Chapitre 1

Fonctionnement du WEB

1.1 Introduction au WEB

1.1.1 Un peu d'histoire...

Le « World Wide Web », plus communément appelé « Web », a été développé au CERN (Conseil Européen pour la Recherche Nucléaire) par le Britannique Sir Timothy John Berners-Lee et le Belge Robert Cailliau **au début des années 90**. À cette époque, les principaux centres de recherche mondiaux étaient déjà connectés les uns aux autres grâce à Internet, mais pour faciliter les échanges d'informations, Tim Berners-Lee met au point le **système hypertexte**. Grâce à ce système public accessible via Internet, il est possible, à partir d'un document, de consulter d'autres documents en cliquant sur des mots clés.

Pour accéder au WEB, on utilise un **navigateur WEB** (souvent improprement appelé *navigateur Internet*) permettant d'accéder à des **sites WEB**. Un site WEB est un ensemble de pages reliées entre elles par des hyperliens ou liens hypertexte (simplement appelés aujourd'hui liens).

Question 01 Citer le nom de plusieurs navigateurs WEB.

Question 02 Visiter la « première page WEB historique »à l'adresse suivante :

`http://info.cern.ch/hypertext/WWW/TheProject.html`

Comment reconnait-on les liens hypertexte sur cette page ?

Question 03 Quelle est la différence entre Internet et le WEB ?

1.1.2 Le rôle du navigateur

Techniquement, le WEB repose sur les 3 acronymes suivants :

HTTP **URL** **HTML**

1. **HTTP** (pour *HyperText Transfert Protocol*) : c'est le **protocole** de communication utilisé sur le WEB. Aujourd'hui, on utilise de plus en plus le protocole HTTPS, semblable à HTTP mais sécurisé. Nous étudierons ces deux protocoles en détail dans un prochain chapitre.
2. **URL** (pour *Uniform Ressource Locator*) : c'est ce que l'on appelle couramment l'**adresse WEB**. Elle permet d'identifier une ressource sur le WEB.
3. **HTML** (pour *HyperText Markup Language*) : c'est un **langage de description** conçu pour représenter les pages WEB dans le navigateur.

Pour afficher une page WEB, voici le déroulement des opérations :

1. On fournit au navigateur l'adresse WEB ou URL d'une page WEB. Par exemple :

   `http://info.cern.ch/hypertext/WWW/TheProject.html`

2. Ce dernier récupère, via le protocole HTTP, les données de la page WEB correspondante à l'adresse donnée.
3. Il traite les données HTML reçues et les transforme en une page lisible.

▷ ***Peut-on voir ce mystérieux code HTML interprété par le navigateur ?***

Bien entendu, et comme nous le découvrirons par la suite, il n'est pas si mystérieux que ça. Il suffit pour cela de demander au navigateur d'**afficher le code source** de la page WEB, en faisant un clic-droit sur la page et en sélectionnant l'option idoine.

Question 04 Afficher le code source (code HTML) de la page :

`http://info.cern.ch/hypertext/WWW/TheProject.html`

☞ *Nous reviendrons en détails sur ce code dans la seconde partie de ce cours.*

Question 05 Se rendre à la page Wikipédia de Tim Berners-Lee à l'adresse suivante :

`https://fr.wikipedia.org/wiki/Tim_Berners-Lee`

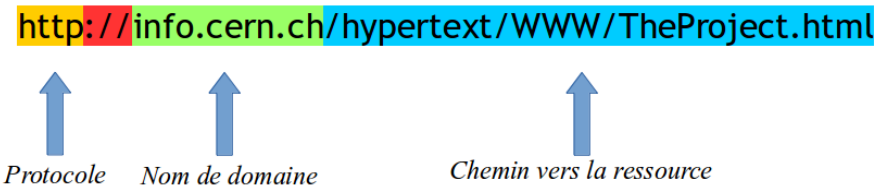
Quelles sont les différences notables entre la page Wikipédia précédente et la « première page WEB » ?
Qu'en est-il du code HTML ?

1.1.3 Précisions sur les URL

Une URL est composée de plusieurs informations qu'il faut savoir reconnaître.

`http://info.cern.ch/hypertext/WWW/TheProject.html`

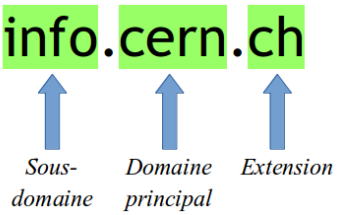
On peut décomposer cette adresse de la manière suivante :



- Le **protocole** utilisé ici est HTTP. Les échanges de données ne se font alors pas de manière sécurisée. Nous n'envoyons ici aucune donnée personnelle, l'utilisation de HTTPS n'est donc pas nécessaire.
- Le **nom de domaine** est l'adresse du serveur à contacter pour accéder aux données.
- Le **chemin vers la ressource** est la localisation de la ressource sur le site WEB.

Noms de domaines

Un **nom de domaine** est une série de caractères sans espaces, toujours associé à une **adresse IP**. La correspondance entre noms de domaine et adresses IP est assurée par le **protocole DNS** que nous avons déjà vu dans la partie Internet.
Lorsqu'on entre un nom de domaine, le navigateur va, de façon transparente, transformer cette adresse en une adresse IP en utilisant le protocole DNS.
Un nom de domaine se décompose de la manière suivante :



- Le **sous-domaine** nous renseigne sur la partie du site WEB visité (ici une partie probablement relative à l'information).
- Le **domaine principal** est... le nom principal du nom de domaine ! Ici, il s'agit du CERN (Organisation européenne pour la recherche nucléaire).
- L'**extension** est associée à un territoire (*.fr* pour la France, *.ch* pour la Suisse...) ou à une activité (*.com* pour une activité commerciale, *.edu* pour l'éducation...).

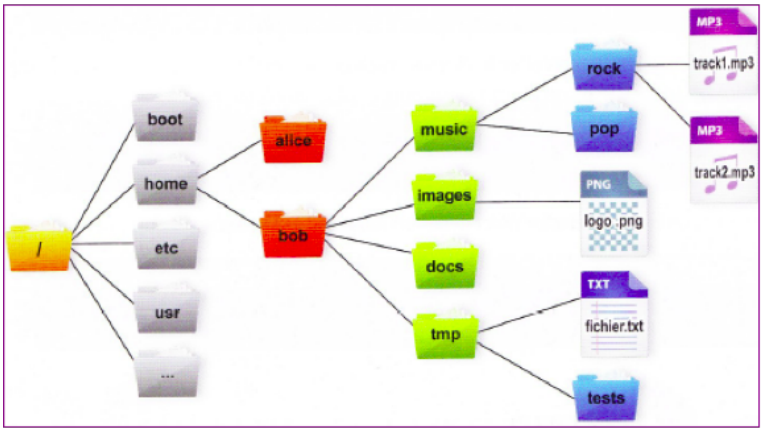
Question 06 Décrire le nom de domaine correspondant à l'adresse suivante :

```
https://docs.python.org/3/library/index.html
```

Question 07 Qu'est ce qu'un *registrar* ? (terme anglais)

Chemin vers une ressource

Comme sur un ordinateur, les données stockées sur les serveurs WEB se trouvent à l'intérieur d'une succession de répertoires ou dossiers. On parle alors d'**arborescence de fichiers**, car on peut représenter une telle situation par un arbre.



Un exemple d'arborescence de fichiers

Dans l'exemple ci-dessus, le chemin vers le fichier *logo.png* est le suivant :

/home/bob/images/logo.png

Dans un chemin, un slash / désigne un répertoire. Le premier / désigne la **racine** de l'arborescence.

Question 08 Donner le chemin des fichiers *fichier.txt* et *track2.mp3*.

Un chemin écrit depuis la racine (en commençant par /) est appelé **chemin absolu**. Mais le chemin d'accès à un fichier peut également se faire de manière **relative**, c'est à dire relativement à un répertoire particulier.
Par exemple, **depuis le répertoire *bob***, le chemin vers le fichier *logo.png* s'écrit :

images/logo.png

Question 09 Donner les chemins relatifs des fichiers *fichier.txt* et *track2.mp3* depuis le répertoire *bob*.

Depuis le répertoire *alice*, on ne peut pas directement accéder aux fichiers précédents. Il faut au préalable « remonter »dans le **répertoire parent** (celui situé « au dessus »dans l'arborescence) en écrivant deux points suivis d'un slash : **../**.
Depuis le répertoire *alice*, le chemin vers *logo.png* est :

../bob/images/logo.png

Le « ../ »signifie « remonter au répertoire parent », c'est à dire le répertoire *home* depuis le dossier *alice*. On suit ensuite l'arborescence normale.

Question 10 Donner les chemins relatifs des fichiers *fichier.txt* et *track2.mp3* depuis le répertoire *alice*.

Question 11 Donner le chemin relatif du fichier *fichier.txt* :

1. À partir du répertoire *etc*
2. À partir du répertoire *rock*

Si l'on reprend l'adresse WEB :

`http://info.cern.ch/hypertext/WWW/TheProject.html`

le chemin vers la ressource est :

/hypertext/WWW/TheProject.html

Question 12 Quel est le fichier demandé ici ? Où se situe-t-il depuis la racine du serveur ?

Question 13 Décrire entièrement l'adresse suivante (protocole / domaine / ressource) :

`http://nsi.dellasantina.corsica/fichiers/03.WEB/01.fonctionnement_web_html_css.pdf`

Quelques précisions

- Lorsque vous vous rendez à l'adresse *http ://nsi.dellasantina.corsica/*, aucune ressource n'est précisée. Plus précisement, on demande la racine du site WEB (le slash tout à la fin de l'adresse) mais rien de plus. Dans ce cas, le fichier chargé par défaut est un fichier généralement appelé *index.html*.
- De nos jours, les serveurs WEB (les ordinateurs chargés de vous transmettre le code HTML et tout ce qui va avec, nous reviendrons plus tard sur cette notion) utilisent une technique appelée **URL Rewriting** permettant d'associer à une ressource une certaine adresse « clarifiée ». Prenons l'adresse suivante :

  `https://fr.wikipedia.org/wiki/Tim_Berners-Lee`

  Il semble y avoir un fichier *Tim_Berners-Lee* (sans extension) dans un dossier *wiki*... Mais il n'en est rien ! Lorsque cette ressource est demandée, le serveur sait qu'il faut charger un page bien précise, mais dont le nom et/ou la localisation peuvent être compliqués. C'est un peu comme un raccourci placé sur votre bureau (clair, visible, accessible) pointant vers un programme enfoui dans votre ordinateur...

  Et ce qui est clair pour vous l'est aussi pour les **moteurs de recherche** : ces derniers indexent plus facilement et plus efficacement une adresse « clarifiée »qu'une adresse brut. Un exemple (fictif) :

Adresse brute
http ://monsite.fr/pages/galerie/album.html ?idAlbum=27&idPage=2

Adresse « clarifiée »
http ://monsite.fr/galerie/photos-du-bresil/page-2/

1.2 Les langages HTML et CSS

Nous l'avons vu dans la partie précédente, le langage HTML est un langage interprété par notre navigateur et retranscrit sous forme de texte, de liens hypertextes, d'images, de tableaux... C'est un langage qui est utilisé pour **décrire la structure** des pages WEB.

▷ ***Ok, le HTML est un langage comme Python...***

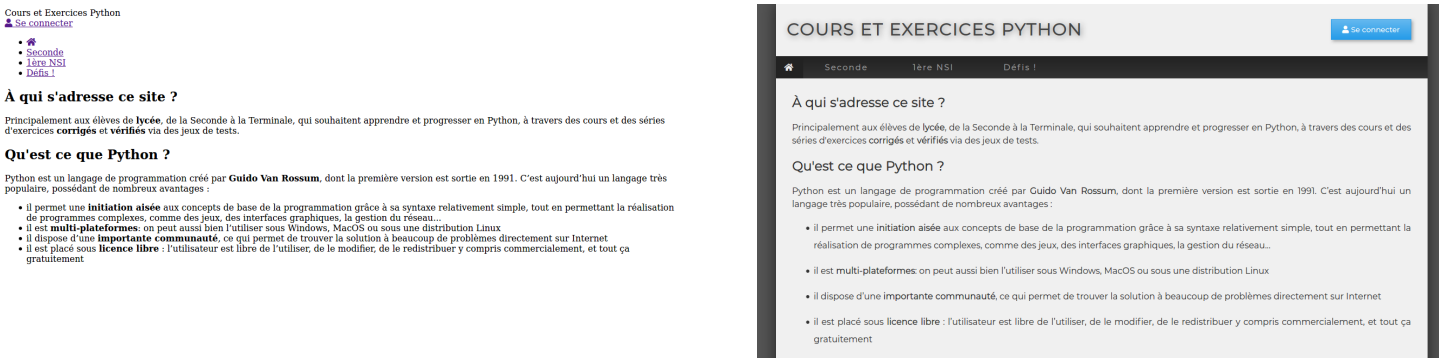
Et bien non ! Contrairement à Python qui est un langage de programmation, HTML est un **langage de description**. Impossible en HTML de déclarer une variable, de définir une fonction et encore moins de faire une boucle. Son rôle est de gérer et organiser le contenu d'une page WEB. Rien de plus.

▷ ***Dans ce cas, toutes les pages Internet devraient être strictement identiques pour tout le monde...***

Déjà, il est plus juste de parler de **pages WEB**. Mais la remarque est juste. Seulement, on peut utiliser un langage de programmation pour générer du code HTML de façon dynamique, comme **PHP** ou **Javascript**.

▷ ***Et CSS là dedans ? Encore un langage ?***

Oui, encore un langage de description, comme HTML. Le langage HTML permet de décrire la structure brute des pages, et le langage CSS (pour *Cascading Style Sheets*, ou *Feuilles de style* en français) permet de mettre le tout en forme : agencement, couleurs, taille du texte... CSS n'est pas obligatoire, mais sans lui, les informations sont affichées de manière « brute ».



Le site WEB python.dellasantina.corsica, sans CSS et avec CSS

1.2.1 Mise en place des éléments avec HTML

Le code HTML n'est pas du simple texte. Pour que le navigateur différencie le menu du site WEB, un titre ou encore une photo, on utilise des **balises**. Les pages HTML en sont remplies : elles sont invisibles pour vous, mais pas pour le navigateur. Les balises se repèrent facilement dans le code source. Elles sont entourées de *chevrons*, c'est-à-dire des symboles `<` et `>`, et s'écrivent sous la forme `<balise>`.

Question 14 Ouvrir la page `http://info.cern.ch/hypertext/WWW/TheProject.html` et afficher le code source. Quelle est la première balise utilisée ?

Les balises servent à indiquer la nature du texte qu'elles entourent. Très souvent, elles vont par paires : la première balise est **ouvrante**, et plus loin dans le code suit la même balise **fermante**. C'est le cas de la balise *title* :

```
<TITLE>The World Wide Web project</TITLE>
```

On indique ainsi que le titre (title) de la page est *The World Wide Web project.*

☞ *Il est courant aujourd'hui d'écrire les balises en minuscules, mais ce n'était pas le cas au début des années 90.*

Question 15 Où apparaît ce titre dans le navigateur ? (Ne pas regarder dans le code source mais sur la page elle-même)

Question 16 Quelle balise contient l'ensemble des informations d'une page WEB affichées par le navigateur, appelé également le *corps* de la page ?

Question 17 À quoi correspond la balise *H1* ? (ligne 6 dans le code HTML de la page précédente)

Certaines balises contiennent des informations appelées **attributs**. Les attributs d'une balise permettent d'associer un ou plusieurs paramètres à la balise. On écrit alors :

```
<balise attribut1="valeur" attribut2="valeur">...</balise>
```

C'est le cas notamment de la balise `<a>` qui permet de déclarer un **lien hypertexte**. Si l'on regarde les lignes 6 à 8 du code source précédent, on remarque l'utilisation d'une balise `<a>` pour signaler un lien. En écrivant la balise sur une seule ligne :

```
<A NAME=0 HREF="WhatIs.html">hypermedia</A>
```

On remarque deux attributs : un attribut *NAME* de valeur 0, et un attribut *HREF* dont la valeur est *WhatIs.html* : c'est précisement le nom de la page vers laquelle renvoie notre lien ! Que voit-on à l'écran ? Seulement le texte hypermedia, mais écrit en bleu et souligné pour mettre le lien en évidence.

Question 18 Afficher le code source de la page `https://python.dellasantina.corsica/` et lister quelques balises HTML présentes (sans chercher à déterminer leur rôle précis). Quelle balise apparaît en premier ?

Question 19 En étudiant le code HTML, préciser le rôle des balises `<ul>` et `<li>`.

☞ *Il existe quantité de balises HTML : le but de ce cours n'est pas de toutes les présenter en détails, mais simplement de donner un premier aperçu du langage HTML*

1.2.2 Mise en forme avec CSS

Le « premier site WEB »paraît bien fade... Et pour cause, c'est du HTML « brut », sans mise en page. Le navigateur n'a pa reçu d'informations concernant la taille du texte, la couleur, d'eventuels effets de style. Disons que le but n'était pas de créer de beaux sites WEB en 1991, mais plutôt de créer des sites WEB tout court ! La première version de CSS n'apparaîtra que plusieurs années plus tard, en 1996.
Mais aujourd'hui, CSS est incontournable. Personne n'envisagerait sérieusement un site WEB sans CSS.

☞ *En réalité, il existe des balises HTML qui permettent de faire de la mise en forme, comme la balise* `<font color="...">` *qui permet de colorer du texte... Mais pour des raisons pratiques, il est préférable que HTML ne s'occupe que de la structure, et CSS que de la mise en forme.*

Le CSS peut s'écrire directement dans les balises HTML (même si cette méthode est peu recommandée). On utilise pour cela l'attribut *style* suivi du code CSS. Par exemple, pour appliquer un style CSS à un lien hypertexte, on pourra écrire :

```
<a href="mapage.html" style="color:red;">Cliquez ici !</a>
```

Le `color:red;` permet de préciser la couleur de notre lien, ici rouge. Attention, le point-virgule `;` est obligatoire ! Il permet entre autres de pouvoir enchaîner plusieurs **propriétés CSS** :

```
<a href="mapage.html" style="color:red; background:black;">Cliquez ici !</a>
```

On demande donc au navigateur d'écrire notre lien en rouge (`color:red;`) sur fond noir (`background: black;`).

▷ ***Si je veux mettre tous les liens en rouge, je dois faire cela dans toutes mes balises ?***

Ce n'est pas nécessaire, c'est d'ailleurs pour cette raison que mélanger le CSS dans les balises HTML n'est pas recommandé. L'idée est de réellement séparer les deux langages. Une seconde manière de faire est de placer le CSS dans un fichier externe, nommé par exemple *style.css* et appelé dans la page HTML avec la balise `<link>` :

```
<link rel="stylesheet" type="text/css" href="style.css" />
```

☞ *La balise* `link` *est spéciale : c'est une balise* ***orpheline****, car elle ne va pas par paire, elle est seule. On le précise en utilisant un slash / avant la fermeture du chevron.*

▷ ***Qu'écrit-on dans ce fichier CSS ?***

On va lister les propriétés CSS associées à chacune des balises de la page HTML. Par exemple, pour faire apparaître tous les liens en rouge sur fond noir, on écrira :

```
a
{
  color: red;
  background: black;
}
```

On peut également tout écrire sur une seule ligne :

```
a { color: red; background: black; }
```

Question 20 Ouvrir le site WEB `https://python.dellasantina.corsica/` et repérer (dans le code source) les lignes correspondant à l'utilisation de fichiers CSS. Elles sont de la forme :

```
<link rel="stylesheet" type="text/css" href="lienVersFichier.css" />
```

Question 21 Ouvrir le fichier `clair.css` ou `sombre.css` en cliquant dessus dans le code source.
Quelle ligne permet de choisir la couleur du texte dans la balise `<html>` ?

La valeur associée à cette dernière propriété est une **valeur héxadécimale** représentant une couleur. Pour représenter une couleur, on indique la quantité de rouge, de vert et de bleu (dans cet ordre). Avec 3 caractères héxadécimaux (comme c'est le cas pour *#555*), le premier caractère représente la quantité de rouge, le second la quantité de vert et le troisième la quantité de bleu. Un caractère héxadécimal permet de réprésenter 16 valeurs (de 0 à 15), nous avons donc ici une couleur utilisant les niveaux suivants :

- Rouge : 5/15
- Vert : 5/15
- Bleu : 5/15

En sachant que *#000* représente le noir et *#fff* représente le blanc, on en déduit que *#555* représente un gris foncé.
C'est cette couleur qui est affichée pour le fond de page.

Question 22 Toujours depuis le code CSS précédent, quelle ligne permet de choisir la couleur de fond du `footer` (pied de page) ?.

Dans cette dernière balise, la couleur est représentée par 6 caractères héxadécimaux : les deux premiers codent le rouge, puis le vert, puis le bleu. La précision est ici accrue : avec deux caractères héxadécimaux, on peut représenter 256 valeurs différentes (de 0 à 255). En héxadécimal, *F0 = 240*, ainsi les niveaux utilisés dans la couleur *#f0f0f0* sont :

- Rouge : 240/255
- Vert : 240/255
- Bleu : 240/255

Il s'agit ici un gris très clair, quasiment blanc.

Question 23 À l'aide du site WEB `https://html-color.codes/`, décrire la couleur héxadécimale *#e9942e*.

1.3 QCM

Chaque question admet **une seule bonne réponse**.

Questions	Réponses
1. Quelle personne est connue pour être l'inventeur du WEB ?	☐ Linus Torvalds ☐ Tim Berners-Lee ☐ Alan Turing ☐ Bill Gates
2. Quel est l'intrus ?	☐ Firefox ☐ Safari ☐ Google ☐ Internet Explorer
3. Quel est le protocole utilisé pour naviguer sur le WEB ?	☐ HTTP ☐ URL ☐ HTML ☐ DNS
4. Le code source d'une page WEB est :	☐ Un ensemble de règles à suivre sur le WEB ☐ Un code utilisé par le navigateur WEB pour représenter une page WEB ☐ Un code secret pour accéder au WEB ☐ Le code de la matrice
5. On considère l'URL : https ://photos.monsite.fr/village/photo3.jpg On peut dire que :	☐ Le nom de domaine utilisé est *monsite* ☐ L'extension du nom de domaine est *.jpg* ☐ Les échanges avec le site *photos.monsite.fr* ne sont pas sécurisés ☐ Il existe un dossier village sur le site *photos.monsite.fr*
6. Quel est l'intrus ?	☐ HTML ☐ Python ☐ PHP ☐ Javascript
7. La balise HTML permettant de créer un lien hypertexte est :	☐ ¡href¿ ☐ ¡link¿ ☐ ¡a¿ ☐ ¡title¿
8. L'utilisation du CSS est obligatoire dans une page WEB :	☐ Vrai ☐ Faux

1.4 Envie d'en savoir (beaucoup) plus ?

Ce cours n'est qu'un survol à très haute altitude du HTML et du CSS. Pour en savoir plus sur ces deux langages, je ne peux que conseiller l'excellent site *Openclassrooms* qui regorge d'informations à ce sujet. On pourra notamment suivre le lien suivant :

`https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3`