CHAPITRE I : LES BASES DU WEB

1.1 DEFINITION ET HISTORIQUE :

Le World Wide Web, littéralement la « toile (d'araignée) mondiale », communément appelé le Web, le web parfois la Toile ou le WWW, est un système hypertexte public fonctionnant sur Internet qui permet de consulter, avec un navigateur, des pages accessibles sur des sites.

Le Web n'est qu'une des applications d'Internet. D'autres applications d'Internet sont le courrier électronique, la messagerie instantanée, …etc.

Historique (source a little history of world wide web)

1989 : Tim Berners-Lee lance l'idée de la Toile
En tant qu'utilisateur de CERNET, le réseau du **CERN**, le chercheur **Tim Berners-Lee** conçoit l'idée de naviguer simplement d'un espace à un autre d'Internet à l'aide de liens hypertextes et grâce à un navigateur. **Tim Berners-Lee** parle de la création d'une toile, tout internaute pouvant aller d'un contenu à l'autre suivant des voies multiples. Il présentera son projet au **CERN** en Novembre 1990. Pendant les trois années suivantes, il travaillera à l'apparition du World Wide **Web**, « toile d'araignée mondiale ».

1990 : L'Université de l'Illinois présente Mosaic
L'université de l'Illinois présente son navigateur **Web** graphique, reposant sur les principes de la Toile tels qu'ils ont été formulés par l'équipe du **CERN** de **Tim Berners-Lee**, notamment le HTTP. Nommée **Mosaic**, l'application fonctionnant sur **Windows** simplifie considérablement la navigation. Elle annonce le développement ultérieur de **Netscape** et autres navigateurs.

1994 : Création de Yahoo!
Deux étudiants de Stanford, David Filo et Jerry Yang, créent l'annuaire Internet Yahoo! Celui-ci doit permettre aux Internautes de trouver rapidement des sites grâce à un classement hiérarchique. L'entreprise sera fondée en 1995 et connaîtra un rapide succès.

1994 : Naissance du W3C
Tim Berners-Lee fonde le World Wide **Web** Consortium, également appelé **W3C**. Cet organisme a pour objectif et fonction d'émettre des recommandations afin de promouvoir et d'assurer la compatibilité des technologies utilisées sur le **Web**. Toutefois les standards proposés ne sont pas des normes absolues. L'organisme, essentiel pour assurer l'efficacité des applications tels que les navigateurs, est géré conjointement par des universités et centres de recherche américains, européens et japonais.

1995 : Le succès de Yahoo.
Moins de deux ans et demi après sa naissance, Yahoo **fait son entrée en bourse. Transformant son statut de simple annuaire en celui de portail aux contenus divers, implanté dans différents pays.**

2010 : 2 Milliards d'utilisateurs, et plus de 250 millions de sites.

1.2 LES CONCEPTS DE BASE :

Le World Wide Web, littéralement la « toile (d'araignée) mondiale », communément appelé le Web, le web parfois la Toile ou le WWW, est un système hypertexte public fonctionnant sur Internet qui permet de consulter, avec un navigateur, des pages accessibles sur des sites.

Une **ressource** du web est une entité informatique (texte, image, forum Usenet, boîte aux lettres électronique, etc.) accessible indépendamment d'autres ressources. Une ressource en accès public est librement accessible depuis Internet. Une ressource locale est présente sur l'ordinateur utilisé, par opposition à une ressource distante (ou en ligne), accessible à travers un réseau.

On ne peut accéder à une ressource distante qu'en respectant un **protocole de communication**. Les fonctionnalités de chaque protocole varient : réception, envoi, voire échange continu d'informations.

HTTP (pour HyperText Transfer Protocol) est le protocole de communication communément utilisé pour transférer les ressources du Web. **HTTPS** est la variante sécurisée de ce protocole.

Une **URL** (pour Uniform Resource Locator) pointe sur une ressource. C'est une chaîne de caractères permettant d'indiquer un protocole de communication et un emplacement pour toute ressource du Web. (exemple : http://www.univ-chlef.dz).

Un **hyperlien** (ou lien) est un élément dans une ressource associé à une URL. Les hyperliens du Web sont orientés : ils permettent d'aller d'une source à une destination.

HTML (pour HyperText Markup Language) et XHTML (Extensible HyperText Markup Language) sont les langages informatiques permettant de décrire le contenu d'un document (titres, paragraphes, disposition des images, etc.) et d'y inclure des hyperliens. Un document HTML est un document décrit avec le langage HTML. Les documents HTML sont les ressources les plus consultées du Web.

Dans un mode de communication **client-serveur**, un serveur est un hôte sur lequel fonctionne un logiciel serveur auquel peuvent se connecter des logiciels clients fonctionnant sur des hôtes clients.

Un serveur Web est un hôte sur lequel fonctionne un serveur HTTP (ou serveur Web). Un serveur Web héberge les ressources qu'il dessert.

Un **navigateur Web** est un logiciel client HTTP conçu pour accéder aux ressources du Web. Sa fonction de base est de permettre la consultation des documents HTML disponibles sur les serveurs HTTP. Le support d'autres types de ressource et d'autres protocoles de communication dépend du navigateur considéré.

Une **page Web** (ou page) est un document destiné à être consulté avec un navigateur Web. Une page Web est toujours constituée d'une ressource centrale (généralement un document HTML) et d'éventuelles ressources liées automatiquement accédées (typiquement des images).

Un **éditeur HTML** (ou éditeur Web) est un logiciel conçu pour faciliter l'écriture de documents HTML et de pages Web en général.

Un **site Web** (ou site) est un ensemble de pages Web et d'éventuelles autres ressources, liées dans une structure cohérente, publiées par un propriétaire (une entreprise, une administration, une association, un particulier, etc.) et hébergées sur un ou plusieurs serveurs Web.

Une **adresse Web** est une URL de page Web, généralement écrite sous une forme simplifiée limitée à un nom d'hôte. Une adresse de site Web est en fait l'adresse d'une page du site prévue pour accueillir les visiteurs.

Un **hébergeur Web** est une entreprise de services informatiques hébergeant (mettant en ligne) sur ses serveurs Web les ressources constituant les sites Web de ses clients.

Une **agence Web** est une entreprise de services informatiques réalisant des sites Web pour ses clients.

Un **annuaire Web** est un site Web répertoriant des sites Web.

Un **portail Web** est un site Web tentant de regrouper la plus large palette d'informations et de services possibles dans un site Web. Certains portails sont thématiques.

Un **service Web** est une technologie client-serveur basée sur les protocoles du Web.

1.3 ARCHITECTURE DU WEB :

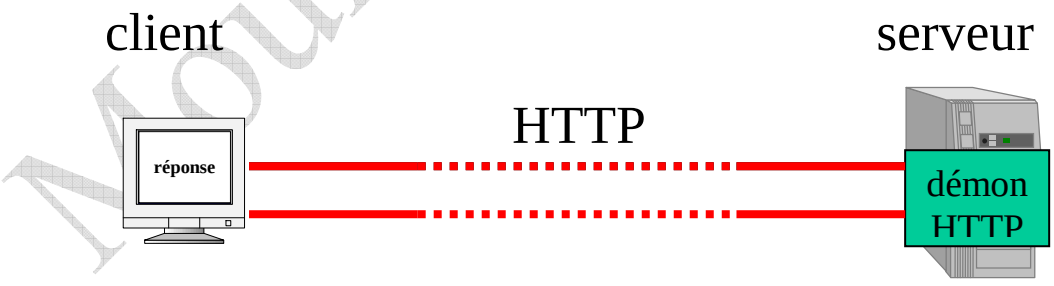
L'architecture du web se base sur les modèles de **Serveur/Client** . Le client envoie des requêtes au serveur, comme :

- transfert de fichiers
- exécution de programmes sur le serveur
- mise à jour de fichiers

Les objets manipulés sont repérés par leur URL.

Le transfert se fait en utilisant le protocole http. Il définit le langage utilisé pour les échanges entre client et serveur Web. Ce protocole n'exige pas de session permanente entre client/serveur

1.4 DEROULEMENT D'UNE REQUETE :



1. Demande d'une connexion
2. Attente de la réponse du serveur
3. Etablissement de la connexion
4. Envoi d'une requête URL
5. Réponse du serveur
6. Affichage de la réponse
7. Fermeture de la connexion

1.5 LE PROTOCOLE HTTP

Le HTTP est un protocole simple, basé sur l'émission d'une requête vers un **serveur HTTP**, en vue de l'obtention d'une **ressource**. Cette ressource est identifiée par un URL = une position "logique" dans le réseau. L'objectif initial du HTTP est la requête sur des pages hypertexte (HTML), lesquelles servent de base à la construction d'un document composite, par combinaison de divers types de ressources annexes :

- des images
- des feuilles de style
- des scripts clients attachés

L'un des plus importants avantages du protocole http est de « découper » les documents à transférer en blocs. Ces blocs peuvent être acheminés de manière indépendante du serveur vers le client. . Ils seront ensuite assemblés au niveau du récepteur.

Gestion des cas d'erreur

Comme tout protocole se doit d'être complet, HTTP doit permettre de renseigner le client sur les multiples cas d'erreur qui peuvent se rencontrer au moment de la résolution de l'adresse "logique". Les codes d'erreur du HTTP sont le résultat du catalogage des erreurs possibles (ressource manquante, URL non conforme, domaine inexistant, accès interdit, etc...).

Gestion des types

Le rôle premier du protocole HTTP est l'acheminement de documents (ressources). Ce n'est que plus tard qu'il a pris en charge d'autre types d'échange utilisateur/réseau, tels que les applications en ligne. Les ressources nécessitent, pour être visulisées, que le client les décode. Chaque format de ressource suppose donc un décodeur adéquat. Le HTTP prend en charge la transmission des méta-informations nécessaires pour que le client arrive à détecter de quel décodeur il aura besoin pour visualiser correctement la ressource.

- Type (extension du document, type MIME)
- Encodage du document (charsets)
- Formatage du document (compression)

GESTION DE L'ACHEMINEMENT

Le HTTP prend en charge, pour la gestion de l'acheminement des données, certains aspects techniques :

- **Gestion des caches :** Le transport de documents en HTTP s'effectue à travers un réseau informatique qui peut compter un certain nombre de relais. Ces relais sont des organes physiques (hors câbles) qui peuvent faire preuve d'une certain intelligence vis à vis du contenu (filtres, proxies, routeurs). Le protocole HTTP prend en charge un certain nombre d'aspects de cette chaîne d'acheminement, en donnant des indications sur les réactions souhaitées de cette chaîne.
- **Gestion des redirections :** Les ressources sont des document "propriété" de l'émetteur. Le problème du Web aujourd'hui est autant dans les technologies clientes (accroissement des possibilités) que dans celle de l'organisation de la masse de documents qu'il représente. Les ressources sont donc souvent réorganisées, mobiles et parfois fugitives. C'est de plus en plus le cas avec les applications dynamiques. Le protocole HTTP contient des primitives qui permettent de donner des informations pertinentes sur ces déplacements.

GESTION D'UNE PERSISTANCE D'INFORMATION

Le protocole HTTP, à travers les Cookies, fournit un moyen pour "marquer" l'agent client de façon suffisamment précise :

- L'application est marquée, puisque c'est dans le container de cette application que l'information de marquage est stockée.
- L'utilisateur d'une machine multi-utilisateur est marqué, puisque les informations de marquage sont stockés dans le profil particulier de cet utilisateur.

STRUCTURE DU HTTP

Notion d'entité : Le HTTP est un protocole destiné à transporter des entités document . Il est constitué de messages-requêtes (du client au serveur) et de messages-réponse (du serveur au client). Il implémente donc parfaitement un paradigme client-serveur en ce sens que :

- Le client (agent utilisateur) est toujours l'initiateur de la requête (PULL)
- Le serveur est toujours en attente d'un client

Les messages transportent des **entités**. Le cas le plus fréquent est celui d'une entité unique. L'entité est toujours définie par :

- une en-tête d'entité (méta informations sur le contenu et la transmission).
- un corps d'entité (le contenu).

Le protocole admet cependant que plusieurs entités puissent être transportées par le même message. C'est le cas du téléchargement de fichiers accompagnant des données de formulaire, ou de téléchargements multiples. (format **multipart**)

Dans ce cas, le message doit permettre de séparer proprement les différentes entités qu'il transporte.

CONSTRUCTION DE LA REQUETE

Le libellé de requête :
La requête HTTP, comme son nom l'indique est une demande à un serveur pour qu'il exécute un ordre (principalement d'obtenir un document). Ces ordres sont symbolisés par des VERBES.

Les verbes les plus connus sont :
- HEAD : "donne moi les méta informations conernant un document"
- GET : "donne moi le contenu du document (et ses méta informations par la même occasion)"
- POST : "fais ce que tu veux (ou tu peux avec les données que je t'envoie)"
- PUT : "mets le document que je t'envoie où je te le dis"

La cible :
Elle est représentée par une URL absolue ou relative. Le serveur est chargé de "consommer" cette cible, la traduire en une réalité physique tangible (un fichier physique, un exécutable, un répertoire).
Les champs d'en-tête

La requête peut informer le serveur d'un certain nombre de prédisposition de l'agent utilisateur. Des mécanismes de gestion de préférences, convenues entre le client et le serveur peuvent permettre d'améliorer le service rendu, sans faire appel à des choix explicites de l'utilisateur. Ces préférences sont, par exemple :

- Le choix de la langue la plus appropriée.
- Le choix d'un encodage acceptable du côté du client.
- Le choix d'une version suffisamment récente, ou au contraire de versions plus anciennes.
- Les champs de requête sont placés à la suite de la "ligne VERBE" sous la syntaxe :

```
NomChamp:[ValeurChamp]LF
```

L'en-tête doit se terminer par une ligne vide, pour indiquer que ce qui suit est un corps d'entité (ou que le message est fini).

Un corps de requête
Certains verbes (PUT,POST) supposent très fortement une transmission de contenu du client vers le serveur. Ce contenu pouvant être à priori d'une taille quelconque, il s'agit bien d'un corps d'entité. Dans ce cas, on doit alors considérer potentiellement une requête comme étant une transmission d'un document à part entière. Le client pousse un document local vers le serveur.

Détermination de la longueur du corps d'entité
Il n'existe pas en HTTP de bloc de fin de message identifiable. D'autre part, virtuellement, le corps d'entité peut contenir n'importe quelle séquence binaire, y compris une séquence qui pourrait se confondre avec un tel bloc. La question de prévoir la fin du corps d'entité DOIT être réglée dès l'en-tête. Toute requête portant un contenu d'une certaine longueur doit informer le serveur de cette longueur. On utilise le champ :

```
Content-Length:
```

Résumé
Une requête mono-entité, dans le cas général se contruit ainsi :

```
[verbe] [Url] [version HTTP] LF
Content-Length:[longueur du corps]
[Nom Champ]:[Valeur Champ]LF
...
LF (sépare l'en-tête et le corps d'entité)
[Octets de l'entité]
```

CONSTRUCTION DE LA REPONSE

De cette dernière conclusion, on déduit que le schéma de réponse est symétrique à celui de la requête, à la différence près que le corps d'entité est le plus souvent non vide.

1.6 LE LANGAGE HTML

Le **HTML** (« *HyperText Mark-Up Language* ») est un langage dit de « marquage » (de « structuration » ou de « balisage ») dont le rôle est de formaliser l'écriture d'un document avec des balises de formatage. Les balises permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents.

Le HTML n'est pas un langage de programmation. Il s'agit d'un langage permettant de décrire la mise en page et la forme d'un contenu rédigé en texte simple.

Une page HTML est ainsi un simple fichier texte contenant des balises (parfois appelées marqueurs ou repères ou tags en anglais) permettant de mettre en forme le texte, les images, etc.

Par convention l'extension donnée au fichier est .htm ou .html, mais une page web peut potentiellement porter n'importe quelle extension.

Une page web peut être construite à partir du plus basique des éditeurs de texte (une application *bloc-note* par exemple), mais il existe des éditeurs beaucoup plus évolués.

COMMENT UTILISER LES BALISES HTML ?

Une balise est un élément de texte (un nom) encadrée par le caractère inférieur (<) et le caractère supérieur (>). par exemple « <*H1*> ».

Les balises HTML fonctionnent par paire afin d'agir sur les éléments qu'elles encadrent. La première est appelée « *balise d'ouverture* » (parfois *balise ouvrante*) et la seconde « *balise de fermeture* » (ou *fermante*). La balise fermante est précédé du caractère */*) :

<marqueur> Votre texte formaté </marqueur>

A titre d'exemple, les balises et permettent de mettre en gras le texte qu'elles encadrent :

 Ce texte est en gras

Les balises HTML peuvent parfois être uniques : la balise
 représente par exemple un retour à la ligne.

Afin d'être le plus proche possible du standard XHTML (beaucoup plus stricte que le standard HTML), il est conseillé d'utiliser la notation suivante : *
*.

IMBRICATION DES BALISES

Les balises HTML ont la particularité de pouvoir être imbriquées de manière hiérarchique afin de permettre le cumul de leur propriétés. En contrepartie le chevauchement de balises n'est pas toléré par le standard HTML. Voici un exemple de texte formaté avec des balises imbriquées :

<u>Comment ça Marche</u>, encyclopédie informatique libre

L'exemple ci-dessus donne le résultat suivant :
<u>Comment ça Marche</u>, encyclopédie informatique libre

En contrepartie l'exemple ci-dessous n'est pas correct :
Comment ça Marche, encyclopédie informatique libre

NOTION D'ATTRIBUT

Un attribut est un élément, présent au sein de la balise ouvrante, permettant de définir des propriétés supplémentaires. Les attributs se présentent la plupart du temps comme une paire clé=valeur, mais certains attributs ne sont parfois définis que par la clé.

Voici un exemple d'attribut pour la balise <p> (balise définissant un paragraphe), permettant de spécifier que le texte doit être aligné sur la droite :

```
<p align="right">Exemple de paragraphe</p>
```

Chaque balise peut comporter un ou plusieurs attributs, chacun pouvant avoir (aucune,) une ou plusieurs valeurs.

ESPACES, SAUT DE LIGNE ET TABULATIONS

`Le langage HTML ne tient pas compte des espaces, des tabulations et des` sauts de ligne (ci-après appelés) ou plus exactement il considère une suite d'un ou plusieurs espaces/tabulations/saut de ligne comme une seule espace.

Le langage HTML possède par contre des éléments permettant expressément de définir chacun de ces éléments de mise en forme :

- **Espace insécable** : il s'agit d'une espace ne pouvant être brisée par une fin de ligne. Sa représentation en HTML est * *.
- **Saut de ligne manuel** : il s'agit d'un saut de ligne explicite. Sa représentation en HTML est <*br*> (<*br /*> pour être conforme au XHTML).

A noter: La balise <NOBR> </NOBR> permet à l'inverse d'empêcher le retour automatique à la ligne réalisé par le navigateur !

COMMENTAIRES

Il est possible d'ajouter des éléments d'information dans une page web sans que ceux-ci soient affichés à l'écran grâce à un jeu de balises spécifique, appelé *balises de commentaires*.

```
<!-- Voici un commentaire -->
```

Les balises de commentaires permettent de mettre en commentaire du texte mais peuvent également servir à commenter du code HTML.

STRUCTURE DU DOCUMENT HTML

Un document HTML commence par la balise <*HTML*> et finit par la balise <*/HTML*>. Il contient également un *en-tête* décrivant le titre de la page, puis un *corps* dans lequel se trouve le contenu de la page.

L'en-tête est délimité par les balises <*HEAD*> et <*/HEAD*>. Le corps est délimité par les balises <*BODY*> et <*/BODY*>.

Voici par exemple une page HTML minimaliste :

```
<HTML>
 <HEAD>
  <TITLE>Titre de la page</TITLE>
 </HEAD>

 <BODY>
  Contenu de la page
    </BODY>
</HTML>
```

DECLARATION DU TYPE DE DOCUMENT

Il est conseillé d'indiquer dans la page HTML le *prologue du type de document*, c'est-à-dire une référence à la norme HTML utilisée, afin de spécifier le standard utilisé pour le codage de la page. Cette déclaration se fait par une ligne du type :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
 <HEAD>...</HEAD>
 <BODY>Contenu de la page</BODY>
</HTML>
```

La déclaration du document indique la DTD (*Document Type Definition*) utilisée, c'est-à-dire la référence des caractéristiques du langage utilisé.

LES BALISES DE STYLE

Les balises de style modifient la typographie du texte. Elles peuvent être imbriquées dans d'autres balises de style de la même façon qu'on le ferait avec un traitement de texte.

Exemples :

Balise de style	Effet Visuel
<ABBREV>	Abréviation
<ACRONYM>	Acronyme
<AU>	L'auteur
<B>	**Met la police en gras**
<BIG>	Police plus grande
<BLINK>	Clignote (propre à Netscape)
<CITE>	*Citation*
<CODE>	`Instruction`
<DEL>	
<DFN>	*Définition d'instance*
<EM>	*Emphase*
<I>	*Italique*
<INS>	Nouveau texte inséré a cet endroit
<KBD>	`Clavier - Suite de caractères devant être tapés tel quel`
<PERSON>	Accentuation du nom d'une personne
<Q>	Encadre le texte par des guillemets

<S>	~~Comme strike (barré)~~
<SAMP>	`Exemple`
<SMALL>	Police plus petite
<STRONG>	**Forte accentuation rendue par du gras**
<STRIKE>	~~Texte barré (comme S)~~
<SUB>	Texte en $_{\text{Indice}}$
<SUP>	Texte en $^{\text{Exposant}}$
<TT>	`Caractère de machine à écrire`
<VAR>	*Nom d'une variable*

NIVEAUX DE TITRE

Le langage HTML définit 6 niveaux de titre (en anglais *heading*), afin de définir une structuration hiérarchique des paragraphes dans un texte :

Balise Effet Visuel

H1 Test

H2 Test

H3 Test

H4 Test

H5 Test

H6 Test