Formation web IM@GE 2019-2020

# Introduction au web

[Introduction au web 1](#_Toc14804940)

[Fonctionnement d’un site internet 2](#_Toc14804941)

[Principe client/serveur 2](#_Toc14804942)

[Principe de Front End et Back End 2](#_Toc14804943)

[Les gestionnaires de code source 3](#_Toc14804944)

[Le front-end 5](#_Toc14804945)

[HTML/CSS et rôle du navigateur 5](#_Toc14804946)

[HTML 5](#_Toc14804947)

## Fonctionnement d’un site internet

### Principe client/serveur

Tout le monde a déjà utilisé un site internet, mais derrière l’utilisation facile et intuitive d’une page web, comment tout cela fonctionne-t-il ?

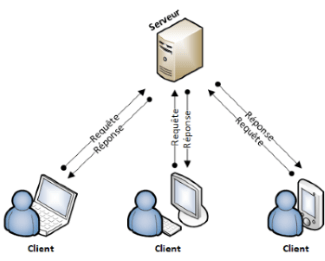
Pour répondre à cette question il faut comprendre le principe de client/serveur.

Lorsque vous consultez un site avec votre machine (ordinateur, téléphone, tablette, etc), vous accéder à des fichiers qui ne sont pas sur votre ordinateur mais sur un serveur qui est bien souvent à l’autre bout du monde. Ces fichiers contiennent le code qui permet de faire fonctionner le site tel que vous le voyez, lui donner son aspect et gérer toutes ses fonctionnalités. Le serveur contient également la base de données.

Mais c’est quoi un serveur au juste ?

Pour faire simple, c’est un ordinateur très puissant, qui est allumé continuellement et qui stocke des ressources accessibles sur internet (les sites internet entre autres).

On dit que votre machine est le **client**, et que le site internet est placé côté **serveur**. Ainsi, le client et le serveur vont communiquer pour réaliser des actions sur votre écran (affichage d’une page stockée sur le serveur, chargement de données stockées sur le serveur, etc). Pour réaliser cette communication, il faut qu’ils se comprennent, qu’ils parlent la même « langue ». Cette langue, c’est ce qu’on appelle le « protocole réseau ».



Vous trouverez des explications beaucoup plus détaillées au sujet des protocoles de communication dans le cours de réseau et cybersécurité, ce n’est pas l’objet de ce cours.

### Principe de Front End et Back End

Maintenant que nous savons que notre machine communique avec un serveur, nous aimerions comprendre quels sont ces fameux fichiers qui sont stockés là-bas. Un peu de patience nous y venons !

D’abord, sachez qu’il existe de nombreux métiers qui gravitent autour de la création d’un site internet, en voici quelques-uns :

* Développeur front-end
* UI designer
* UX designer
* Web designer
* Développeur back-end
* Architecte réseau
* Administrateur de bases de données
* Développeur full-stack
* …

Ces métiers se comptent par dizaines, certains sont spécialisés dans un domaine précis, d’autres sont plus polyvalents.

Très bien mais quel est le rapport avec tout ce que l’on a pu dire avant ? Et bien il faut pouvoir faire travailler tout ce beau monde en même temps !

Pour cela, on va diviser la création de notre site en deux catégories : le **front-end** et le **back-end**.

Le développement front-end concerne tout ce qui attrait à l’apparence, au design de votre site internet ! C’est donc tout ce qui sera affiché à l’écran. Si votre site n’est pas joli, il faudra taper sur les doigts du développeur front-end et des designers !

Si l’on prend l’exemple d’une maison, le front-end, c’est le décorateur, le maçon, le charpentier et le jardinier qui s’en occupent !

Le back-end est la partie complémentaire au front-end, il s’agit de la partie immergée de l’iceberg ! C’est tout ce que l’on ne voit pas au premier coup d’œil mais qui est néanmoins essentiel au fonctionnement de votre site*.*

En effet, comment afficher des données si ces dernières sont sur le serveur ?

Il faut les charger et c’est le rôle du back-end. Le back-end permet aussi entre autres de réaliser des connexions à un compte utilisateur, de modifier la base de données, etc.

Si on reprend l’analogie de notre maison, le back-end c’est le plombier qui raccorde l’eau au bâtiment ou l’électricien qui réalise le câblage électrique.

Cette division du code en deux parties permet de bien séparer le visuel du fonctionnel. Ainsi, si l’on veut faire une refonte graphique de notre site, il suffira de modifier le front-end sans que le back-end n’en soit altéré.

Remarque : Si vous entendez parler de « full stack », ce terme regroupe le front-end et le back-end (ex : développeur full stack).

### Les gestionnaires de code source

Cette séparation du code c’est pratique pour structurer notre projet et cela facilite un peu le travail en équipe. Mais en pratique, ce sont souvent des équipes de dizaines d’employés qui travaillent sur un même projet. Si tout le monde modifie les mêmes fichiers en même temps, cela créé des conflits de fichier et tout peut-être endommagé.

Comment faire alors pour faire travailler simultanément plusieurs personnes sur un même projet ?

C’est là qu’interviennent les gestionnaires de code source. Cette partie du cours ne concerne pas à proprement parlé le développement web mais sera utile quel que soit le projet informatique que vous comptez faire. Mieux vaut prendre les bonnes habitudes dès le début !

Dans ce cas, qu’est-ce qu’un gestionnaire de code source ?

**PARTIE DU COURS SUR GIT A FAIRE VENDREDI**

## Le front-end

### HTML/CSS et rôle du navigateur

Le front-end, rappelez-vous, est la mise en forme de notre site. Pour cela, on dispose de deux principaux langages informatique, le HTML et le CSS.

Le HTML (*HyperText Markup Language*), c’est le langage qui donne sens à votre page. C’est lui qui définit les éléments de la page, qui dit que tel élément est un titre, tel autre est un paragraphe, que telle partie est un entête ou tel autre un pied de page, etc. Le HTML est interprété par le navigateur web avant d’afficher votre page. Si votre code HTML est bien réalisé, le navigateur va facilement comprendre comment votre code est structuré et le référencement de votre site internet sera amélioré. Le référencement, vous savez, c’est le fait que votre site internet soit dans les premiers résultats lors d’une recherche dans un moteur de recherche ou qu’au contraire il soit plusieurs pages plus loin. N’oubliez jamais que le référencement c’est très important, on fait un site internet pour qu’il soit consulté ! De plus, si votre code HTML est propre, il vous sera plus facile de le modifier par la suite.

Le CSS (*Cascading Style Sheets*) quant à lui est le langage qui permet de faire la mise en page. Ainsi, nous allons pouvoir désigner des éléments du code HTML et modifier leur positionnement, leur taille leur couleur et bien d’autres ! Depuis quelques années, le CSS permet d’adapter la mise en page de notre site en fonction de la taille d’écran. Il s’agit d’un nouvel enjeu devenu très important avec l’essor des tablettes et des téléphones portables. Le fait qu’un site s’adapte aux différentes tailles d’écran est désigné par le terme de site **responsive** ou **implémentant du responsive design**.

Cependant, ce code HTML et CSS n’est pas suffisant tel quel pour pouvoir visualiser un site internet. Pour ce faire, ils doivent être interprétés par un navigateur web. Des navigateurs web, il en existe une multitude dont en voici les principaux : Internet Explorer, Mozilla Firefox, Google Chrome, Safari, Microsoft Edge ou encore Opéra. Les fonctionnalités de ces derniers sont très similaires, cependant on s’assurera que le rendu de nos pages web est correct sur plusieurs navigateurs car on peut parfois avoir de mauvaises surprises.

**SCHEMA ROLE DU NAVIGATEUR WEB**

### HTML

Un langage à balise

Le HTML est un langage à balise, la plupart de ses balises s’utilisent avec une balise ouvrante, un contenu puis une balise fermante :

<balise ouvrante>Contenu de la balise<balise fermante>

Exemple : Le paragraphe

<p>Ceci est un paragraphe</p>

Certaines balises sont auto-fermantes, aussi appelées balises orphelines, elles n’ont pas de contenu :

**<balise />**

Exemple : Le retour à la ligne

**<br />**

Ensuite, les balises ont des attributs qui permettent de donner des informations à leur sujet.

<balise ouvrante attribut1="valeur1", attribut2="valeur2">Contenu de la balise<balise fermante>

Exemple : Une image se nommant *mon\_image.jpg*

**<img src=**"mon\_image.jpg" />

#### Structure générale d’une page HTML

Avant toute chose, sachez qu’une page HTML a toujours la même structure de base

**<!DOCTYPE html>**

**<html>**

**<head>**

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

**<title>Titre</title>**

**</head>**

**<body>**

**</body>**

**</html>**

Ne vous inquiétez pas, ça à l’air compliqué comme ça mais il n’y a rien de difficile. Avec le temps, écrire ce code de base devient même un automatisme !

Commençons par la première ligne :

**<!DOCTYPE html>**

Rien de plus simple, cela indique au navigateur web que le document que nous sommes en train d’écrire est de type HTML.

Ensuite :

**<html></html>**

Cela délimite le début et la fin de notre code HTML.