui détermine comment les caractères spéciaux vont s'afficher (accents, idéogrammes chinois et japonais, caractères arabes, etc.).

Il y a plusieurs techniques d'encodage portant des noms bizarres et utilisées en fonction des langues : ISO-88591, OEM 775, Windows-1253... Une seule cependant devrait être utilisée aujourd'hui autant que possible : UTF

8. Cette méthode d'encodage permet d'afficher sans aucun problème pratiquement tous les symboles de toutes les langues de notre planète ! C'est pour cela que j'ai indiqué utf-8 dans cette balise.

Il ne suffit pas de *dire* que votre fichier est en UTF-8. Il faut aussi que votre fichier soit bien *enregistré* en UTF

8. C'est heureusement le cas désormais par défaut dans la plupart des éditeurs de texte.

Si les accents s'affichent mal par la suite, c'est qu'il y a un problème avec l'encodage. Vérifiez que la balise meta indique bien UTF-8 et que votre fichier est enregistré en UTF-8 (sous Sublime Text, allez dans le menu File > Save with Encoding > UTF-8 pour vous assurer que votre fichier est enregistré en UTF-8.).

Le titre principal de la page

```
<title>
```

C'est le titre de votre page, probablement l'élément le plus important ! Toute page doit avoir un titre qui décrit ce qu'elle contient. Il est conseillé de garder le titre assez court (moins de 100 caractères en général).

Le titre ne s'affiche pas dans votre page mais en haut de celle-ci (souvent dans l'onglet du navigateur). Enregistrez votre page web et ouvrez-la dans votre navigateur. Vous verrez que le titre s'affiche dans l'onglet, comme sur la figure suivante.



Le titre de la page apparaît en haut du navigateur

Il faut savoir que le titre apparaît aussi dans les résultats de recherche, comme sur Google (figure suivante).



Le titre de la page apparaît dans les recherches Google

Autant vous dire que bien choisir son titre est important !

Les commentaires

Nous avons appris à créer notre première *vraie* page HTML dans ce chapitre. Avant de terminer, j'aimerais vous présenter le principe des commentaires.

Un **commentaire** en HTML est un texte qui sert simplement de mémo. Il n'est pas affiché, il n'est pas lu par l'ordinateur, cela ne change rien à l'affichage de la page.

Bref, cela ne sert à rien ?

Eh bien si ! Cela sert à *vous* et aux personnes qui liront le code source de votre page. Vous pouvez utiliser les commentaires pour laisser des indications sur le fonctionnement de votre page.

Quel intérêt ? Cela vous permettra de vous rappeler comment fonctionne votre page si vous revenez sur votre code source après un long moment d'absence. Ne rigolez pas, cela arrive à tous les webmasters.

Insérer un commentaire

Un commentaire est une balise HTML avec une forme bien spéciale :

`<!--Ceci est un commentaire -->`

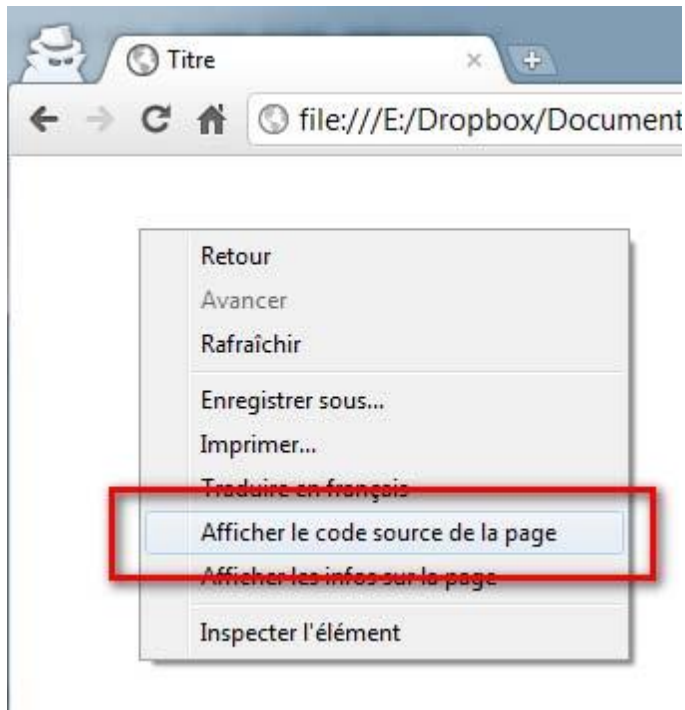
Vous pouvez le mettre où vous voulez au sein de votre code source : il n'a aucun impact sur votre page, mais vous pouvez vous en servir pour vous aider à vous repérer dans votre code source (surtout s'il est long).

```
<!DOCTYPE html>
<html>
  <head>
    <!-- En-tête de la page -->
    <meta charset="utf-8" />
    <title>Titre</title>
  </head>

  <body>
    <!-- Corps de la page -->
  </body>
</html>
```

Tout le monde peut voir vos commentaires... et tout votre code HTML !

Terminons par une remarque importante : *tout le monde peut voir le code HTML de votre page* une fois celle-ci mise en ligne sur le Web. Il suffit de faire un clic droit sur la page et de sélectionner « Afficher le code source de la page » (l'intitulé peut changer selon votre navigateur), comme le montre la figure suivante.



Menu afficher le code source

Le code source s'affiche alors (figure suivante).



Affichage du code source

Vous pouvez tester cette manipulation sur n'importe quel site web, cela marche ! Garanti à 100%. Cela s'explique assez facilement : le navigateur *doit* obtenir le code HTML pour savoir ce qu'il faut afficher. Le code HTML de tous les sites est donc public.

La morale de l'histoire ? Tout le monde pourra voir votre code HTML et vous ne pouvez pas l'empêcher. Par conséquent, ne mettez pas d'informations sensibles comme des mots de passe

dans les commentaires... et soignez votre code source, car je pourrai venir vérifier si vous avez bien suivi mon cours à la lettre !

Lorsque vous regarderez le code de certains sites web, ne prenez pas peur s'il vous paraît long ou ne pas

respecter les mêmes règles que celles que je vous présente dans ce livre. Tous les sites ne sont pas écrits en HTML5 (loin de là) et, parfois, certains webmasters rédigent très mal leur code, ce ne sont pas toujours des exemples à suivre !

En résumé

- On utilise l'éditeur de texte (Sublime Text, Notepad++, jEdit, vim...) pour créer un fichier ayant l'extension .html (par exemple : test.html). Ce sera notre page web.
- Ce fichier peut être ouvert dans le navigateur web simplement en faisant un double-clic dessus.
- À l'intérieur du fichier, nous écrivons le contenu de notre page, accompagné de balises HTML.
- Les balises peuvent avoir plusieurs formes :
 - o `<balise> </balise>`: elles s'ouvrent et se ferment pour délimiter le contenu (début et fin d'un titre, par exemple).
 - o `<balise />`: balises orphelines (on ne les insère qu'en un seul exemplaire), elles permettent d'insérer un élément à un endroit précis (par exemple une image).
- Les balises sont parfois accompagnées d'attributs pour donner des indications supplémentaires (exemple : `<image nom="photo.jpg" />`).
- Une page web est constituée de deux sections principales : un en-tête (`<head>`) et un corps (`<body>`).
- On peut afficher le code source de n'importe quelle page web en faisant un clic droit puis en sélectionnant Afficher le code source de la page.

Organiser son texte

Visionner la vidéo du Chapitre 3 de la Partie 1 sur Vimeo

Bon, la page blanche c'est bien joli, mais votre site web risque d'avoir un succès mitigé si vous le laissez comme cela.

Nous allons découvrir de nombreuses balises HTML dans ce chapitre. Certaines existent depuis la toute première version de HTML, d'autres ont été introduites plus récemment dans HTML5.

Nous allons voir successivement dans ce chapitre :

- comment rédiger des paragraphes ;
- comment structurer sa page avec les titres ;
- comment donner de l'importance à certains mots de son texte ;
- comment organiser les informations sous forme de listes.

Motivés ? Allez, vous allez voir, ce n'est pas compliqué.

Les paragraphes

La plupart du temps, lorsqu'on écrit du texte dans une page web, on le fait à l'intérieur de paragraphes. Le langage HTML propose justement la balise `<p>` pour délimiter les paragraphes.

```
<p>Bonjour et bienvenue sur mon site !</p>
```

- `<p>` signifie « Début du paragraphe » ;
- `</p>` signifie « Fin du paragraphe ».

Comme je vous l'ai dit au chapitre précédent, on écrit le contenu du site web entre les balises `<body></body>`. Il nous suffit donc de mettre notre paragraphe entre ces deux balises et nous aurons enfin notre première vraie page web avec du texte !

Je reprends donc exactement le même code qu'au chapitre précédent et j'y ajoute mon paragraphe :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Paragraphes</title>
  </head>

  <body>
    <p>Bonjour et bienvenue sur mon site !</p>
  </body>
</html>
```

Essayez, vous allez voir le résultat ! Bon, ok, ce n'est pas encore le nirvana mais c'est un bon début.

Mais ne nous arrêtons pas en si bon chemin. Nous allons voir maintenant quelque chose d'un peu particulier en HTML : le saut de ligne. Cela paraît simple et pourtant, cela ne fonctionne pas vraiment comme dans un traitement de texte habituel...

Sauter une ligne

En HTML, si vous appuyez sur la touche Entrée, cela ne crée pas une nouvelle ligne comme vous en avez l'habitude. Essayez donc ce code :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Essais de sauts de ligne</title>
  </head>

  <body>
    <p>Bonjour et bienvenue sur mon site !
      Ceci est mon premier test alors soyez indulgents s'il vous
plaît, j'apprends petit à petit comment cela marche.
      Pour l'instant c'est un peu vide, mais revenez dans 2-3 jours
quand j'aurai appris un peu plus de choses, je vous assure que vous allez
être surpris !</p>
  </body>
</html>
```

Tout le texte s'affiche sur la même ligne alors qu'on est bien allé à la ligne dans le code !

Taper frénétiquement sur la touche Entrée dans l'éditeur de texte ne sert donc strictement à rien.

Comme vous devez vous en douter, il y a pourtant bien un moyen de faire des sauts de ligne en HTML.

En fait, si vous voulez écrire un deuxième paragraphe, il vous suffit d'utiliser une deuxième balise <p>. Votre code HTML devrait donc être au final rempli de balises de paragraphe !

Un exemple :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Paragraphes</title>
  </head>

  <body>
    <p>Bonjour et bienvenue sur mon site !
    Ceci est mon premier test alors soyez indulgents s'il vous plaît,
    j'apprends petit à petit comment cela marche. </p>

    <p>Pour l'instant c'est un peu vide, mais revenez dans 2-3 jours
    quand j'aurai appris un peu plus de choses, je vous assure que vous allez
    être surpris ! </p>
  </body>
</html>
```

Le résultat se trouve à la figure suivante.



Deux paragraphes avec 2 balises <p>

Oui, mais si je veux juste aller à la ligne dans un paragraphe et non pas sauter une ligne ?

Eh bien devinez quoi : il existe une balise « Aller à la ligne » ! C'est une balise **orpheline** qui sert juste à indiquer qu'on doit aller à la ligne : `
`. Vous devez obligatoirement la mettre à l'intérieur d'un paragraphe.

Voici comment l'utiliser dans un code :


```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Sauts de ligne</title>
  </head>

  <body>
    <p>Bonjour et bienvenue sur mon site !<br />
    Ceci est mon premier test alors soyez indulgents s'il vous plaît,
    j'apprends petit à petit comment cela marche. </p>

    <p>Pour l'instant c'est un peu vide, mais revenez dans 2-3 jours
    quand j'aurai appris un peu plus de choses, je vous assure que vous allez
    être surpris ! </p>
  </body>
</html>
```

Vous pouvez théoriquement mettre plusieurs balises `
` d'affilée pour faire plusieurs sauts de lignes, mais on considère que c'est une mauvaise pratique qui rend le code délicat à maintenir. Pour décaler un texte avec plus de précision, on utilisera le CSS, ce langage qui vient compléter le HTML et dont je vous parlerai un peu plus loin.

Donc c'est compris ?

- `<p> </p>`: pour organiser son texte en paragraphes ;
- `
`: pour aller à la ligne.

Maintenant qu'on sait écrire des paragraphes, voyons voir comment on crée des titres.

Les titres

Lorsque le contenu de votre page va s'étoffer avec de nombreux paragraphes, il va devenir difficile pour vos visiteurs de se repérer. C'est là que les titres deviennent utiles.

En HTML, on est verni, on a le droit d'utiliser six niveaux de titres différents. Je veux dire par là qu'on peut dire « Ceci est un titre très important », « Ceci est un titre un peu moins important », « Ceci est un titre encore moins important », etc. On a donc six balises de titres différentes :

- `<h1> </h1>`: signifie « titre très important ». En général, on s'en sert pour afficher le titre de la page au début de celle-ci.
- `<h2> </h2>`: signifie « titre important ».
- `<h3> </h3>`: pareil, c'est un titre un peu moins important (on peut dire un « sous-titre » si vous voulez).
- `<h4> </h4>`: titre encore moins important.

- `<h5>` `</h5>`: titre pas important.
- `<h6>` `</h6>`: titre vraiment, mais alors là vraiment pas important du tout.

Attention : ne confondez pas avec la balise `<title>` ! La balise `<title>` affiche le titre de la page dans la barre de titre du navigateur comme nous l'avons vu. Les titres `<h1>` et compagnie, eux, servent à créer des titres qui seront affichés *dans* la page web.

Ne vous laissez pas impressionner par toutes ces balises. En fait, six niveaux de titres, c'est beaucoup. Dans la pratique, personnellement, je n'utilise que les balises `<h1>`, `<h2>` et `<h3>`, et très rarement les autres (je n'ai pas souvent besoin de six niveaux de titres différents). Votre navigateur affiche le titre très important en très gros, le titre un peu moins important en un peu moins gros, etc.

Ne choisissez pas votre balise de titre en fonction de la taille qu'elle applique au texte ! Il faut impérativement bien structurer sa page en commençant par un titre de niveau 1 (`<h1>`), puis un titre de niveau 2 (`<h2>`), etc. Il ne devrait pas y avoir de sous-titre sans titre principal ! Si vous voulez modifier la taille du texte, sachez que nous apprendrons à faire cela en CSS un peu plus tard.

Essayez de créer une page web avec des titres pour voir ce que cela donne :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Niveaux de titres</title>
  </head>

  <body>
    <h1>Titre super important</h1>
    <h2>Titre important</h2>
    <h3>Titre un peu moins important (sous-titre)</h3>

    <h4>Titre pas trop important</h4>
    <h5>Titre pas important</h5>
    <h6>Titre vraiment pas important du tout</h6>
  </body>
</html>
```

Allez, je vous donne un exemple d'utilisation des titres dans une page web (vous allez voir que je ne me sers pas de toutes les balises) :

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Présentation d'OpenClassrooms</title>
</head>

<body>
  <h1>Bienvenue sur OpenClassrooms !</h1>

  <p>Bonjour et bienvenue sur mon site : OpenClassrooms.<br />
  OpenClassrooms, qu'est-ce que c'est ? </p>

  <h2>Des cours pour débutants</h2>

  <p>OpenClassrooms vous propose des cours (tutoriels) destinés aux
  débutants : aucune connaissance n'est requise pour lire ces cours !</p>

  <p>Vous pourrez ainsi apprendre, sans rien y connaître auparavant, à
  créer un site web, à programmer, à construire des mondes en 3D !</p>

  <h2>Une communauté active</h2>

  <p>Vous avez un problème, un élément du cours que vous ne comprenez pas
  ? Vous avez besoin d'aide pour créer votre site ?<br />

  Rendez-vous sur les forums ! Vous y découvrirez que vous n'êtes pas le
  seul dans ce cas et vous trouverez très certainement quelqu'un qui vous
  aidera aimablement à résoudre votre problème.</p>
</body>
</html>
```

Voilà une page web qui prend forme !

Oui, mais moi je veux centrer mon titre, l'écrire en rouge et le souligner !

Nous ferons tout cela lorsque nous apprendrons le CSS (dès la deuxième partie du cours). Il faut savoir que<h1>ne signifie pas « Times New Roman, taille 16 pt », mais « *Titre important* ».

Grâce au langage CSS, vous pourrez dire « Je veux que mes titres importants soient centrés, rouges et soulignés ». Pour le moment, en HTML, nous ne faisons que structurer notre page. *Nous rédigeons le contenu avant de nous amuser à le mettre en forme.*

La mise en valeur

Au sein de vos paragraphes, certains mots sont parfois plus importants que d'autres et vous aimeriez les faire ressortir. HTML vous propose différents moyens de mettre en valeur le texte de votre page.

Mettre un peu en valeur

Pour mettre *un peu* en valeur votre texte, vous devez utiliser la balise . Son utilisation est très simple : encadrez les mots à mettre en valeur avec ces balises et c'est bon ! Je reprends un peu l'exemple de tout à l'heure et j'y mets quelques mots en évidence :

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Emphase</title>
</head>

<body>
  <p>Bonjour et bienvenue sur mon site !<br />
  Ceci est mon premier test alors <em>soyez indulgents</em> s'il vous
  plaît, j'apprends petit à petit comment cela marche.</p>
</body>
</html>
```

Comme vous pouvez le voir, utiliser la balise `` a pour conséquence de mettre le texte en *italique*. En fait, c'est le navigateur qui choisit comment afficher les mots. On lui dit que les mots sont assez importants et, pour faire ressortir cette information, il change l'apparence du texte en utilisant l'italique.

Mettre bien en valeur

Pour mettre un texte bien en valeur, on utilise la balise `` qui signifie « fort », ou « important » si vous préférez. Elle s'utilise exactement de la même manière que ``:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Forte emphase</title>
</head>

<body>
  <p>Bonjour et bienvenue sur mon site !<br />
  Ceci est mon premier test alors <strong>soyez indulgents</strong> s'il
  vous plaît, j'apprends petit à petit comment cela marche.</p>
</body>
</html>
```

Vous voyez sûrement le texte s'afficher en gras. Là encore, le gras n'est qu'une *conséquence*. Le navigateur a choisi d'afficher en gras les mots importants pour les faire ressortir davantage.

La balise `` ne signifie pas « mettre en gras » mais « important ». On pourra décider plus tard, en CSS, d'afficher les mots « importants » d'une autre façon que le gras si on le souhaite.

Marquer le texte

La balise `<mark>` permet de faire ressortir visuellement une portion de texte. L'extrait n'est pas forcément considéré comme important mais on veut qu'il se distingue bien du reste du texte. Cela peut être utile pour faire ressortir un texte pertinent après une recherche sur votre site par exemple.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Marquage du texte</title>
</head>

<body>
  <p>Bonjour et bienvenue sur mon site !<br />
  Ceci est mon premier test alors <mark>soyez indulgents</mark> s'il vous
  plaît, j'apprends petit à petit comment cela marche.</p>
</body>
</html>
```

Par défaut, `<mark>` a pour effet de surligner le texte. On pourra changer l'affichage en CSS (décider de surligner dans une autre couleur, d'encadrer le texte, etc.). C'est le même principe que ce que je vous disais pour les balises précédentes : elles indiquent le *sens* des mots et non pas comment ceux-ci doivent s'afficher.

N'oubliez pas : HTML pour le fond, CSS pour la forme

Je vais peut-être vous sembler un peu lourd mais il est très important qu'on se comprenne bien car les débutants font souvent la même grosse erreur à ce stade. Ils ont vu les balises ``, ``, `<mark>`... et ils se disent : « Chouette, j'ai découvert comment mettre en italique, en gras et comment surligner du texte en HTML ! ».

Et pourtant... ce n'est pas à cela que servent ces balises ! Je sais, je sais, vous allez me dire « Oui mais quand j'utilise `` le texte apparaît en gras, donc c'est pour mettre en gras. » et pourtant, c'est une erreur de croire que cette balise sert à cela.

Le rôle des balises est d'indiquer le *sens* du texte. Ainsi, `` indique à l'ordinateur « Ce texte est important ». C'est tout. Et pour *montrer* que le texte est important, l'ordinateur décide de le mettre en gras (mais il pourrait aussi bien l'écrire en rouge !). La plupart des navigateurs affichent les textes importants en gras, mais rien ne les y oblige.

Je ne comprends pas. À quoi cela sert-il que l'ordinateur sache qu'un texte est important ?

Il n'est pas assez intelligent pour comprendre !

Détrompez-vous ! De nombreux programmes analysent le code source des pages web, à commencer par les robots de moteurs de recherche. Ces robots parcourent le Web en lisant le code HTML de tous les sites. C'est le cas des robots de Google et de Bing, par exemple. Les mots-clés « importants » ont tendance à avoir plus de valeur à leurs yeux, donc si quelqu'un fait une recherche sur ces mots, il a plus de chances de tomber sur votre site. Bien entendu, c'est une explication grossière et il ne faut pas croire qu'utiliser la balise `` à tout-va améliorera votre référencement. Il faut simplement faire confiance aux ordinateurs : ils comprennent ce qu'un texte « important » veut dire et peuvent se servir de cette information.

Mais alors, comment fait-on pour mettre spécifiquement en gras, pour écrire en rouge, et tout et tout ?

Tout cela se fait en CSS. Souvenez-vous :

- le HTML définit le fond (contenu, logique des éléments) ;
- le CSS définit la forme (apparence).

Nous verrons le CSS plus loin, pour l'instant nous nous concentrons sur le HTML et ses balises, qui ont chacune un sens particulier.

Les listes

Les listes nous permettent souvent de mieux structurer notre texte et d'ordonner nos informations. Nous allons découvrir ici deux types de listes :

- les listes non ordonnées ou listes à puces ;
- les listes ordonnées ou listes numérotées ou encore énumérations.

Liste non ordonnée

Une liste non ordonnée ressemble à ceci :

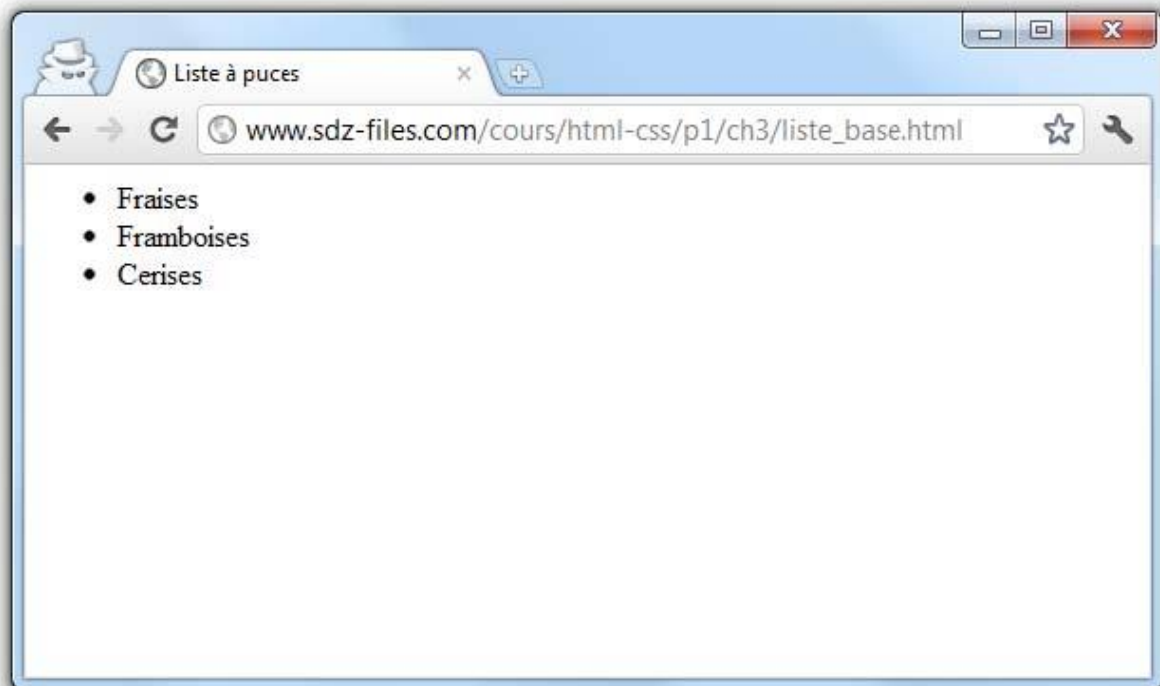
- Fraises
- Framboises
- Cerises

C'est un système qui nous permet de créer une liste d'éléments sans notion d'ordre (il n'y a pas de « premier » ni de « dernier »). Créer une liste non ordonnée est très simple. Il suffit d'utiliser la balise `` que l'on referme un peu plus loin avec ``. Commencez donc à taper ceci :

```
<ul></ul>
```

Et maintenant, voilà ce qu'on va faire : on va écrire chacun des éléments de la liste entre deux balises ``. Chacune de ces balises doit se trouver entre `` et ``. Vous allez comprendre de suite avec cet exemple : Le résultat se trouve à la figure suivante.

```
<ul>  
  <li>Fraises</li>  
  <li>Framboises</li>  
  <li>Cerises</li>  
</ul>
```



Une liste non ordonnée

Notez que la liste doit être placée à l'intérieur de `<body></body>`. À partir de maintenant, je ne mets pas tout le code de la page pour rester lisible.

Retenez donc ces deux balises :

- `` délimite toute la liste ;
- `` délimite un élément de la liste (une puce).

Vous pouvez mettre autant d'éléments que vous voulez dans la liste à puces, vous n'êtes pas limités à trois éléments.

Et voilà, vous savez créer une liste à puces ! Pas si dur une fois qu'on a compris comment imbriquer les balises.

Pour ceux qui ont besoin de faire des listes complexes, sachez que vous pouvez *imbriquer* des listes à puces (créer une liste à puces **dans** une liste à puces). Si vous voulez faire ça, ouvrez une seconde balise `` à l'intérieur d'un élément ``. Si vous fermez les balises dans le bon ordre, vous n'aurez pas de problème. Attention néanmoins, cette technique est un peu compliquée à maîtriser.

Liste ordonnée

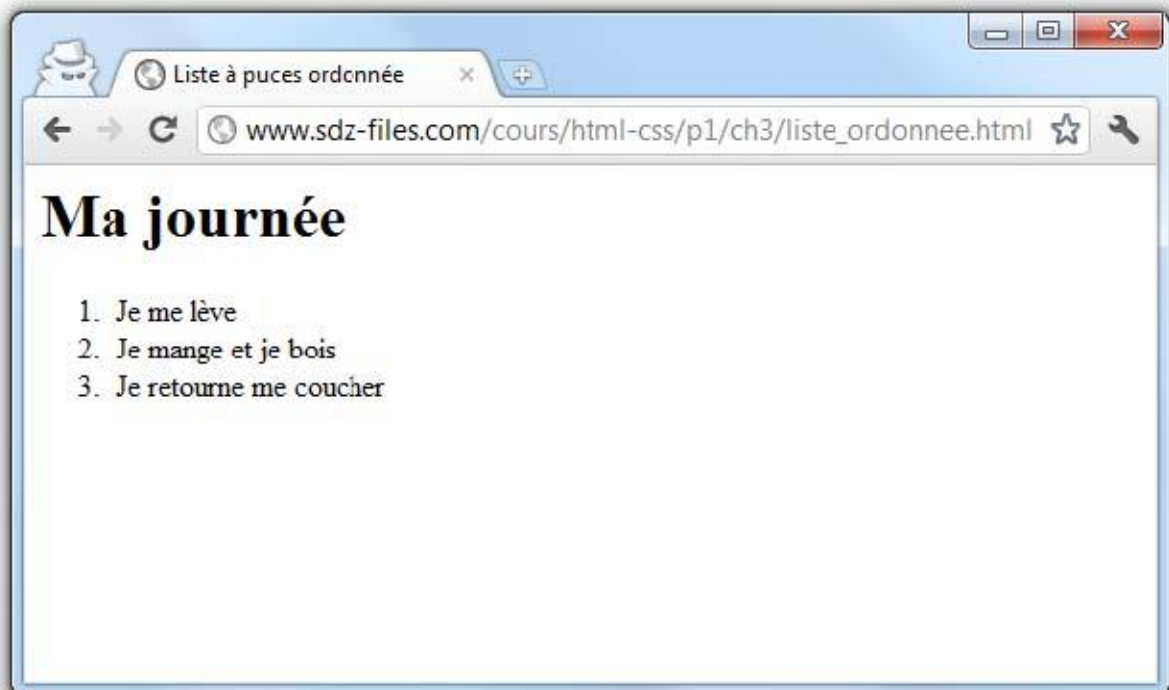
Une liste ordonnée fonctionne de la même façon, seule une balise change : il faut remplacer `` par ``. À l'intérieur de la liste, on ne change rien : on utilise toujours des balises `` pour délimiter les éléments.

L'ordre dans lequel vous placez les éléments de la liste est important. Le premier `` sera l'élément n° 1, le second sera le n°2 etc...

Comme c'est particulièrement intuitif, je vous laisse admirer la simplicité de cet exemple (résultat à la figure suivante) :

```
<h1>Ma journée</h1>

<ol>
  <li>Je me lève.</li>
  <li>Je mange et je bois.</li>
  <li>Je retourne me coucher.</li>
</ol>
```



Une liste ordonnée

Par rapport à l'exemple précédent, tout ce qu'on a eu à changer est donc la balise ``.

Pour information, il existe un troisième type de liste, beaucoup plus rare : la liste de définitions. Elle fait intervenir les balises `<dl>` (pour délimiter la liste), `<dt>` (pour délimiter un terme) et `<dd>` (pour délimiter la définition de ce terme).

En résumé

- Le HTML comporte de nombreuses balises qui nous permettent d'organiser le texte de notre page. Ces balises donnent des indications comme « Ceci est un paragraphe », « Ceci est un titre », etc.
- Les paragraphes sont définis par la balise `<p>` `</p>` et les sauts de ligne par la balise `
`.
- Il existe six niveaux de titre, de `<h1>` `</h1>` à `<h6>` `</h6>`, à utiliser selon l'importance du titre.
- On peut mettre en valeur certains mots avec les balises ``, `` et `<mark>`.
- Pour créer des listes, on doit utiliser la balise `` (liste à puces, non ordonnée) ou `` (liste ordonnée). À l'intérieur, on insère les éléments avec une balise `` pour chaque item.

Créer des liens

Visionner la vidéo du Chapitre 4 de la Partie 1 sur Vimeo

Au chapitre précédent, vous avez appris à créer une page HTML toute simple. D'accord, elle n'était pas franchement magnifique, mais c'était une vraie page HTML quand même.

Comme vous le savez, un site web est composé de plusieurs pages. Comment faire pour aller d'une page vers une autre ? À l'aide de liens pardi ! Dans ce chapitre, nous allons justement apprendre à créer des liens entre nos pages.

Je suppose que chacun d'entre vous sait ce qu'est un lien : il s'agit d'un texte sur lequel on peut cliquer pour se rendre sur une autre page. On peut faire un lien d'une page `a.html` vers une page `b.html`, mais on peut aussi faire un lien vers un autre site (par exemple, `http://www.siteduzero.com`). Dans les deux cas, nous allons voir que le fonctionnement est le même.

Un lien vers un autre site

Il est facile de reconnaître les liens sur une page : ils sont écrits d'une façon différente (par défaut, en bleu et soulignés) et un curseur en forme de main apparaît lorsqu'on pointe dessus.

Je vous propose d'essayer de coder le lien qui amène vers OpenClassrooms, comme à la figure suivante.

Bonjour. Vous souhaitez visiter [OpenClassrooms](#) ?
C'est un bon site ;o)



Lien vers OpenClassrooms

INSTITUTS UNIVERSITAIRE ANNEE ACCADEMIQUE 2023-2024

Pour faire un lien, la balise que nous allons utiliser est très simple à retenir : `<a>`. Il faut cependant lui ajouter un attribut, `href`, pour indiquer vers quelle page le lien doit conduire.

Par exemple, le code ci-dessous est un lien qui amène vers OpenClassrooms, situé à l'adresse `https://openclassrooms.com`:

```
<a href="https://openclassrooms.com">OpenClassrooms</a>
```

Nous allons placer ce lien au sein d'un paragraphe. Voici donc comment reproduire l'exemple de la figure précédente :

```
<p>Bonjour. Vous souhaitez visiter <a  
href="https://openclassrooms.com">OpenClassrooms</a> ?<br />  
C'est un bon site ;o)</p>
```

Nous verrons comment changer cette apparence lorsque nous étudierons le CSS.

Par défaut, le lien s'affiche en bleu souligné. Si vous avez déjà ouvert la page, le lien s'affiche en violet.

Si vous voulez faire un lien vers un autre site, il suffit donc de copier son adresse (on parle d'URL) en `http://`. Notez que certains liens commencent parfois par `https://` (sites sécurisés, comme OpenClassrooms) ou d'autres préfixes (`ftp://`, ...).

Si vous faites un lien vers un site qui comporte une adresse un peu bizarre avec des `&`, comme : `http://www.site.com/?data=15&name=mateo21`, vous devrez remplacer tous les « `&` » par « `&` » dans votre lien comme ceci : `http://www.site.com/?data=15&name=mateo21`. Vous ne verrez pas la différence, mais cela est nécessaire pour avoir une page web correctement construite en HTML5.

Les liens que nous venons de voir sont appelés **liens absolus** car on indique l'adresse complète. Nous allons maintenant voir que l'on peut écrire les liens d'une façon un peu différente, ce qui va nous être utile pour faire des liens entre les pages de notre site.

Un lien vers une autre page de son site

Nous venons d'apprendre à créer des liens vers des sites existants. Mais je suis sûr que vous aimeriez faire des liens entre les différentes pages de votre site, non ?

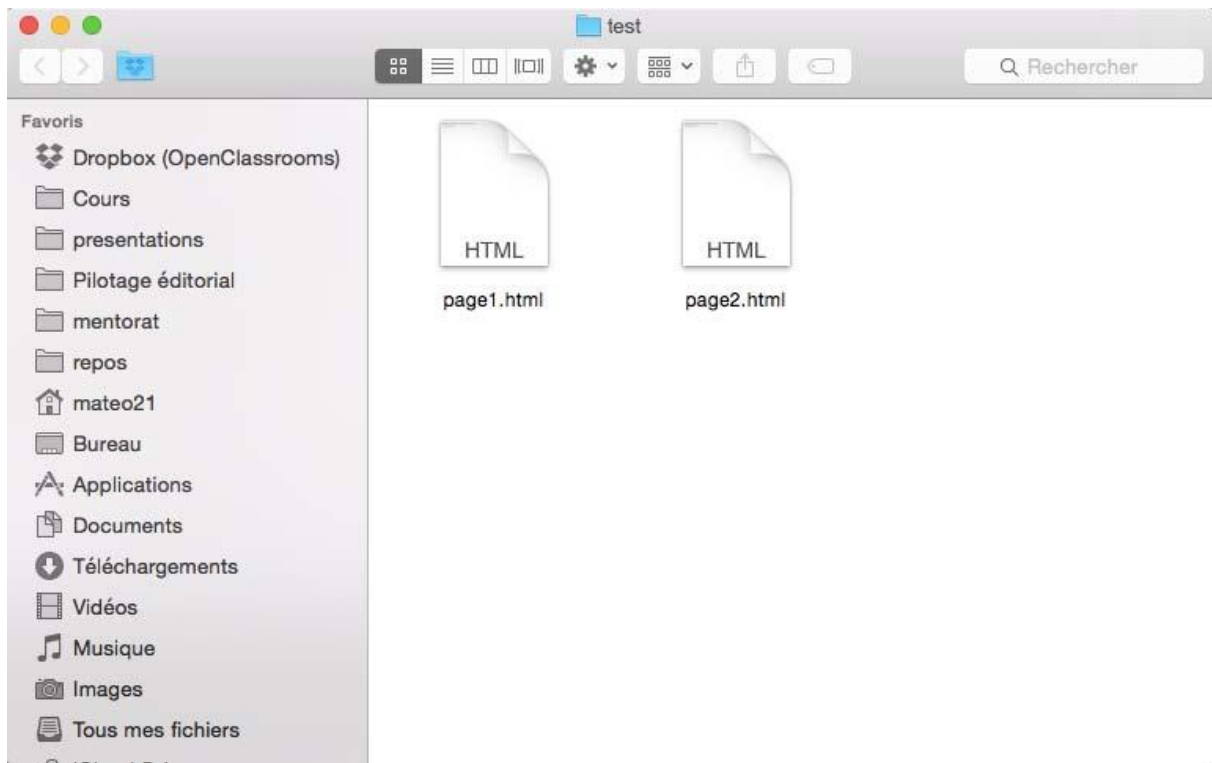
Oui, justement, comment je fais pour faire un lien vers une autre page de mon site ? Je ne connais pas son

adresse en `http://...`, je commence à peine à créer mon site là ! Je n'ai pas d'adresse.

En effet, pour le moment, vous êtes en train de créer votre site sur votre ordinateur. Vous êtes le seul à pouvoir le voir et il n'a pas encore « d'adresse web » qui commence en `http://` comme la plupart des sites. Heureusement, cela ne va pas nous empêcher de travailler.

Deux pages situées dans un même dossier

Pour commencer, nous allons créer deux fichiers correspondant à deux pages HTML différentes. Comme je suis très inspiré, je vous propose de les appeler `page1.html` et `page2.html`. Nous aurons donc ces deux fichiers sur notre disque *dans le même dossier* (figure suivante).



Plusieurs fichiers HTML dans un même dossier

Comment faire un lien de la page 1 vers la page 2, sans avoir d'adresse en `http://`? En fait, c'est facile : si les deux fichiers sont situés dans le même dossier, il suffit d'écrire comme cible du lien le nom du fichier vers lequel on veut amener. Par exemple : ``. On dit que c'est un **lien relatif**.

Voici le code que nous allons utiliser dans nos fichiers `page1.html` et `page2.html`.

page1.html

```
<p>Bonjour. Souhaitez-vous consulter <a href="page2.html">la page 2</a>?</p>
```

page2.html

La page 2 (page d'arrivée) affichera simplement un message pour indiquer que l'on est bien arrivé sur la page 2 :

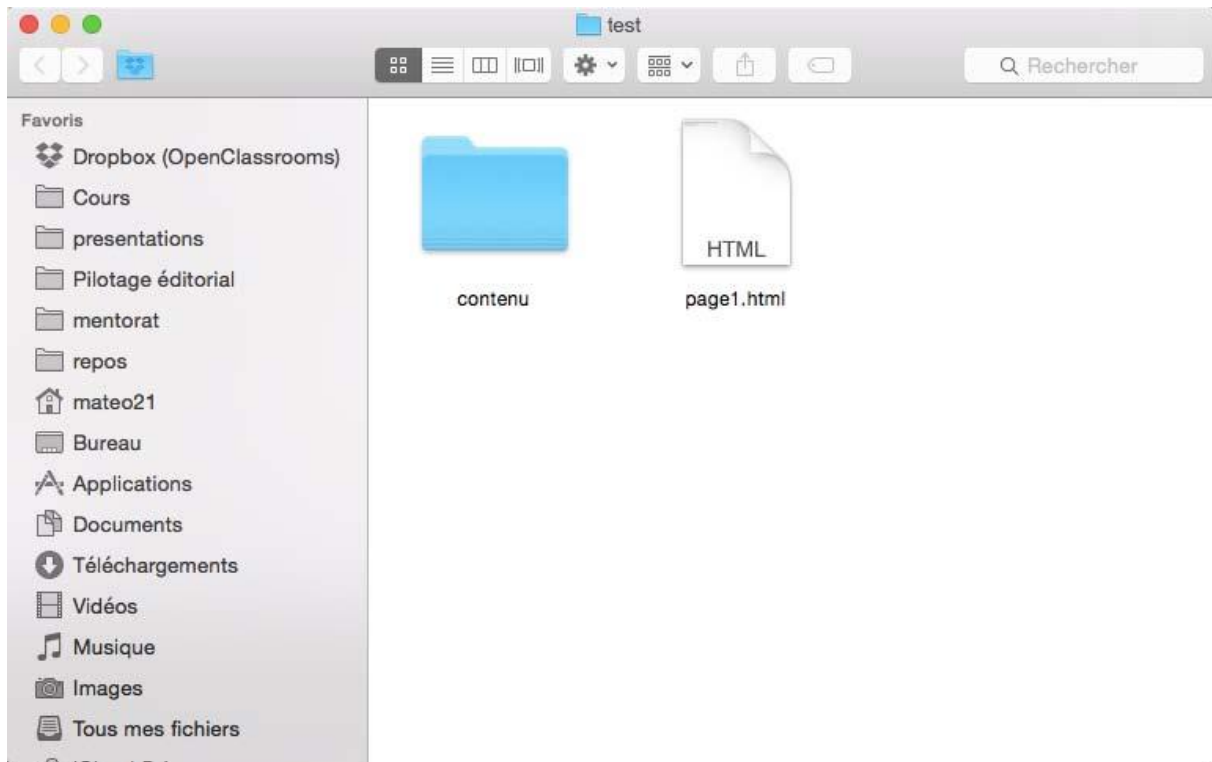
```
<h1>Bienvenue sur la page 2 !</h1>
```

Deux pages situées dans des dossiers différents

INSTITUTS UNIVERSITAIRE ANNEE ACCADEMIQUE 2023-2024

Les choses se corsent un petit peu si les pages sont situées dans des dossiers différents. Idéalement, elles ne devraient pas être trop loin l'une de l'autre (dans un sous-dossier par exemple).

Imaginons que `page2.html` se trouve dans un sous-dossier appelé contenu, comme à la figure suivante.



Le fichier `page2.html` se trouve à l'intérieur du dossier contenu

Dans ce cas de figure, le lien doit être rédigé comme ceci :

```
<a href="contenu/page2.html">
```

S'il y avait plusieurs sous-dossiers, on écrirait ceci :

```
<a href="contenu/autredossier/page2.html">
```

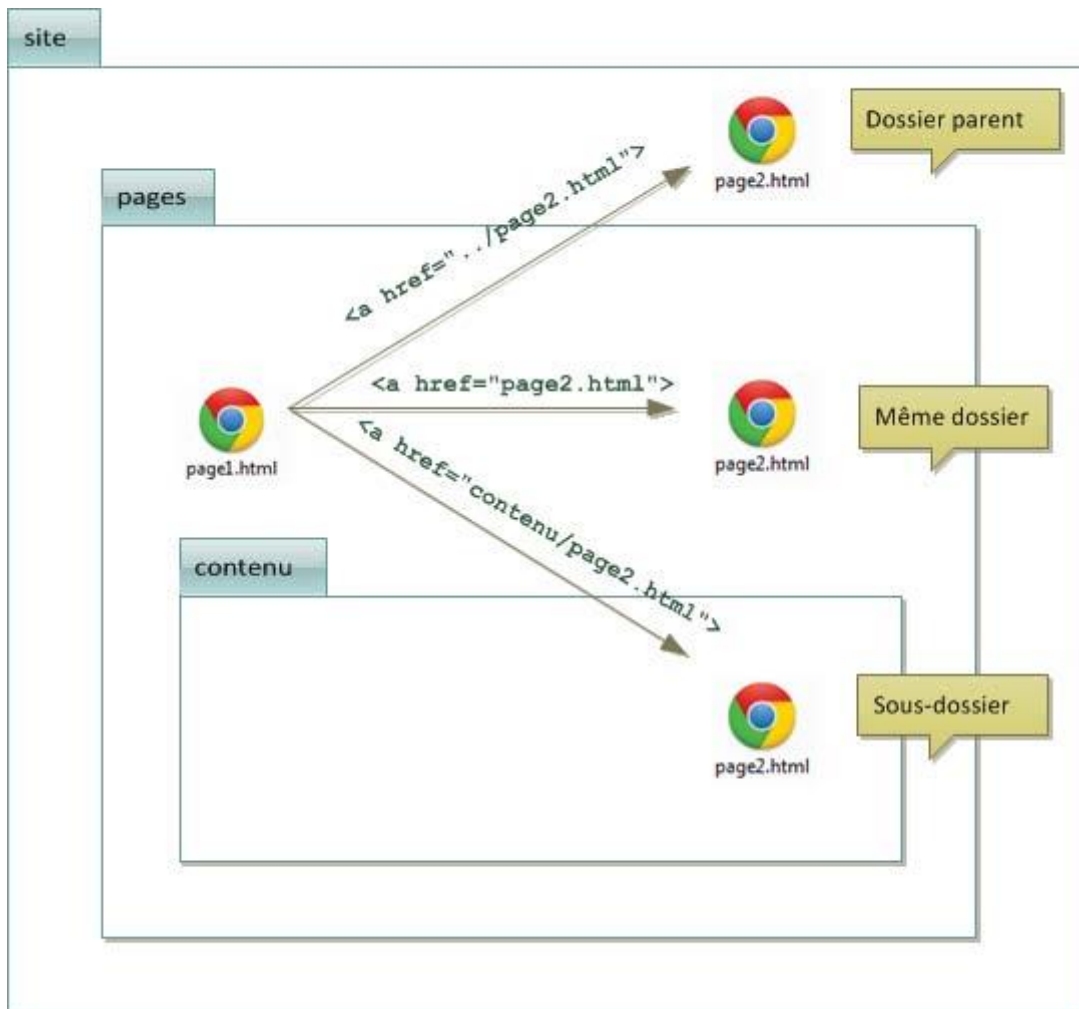
Et si le fichier ne se trouve pas dans un sous-dossier mais dans un dossier « parent », on fait comment ?

Si votre fichier cible est placé dans un dossier qui se trouve « plus haut » dans l'arborescence, il faut écrire deux points comme ceci :

```
<a href="../page2.html">
```

Résumé en images

Les liens relatifs ne sont pas bien compliqués à utiliser une fois qu'on a compris le principe. Il suffit de regarder dans quel « niveau de dossier » se trouve votre fichier cible pour savoir comment écrire votre lien. La figure suivante fait la synthèse des différents liens relatifs possibles.



Les différents liens relatifs

Un lien vers une ancre

Une **ancre** est une sorte de point de repère que vous pouvez mettre dans vos pages HTML lorsqu'elles sont très longues. En effet, il peut alors être utile de faire un lien amenant plus bas dans la même page pour que le visiteur puisse sauter directement à la partie qui l'intéresse.

Pour créer une ancre, il suffit de rajouter l'attribut `id` à une balise qui va alors servir de repère. Ce peut être n'importe quelle balise, un titre par exemple. Utilisez l'attribut `id` pour donner un nom à l'ancre. Cela nous servira ensuite pour faire un lien vers cette ancre. Par exemple :

```
<h2 id="mon_ancre">Titre</h2>
```


Ensuite, il suffit de créer un lien comme d'habitude, mais cette fois l'attribut href contiendra un dièse (#) suivi du nom de l'ancre. Exemple :

```
<a href="#mon_ancre">Aller vers l'ancre</a>
```

Normalement, si vous cliquez sur le lien, cela vous amènera plus bas dans la même page (à condition que la page comporte suffisamment de texte pour que les barres de défilement se déplacent automatiquement).

Voici un exemple de page comportant beaucoup de texte et utilisant les ancres (j'ai mis n'importe quoi dans le texte pour remplir) :

```
<h1>Ma grande page</h1>

<p>
  Aller directement à la partie traitant de :<br />
  <a href="#cuisine">La cuisine</a><br />
  <a href="#rollers">Les rollers</a><br />
  <a href="#arc">Le tir à l'arc</a><br />
</p>
<h2 id="cuisine">La cuisine</h2>

<p>... (beaucoup de texte) ...</p>

<h2 id="rollers">Les rollers</h2>

<p>... (beaucoup de texte) ...</p>

<h2 id="arc">Le tir à l'arc</h2>

<p>... (beaucoup de texte) ...</p>
```

S'il ne se passe rien quand vous cliquez sur les liens, c'est qu'il n'y a pas assez de texte. Dans ce cas, vous pouvez soit rajouter du blabla dans la page pour qu'il y ait (encore) plus de texte, soit réduire la taille de la fenêtre de votre navigateur pour faire apparaître les barres de défilement sur le côté.

L'attribut `id` sert à donner un nom « unique » à une balise, pour s'en servir de repère. Et, croyez-moi, vous n'avez pas fini d'entendre parler de cet attribut. Ici, on s'en sert pour faire un lien vers une ancre mais, en CSS, il nous sera très utile pour « repérer » une balise précise, vous verrez. Évitez cependant de créer des `id` avec des espaces ou des caractères spéciaux, utilisez simplement, dans la mesure du possible, des lettres et chiffres pour que la valeur soit reconnue par tous les navigateurs.

Lien vers une ancre située dans une autre page

Alors là je vous préviens, on va faire le Mégamix ! L'idée, c'est de faire un lien qui ouvre une autre page ET qui amène directement à une ancre située plus bas sur cette page.

En pratique c'est assez simple à faire : il suffit de taper le nom de la page, suivi d'un dièse (#), suivi du nom de l'ancre. Par exemple : ` ...` vous amènera sur la page `ancres.html`, directement au niveau de l'ancre appelée `rollers`. Voici une page qui contient trois liens, chacun amenant vers une des ancres de la page de l'exemple précédent :


```
<h1>Le Mégamix</h1>
<p>
  Rendez-vous quelque part sur une autre page :<br />
  <a href="ancres.html#cuisine">La cuisine</a><br />
  <a href="ancres.html#rollers">Les rollers</a><br />
  <a href="ancres.html#arc">Le tir à l'arc</a><br />
</p>
```

Cas pratiques d'utilisation des liens

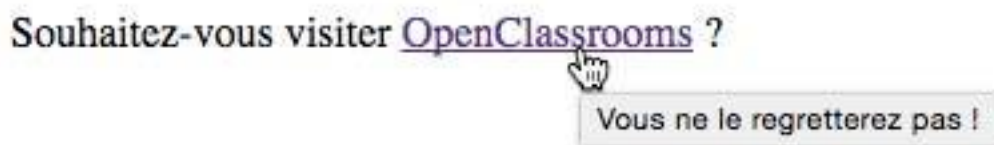
Je vais essayer de vous montrer ici quelques cas pratiques d'utilisation des liens. Par exemple, saviez-vous qu'il est très facile de faire des liens qui lancent un téléchargement ? Qui créent un nouvel e-mail ? Qui ouvrent une nouvelle fenêtre ?

Non ? Eh bien nous allons voir tout cela ici.

Un lien qui affiche une infobulle au survol

Vous pouvez utiliser l'attribut `title` qui affiche une bulle d'aide lorsqu'on pointe sur le lien. Cet attribut est facultatif.

Vous aurez un résultat ressemblant à la figure suivante.



Une infobulle

La bulle d'aide peut être utile pour informer le visiteur avant même qu'il n'ait cliqué sur le lien. Voici comment reproduire ce résultat :

```
<p>Bonjour. Souhaitez-vous visiter <a href="https://openclassrooms.com"
title="Vous ne le regretterez pas !">OpenClassrooms</a> ?</p>
```

Un lien qui ouvre une nouvelle fenêtre

Il est possible de « forcer » l'ouverture d'un lien dans une nouvelle fenêtre. Pour cela, on rajoutera `target="_blank"` à la balise `<a>`:

```
<p>Bonjour. Souhaitez-vous visiter <a href="https://openclassrooms.com"
title="Vous ne le regretterez pas !" target="_blank">OpenClassrooms</a>
?</p>
```

Selon la configuration du navigateur, la page s'affichera dans une nouvelle fenêtre ou un nouvel onglet. Vous ne pouvez pas choisir entre l'ouverture d'une nouvelle fenêtre ou d'un nouvel onglet.

Notez cependant qu'il est déconseillé d'abuser de cette technique car elle perturbe la navigation. Le visiteur lui-même peut décider s'il veut ouvrir le lien dans une nouvelle fenêtre.

Il fera Maj+ Clic sur le lien pour ouvrir dans une nouvelle fenêtre ou Ctrl+ Clic pour ouvrir dans un nouvel onglet.

Un lien pour envoyer un e-mail

Si vous voulez que vos visiteurs puissent vous envoyer un e-mail, vous pouvez utiliser des liens de typemailto. Rien ne change au niveau de la balise, vous devez simplement modifier la valeur de l'attribut href comme ceci : Il suffit donc de faire commencer le lien par mailto:et d'écrire l'adresse e-mail où on peut vous contacter. Si vous cliquez sur le lien, un nouveau message vide s'ouvre, prêt à être envoyé à votre adresse e-mail.

```
<p><a href="mailto:votrenom@bidule.com">Envoyez-moi un e-mail !</a></p>
```

Un lien pour télécharger un fichier

Beaucoup d'entre vous se demandent comment cela se passe pour le téléchargement d'un fichier... En fait, il faut procéder exactement comme si vous faisiez un lien vers une page web, mais en indiquant cette fois le nom du fichier à télécharger.

Par exemple, supposez que vous vouliez faire télécharger monfichier.zip. Placez simplement ce fichier dans le même dossier que votre page web (ou dans un sous-dossier) et faites un lien vers ce fichier :

```
<p><a href="monfichier.zip">Télécharger le fichier</a></p>
```

C'est tout ! Le navigateur, voyant qu'il ne s'agit pas d'une page web à afficher, va lancer la procédure de téléchargement lorsqu'on cliquera sur le lien.

En résumé

- Les liens permettent de changer de page et sont, par défaut, écrits en bleu et soulignés.
- Pour insérer un lien, on utilise la balise <a> avec l'attribut href pour indiquer l'adresse de la page cible. Exemple : .
- On peut faire un lien vers une autre page de son site simplement en écrivant le nom du fichier : .
- Les liens permettent aussi d'amener vers d'autres endroits sur la même page. Il faut créer une ancre avec l'attribut id pour « marquer » un endroit dans la page, puis faire un lien vers l'ancre comme ceci : .

Les images

Visionner la vidéo du Chapitre 5 de la partie 1 sur Vimeo

Insérer une image dans une page web ? Vous allez voir, c'est d'une facilité déconcertante... Enfin presque. Il existe différents **formats** d'image que l'on peut utiliser sur des sites web, et on ne doit pas les choisir au hasard. En effet, les images sont parfois volumineuses à télécharger, ce qui ralentit le temps de chargement de la page (beaucoup plus que le texte !).

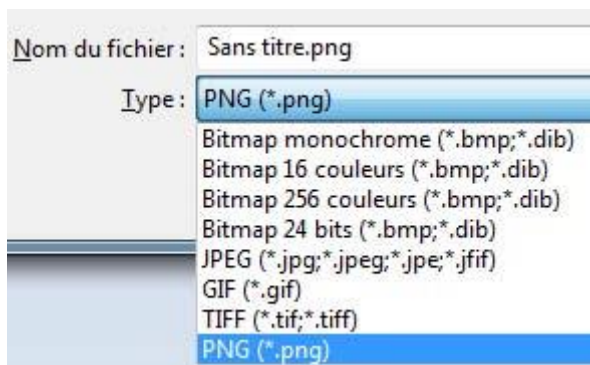
Pour faire en sorte que vos pages restent lisibles et rapides à télécharger, suivez donc activement mes conseils !

Les différents formats d'images

Savez-vous ce qu'est un format d'image ?

Quand vous avez une image « entre les mains », vous avez la possibilité de l'enregistrer dans plusieurs « formats » différents. Le poids (en Ko, voire en Mo) de l'image sera plus ou moins élevé selon le format choisi et la qualité de l'image va changer.

Par exemple, le logiciel de dessin Paint (même si c'est loin d'être le meilleur) vous propose de choisir entre plusieurs formats lorsque vous enregistrez une image (figure suivante).



Différents formats d'image proposés par Paint

Certains formats sont plus adaptés que d'autres selon l'image (photo, dessin, image animée...). Notre but ici est de faire le tour des différents formats utilisés sur le Web pour que vous les connaissiez et sachiez choisir celui qui convient le mieux à votre image. Rassurez-vous, il n'y a pas beaucoup de formats différents, cela ne sera donc pas bien long.

Toutes les images diffusées sur Internet ont un point commun : elles sont **compressées**. Cela veut dire que l'ordinateur fait des calculs pour qu'elles soient moins lourdes et donc plus rapides à charger.

Le JPEG

Les images au format JPEG (*Joint Photographic Expert Group*) sont très répandues sur le Web. Ce format est conçu pour réduire le poids des photos (c'est-à-dire la taille du fichier associé), qui peuvent comporter plus de 16 millions de couleurs différentes. La figure suivante est une photo enregistrée au format JPEG.



Une photo de montagne en JPEG

Les images JPEG sont enregistrées avec l'extension .jpg ou .jpeg.

Notez que le JPEG détériore un peu la qualité de l'image, d'une façon généralement imperceptible. C'est ce qui le rend si efficace pour réduire le poids des photos. Quand il s'agit d'une photo, on ne peut généralement pas détecter la perte de qualité. Par contre, si ce n'est pas une photo, vous risquez de voir l'image un peu « baver ». Dans ce cas, il vaut mieux utiliser le format PNG.

Le PNG

Le format PNG (*Portable Network Graphics*) est le plus récent de tous. Ce format est adapté à la plupart des graphiques (je serais tenté de dire « à tout ce qui n'est pas une photo »). Le PNG a deux gros avantages : il peut être rendu transparent et il n'altère pas la qualité de l'image.

Le PNG a été inventé pour concurrencer un autre format, le GIF, à l'époque où il fallait payer des royalties pour pouvoir utiliser des GIF. Depuis, le PNG a bien évolué et c'est devenu le format le plus puissant pour enregistrer la plupart des images.

Le PNG existe en deux versions, en fonction du nombre de couleurs que doit comporter l'image :

- PNG 8 bits : 256 couleurs ;
- PNG 24 bits : 16 millions de couleurs (autant qu'une image JPEG).

La figure suivante est une image PNG en 24 bits, représentant Zozor, qui sera notre mascotte tout au long de ce cours.



Zozor en PNG

Au fait, si le PNG 24 bits peut afficher autant de couleurs qu'une image JPEG, et qu'en plus il peut être rendu transparent sans modifier la qualité de l'image... quel est l'intérêt du JPEG ?

La compression du JPEG est plus puissante sur les photos. Une photo enregistrée en JPEG se chargera toujours beaucoup plus vite que si elle était enregistrée en PNG. Je vous conseille donc toujours de réserver le format JPEG aux photos.

Le GIF

C'est un format assez vieux, qui a été néanmoins très utilisé (et qui reste très utilisé par habitude). Aujourd'hui, le PNG est globalement bien meilleur que le GIF : les images sont généralement plus légères et la transparence est de meilleure qualité. Je vous recommande donc d'utiliser le PNG autant que possible.

Le format GIF est limité à 256 couleurs (alors que le PNG peut aller jusqu'à plusieurs millions de couleurs).

Néanmoins, le GIF conserve un certain avantage que le PNG n'a pas : il peut être animé. D'où l'explosion ces dernières années des GIF animés sur le web (aussi appelé "*reaction gifs*").



Un GIF animé

Il existe un format adapté à chaque image

Si on résume, voici quel format adopter en fonction de l'image que vous avez :

- **Une photo** : utilisez un JPEG.
- **N'importe quel graphique avec peu de couleurs** (moins de 256) : utilisez un PNG 8 bits ou éventuellement un GIF.
- **N'importe quel graphique avec beaucoup de couleurs** : utilisez un PNG 24 bits.
- **Une image animée** : utilisez un GIF animé. **Les erreurs à éviter Bannissez les autres formats** Les autres formats non cités ici, comme le format BITMAP (*.bmp) sont à bannir car bien souvent ils ne sont pas compressés, donc trop gros. Ils ne sont pas du tout adaptés au Web. On peut en mettre sur son site mais le chargement sera vraiment *extrêmement* long !

Choisissez bien le nom de votre image

Si vous voulez éviter des problèmes, prenez l'habitude d'enregistrer vos fichiers avec des noms en minuscules, sans espace ni accent, par exemple : mon_image.png.

Vous pouvez remplacer les espaces par le caractère *underscore* (« _ ») comme je l'ai fait ici.

Insérer une image

Revenons maintenant au code HTML pour découvrir comment placer des images dans nos pages web !

Insertion d'une image

Quelle est la fameuse balise qui va nous permettre d'insérer une image ? Il s'agit de... !

C'est une balise de type **orpheline** (comme
). Cela veut dire qu'on n'a pas besoin de l'écrire en deux exemplaires comme la plupart des autres balises que nous avons vues jusqu'ici. En effet, nous n'avons pas besoin de délimiter une portion de texte, nous voulons juste insérer une image à un endroit précis.

La balise doit être accompagnée de deux attributs obligatoires :

- **src** : il permet d'indiquer où se trouve l'image que l'on veut insérer. Vous pouvez soit mettre un chemin absolu (ex. : `http://www.site.com/fleur.png`), soit mettre le chemin en relatif (ce qu'on fait le plus souvent). Ainsi, si votre image est dans un sous-dossier images, vous devrez taper : `src="images/fleur.png"`

- alt: cela signifie « texte alternatif ». On doit *toujours* indiquer un texte alternatif à l'image, c'est-à-dire un court texte qui décrit ce que contient l'image. Ce texte sera affiché à la place de l'image si celle-ci ne peut pas être téléchargée (cela arrive), ou dans les navigateurs de personnes handicapées (non-voyants) qui ne peuvent malheureusement pas « voir » l'image. Cela aide aussi les robots des moteurs de recherche pour les recherches d'images. Pour la fleur, on mettrait par exemple : alt="Une fleur".

Les images doivent se trouver obligatoirement à l'intérieur d'un paragraphe (<p></p>). Voici un exemple d'insertion d'image :

```
<p>
  Voici une photo que j'ai prise lors de mes dernières vacances à la
  montagne :<br />
  
</p>
```

Bref, l'insertion d'image est quelque chose de très facile pour peu qu'on sache indiquer où se trouve l'image, comme on avait appris à le faire avec les liens. La plus grosse « difficulté » (si on peut appeler cela une difficulté) consiste à choisir le bon format d'image. Ici, c'est une photo donc c'est évidemment le format JPEG qu'on utilise.

Il faudrait supprimer les espaces (ou les remplacer par le symbole « _ »), supprimer les accents et tout mettre en minuscules comme ceci :

"Images du site/Image toute bête.jpg".

Sachez donc que, si votre image ne s'affiche pas, c'est très certainement parce que le chemin est incorrect ! Simplifiez au maximum vos noms de fichiers et de dossiers, et tout ira bien.

"images_du_site/image_toute_bete.jpg".

Ajouter une infobulle

L'attribut permettant d'afficher une bulle d'aide est le même que pour les liens : il s'agit de title. Cet attribut est facultatif (contrairement à alt).

Voici ce que cela peut donner :

Je le répète : évitez à tout prix les accents, majuscules et espaces dans vos noms de fichiers et de dossiers. Voici un chemin qui va poser problème :

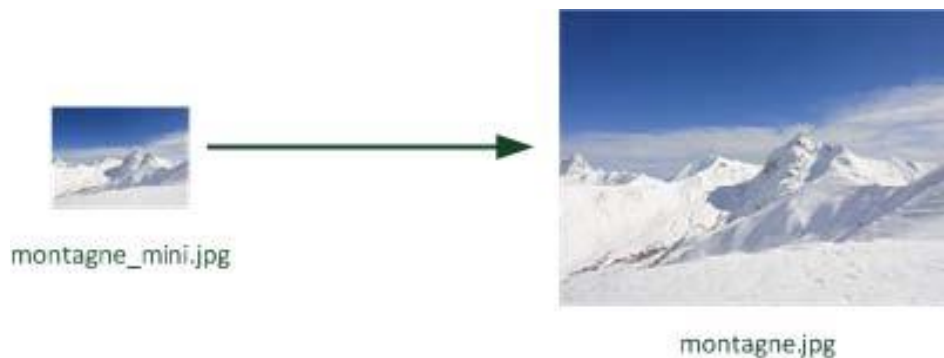
```
<p>
  Voici une photo que j'ai prise lors de mes dernières vacances à la
  montagne :<br />
  
</p>
```


Survolez la photo avec la souris pour voir l'infobulle apparaître.

Miniature cliquable

Si votre image est très grosse, il est conseillé d'en afficher la miniature sur votre site. Ajoutez ensuite un lien surcette miniature pour que vos visiteurs puissent afficher l'image en taille originale.

Il existe des millions de logiciels permettant de créer des miniatures d'images. J'utilise personnellement Easy Thumbnails. Je vais ainsi disposer de deux versions de ma photo, comme à la figure suivante) : la miniature et l'image d'origine.



La miniature et son image d'origine

Je les place toutes les deux dans un dossier appelé par exemple img. J'affiche la version montagne_mini.jpg sur ma page et je fais un lien vers montagne.jpg pour que l'image agrandie s'affiche lorsqu'on clique sur la miniature.

Voici le code HTML que je vais utiliser pour cela :

```
<p>
  Vous souhaitez voir l'image dans sa taille d'origine ? Cliquez dessus
  !<br />
  <a href="img/montagne.jpg"></a>
</p>
```

Parfois, certains navigateurs choisissent d'afficher un cadre bleu (ou violet) pas très esthétique autour de votre image cliquable. Heureusement, nous pourrions retirer ce cadre dans peu de temps grâce au CSS.

Les figures

Au cours de la lecture de ce livre, vous avez déjà rencontré plusieurs fois des **figures**. Ce sont des éléments qui viennent enrichir le texte pour compléter les informations de la page.

Les figures peuvent être de différents types :

- images ;

- codes source ;
- citations ;
- etc.

Bref, tout ce qui vient *illustrer* le texte est une figure. Nous allons ici nous intéresser aux images mais, contrairement à ce qu'on pourrait croire, les figures ne sont pas *forcément* des images : un code source aussi illustre le texte.

Création d'une figure

Reprenons par exemple cette capture d'écran du premier chapitre, représentée à la figure suivante.



Le logiciel Bloc-Notes

En HTML5, on dispose de la balise `<figure>`. Voici comment on pourrait l'utiliser :

```
<figure>
  
</figure>
```

Une figure est le plus souvent accompagnée d'une légende. Pour ajouter une légende, utilisez la balise `<figcaption>` à l'intérieur de la balise `<figure>`, comme ceci :

```
<figure>
  
  <figcaption>Le logiciel Bloc-Notes</figcaption>
</figure>
```

Bien comprendre le rôle des figures

Un peu plus tôt dans ce chapitre, je vous ai dit que les images devaient être situées dans des paragraphes (placées à l'intérieur d'une balise `<p></p>`). Ce n'est pas tout à fait vrai.

Si vous faites de votre image une figure, l'image peut être située en-dehors d'un paragraphe.

```
<p>Connaissez-vous le logiciel Bloc-Notes ? On peut faire des sites web avec !</p>

<figure>
  
  <figcaption>Le logiciel Bloc-Notes</figcaption>
</figure>
```

Je ne vois pas vraiment de changement. Quand dois-je placer mon image dans un paragraphe et quand dois-je la placer dans une figure ?

Bonne question ! Tout dépend de ce que votre image apporte au texte :

- Si elle n'apporte aucune information (c'est juste une illustration pour décorer) : placez l'image dans un paragraphe.
- Si elle apporte une information : placez l'image dans une figure.

La balise `<figure>` a un rôle avant tout **sémantique**. Cela veut dire qu'elle indique à l'ordinateur que l'image a du sens et qu'elle est importante pour la bonne compréhension du texte. Cela peut permettre à un programme de récupérer toutes les figures du texte et de les référencer dans une table des figures, par exemple.

Enfin, sachez qu'une figure peut très bien comporter plusieurs images. Voici un cas où cela se justifie :

```
<figure>
  
  
  
  <figcaption>Logos des différents navigateurs</figcaption>
</figure>
```

En résumé

- Il existe plusieurs formats d'images adaptées au Web :
 - o JPEG : pour les photos ;

- o PNG : pour toutes les autres illustrations ;
- o GIF : similaire au PNG, plus limité en nombre de couleurs mais qui peut être animé.
- On insère une image avec la balise ``. Elle doit obligatoirement comporter au moins ces deux attributs : `src`(nom de l'image) et `alt`(courte description de l'image).
- Si une image illustre le texte (et n'est pas seulement décorative), il est conseillé de la placer au sein d'une balise `<figure>`. La balise `<figcaption>` permet d'écrire la légende de l'image.