



[← Revenir au sommaire.](#)

# Introduction au cours d'HTML

- 2. Historique
- 3. Schéma de la communication entre deux personnes
  - 3.1 Exemple 1: un client et une serveuse
  - 3.2 Exemple 2: une serveuse et un cuisinier
  - 3.3 Exemple 3: un client, une serveuse et un cuisinier
- 4. Schéma de communication entre un navigateur web et un serveur web
- Le HTML
- Les versions du HTML
- 5. Deux organismes de standardisation du Web
  - 5.1 W3C (World Wide Web Consortium)
    - Définition
    - Rôle
  - 5.2 Le WHATWG
    - Définition
    - Rôle
- 6. Les navigateurs
- 7. Les serveurs webs
- 8. Les éditeurs de code
- 9. Sites de référence

# 1. Introduction

Aujourd'hui, **Internet** est devenu une partie intégrante de nos vies. Nos smartphones, nos télévisions, nos montres, nos voitures, nos maisons... tous sont connectés à Internet. Nous utilisons Internet pour communiquer avec nos amis, pour regarder des vidéos, écouter de la musique, jouer à des jeux, effectuer des recherches, et bien plus encore. De nombreuses applications sont désormais basées sur le web, car le web d'antan a évolué pour devenir le web d'aujourd'hui : moderne, rapide et performant.

Les applications web sont des programmes qui s'exécutent dans un navigateur web. Toutefois, il est désormais possible pour certaines applications web d'être installées directement sur un ordinateur ou un smartphone, sans nécessiter un navigateur pour être lancées à chaque utilisation. On parle alors de **PWA** (*Progressive Web App*). L'avantage des PWA est qu'elles peuvent ressembler de très près à des applications natives en termes de performance et de convivialité. Bien que nous n'aborderons pas les PWA en détail dans ce cours, il est intéressant de savoir qu'elles existent et qu'elles représentent une évolution importante du web.

Le **HTML** est un langage de balisage qui permet de structurer le contenu d'une page web. Utilisé en combinaison avec le CSS, qui gère la mise en forme, et le JavaScript, qui ajoute de l'interactivité, le HTML est utilisé sur tous les sites web que nous visitons.

Pour en revenir au CSS et au JavaScript, on pourrait dire que le CSS habille la page web et que le JavaScript lui donne vie.

Il est important de clarifier une idée reçue courante : Internet n'est pas simplement Google ou votre navigateur préféré. Quand quelqu'un affirme : "Je suis sur Internet", il est probable qu'il utilise le World Wide Web, un service d'Internet, à travers un navigateur. Mais Internet est bien plus vaste, offrant une multitude de services tels que le courrier électronique, le FTP, le P2P, les jeux en ligne, et bien d'autres.

Internet est souvent décrit comme le réseau des réseaux. C'est un système mondial de réseaux d'ordinateurs interconnectés, utilisant le protocole TCP/IP pour communiquer entre eux. TCP/IP est l'ensemble des règles qui définissent comment les données sont échangées sur Internet. Il ne s'agit pas seulement d'une interaction client-serveur, mais d'un ensemble complexe de protocoles de communication.

Enfin, Internet est structuré en plusieurs couches, allant de la couche physique, qui concerne le matériel et la transmission de données brutes, à la couche applicative, où fonctionnent des

protocoles tels que HTTP (pour le web), HTTPS (pour le web sécurisé), SMTP (pour l'email), etc. Le HTML, le CSS et le JavaScript opèrent à ce niveau applicatif, formant les piliers de la création de pages web interactives.

J'ajouterais encore (et oui je sais, je parle beaucoup !) que maintenant on parle de HTTP/2 qui est la nouvelle version du protocole HTTP/1.1. Il apporte de grandes améliorations et terme de performance.

Normalement, il n'est pas obligatoire d'utiliser HTTPS pour bénéficier de HTTP/2. Cependant, en pratique, tous les principaux navigateurs (comme Chrome, Firefox, Safari, etc.) ont choisi de n'implémenter le support de HTTP/2 que sur des connexions sécurisées (HTTPS). Cette décision a été prise pour plusieurs raisons, notamment pour encourager une adoption plus large de HTTPS et pour éviter certains problèmes potentiels avec les intermédiaires sur le réseau qui ne comprennent pas ou ne s'attendent pas à HTTP/2.

Cela signifie que, bien que techniquement possible, si vous mettez en place un serveur HTTP/2 sans HTTPS, la plupart des navigateurs modernes ne se connecteront pas à lui en utilisant HTTP/2. Ils retomberont à HTTP/1.1.

## 2. Historique

Le HTML a été conçu par Tim Berners-Lee dans le cadre de son projet du World Wide Web qu'il a proposé en 1989. Il avait pour objectif de permettre aux scientifiques du CERN de partager leurs recherches. Robert Cailliau, un Belge, a collaboré avec Tim Berners-Lee pour donner vie au Web. Ensemble, ils ont développé le premier navigateur web, le premier serveur web, et ont mis en ligne le premier site web le 6 août 1991. Ce site est toujours accessible et peut être consulté à l'adresse : <http://info.cern.ch/hypertext/WWW/TheProject.html>.

Le protocole HTTP a été introduit en même temps que le World Wide Web. Il sert de fondement au transfert de données sur le web. Dans sa forme la plus simple, le protocole HTTP est composé d'un client qui envoie une requête à un serveur, qui à son tour renvoie une réponse. HTTP est un protocole sans état, signifiant qu'il ne conserve pas d'informations sur les requêtes précédentes.

Un "protocole", dans le contexte des réseaux et de l'informatique, est un ensemble de règles et de conventions que les ordinateurs suivent pour communiquer entre eux. C'est comme une langue ou un code que les machines utilisent pour s'assurer qu'elles se comprennent

mutuellement lorsqu'elles échangent des informations.

Prenant l'exemple du protocole HTTP: c'est un ensemble spécifique de règles que les navigateurs web et les serveurs suivent pour échanger des informations, principalement des pages web.

Je vous invite à visionner la vidéo [La méconnue naissance du Web](#) sur Auvio pour en apprendre davantage. (Notez qu'un compte Auvio est nécessaire pour y accéder).

De plus, voici l'[audio](#) du livre [Alexandria: Les pionniers oubliés du web](#). Je vous recommande grandement ce livre. Il est très intéressant et vous permettra de mieux comprendre l'histoire du web.

## 3. Schéma de la communication entre deux personnes

### 3.1 Exemple 1: un client et une serveuse

Dans un restaurant, nous avons un client demande à la serveuse: **"Puis-je avoir une pizza 4 fromages svp ?"**.

La serveuse répond: **"Oui, bien sûr"**.

La serveuse va ensuite transmettre la commande au cuisinier. Le cuisinier va préparer la pizza.

La serveuse va ensuite apporter la pizza au client. Le client va manger la pizza.

Représentons le schéma de la communication entre la serveuse et le client:

Émetteur	Récepteur	Message
1. CLIENT	SERVEUSE	<i>Puis-je avoir une pizza 4 fromages ?</i>
2. SERVEUSE	CLIENT	<i>Oui, bien sûr !</i>

Version avec un [schéma graphique](#) (non accessible).

Dans la communication entre le client et la serveuse, les deux personnes vont devoir utiliser un langage commun: **un protocole de communication**. Le langage commun est le français. Le client et la serveuse vont devoir utiliser la même langage pour pouvoir communiquer. Une fois la

même langue utilisée, le client va pouvoir demander à la serveuse une pizza. La serveuse va pouvoir répondre au client qu'elle va lui apporter une pizza. Le client va pouvoir manger la pizza. Grâce à cette langue commune, ils ont pu échanger des messages.

La communication entre le client et la serveuse est **bidirectionnelle**. Cela signifie que le client et la serveuse peuvent envoyer et recevoir des messages.

Un message utilise un canal de communication. Dans notre exemple, le canal de communication est l'air. Le client et la serveuse vont utiliser l'air pour envoyer et recevoir des messages. Le canal de communication peut être un fil, l'air, un câble, etc.

## 3.2 Exemple 2: une serveuse et un cuisinier

La serveuse va ensuite transmettre la commande au cuisinier: "**Une pizza 4 fromages pour la table 4 !**"

Le cuisinier va préparer la pizza.

Quand la pizza sera préparée, le cuisinier fera sonner une cloche et pourrait dire: "**La pizza 4 fromages de la table 4 est prête !**"

La serveuse va ensuite apporter la pizza au client et lui dire: "**Voici votre pizza, bon appétit !**"  
Le client mange ensuite la pizza.

Représentons le schéma de la communication entre la serveuse et le client:

Émetteur	Récepteur	Message
1. SERVEUSE	CUISINIER	<i>Une pizza 4 fromages pour la table 4 !</i>
2. CUISINIER	SERVEUSE	<i>La pizza 4 fromages de la table 4 est prête !</i>

Version avec un [schéma graphique](#) (non accessible).

## 3.3 Exemple 3: un client, une serveuse et un cuisinier

Nous allons reprendre les deux exemples précédents et les mettre ensemble.

Le client va demander à la serveuse: "**Puis-je avoir une pizza 4 fromages svp ?**". La serveuse va répondre: "**Oui, bien sûr**". La serveuse va ensuite transmettre la commande au cuisinier: "**Une pizza 4 fromages pour la table 4 !**" Le cuisinier va préparer la pizza. Quand la

pizza sera préparée, le cuisinier fera sonner une cloche et pourrais dire: "**La pizza 4 fromages de la table 4 est prête !**" La serveuse va ensuite apporter la pizza au client et lui dire: "**Voici votre pizza, bon appétit !**" Le client mange ensuite la pizza.

Représentons le schéma de la communication entre le client, la serveuse et le cuisinier:

Émetteur	Récepteur	Message
1. CLIENT	SERVEUR	<i>Puis-je avoir une pizza 4 fromages ?</i>
2. SERVEUSE	CLIENT	<i>Oui, bien sûr !</i>
3. SERVEUSE	CUISINIER	<i>Une pizza 4 fromages pour la table 4 !</i>
4. CUISINIER	SERVEUSE	<i>La pizza 4 fromages de la table 4 est prête !</i>
5. SERVEUSE	CLIENT	<i>Voici votre pizza, bon appétit !</i>

Version avec un [schéma graphique](#) (non accessible).

## 4. Schéma de communication entre un navigateur web et un serveur web

Nous pourrions comparer la serveuse à un navigateur web, le cuisinier à un serveur web, le client à un utilisateur et la pizza à une page web.

Le navigateur va initier la communication en demandant au serveur web d'envoyer réponse: une page web. Le serveur web va ensuite envoyer la page web au navigateur web. Le navigateur web va ensuite afficher la page web.

Représentons le schéma de la communication entre le navigateur web et le serveur web:

Émetteur	Récepteur	Message
1. NAVIGATEUR	SERVEUR	<i>Donne-moi la page web suivante</i>
2. SERVEUR	NAVIGATEUR	<i>Voici la page web</i>

Version avec un [schéma graphique](#) (non accessible).

Ici, le schéma a été réduit à sa plus simple expression. En réalité, la communication entre le navigateur et le serveur est plus complexe car le protocole http inclu des informations supplémentaires. Par exemple, le navigateur va envoyer au serveur des informations sur le navigateur, le système d'exploitation, etc. Le serveur va ensuite envoyer une réponse au navigateur avec des informations sur le serveur, le type de contenu, etc. Le navigateur va ensuite afficher la réponse.

Dans une communication de type client/serveur le canal de communication est un réseau local ou sur Internet. Le client et le serveur vont utiliser le réseau pour envoyer et recevoir des messages. Le canal de communication peut être un fil, l'air, un câble, etc.

## Le HTML

Le HTML (**H**ypertext **M**arkup **L**anguage) est un langage de balisage qui permet de structurer le contenu d'une page web. Utilisé en combinaison avec le CSS, qui gère la mise en forme, et le JavaScript, qui ajoute de l'interactivité, le HTML forme la base de la plupart des sites web que nous visitons.

Le HTML est un langage de balisage, ce qui signifie qu'il utilise des balises pour décrire le contenu d'une page web. Une balise est un élément qui permet de structurer un document. Une balise est composée d'un nom et d'un contenu. Le nom d'une balise est entouré de chevrons: symbole plus petit < et symbole plus grand >. Le contenu d'une balise est placé entre la balise ouvrante et la balise fermante. Une balise peut être vide, c'est-à-dire qu'elle n'a pas de contenu.

Mais je vous en dirai plus sur le HTML dans le cours...

## Les versions du HTML

Le HTML, ou HyperText Markup Language, est le langage standard de description de pages web. Depuis sa création, il a connu de nombreuses évolutions qui ont reflété les besoins croissants et changeants du web. Voici un bref aperçu, qu'il ne faut pas retenir, des versions majeures du HTML et de leurs innovations principales :

- **HTML 1.0** (1991) :
  - Description : Il s'agit de la première version officielle du HTML, conçue principalement pour créer des documents basiques avec des liens et du

formatage simple.

- Innovations : Introduction des éléments fondamentaux comme les liens hypertextes.
- **HTML 2.0** (1995) :
  - Description : Cette version a établi un standard plus formel pour le HTML après que différentes versions du langage ont été utilisées par différents navigateurs.
  - Innovations : Introduction des formulaires pour la collecte de données sur le web.
- **HTML 3.2** (1997) :
  - Description : Il s'agit d'une expansion du HTML pour inclure des fonctionnalités comme les scripts et les applets.
  - Innovations : Support pour les scripts (comme JavaScript) et introduction de tables pour la mise en forme.
- **HTML 4.01** (1999) :
  - Description : Cette version a apporté une séparation plus nette entre le contenu et la présentation, encourageant l'utilisation de CSS pour le design.
  - Innovations : Meilleur support pour les feuilles de style en cascade (CSS), améliorations en matière d'accessibilité et introduction des entités de caractères UTF-8.
- **XHTML 1.0** (2000) :
  - Description : En combinant HTML avec XML, cette version visait à créer un langage plus strict et plus facile à analyser.
  - Innovations : Règles syntaxiques strictes pour une meilleure interopérabilité et compatibilité.
- **HTML5** (2014) :
  - Description : Probablement la mise à jour la plus importante à ce jour, elle a introduit une multitude de fonctionnalités pour faire du web une plateforme de développement complète.
  - Innovations : Éléments pour la vidéo et l'audio, support pour les applications web hors ligne, meilleures sémantiques avec des éléments comme <article>, <section>, <nav>, et <header>.

Au fil du temps, le HTML a évolué pour répondre aux besoins d'un web en constante évolution. Des simples documents avec des liens hypertextes à de véritables applications web riches et



interactives, la progression du HTML reflète la croissance et la maturation du World Wide Web lui-même.

Toutes ces versions (sauf la 1.0) ont été validées par le W3C. Le HTML5 a d'abord été développé par le WHATWG puis validé par le W3C. Le W3C a ensuite créé le groupe de travail HTML pour continuer à développer le HTML5. Le HTML5 est maintenant terminé. Le W3C travaille maintenant sur le HTML6.

## **5. Deux organismes de standardisation du Web**

### **5.1 W3C (World Wide Web Consortium)**

#### **Définition**

Le W3C est un organisme de standardisation à but non lucratif qui a pour mission de développer des standards pour le web.

Le W3C a été fondé par Tim Berners-Lee en 1994. Le W3C est composé de membres qui sont des entreprises, des organisations et des universités. Le W3C est composé de 400 membres. Le W3C est basé à Boston aux Etats-Unis. Le W3C a des bureaux dans 10 pays dont la France.

#### **Rôle**

Le rôle principal du W3C est d'établir des standards ouverts pour le Web afin d'assurer sa croissance et sa stabilité à long terme. Il développe des recommandations techniques, des directives et des outils pour créer et interpréter le contenu Web. Historiquement, il a été responsable de la standardisation de nombreuses technologies Web clés, telles que HTML, CSS et XML.

### **5.2 Le WHATWG**

#### **Définition**

Le WHATWG est un groupe de travail formé en 2004 par des représentants d'Apple, Mozilla, et Opera. Il est né en réponse à la perception que le W3C n'évoluait pas suffisamment vite dans le développement des standards du Web, en particulier pour les besoins des applications Web modernes.

## Rôle

Le WHATWG se concentre sur le développement et la maintenance de standards vivants pour des technologies Web clés, en particulier HTML. Contrairement à l'approche traditionnelle basée sur des versions, les standards vivants sont constamment mis à jour (Living Standard) pour refléter les besoins actuels et futurs du Web. Cela permet une adaptation plus rapide aux innovations technologiques et aux besoins des utilisateurs.

## 6. Les navigateurs

Les navigateurs sont des logiciels qui permettent d'afficher des pages web. Les navigateurs les plus connus sont Google Chrome, Mozilla Firefox, Microsoft Edge et Safari.

N'utilisez plus Internet Explorer car il n'est plus maintenu par Microsoft. Il est remplacé par Microsoft Edge. De plus, il n'est pas complètement compatible avec les dernières versions du HTML, du CSS et du JavaScript.

Pour les développeurs c'était un cauchemar de devoir faire fonctionner leur site web sur Internet Explorer. C'est pour cela que Microsoft a décidé de remplacer Internet Explorer par Microsoft Edge. Microsoft Edge est basé sur Chromium, le moteur de Google Chrome.

## 7. Les serveurs webs

Les serveurs webs sont des logiciels qui permettent d'envoyer des pages web aux navigateurs. Les serveurs webs les plus connus sont Apache, Nginx et IIS. Mais il existe aussi des serveurs webs écrits en JavaScript comme Node.js, etc.

## 8. Les éditeurs de code

Les éditeurs de code sont des logiciels qui permettent d'écrire du code. Les éditeurs de texte les plus connus sont:

- [Visual Studio Code](#) (VS Code) : Développé par Microsoft, c'est l'un des éditeurs de code les plus populaires du moment. Il offre une vaste collection d'extensions qui le rendent adapté au développement web moderne.
- [Sublime Text](#) : Connu pour sa vitesse et son interface, c'est un choix populaire parmi

les développeurs web.

- [Notepad++](#) : Un éditeur de texte avancé basé sur Scintilla, il est léger et puissant.
- [Brackets](#) : Développé par Adobe, cet éditeur open source est spécialement conçu pour le développement web, avec des fonctionnalités comme l'aperçu en direct.
- [WebStorm](#) : Développé par JetBrains, c'est un IDE spécifiquement pour le développement web, y compris JavaScript, CSS, et HTML.
- [PHP Storm](#) : Développé par JetBrains, c'est un IDE spécifiquement pour le développement PHP.

Pour ce cours, nous allons utiliser PHP Storm car vous l'utiliserez aussi pour le cours de PHP. Autant commencer à l'utiliser dès maintenant.

## 9. Sites de référence

- [MDN Web Docs](#): C'est LE site de référence pour le HTML, CSS, Javascript, etc.
- [MDN Web Docs - HTML](#): Je vous donne le lien direct pour la partie HTML.
- [Can I Use](#): Ce site vous permet de savoir si une fonctionnalité est supportée par les navigateurs. Rappelez-vous que tous les navigateurs ne supportent pas toutes les fonctionnalités. Il est important de vérifier si une fonctionnalité est supportée par les navigateurs avant de l'utiliser.
- [Stack Overflow](#): C'est un site de questions/réponses pour les développeurs. Si vous avez une question, vous pouvez la poser sur Stack Overflow. Si vous avez une réponse à une question, vous pouvez la partager sur Stack Overflow. C'est un site très utile pour les développeurs.
- [W3Schools](#): C'est un site qui propose des tutoriels sur le HTML, CSS, Javascript, etc. C'est un site très utile pour les débutants.
- [CSS Tricks](#): C'est un site qui propose des tutoriels sur le CSS.
- [Smashing Magazine](#): C'est un site qui propose des tutoriels sur le HTML, CSS, Javascript, etc.
- [A List Apart](#): C'est un site qui propose des tutoriels sur le HTML, CSS, Javascript, etc.
- [CSS Zen Garden](#): C'est un site qui propose des exemples de design de sites web. C'est un site très utile pour les débutants.
- [W3C](#): C'est le site officiel du W3C.
- [WHATWG](#): C'est le site officiel du WHATWG.

[← Revenir au sommaire.](#)

---

© 2023 [Johnny Piette](#).



Ce travail est licencié sous [Creative Commons Attribution 4.0 International License](#).

*Vous pouvez copier, modifier, distribuer et représenter ce travail, même à des fins commerciales, à condition de donner le crédit approprié, fournir un lien vers la licence, et indiquer si des modifications ont été effectuées.*