

Table des matières

TABLE DES MATIÈRES	1
INTRODUCTION AU DÉVELOPPEMENT WEB	1
<i>INTERNET, COMMENT ÇA FONCTIONNE ?</i>	<i>2</i>
<i>LES NAVIGATEURS WEB</i>	<i>3</i>
<i>HTML ET CSS, C'EST QUOI ?</i>	<i>4</i>
<i>ORIGINE DU LANGAGE HTML : UNE RÉVOLUTION DANS LA COMMUNICATION NUMÉRIQUE</i>	<i>6</i>
<i>LES DIFFÉRENTES VERSIONS DE HTML ET CSS</i>	<i>7</i>
<i>LA STANDARDISATION DU WEB : ENTRE RECOMMANDATIONS ET RÉALITÉS TECHNIQUES</i>	<i>8</i>
<i>LA DIVERSITÉ NUMÉRIQUE : UN DÉFI POUR LES DÉVELOPPEURS WEB</i>	<i>9</i>
<i>IMPORTANCE DES STANDARDS POUR L'INTEROPÉRABILITÉ</i>	<i>10</i>
<i>PRÉALABLES À LA CRÉATION D'UN DOCUMENT HTML</i>	<i>12</i>
<i>LES ÉLÉMENTS STRUCTURAUX DE BASE POUR TOUS FICHIERS HTML</i>	<i>12</i>
<i>LES TYPES DE BALISES</i>	<i>14</i>

Introduction au développement web

Le développement d'un site web est un processus méthodique et réfléchi qui nécessite la collaboration étroite entre programmeurs et entrepreneurs. Derrière chaque page web que vous consultez se cachent des professionnels qui ont soigneusement conçu chaque élément visuel et fonctionnel, transformant l'idée d'un client en une vitrine numérique accessible au monde entier.

Ce chapitre a pour objectif de vous faire comprendre les mécanismes essentiels de la création d'un site web. Nous explorerons les origines historiques des langages de programmation web, décrirons leurs fonctions spécifiques et mettrons en lumière les différents outils utilisés par les développeurs pour créer des pages web dynamiques et interactives.

Enfin, nous découvrirons Visual Studio Code, un environnement de développement intégré gratuit et puissant qui permet aux programmeurs de concevoir et de réaliser leurs projets web avec efficacité et créativité.

OBJECTIFS

Découvrir les origines du développement Web.

Identifier les principaux langages utilisés dans le développement d'une page Web.

Découvrir et utiliser l'environnement de développement intégré Visual Studio Code.

Internet, comment ça fonctionne ?

Lorsque vous visitez un site Web, vous saisissez son adresse dans votre navigateur ou fureteur (browser). Que ce soit Internet Explorer, Microsoft Edge, Mozilla Firefox, Opera, Safari ou Google Chrome, pour ne nommer que ceux-là, la page Web que vous demandez ne s'affiche pas par magie et, le tout demeure transparent à l'utilisateur.



Figure 1 Icônes des principaux navigateurs Internet sur le marché logiciel

Deux cas se présentent à nous. Dans un premier temps, nous pouvons demander à un site Web de nous retourner une page dite statique. Ce type de page ne contient que de l'information brute qui doit être mise-à-jour manuellement, c'est-à-dire, dans le code même de la page.

Dans un tout autre ordre d'idées, les pages peuvent être dynamiques, c'est-à-dire qu'elles affichent l'information que nous désirons. Cette information est normalement entreposée dans une base de données et retournée pour affichage selon le ou les formats programmés par vous en tant que programmeur de site Web.

Peu importe l'appareil que vous utilisez lors de la requête d'une page Web, le principe de communication demeure le même.

1. Une requête de page web est envoyée par le client, votre ordinateur.
2. La requête est acheminée vers le serveur web via le réseau Internet.
3. Le serveur récupère le contenu de la page web demandée.
4. Le serveur envoie la page au format HTML via le réseau Internet.
5. Votre navigateur affiche la page Web reçue.

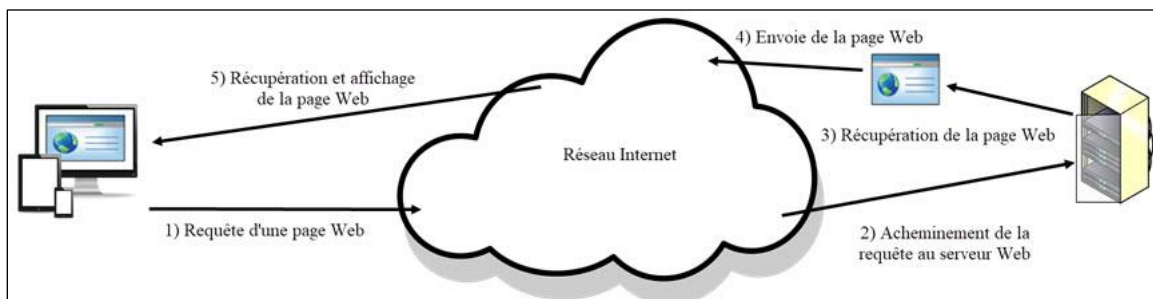


Figure 2 Requête entre un client et un serveur

Votre navigateur a le rôle d'interpréter et d'afficher le code retourné par le serveur qui entrepose la page Web.

Les navigateurs web

Voici un aperçu des principales fonctionnalités et différences entre les navigateurs web les plus populaires : Google Chrome, Mozilla Firefox, Microsoft Edge et Apple Safari.

Google Chrome

- **Performance** : Chrome est connu pour sa rapidité et son efficacité. Il utilise le moteur de rendu Blink, ce qui lui permet de charger les pages rapidement.
- **Extensions** : Chrome dispose d'une vaste bibliothèque d'extensions disponibles sur le Chrome Web Store, permettant de personnaliser l'expérience utilisateur.
- **Synchronisation** : Si vous êtes connecté avec un compte Google, vous pouvez synchroniser vos favoris, mots de passe et historiques entre différents appareils.
- **Mises à jour fréquentes** : Chrome reçoit des mises à jour régulières qui améliorent la sécurité et ajoutent de nouvelles fonctionnalités.
- **Interface utilisateur** : Simple et minimaliste, avec une barre d'adresse intégrée et une gestion des onglets fluide.

Mozilla Firefox

- **Vie privée** : Firefox se concentre sur la protection de la vie privée des utilisateurs, avec des fonctionnalités telles que le blocage par défaut des traqueurs et une navigation privée renforcée.
- **Personnalisation** : Firefox offre une grande flexibilité en matière de personnalisation, avec de nombreux thèmes et options de configuration.
- **Extensions** : Bien que moins nombreuses que celles de Chrome, Firefox a une bonne sélection d'extensions disponibles sur le site officiel.
- **Moteur de rendu** : Utilise le moteur Gecko, qui assure une bonne compatibilité avec les normes web.
- **Fonctionnalités avancées** : Intègre des fonctionnalités comme le mode Picture-in-Picture et des outils de capture d'écran.

Microsoft Edge

- **Performance** : Basé sur Chromium (comme Chrome), Edge est rapide et performant, avec une intégration améliorée dans Windows 10 et 11.
- **Fonctionnalités de productivité** : Edge propose des fonctionnalités comme la collection (pour organiser des informations) et une lecture immersive (pour simplifier la lecture d'articles).
- **Sécurité** : Inclut des fonctionnalités de sécurité avancées, comme la protection contre les sites malveillants et un mode de navigation InPrivate.
- **Intégration avec Windows** : Profite d'une intégration native avec le système d'exploitation Windows, y compris les fonctionnalités de synchronisation avec un compte Microsoft.
- **Extensions** : Peut utiliser les extensions de Chrome en plus de celles disponibles sur le Microsoft Store.

Apple Safari

- **Optimisation pour macOS et iOS** : Safari est conçu pour fonctionner de manière optimale sur les appareils Apple, offrant une excellente performance et une bonne gestion de la batterie.
- **Vie privée** : Met l'accent sur la protection de la vie privée, avec des fonctionnalités comme Intelligent Tracking Prevention (ITP) pour limiter le suivi des utilisateurs.
- **Interface utilisateur** : Interface épurée et minimaliste, avec des fonctionnalités comme le mode plein écran et des onglets regroupés.
- **Fonctionnalités de synchronisation** : Permet de synchroniser les signets, l'historique et les onglets ouverts entre les appareils Apple via iCloud.
- **Extensions** : Moins d'extensions disponibles par rapport à Chrome et Firefox, mais les options s'élargissent progressivement.

En résumé :

- **Performance** : Chrome et Edge sont souvent considérés comme les plus rapides, tandis que Safari est optimisé pour les appareils Apple.
- **Vie privée** : Firefox et Safari se concentrent davantage sur la protection de la vie privée comparé à Chrome et Edge.
- **Extensions** : Chrome offre la plus grande bibliothèque d'extensions, suivi de Firefox, Edge et enfin Safari.
- **Intégration système** : Edge et Safari sont plus intégrés dans leurs systèmes respectifs (Windows et macOS/iOS).

En fin de compte, le meilleur choix de navigateur dépend des préférences personnelles, des besoins en matière de fonctionnalités et des préoccupations en matière de confidentialité. Malgré ces différences entre tous ces navigateurs, il est important de vérifier régulièrement l'affichage de nos pages web sur ces derniers afin de s'assurer que le rendu visuel sur chacun d'eux soit uniforme. Il est toujours possible qu'il y ait certaines différences minimales dans les affichages d'un navigateur à un autre puisque chacun des navigateurs interprète le code d'une page différemment.

Dans ce premier cours, nous allons nous limiter à la composition de sites Web statique en mettant l'accent sur le système de balisage du langage HTML5 de concert avec le langage CSS3 pour styliser l'affichage de vos pages. Nous étudierons, dans un prochain cours, comment dynamiser une page web en utilisant le langage JavaScript.

HTML et CSS, c'est quoi ?

HTML signifie « *HyperText Markup Language* », ou langage hypertexte. Un document HTML est un fichier ASCII contenant le texte à afficher et les codes ou **balises** assurant la logique d'affichage, la **mise en forme** se fera avec le langage **CSS** que nous découvrirons dans ce cours. **HTML** est le langage utilisé pour créer des pages destinées au World Wide Web (WWW), souvent appelé le **Web**. **HTML** est un langage de programmation très simple. En fait, il s'agit simplement d'un ensemble de balises spéciales qui indiquent aux différents navigateurs Web comment disposer les éléments

que contient une page Web. Pour comprendre un fichier **HTML**, il suffit de savoir comment utiliser et positionner nos différents éléments dans un fichier .html.

L'hypertexte est un ensemble de codes sous forme écrite qui servent à relier des documents. Les liens et les ancres sont des concepts que nous découvrirons dans ce cours. Ces deux éléments servent de connecteurs entre les pages Web.

Le langage de balisage indique aux navigateurs comment ils doivent positionner les éléments contenus dans un fichier **HTML**. Ces éléments peuvent être du texte, des images, de la vidéo ou des sons.

Le **CSS** est le langage utilisé afin de styliser le contenu de votre page. Vous pourrez dire : je veux que mon menu soit à gauche et occupe telle largeur, que l'entête de mon site soit calé en haut et qu'il soit toujours visible, que le contenu textuel de ma page s'affiche en utilisant cette couleur, etc.

Imaginez-vous un squelette sur lequel vous ajoutez de la peau et un habillement, vous obtenez ainsi un homme bien présenté. Vous obtiendrez un résultat semblable en combinant un fichier HTML avec le CSS. En rassemblant tous ces éléments, on obtient un outil très puissant qui nous permet de créer une présence visuelle assez percutante sur Internet.

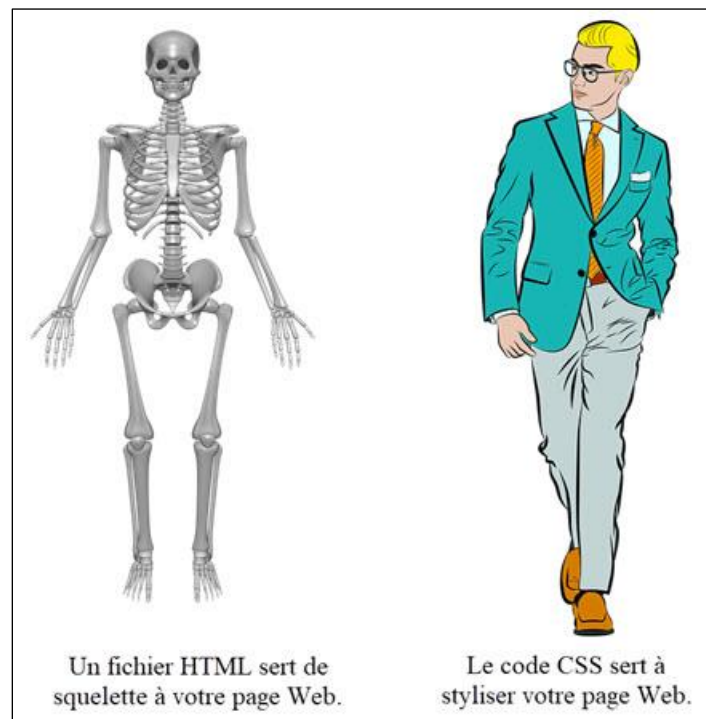


Figure 3 Interaction entre le HTML et le CSS

Pour terminer, le langage JavaScript, qui sera couvert dans un autre cours, nous permettra d'apporter un souffle de vie, un peu de dynamisme à nos pages web, en y ajoutant la possibilité pour l'utilisateur de nos pages d'interagir avec celle-ci.

En quelques lignes, un fichier **HTML** est un simple fichier texte qui contient les informations à afficher ainsi que les balises qui identifient les éléments d'une page Web comme les entêtes, les menus, les titres, les liens et les ancrs, les fichiers multimédia (images, vidéo et sons) et, un fichier **CSS** nous permet de styliser l'affichage du contenu d'un fichier **HTML**. Tous les fichiers **HTML** doivent avoir l'extension **.htm** ou **.html** sans quoi, ils ne seront pas reconnus par les navigateurs qui les afficheront alors comme de simples fichiers texte sans mise en forme.

Origine du langage HTML : Une révolution dans la communication numérique

À l'aube d'Internet, le partage de fichiers était complexe et chaotique. Les différents navigateurs interprétaient les formats de manière disparate, engendrant des problèmes d'affichage récurrents : marges incohérentes, polices déformées, styles mal rendus. Cette incompatibilité rendait la lecture des documents numériques difficile et frustrantes.

Pour résoudre ces défis, le langage SGML (Standard Generalized Markup Language) fut développé. Cette première tentative utilisait des marqueurs spéciaux pour définir et structurer les contenus, permettant une interprétation plus uniforme des documents. Bien que représentant une avancée significative, SGML ne résolvait pas totalement les problèmes d'interopérabilité.

C'est alors qu'intervint Sir Tim Berners-Lee, créateur du HTML (HyperText Markup Language). Ce langage reprend le principe des marqueurs, mais les simplifie et les rend universellement compréhensibles. Les balises HTML permettent aux développeurs de formater et structurer les informations de manière précise, garantissant un affichage cohérent quel que soit le navigateur utilisé.

Visionnaire, il continue d'influencer l'écosystème web. Il a fondé le World Wide Web Consortium (W3C), organisme qui définit et standardise les évolutions des langages web. Plus récemment, il a créé la World Wide Web Foundation, qui analyse et accompagne la transformation numérique mondiale.

Aujourd'hui, les programmeurs web utilisent simplement ces balises HTML pour donner aux pages l'apparence exacte souhaitée. Les navigateurs modernes interprètent ces instructions avec une précision remarquable, transformant des lignes de code en expériences visuelles riches et dynamiques.



Figure 4 Sir Tim Berners-Lee

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 en commençant par le HTML :

- HTML1 : c'est la toute première version créée par Sir Tim Berners-Lee en 1991.
- HTML2 : la deuxième version du HTML apparaît en 1994 et se terminera en 1996 avec l'apparition du HTML 3.0. C'est cette version qui posera en fait les bases des prochaines versions du HTML. Les règles et le fonctionnement de cette version sont donnés par le W3C tandis que, la première version a été créée par un seul homme.
- HTML3 : 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...
- HTML4 : il s'agit de la version la plus répandue de HTML, plus précisément il s'agit de HTML 4.01. Elle apparaît pour la première fois en 1998, et propose l'utilisation de cadres ou 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 l'utilisation de feuilles de style, notre fameux **CSS** !
- HTML5 : 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.

Au tour du langage CSS :

- CSS1 : dès 1996, la première version du CSS est utilisable. Elle pose les bases de ce langage qui permet de styliser sa page web, comme les couleurs, les marges, les polices de caractères etc...
- CSS2 : apparue en 1999 puis complétée ensuite 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.
- CSS3 : c'est la dernière version qui apporte des fonctionnalités particulièrement attendues comme les bordures arrondies, les dégradés, les ombres, etc.

La standardisation du web : entre recommandations et réalités techniques

Le W3C (World Wide Web Consortium) joue un rôle central dans la standardisation des technologies web. Voici ses principales fonctions et responsabilités :

Élaboration de standards web ouverts

Le W3C développe et publie des normes pour les technologies web afin de garantir leur compatibilité, leur accessibilité et leur interopérabilité sur différentes plateformes et appareils. Ces normes incluent, entre autres :

- HTML (HyperText Markup Language),
- CSS (Cascading Style Sheets),
- XML (Extensible Markup Language),
- SVG (Scalable Vector Graphics).

Favoriser l'interopérabilité

Le W3C garantit que les technologies web fonctionnent de manière cohérente sur différents navigateurs, systèmes d'exploitation et appareils. Cela permet aux développeurs de créer des applications utilisables par le plus grand nombre sans avoir à adapter leurs solutions pour des environnements spécifiques.

Encourager l'accessibilité

L'une des priorités du W3C est de rendre le web accessible à tous, y compris aux personnes ayant des handicaps. Les directives comme les WCAG (Web Content Accessibility Guidelines) offrent des recommandations pour concevoir des sites web et des applications accessibles.

Collaborer avec des parties prenantes

Le W3C regroupe des membres issus de divers horizons : entreprises technologiques, gouvernements, organisations à but non lucratif, chercheurs, et individus. Ces membres travaillent ensemble pour définir les besoins de la communauté web et élaborer des solutions adaptées.

Promouvoir les meilleures pratiques

Le W3C encourage l'utilisation des bonnes pratiques pour le développement web, notamment en matière de sécurité, de performance, de respect de la vie privée et de durabilité.

Tester et valider les implémentations

Le W3C propose des outils et des frameworks pour tester la conformité des sites web et des applications aux standards qu'il publie (par exemple, les validateurs HTML et CSS).

En résumé

Le rôle du W3C est d'assurer que le web reste ouvert, évolutif, et utilisable par tous en définissant des standards techniques qui favorisent une innovation durable et universelle. Ses travaux permettent au web de se développer en tant que plateforme unifiée et collaborative.

Bien que le World Wide Web Consortium (W3C) établisse des recommandations précises pour la création de pages web, la réalité technique est plus complexe. Chaque navigateur interprète ces standards selon sa propre implémentation, créant des variations subtiles mais potentiellement significatives dans l'affichage des pages.

Cette disparité représente un défi majeur pour les développeurs web. Une balise HTML ou une commande CSS qui s'affiche parfaitement sur un navigateur peut apparaître différemment, voire être mal rendue, sur un autre. Ces écarts peuvent altérer l'expérience utilisateur et compromettre la cohérence visuelle d'un site.

Pour garantir une expérience utilisateur optimale, il est essentiel d'adopter une approche proactive. Notre méthode vous guidera vers les meilleures pratiques : tester systématiquement votre code sur plusieurs navigateurs, vérifier chaque rendu et anticiper les potentielles divergences d'affichage.

Nous vous encouragerons donc à développer le réflexe de test multi-navigateurs. Avant de valider définitivement une page web, assurez-vous de la consulter sur différentes plateformes : Chrome, Firefox, Safari, Edge. Cette démarche vous permettra de détecter et de corriger rapidement les éventuelles inconsistances, offrant ainsi une expérience utilisateur fluide et professionnelle.

La diversité numérique : un défi pour les développeurs web

La mondialisation et la démocratisation d'Internet ont profondément transformé la façon dont le contenu web est consulté. Aujourd'hui, un site web doit être accessible depuis n'importe quel appareil : ordinateurs, smartphones, tablettes, téléviseurs connectés. Les utilisateurs choisissent librement leur navigateur et leur support, sans se soucier des contraintes techniques de conception.

Pour les développeurs, cela signifie relever un double défi : garantir une accessibilité universelle tout en préservant l'identité visuelle du site. La compatibilité multi-plateformes n'est plus un choix, mais une nécessité impérative. Chaque page web doit s'adapter dynamiquement, offrant une expérience cohérente et fluide, quelle que soit la technologie utilisée.

Les technologies émergentes repoussent constamment les limites de la navigation numérique. Un design responsive, des interfaces adaptatives et des techniques de développement flexibles sont désormais essentiels pour répondre à cette diversité croissante des supports et des usages.

Le programmeur web moderne doit donc penser **universel** : créer des expériences numériques inclusives, élégantes et fonctionnelles, capables de séduire et de servir un public global, connecté sur une multitude de dispositifs.

Sir Tim Berners-Lee, directeur du W3C et créateur du World Wide Web nous dit :

« The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect. »

Ce message, concernant l'accessibilité du World Wide Web ne peut être plus clair :

« La force du Web réside dans son universalité. L'accès pour tous, indépendamment du handicap, est un aspect essentiel. » (Traduction libre)

Les technologies de la voix, bien que supportées partiellement par certains navigateurs, se développent peu à peu permettant désormais un accès Internet pour les malvoyants.

Les standards web jouent un rôle essentiel dans le développement web, en particulier pour garantir l'interopérabilité et l'accessibilité. Voici pourquoi ces deux aspects sont cruciaux :

Importance des standards pour l'interopérabilité

L'interopérabilité signifie que les technologies web fonctionnent de manière cohérente sur différentes plateformes, navigateurs et appareils. Les standards assurent cette cohérence grâce aux spécifications uniformisées définies par des organismes comme le W3C.

Avantages de l'interopérabilité :

- **Expérience utilisateur uniforme** : Les utilisateurs peuvent accéder à des sites web et des applications sans rencontrer de problèmes, quel que soit leur appareil ou navigateur.
- **Réduction des coûts de développement** : Les développeurs n'ont pas besoin de créer des versions spécifiques pour chaque navigateur ou système d'exploitation, ce qui réduit le temps et les ressources nécessaires.
- **Innovation et compatibilité** : Les standards favorisent un écosystème où les nouvelles technologies, comme les Progressive Web Apps (PWA) ou WebAssembly, fonctionnent de manière transparente avec les anciens systèmes.

Exemple d'interopérabilité :

Les standards HTML5 et CSS3 permettent aux développeurs de créer des interfaces qui fonctionnent de la même manière dans Chrome, Firefox, Safari et autres navigateurs modernes, tout en restant compatibles avec des appareils variés (ordinateurs, tablettes, smartphones).

Importance des standards pour l'accessibilité

L'accessibilité vise à rendre le web utilisable par tous, y compris par les personnes ayant des handicaps visuels, auditifs, moteurs ou cognitifs. Les standards définissent des pratiques et des technologies pour atteindre cet objectif.

Avantages de l'accessibilité :

- ***Inclusion universelle*** : Tout utilisateur, indépendamment de ses capacités, peut naviguer sur le web. Cela inclut des éléments comme le texte alternatif pour les images, les sous-titres pour les vidéos et des interfaces adaptées aux lecteurs d'écran.
- ***Respect des réglementations*** : Dans de nombreux pays, des lois exigent que les sites respectent des normes d'accessibilité, comme les WCAG (Web Content Accessibility Guidelines).
- ***Meilleure expérience utilisateur*** : Les pratiques d'accessibilité, comme une navigation claire et des contrastes adaptés, bénéficient à tous les utilisateurs, même ceux sans handicaps.

Exemple d'accessibilité :

Un formulaire conçu selon les standards WCAG inclura des étiquettes claires, des indications textuelles pour les champs obligatoires et un support pour la navigation clavier. Cela le rendra accessible aussi bien pour les utilisateurs ordinaires que pour ceux utilisant un lecteur d'écran.

Pourquoi ces deux aspects sont liés ?

Un site interopérable est plus facilement accessible, car il fonctionne sur les technologies et outils d'assistance utilisés par les personnes handicapées (lecteurs d'écran, navigateurs spécialisés, etc.).

Les standards permettent d'unifier les efforts des développeurs pour offrir une expérience cohérente, inclusive et universelle.

Les standards web pour l'interopérabilité et l'accessibilité sont essentiels pour garantir un web ouvert, utilisable et inclusif. Ils permettent à tous, quels que soient leur équipement ou leurs capacités, d'accéder au contenu et aux services en ligne. Respecter ces standards n'est pas seulement une bonne pratique ; c'est un engagement envers une société numérique équitable.

Vous désirez obtenir plus d'informations sur les ***Règles pour l'accessibilité des contenus Web*** (WCAG), nous vous encourageons à visiter le site Web suivant :

<https://www.w3.org/Translations/WCAG21-fr/>

Le défi est de taille et nous espérons que vous êtes prêt à le relever tout comme nous avons accepté de le relever afin de vous transmettre les informations contenues dans ce guide.

Préalables à la création d'un document HTML

L'un des avantages de ces langages est qu'ils ne nécessitent pas de ressources particulières. Pour créer un site Web, un simple éditeur de texte comme Bloc-Notes ou Word-Pad de Windows suffit, un navigateur Web afin de visualiser le résultat et bien évidemment, la connaissance des deux langages : HTML et CSS.

Il y a cependant des logiciels plus puissants aujourd'hui et personne n'utilise vraiment le Bloc-Notes pour la création de sites Web. 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 : Nvu, Microsoft Expression Web, Dreamweaver... et même Word ! Leur principal défaut est la qualité du code HTML et CSS qui est automatiquement généré par ces outils. Ce code est souvent d'assez mauvaise qualité. Un bon créateur de site web doit tôt ou tard connaître HTML et CSS, c'est pourquoi nous ne recommandons pas l'usage de ces outils.
- Les environnements de développement intégré : ce sont des programmes dédiés à l'écriture de code comme Notepad++ ou Visual Studio Code que l'on peut télécharger gratuitement sur Internet. 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 !

Note : Lors du développement de ce guide d'études, nous avons utilisé l'environnement de développement **Visual Studio Code**, que l'on peut se procurer gratuitement sur Internet. Il s'agit d'un environnement de développement polyvalent qui est en mesure de vous aider lors de la composition de vos pages Web.

Les éléments structuraux de base pour tous fichiers HTML

Examinez la prochaine figure. Cette figure affiche les éléments de base qui composent une page Web.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9
10 </body>
11 </html>
```

Figure 5 Éléments de base de tout fichier HTML

Cette structure de base est le point de départ pour créer n'importe quelle page web. Elle définit les éléments essentiels requis par le navigateur pour interpréter et afficher correctement le contenu. Reprenons cette structure de base ligne par ligne afin de découvrir en quoi elle consiste.

Sur la ligne 1 : la balise `<!DOCTYPE html>` indique au navigateur que le document est écrit en HTML5, le standard HTML actuel. Elle permet de s'assurer que le navigateur interprète correctement le contenu de la page.

Sur la ligne 2 : la balise `<html lang="fr">` représente l'élément racine de tout document HTML. L'attribut `lang="fr"` spécifie que la langue principale du document est le français. Cette balise doit inclure une balise de fermeture, la balise `</html>` que vous retrouvez à la fin du document, la ligne 11 dans notre exemple.

Sur la ligne 3 : la balise `<head>` contient les métadonnées du document, qui ne sont pas directement affichées sur la page. Cette balise se doit d'être fermée avec une balise de fermeture, la balise `</head>` que vous retrouvez sur la ligne 7. Entre ces deux balise se trouve les informations importantes pour le navigateur et les moteurs de recherche.

Sur la ligne 4 : la balise `<meta charset="UTF-8">` définit l'encodage des caractères utilisé dans le document. `UTF-8` est l'encodage le plus couramment utilisé, car il prend en charge une large gamme de caractères.

Sur la ligne 5 : la balise `<meta name="viewport">` est cruciale pour le design web responsive. Elle dit au navigateur :

- D'adapter automatiquement la largeur de la page à la largeur de l'écran de l'appareil
- De définir le zoom initial à 100%

En d'autres termes, cette balise garantit que votre site web s'affiche correctement sur tous les types d'écrans, des smartphones aux ordinateurs de bureau, en ajustant dynamiquement la mise en page et l'échelle.

Sur la ligne 6 : la balise `<title>Document</title>` définit le titre de la page, qui s'affiche dans l'onglet du navigateur. C'est également un élément important pour le référencement naturel (SEO). Nous discuterons du référencement à plusieurs endroits dans ce guide d'études.

Par exemple, si je donne le titre *Accueil – Animalerie Becs et Museaux* à ma page, `<title>Accueil - Animalerie Becs et Museaux </title>`, le titre sera affiché sur le navigateur du client comme le montre la prochaine figure.

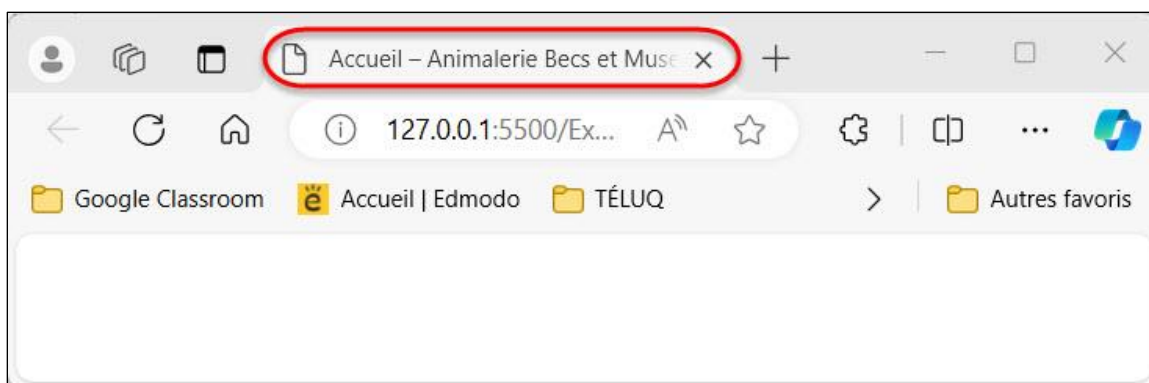


Figure 6 Résultat de la modification du contenu de la balise <title>

Note : Un titre ne peut contenir de mise en forme, d'images ou de liens vers d'autres pages. De plus, il est recommandé de ne pas utiliser les deux points (:) ni la barre oblique inversée (\) dans un titre.

Sur la ligne 8 : la balise **<body>** ouvre la section dans laquelle vous ajouterez les éléments visibles de la page web. Cette balise doit aussi comporter une balise de fermeture, la balise **</body>** que vous retrouvez sur la ligne 10 de notre exemple. C'est entre ces deux balises que vous placerez tous les éléments HTML qui seront affichés à l'utilisateur.

Les types de balises

Comme vous avez été en mesure de le constater avec l'image de la figure 5, il existe deux types de balises en HTML soit :

- Les balises contenantes (ou balises de paire) :
 - Elles ont une balise ouvrante et une balise fermante
 - Le contenu se trouve entre ces deux balises
- Les balises simples (ou balises orphelines) :
 - Elles n'ont pas de balise de fermeture
 - Elles se referment directement dans leur propre définition

Une balise contenante, comme le dit son nom, est une balise qui forme un bloc spécifique. Dans l'exemple de la première figure, nous retrouvons quatre exemples de ce type de balise. Le premier exemple est celui de la balise **<html>** qui doit inclure la balise de fermeture **</html>**. Ces deux balises sont indissociables et forme le bloc principal de notre fichier html.

Le deuxième exemple est celui de la balise **<head>** qui doit inclure la balise de fermeture **</head>**. Encore une fois, ces deux balises sont indissociables et du contenu est ajouté à l'intérieur de ce bloc. Notre troisième exemple est composé des balises **<title>** et **</title>** incluse dans notre bloc composé des balises **<head>** et **</head>**. Ce dernier bloc contient le seul élément qui peut être affiché parmi tous les éléments du bloc d'entête. Pour terminer, nous retrouvons le bloc entouré des balises **<body>** et **</body>** qui, à son tour, contiendra aussi des ensembles de balises afin de composer le contenu de la page Web pour laquelle le fichier sera composé.

Il faut porter attention et s'assurer de ne pas inclure la paire de balises `<body>` et `</body>` à l'intérieur du bloc composé par les balises `<head>` et `</head>` et vice-versa. Assurez-vous de bien séparer ces deux blocs sans quoi, il sera plus facile de commettre des erreurs qui peuvent entraîner un comportement étrange du navigateur.

Une balise simple ou orpheline est une balise qui ne possède aucun contenu mis-à-part des attributs. C'est le cas des balises `<meta>` incluse dans le bloc `<head>` `</head>` de la figure de la page 9. Il en existe d'autres qui peuvent être utilisés, pour l'affichage des fichiers multimédias par exemple et dont nous étudierons l'utilisation dans la suite de ce guide.

Note importante : Par le passé, une balise simple devait être complétée en y ajoutant une barre oblique à la fin, comme le montre cet exemple :

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

Figure 7 Composition d'une balise simple avant les nouvelles normes du W3C

Aujourd'hui, selon les règles du W3C, il n'est plus nécessaire d'ajouter cette barre oblique à la fin d'une balise simple. Nous avons souligné cette petite règle afin que vous ne soyez pas surpris de ce type d'écriture si jamais vous devez revisiter le code d'un site qui a été composé avant que cette norme ne soit modifiée.