Formation web IM@GE 2019-2020

Introduction au web

[Fonctionnement d’un site internet 2](#_Toc15048934)

[Principe client/serveur 2](#_Toc15048935)

[Principe de Front End et Back End 2](#_Toc15048936)

[Les gestionnaires de code source 3](#_Toc15048937)

[Le front-end 5](#_Toc15048938)

[HTML/CSS et rôle du navigateur 5](#_Toc15048939)

[HTML 5](#_Toc15048940)

[*Un langage à balise* 5](#_Toc15048941)

[Structure générale d’une page HTML 6](#_Toc15048942)

[Les commentaires 7](#_Toc15048943)

[Manipulation de texte 7](#_Toc15048944)

[Les titres 7](#_Toc15048945)

[Les paragraphes 8](#_Toc15048946)

[Souligner l’importance d’un mot ou d’une expression 9](#_Toc15048947)

[Les liens 10](#_Toc15048948)

# 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 ?

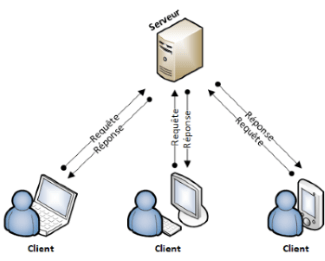
Pour répondre à cette question il faut comprendre le principe de client/serveur.

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 version. 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 version ?

**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 !

La première ligne : <!DOCTYPE html>, indique au navigateur web que le document que nous sommes en train d’écrire est de type HTML. Facile non ?

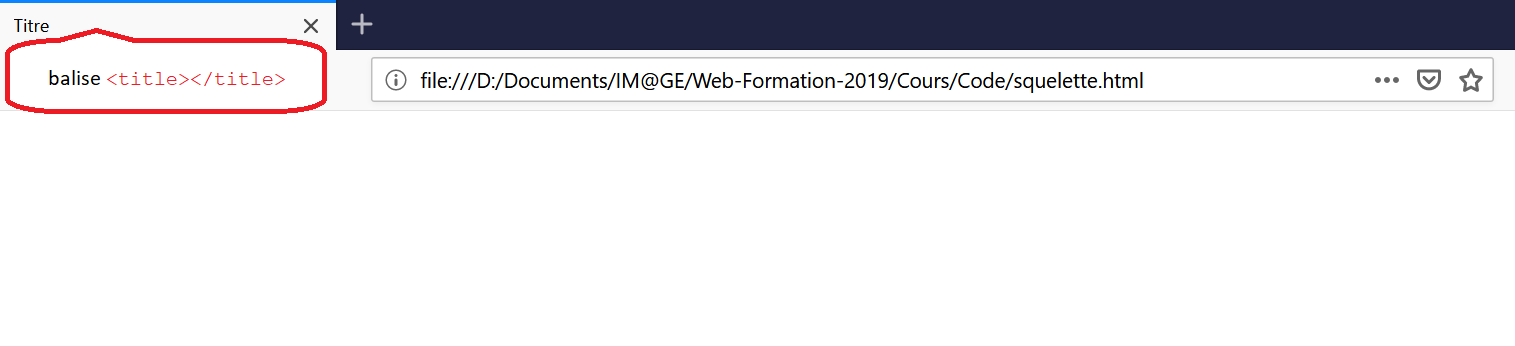
Ensuite les balises <html></html> délimitent le début et la fin du code html.

Au sein de ces deux balises on a deux autres balises, la première est la suivante : <head></head>.

Il s’agit de l’entête du code html. L’entête contient une ou plusieurs balises <meta /> et une balise <title></title>.   
La balise <title></title> c’est le titre de votre page qui s’affiche au niveau de l’onglet dans le navigateur web. Les balises <meta /> quant à elles contiennent des métadonnées, il s’agit en fait de données utilisées par les navigateurs et les moteurs de recherches. Certaines sont utiles pour le référencement et d’autres sont devenues obsolètes. Celle que l’on mettra dans tous les cas est : <meta charset="utf-8" />. Cela permet de donner le type d’encodage des caractères dans notre fichier html, le navigateur pourra alors reconnaître les caractères un peu spéciaux tels que les accents et les apostrophes par exemple.

Enfin viennent les balises <body></body>, elles délimitent le corps du code html. C’est à l’intérieur de ces balises que l’on va pouvoir définir les éléments de notre page.

On obtient alors le résultat suivant :



### Les commentaires

En html, on peut écrire des **commentaires** comme suit : <!-- Contenu du commentaire --> .

Les commentaires sont souvent négligés, quel que soit le langage utilisé, que ce soit par manque de temps ou par flemme Emoji Riez dentés. Cependant, les commentaires permettent d’expliquer certains bouts de votre code, ils sont utiles pour expliciter certains passages un peu compliqués à comprendre à première vue et à faciliter la lecture et la compréhension par un tiers ou par vous-même. Commentez votre code ! Vous me remercierez quand vous le relirez dans deux ans.

On est d’accord, le commentaire en html n’est pas forcément primordial car le langage est très intuitif et proche du langage naturel. Pour d’autres langages il est bien plus important alors prenons les bonnes habitudes dès le départ !

### Manipulation de texte

#### Les titres

Bien ! Nous pouvons maintenant entrer dans le vif du sujet ! Nous allons enfin pouvoir commencer à écrire notre code html à proprement parler. Souvenez-vous, c’est au sein des balises <html></html> que l’on va écrire ce dernier.

Lorsque nous commençons notre page web, la première chose à faire est souvent d’écrire un titre (pas le titre dans l’onglet hein ne confondons pas avec la balise <title></title> vue précédemment). Cependant, tous les titres de notre page n’auront pas la même importance. En effet, il faut pouvoir différencier nos titres par niveau (titre général, sous-titre, titre encore plus petit, etc). Pour ce faire, html nous laisse six balises à notre disposition qui sont les suivantes :

<h1>Ce titre est méga important !</h1>

<h2>Ce titre est super important !</h2>

<h3>Ce titre est important !</h3>

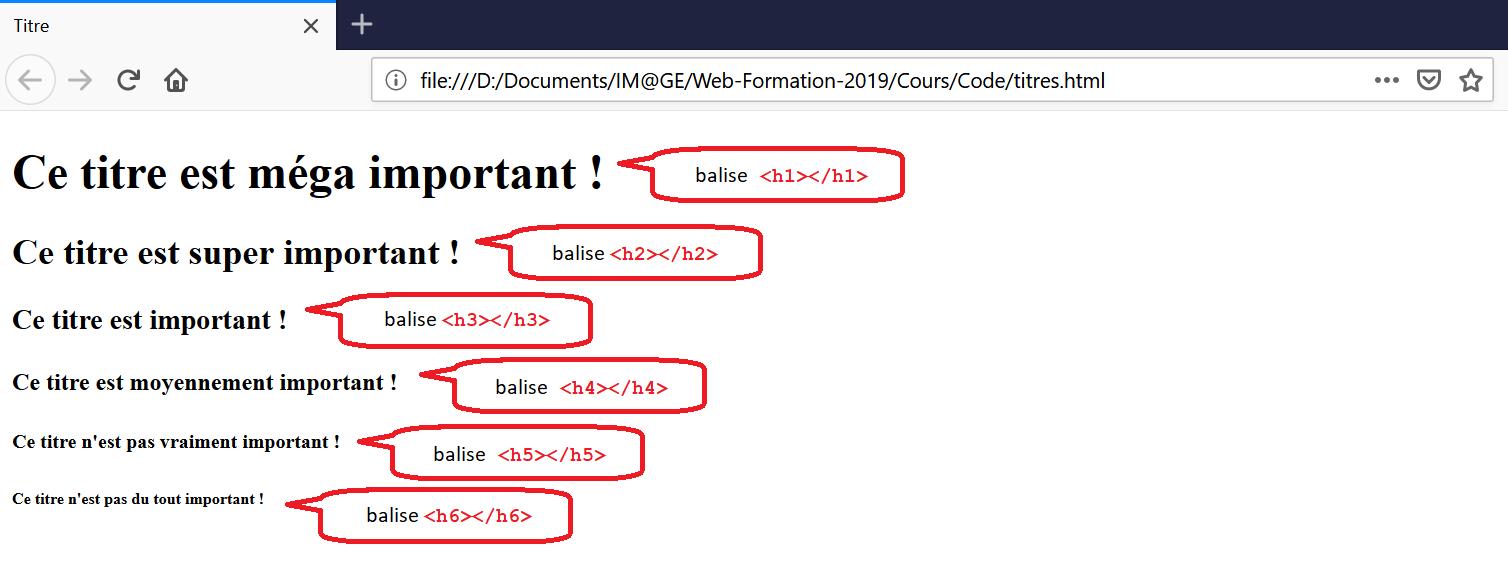
<h4>Ce titre est moyennement important !</h4>

<h5>Ce titre n’est pas vraiment important !</h5>

<h6>Ce titre n’est pas du tout important !</h6>

Le titre <h1> étant le plus important et <h6> le moins important.

Voici par exemple un aperçu des six différents types de titre.



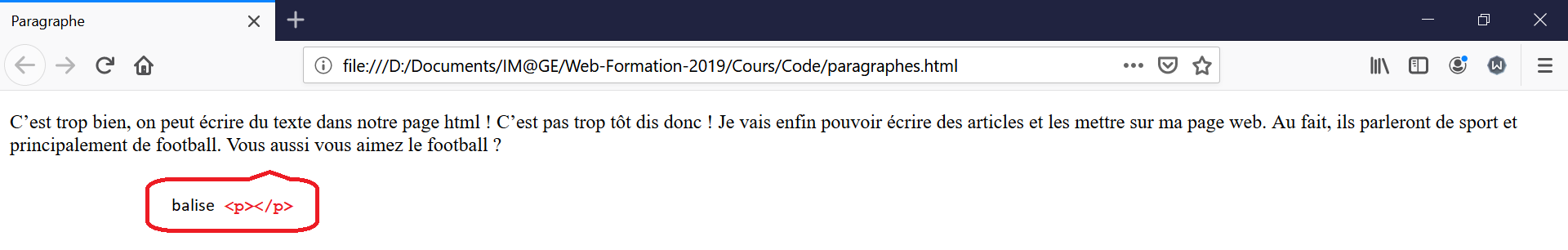
#### Les paragraphes

Mettre des titres c’est bien beau mais on aimerait maintenant écrire un bloc de texte. En html, cela consiste à créer des paragraphes :

<p>C’est trop bien, on peut écrire du texte dans notre page html ! C’est pas trop tôt dis donc !

Je vais enfin pouvoir écrire des articles et les mettre sur ma page web. Au fait, ils parleront de sports et principalement de football. Vous aussi vous aimez le football ?</p>

On obtient alors :



Mince ! J’ai mis un retour à la ligne mais lorsque la page s’affiche à l’écran tout reste sur la même ligne ! Au secouuurs je ne comprends pas !

Pas de panique c’est normal ! En html, un paragraphe est un bloc de texte continu. Appuyer sur la barre « espace » au beau milieu du code n’y changera rien. Mais il y a bien une solution, à vrai dire il y en a même deux. Les deux sont équivalentes, le choix va varier en fonction de la manière dont vous pensez les choses.

En effet, si vous souhaitez retourner à la ligne c’est sûrement parce que votre paragraphe est fini et que vous vous apprêtez tout simplement à en commencer un autre. A ce moment il suffit de créer deux paragraphes au lieu d’un, on a alors un saut de ligne :

<p>C’est trop bien, on peut écrire du texte dans notre page html ! C’est pas trop tôt dis donc !</p>

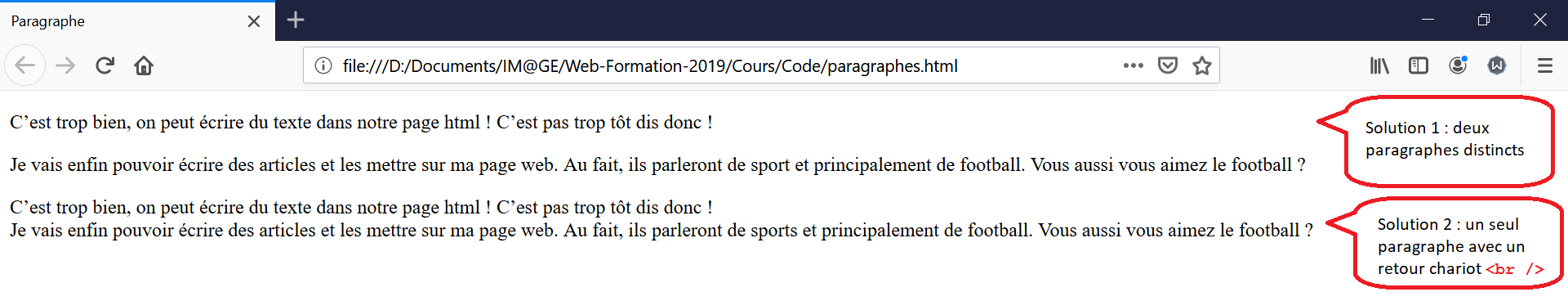
<p>Je vais enfin pouvoir écrire des articles et les mettre sur ma page web. Au fait, ils parleront de sports et principalement de football. Vous aussi vous aimez le football ?</p>

Dans le second cas, on veut rester dans le même paragraphe, il s’agit réellement d’un retour à la ligne. La solution est de mettre la balise qui représente le retour à la ligne en html. Nous l’avions vu en début de cours rappelez-vous, il s’agit de la balise orpheline <br /> :

<p>C’est trop bien, on peut écrire du texte dans notre page html ! C’est pas trop tôt dis donc !<br />

Je vais enfin pouvoir écrire des articles et les mettre sur ma page web. Au fait, ils parleront de sports et principalement de football. Vous aussi vous aimez le football ?</p>

On a donc :



#### Souligner l’importance d’un mot ou d’une expression

Dans notre texte, on va sûrement vouloir accentuer certains mots pour mieux les mettre en valeur car ce sont eux qui dégagent les idées importantes. Prenons pour exemple le premier paragraphe de la page d’Elon Musk sur Wikipedia :

<p>Elon Reeve Musk (prononciation : /ˈiː.lɒn ˈmʌsk/), né le 28 juin 1971 à Pretoria, est un entrepreneur, chef d'entreprise et ingénieur d'origine sud-africaine naturalisé canadien en 1988 puis américain en 2002. Il est actuellement installé à Los Angeles. Il est le PDG et directeur de la technologie de la société SpaceX mais également DG, directeur architecture produit de la société Tesla, et ancien président du conseil d'administration de SolarCity et de Tesla. Il est également le fondateur de The Boring Company, une société de construction de tunnels, et de Neuralink, une société de neurotechnologie.</p>

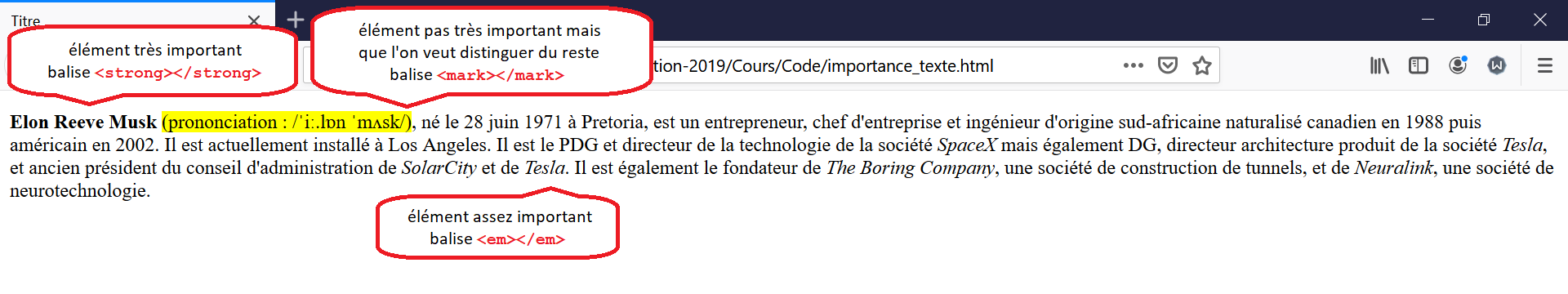
Dans ce paragraphe, ce que l’on trouve important c’est le nom complet d’Elon Musk et les différentes entreprises qu’il a pu diriger.

Il y a trois types de mise en valeur possible qui ont des degrés de force différents.

* On veut signifier qu’une expression est très importante : <strong></strong>
* On veut signifier qu’une expression est assez importante : <em></em>
* L’expression n’est pas spécialement importante, on veut simplement la mettre en valeur pour la distinguer du reste : <mark></mark>

Ici, on va fortement accentuer le nom d’Elon Musk, ensuite on considère que les noms des entreprises sont assez importants. Enfin, la prononciation n’est pas spécialement importante mais on aimerait tout de même la distinguer du reste pour qu’elle soit facilement lisible.

On obtient finalement une page avec les différents mots mis en valeur :



Par défaut, vous pouvez voir que le navigateur mais le contenu des balises <strong></strong> en gras, le contenu des balises <em></em> en italique et enfin surligne en jaune le contenu de <mark></mark>. En effet, le navigateur comprend qu’il s’agit de mot dont on veut accentuer l’impact et change donc l’apparence par défaut. Cependant, n’oublions pas que le html est un langage qui permet de donner la structure de notre page, on utilisera donc ces balises **SEULEMENT** pour accentuer un mot et non pas pour le mettre en gras ou en italique ! La mise en forme est gérée par le css et non par le html. On verra d’ailleurs plus tard qu’il est facilement possible de changer la mise en forme de ces balises en css.

### Les liens

Bon on a maintenant une page avec du texte, c’est cool ! Mais, comme vous le savez, l’intérêt d’internet est de pouvoir naviguer entre des pages.

Oui c’est vrai ça ! Comment lier des pages ?

Pour cela, on va utiliser ce qu’on appelle des liens. Il existe trois types de liens :

* Les liens vers une page d’un autre site
* Les liens vers une autre page de notre propre site, typiquement, on les utilise pour faire des sommaires
* Les liens qui permettent de se déplacer dans la page aussi appelés **ancres**

#### Les liens vers une page d’un autre site

Reprenons notre paragraphe sur Elon Musk. Rendons à César ce qui appartient à César, nous allons donc mettre un lien vers l’article dont nous avons honteusement copié le début.

Un lien est représenté par les balises <a></a>. Nous allons ajouter à la fin de notre paragraphe le code suivant :

<a href=”https://fr.wikipedia.org/wiki/Elon\_Musk”>L’article Wikipédia complet ici</a>

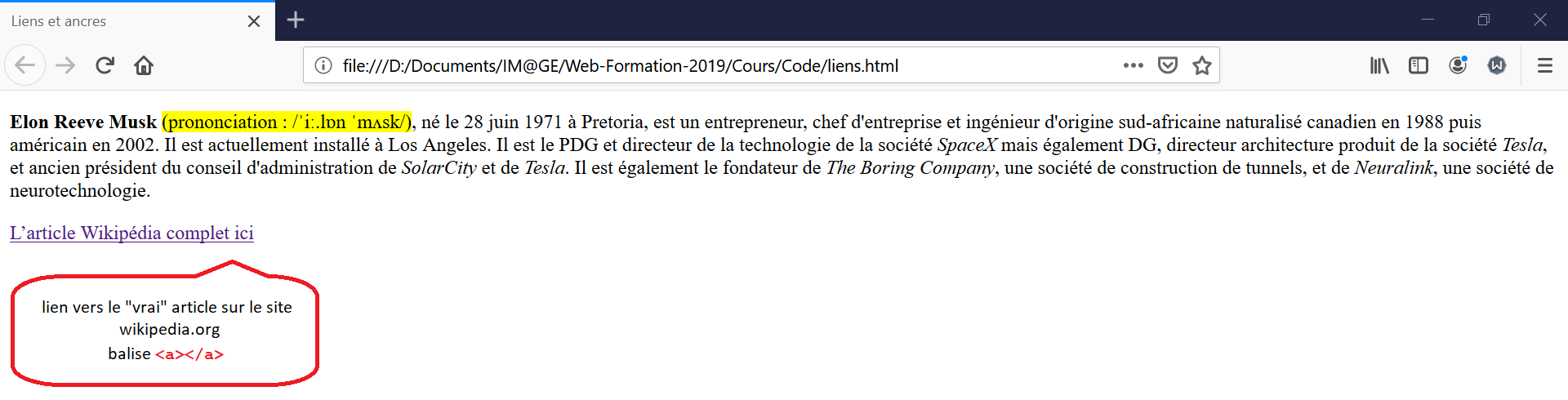
Comme vous pouvez le constater, cette balise a un attribut **href** qui permet de référencer l’URL (l’adresse) de la page sur laquelle on veut aller que l’on appelle **URL destination**.

Un petit mot sur l’URL tout de même : [**https://fr.wikipedia.org/wiki/Elon\_Musk**](https://fr.wikipedia.org/wiki/Elon_Musk). Ce type d’URL est une adresse absolue vers la page destination puisque l’adresse commence à la racine de l’arborescence jusqu’à la fin, chaque fils de l’arbre étant rajouté derrière un « / ».

**FAIRE UN SCHEMA EXPLICATIF**

Ce type de lien utilisant une adresse absolue est appelé **lien absolu**.

On obtient donc un lien qui est souligné est colorié en bleu par défaut pas les navigateurs :



Lorsque l’on clique sur le lien on arrive bien sur l’article en question :

