

Comprendre le World Wide Web

Avant de vous lancer dans l'apprentissage des langages de programmation, il est important de comprendre comment le web fonctionne. Nous allons aborder brièvement l'histoire du Web, son fonctionnement et ce qui vous permet de consulter une page Web grâce à vos téléphones, tablettes ou ordinateur.

Table des matières

- [L'histoire du Web en Bref](#)
- [Qu'est ce qu'un client](#)
 - [Développement Front-End](#)
- [Qu'est ce qu'un serveur ?](#)
 - [Développement Back-End](#)
 - [Notion de serveur local et de web hosting](#)
 - [URLs](#)
 - [Serveur distant](#)
 - [Le serveur local](#)
- [Comment transférer vos fichiers locaux vers un hébergeur](#)
- [Quelle est la différence entre un site statique et un site dynamique ?](#)
 - [Site statique](#)
 - [Site dynamique](#)
- [Les outils de développement](#)
- [Conclusion](#)

L'histoire du Web en Bref

En anglais, WEB (aussi appelé World Wide Web) signifie "la Toile", on peut l'imaginer telle une toile d'araignée tissée reliant tous ses utilisateurs par delà les frontières physiques, de langues et de distances.

Ce réseau inventé entre 1989 et 1990 a permis de donner accès à la connaissance et au partage à travers le monde et fait aujourd'hui partie intégrante de notre quotidien.

Imaginez vous une journée sans accès à votre réseau social favoris? Que feriez vous sans Wikipédia ?

Le Web a permis de créer des liens entre les populations, et de rendre l'information disponible à tous. Actualité, livres, vidéos, musique, jeux...

Mais savez vous à qui vous devez cette merveilleuse invention?

Tim Berners-Lee ainsi que Robert Cailliau ont inventé ce merveilleux outils. On peut donc affirmer que la moitié de la paternité du Web revient à la Belgique, en effet Robert Cailliau est un ingénieur Liégeois, plutôt cool n'est ce pas?



[Revenir au top](#)

Qu'est ce qu'un client

Un client ou client HTTP est l'outil permettant l'affichage des fichiers contenus sur un serveur. Ces fichiers comprennent l'entièreté du contenu d'un site web ou d'une application. En d'autres termes il s'agit d'un Browser, en français: votre navigateur favoris -> Chrome, Firefox, Edge, Safari, Opera.... Internet explorer pour les plus aventureux (rip)!

Un terme compliqué pour un outil simple que vous utilisez déjà tout les jours!

Développement Front-End

Lorsque l'on parle d'affichage client, on parle donc de ce que votre navigateur interprète du serveur.

Pour que votre Browser puisse interpréter ces données, il faut qu'elles aient été rendues disponibles et compréhensibles à la lecture de votre navigateur via le développement Client, c'est à dire le développement **Front-End**.

On code en HTML, CSS et Javascript, qui sont les seuls langages compris et interprété par votre navigateur. Le browser n'est donc que le traducteur entre le code fourni par le développeur (d'abord côté serveur, puis côté client) et l'utilisateur. Un navigateur retranscrit le langage "machine" dans un langage compréhensible par l'humain et permet l'affichage visuel du contenu.

Le client permet d'envoyer des demandes (requêtes) d'actions à performer sur les données contenues par le serveur. Une fois ces demandes reçues par le serveur, c'est le serveur qui décide si oui ou non il va performer cette action et de quelle façon.

Par exemple: vous décidez de changer votre nom et prénom sur Facebook. Facebook vous demande de remplir un formulaire et de l'envoyer. Il s'agit d'une action appelée "Submit" ou "Post". Le serveur reçoit cette demande et en fonction des restrictions programmées par les développeurs et de sa configuration, l'action sera exécutée ou non.

Un client peut également fonctionner de manière indépendante, sans l'aide d'un serveur.

Ce type de développement ne fonctionne uniquement que dans le cadre des sites statiques (nous aborderons plus loin la différence entre un site statique et un site dynamique).

Pour rendre disponibles ces fichiers, ils sont simplement hébergés par une société d'hébergement qui se charge de les rendre disponibles sur internet via une URL unique.

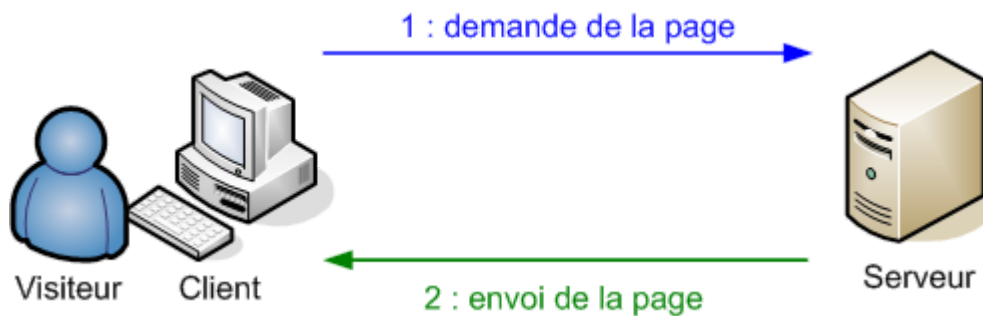


[Revenir au top](#)

Qu'est ce qu'un serveur ?

Nous venons de voir l'utilité d'un client, nous avons abordé la notion de serveur, mais qu'est ce qu'un serveur?

Un serveur HTTP (ou Webservice) est un logiciel utilisé pour stocker, transférer, manipuler, des données (fichiers) à travers le protocole HTTP. Il est le contenant et l'opérateur de toutes les informations d'un site web, ou d'une application (web / mobile / desktop).



Source: Schéma provenant d'OpenClassRoom

Nous avons évoqué dans le chapitre sur le Client que les requêtes / actions envoyées par l'utilisateur sont **gérées par le serveur**.

Développement Back-End

Le programmeur Back-End gère la programmation du serveur, il autorise ou non les requêtes et décide de la manière dont celles-ci seront traitées, la façon dont les données seront manipulées puis retournées au client, il est l'opérateur, c'est lui le cerveau derrière chaque application ou site dynamique. On appelle ce type de programmation "la logique métier".

Par exemple: Lorsque une requête n'abouti à rien de "connu" par le serveur, une erreur 404 est retournée, c'est la fameuse "404 not found" bien connue de tout le monde lorsque vous essayez de contacter une page n'existant pas ou plus.

Retenez simplement que le code fournit par le développeur Back-End contient la logique métier qui permet de performer l'action demandée par une requête (appel) effectuée par le client qui en retour interprétera les données reçues, exécutera des actions sur les données contenues par ces fichiers et que le client enverra ses requêtes et en traduira les réponses (que ce soit l'affichage d'une page ou la modification d'un nom sur facebook) pour l'œil humain via votre navigateur.



[Revenir au top](#)

Notion de serveur local et de web hosting

Comme nous venons de le voir le serveur est un contenant des fichiers programmés et écrit par le développeur.

Nous distinguons 2 types de serveur:

- Le serveur local (ou localhost, disponible par défaut dans votre navigateur à l'URL: `http://localhost:8000`).
- Le web hosting ou serveur distant (disponible à l'URL personnalisée du propriétaire de l'application ou site web).



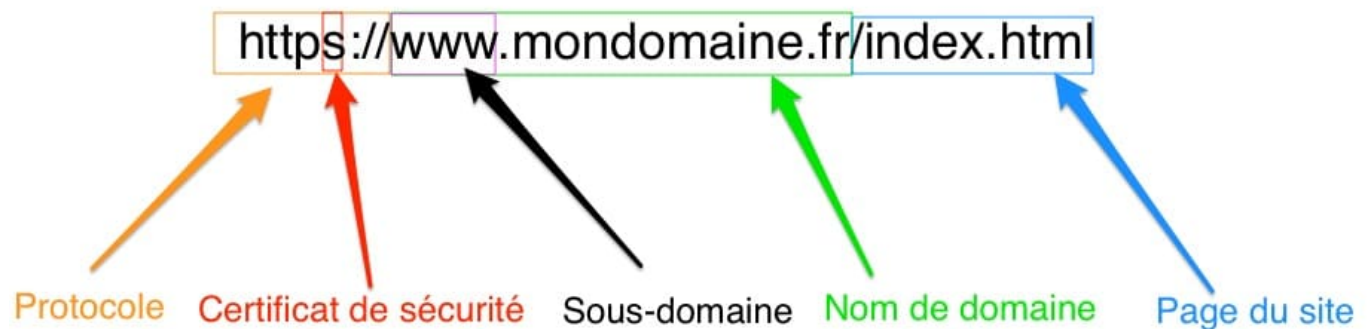
[Revenir au top](#)

URLs

Une URL ou Uniform Resource Locator est l'adresse d'une page, d'un site ou d'une application web. Elle permet de localiser la page dans l'immensité de la toile du web. Une URL est donc personnalisée et unique.

Serveur distant

L'anatomie d'une URL lorsqu'elle est disponible sur un serveur distant (hébergée par un serveur physique: le web hosting) se compose comme suit:



Source: www.mission-internet.fr

- **Le nom du protocole (ou préfixe):** il s'agit du langage utilisé pour communiquer sur le réseau. HTTP est l'acronyme de HyperText Transfer Protocol, il permet de "transférer" les informations des pages Web (HTML, CSS, Javascript) vers le client. Il existe d'autres protocoles mais celui-ci est le plus utilisé.
- **Le certificat de sécurité:** Il s'agit d'une surcouverte ajoutée au protocole assurant la sécurité du site ou de l'application, elle permet par exemple d'effectuer des paiements en ligne, de s'identifier, d'envoyer des informations depuis le client vers le serveur (exemple: lorsque vous envoyez un email, la surcouverte appelée SSL permet l'envoi sécurisé de vos données).
- **Le sous domaine:** Il permet de scinder le site en plusieurs parties et lui dédier des tâches différentes. Il peut se composer comme suit: "www", "ww2" ou "ww3". Il est utilisé pour le référencement SEO, une notion que nous aborderons plus tard dans la formation. Pour l'instant retenez simplement que "www" correspond au sous-domaine principal, celui que vous rencontrerez par défaut et le plus souvent.
- **Le nom de domaine:** Il s'agit du nom personnalisé et choisi par le propriétaire du site web ou de l'application et sert à l'identifier de manière précise. Il permet de contacter le serveur se rapportant au nom de domaine. Il est également possible de contacter le serveur via son adresse IP, ce qui rend la lecture de l'URL plus compliquée. Par exemple: vous pouvez vous rendre sur <http://google.com> à l'aide de son adresse IP physique (ou DNS): <http://74.125.19.147>
- **L'extension:** Permet d'aller plus loin dans la personnalisation de l'URL, on peut la désigner soit sous l'acronyme d'un pays (ex: .be), mais également sous la forme d'une activité (ex: .org pour organisation)
- **Les routes:** Une URL peut également pointer vers un nom d'une page en particulier.

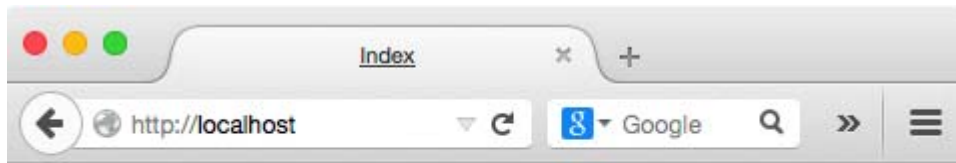
 [Revenir au top](#)

Le serveur local

Par serveur local comprenez "hébergeur local" il ne s'agit pas du serveur "code métier" mais uniquement d'un outil permettant d'héberger localement sur votre ordinateur votre site statique ou dynamique. Lorsque nos fichiers sont hébergés en "local", c'est à dire non pas sur le web mais uniquement sur votre ordinateur,

l'intérêt se situe lors de la conception et de la maintenance de votre site ou de votre application, le serveur local permet de tester l'affichage et le bon fonctionnement de votre code.

Dans ce cadre l'URL peut se représenter comme suit:



It works!

Source: MDN

Vous l'aurez compris le préfixe, le nom de domaine et l'extension sont donc remplacé par le localhost.

Les routes et paramètres sont toujours existant et permettent de vérifier le bon fonctionnement de votre code.

Il peut exister un serveur local dédié au *développement client* (front-end) et un serveur local dédié au *développement serveur* (back-end). Pour permettre à l'ordinateur de lancer le développement de ces deux parties simultanément vous aurez besoin de configurer un port spécifique pour chacune des parties.

[↑ Revenir au top](#)

Comment transférer vos fichiers locaux vers un hébergeur

Pour ce faire il faut donc se procurer un hébergeur. Ceux-ci peuvent être payant ou gratuit (souvent avec pubs ou restrictions). Ensuite, à l'aide d'un client FTP, vous pourrez envoyer vos fichiers sur le serveur de votre hébergeur et accéder à votre site en ligne.

Lors de cette formation, nous n'aurons probablement pas le temps d'aborder ce processus.

[↑ Revenir au top](#)

Quelle est la différence entre un site statique et un site dynamique ?

Site statique

Contrairement aux idées reçues un site statique n'est pas un site dépourvu d'animations.

Lorsque nous parlons de site statique nous parlons de **"pages statiques" visibles telles qu'elles ont été conçues et ne pouvant recevoir de modification extérieur.**

Pour être modifiées ces informations doivent d'abord repasser dans les mains du développeur pour ensuite être codées, uploadées et déployées.

Si non, elles resteront toujours identiques à ce qui a été développé à sa création.

Un site statique ne permet **aucune interaction** avec l'utilisateur et donc aucune modification ou ajout de données de sa part.

 [Revenir au top](#)

Site dynamique

*Contrairement au site statique, un site dynamique **peut être modifié par son utilisateur.***

Reprenons comme exemple facebook, vous pouvez ajouter des publications, likez celles de vos amis, changer de photo de profil, etc.

Dans le panel des sites ou application dynamiques nous pouvons retrouver: réseaux sociaux, blogs, plateforme de téléchargements ou de streaming telles que Netflix, application de gestion de temps (To-do liste), etc...

Pour ce faire l'utilisateur, comme nous l'avons vu précédemment dans les chapitre Client/Serveur, va devoir envoyer une requête au serveur, ces requêtes peuvent uploader un nouveau contenu, récupérer un contenu, le modifier, ou le supprimer. Le serveur traite la demande et effectue l'opération pour ensuite recevoir une réponse contenant le contenu demandé, modifié, ajouté, ou supprimé.

 [Revenir au top](#)

Les outils de développement

Afin de faciliter le développement de vos applications, les différents navigateurs ont développé leur propre outils de développement, disponibles en pressant la touche "inspecteur" F12 de votre clavier.

Je vous renvoie à [cet article agrémenté de vidéos de KhanAcademy](#) reprenant ce sujet, veuillez à le lire et à regarder attentivement les vidéos, faites en faire bon usage!

 [Revenir au top](#)

Conclusion

Nous voici à la fin de cette introduction au vaste monde du Web, comme vous l'aurez compris, ce ne sont que des notions élémentaires que nous viendrons agréments tout au long de votre formation. Pas de panique, tout ceci semble difficile à appréhender au début, mais une fois les mains dans le cambouis je n'ai aucun doute quant à vos capacités d'apprentissage. N'hésitez pas à revenir sur cette théorie dès qu'il s'avérera nécessaire. Petit à petit vous vous rendrez compte du chemin parcouru depuis cette introduction et de vos nouveaux talents de codeur en herbe.

Bon travail à tous!

 [Revenir au top](#)

 [Retour au sommaire du cours](#)

Cours original: Julie Vanderbyse

Modification: Jeremy Scala