

Lexique de l'apprenti développeur

Se comprendre pour mieux apprendre

Le web

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, mais pour faciliter les échanges d’information Tim Berners-Lee met au point le système hypertexte. Le système hypertexte permet, à partir d’un document, de consulter d’autres documents en cliquant sur des mots clés. Ces mots “cliquables” sont appelés hyperliens et sont souvent soulignés et en bleu. Ces hyperliens sont plutôt connus aujourd’hui sous le simple terme de “liens”.

Adresse IP et nom de domaine

Le domaine (ex : impots.gouv.fr) peut généralement être remplacé par une adresse IP.

Testez : 145.242.11.26 pointe vers le même site que impots.gouv.fr

C’est le serveur **DNS** (domain name serveur) qui traduit cette adresse IP en nom de domaine.

HTTP, URL & HTML

Techniquement le web se base sur trois choses :

- Le protocole HTTP (HyperText Transfert Protocol),
- Les URL (Uniform Resource Locator)
- Le langage de description HTML (HyperText Markup Language).

Une chose très importante à bien avoir à l’esprit : beaucoup de personnes confondent “web” et “internet”. Même si le “web” “s’appuie” sur internet, les deux choses n’ont rien à voir puisqu’“internet” est un “réseau de réseau” s’appuyant sur le protocole IP alors que, comme nous venons de le voir, le web est la combinaison de trois technologies : HTTP, URL et HTML. D’ailleurs on trouve autre chose que le “web” sur internet, par exemple, les emails avec le protocole SMTP (Simple Mail Transfert Protocol) et les transferts de fichiers avec le protocole FTP (File Transfert Protocol), le Peer to Peer (Protocol BitTorrent), ect....

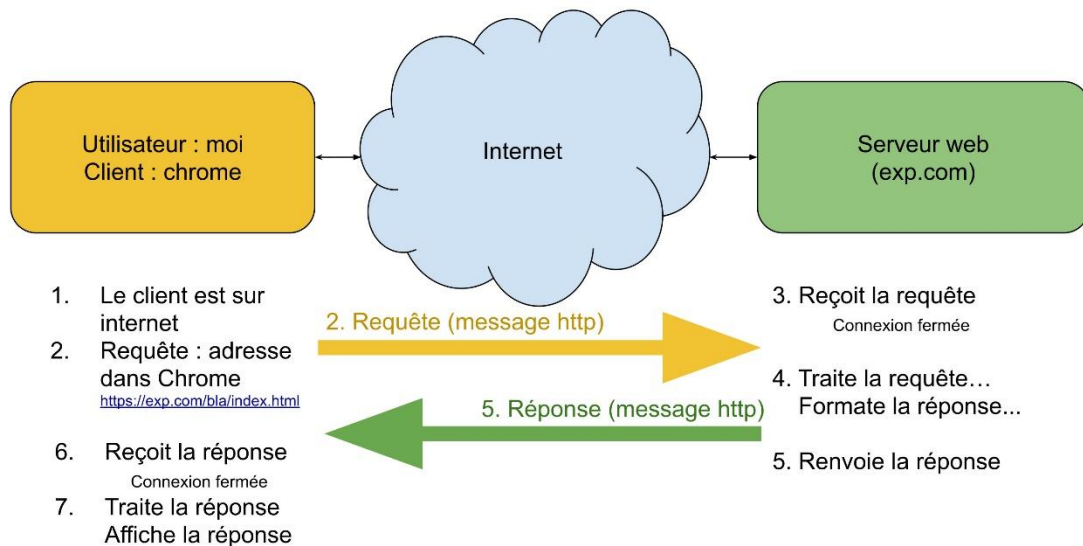
Protocole HTTP

Un **protocole** est un ensemble de règles qui permettent à deux ordinateurs de communiquer ensemble. HTTP (HyperText Transfer Protocol) permet au client d’effectuer des requêtes à destination d’un serveur web. En retour le serveur web envoie une réponse.

- HTTP est utilisé pour échanger des données et fournir des contenus : images, vidéos, textes (pages web) etc.

- HTTP est un protocole client / serveur.
 - Le client (généralement le navigateur) émet une requête,
 - Le serveur renvoie une réponse à la requête.

Le cycle Requête Réponse



Les URL

Dans la barre d'adresse de votre navigateur web vous trouvez, quand vous visitez un site, des choses du genre :

"https://qkzk.xyz/docs/nsi/cours_premiere/programmation/comparaison/".

La partie `"/docs/nsi/cours_premiere/programmation/comparaison/"` s'appelle une URL.

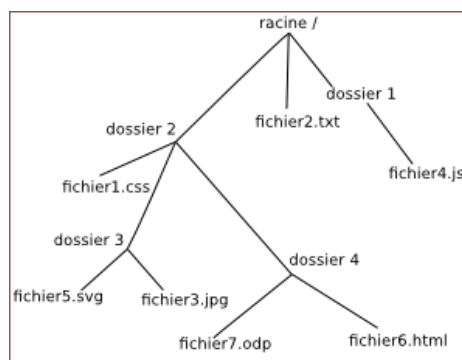
Plus précisément :

protocole domaine

URL

`https://qkzk.xyz/docs/nsi/cours_premiere/programmation/comparaison/`

Une URL (Uniform Resource Locator) permet d'identifier une ressource (par exemple un fichier) sur un réseau.



Comme vous pouvez le constater, la base de l'arbre s'appelle la *racine* de l'arborescence et se représente par un /

Chemin absolu ou chemin relatif ?

Il existe deux types de chemins dans une arborescence : **absolu** et **relatif**.

Le chemin absolu doit indiquer le chemin depuis la racine.

Par exemple l'URL du fichier `fichier3.jpg` sera : `/dossier2/dossier3/fichier3.jpg`

Imaginons maintenant que le fichier `fichier1.css` fasse appel au fichier `fichier3.jpg` (comme un fichier HTML peut faire appel à un fichier CSS). Il est possible d'indiquer le chemin non pas depuis la racine, mais depuis le dossier (`dossier2`) qui accueille le fichier `fichier1.css`, nous parlerons alors de chemin relatif :

`dossier3/fichier3.jpg`

Remarquez l'absence du / au début du chemin (c'est cela qui nous permettra de distinguer un chemin relatif et un chemin absolu).

Chemin absolu

- Débute par /
- Indique l'adresse depuis la racine

Chemin relatif

- Ne débute pas par /
- Indique l'adresse depuis la position actuelle
- ../ signifie un niveau plus haut dans l'arborescence

HTML

HTML est un *langage de balise* et non un *langage de programmation*.

HTML permet de *décrire* le contenu d'un document et d'en donner *la structure*.

HTML n'étant pas un langage de programmation, il n'existe pas de variables ou de conditions, fonctions etc. en HTML.

Voici la définition accessible sur Wikipédia :

L'Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias, dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade).

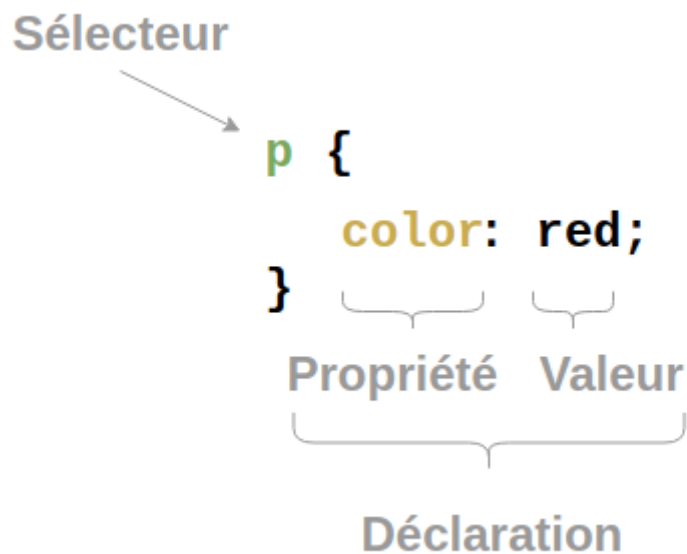
CSS

Si HTML décrit la structure du document, CSS en réalise la mise en forme.

C'est CSS qui permet d'appliquer des couleurs, des changements de fontes, de taille etc.

CSS pour "Cascading Style Sheets" ou "feuilles de style en cascade"

Le principe de CSS est d'appliquer une mise en forme à un ou des éléments sélectionnés.



Sélecteur

C'est le nom de l'élément HTML situé au début de l'ensemble de règles. Il permet de sélectionner les éléments sur lesquels appliquer le style souhaité (en l'occurrence, les éléments p). Pour mettre en forme un élément différent, il suffit de changer le sélecteur.

- Les sélecteurs de balise.

Ils sélectionnent toutes les balises portant ce nom.

```
p {  
  color: red;  
}
```

- Les sélecteurs d'ID

Ils sélectionnent LA balise portant cette id

```
#mon-id {  
  background-color: black;  
}
```

- Les sélecteurs de classes

Ils sélectionnent TOUTES les balises portant cette classe

```
.ma-classe {  
  font-style: verdana;  
}
```

- Citons aussi les sélecteurs de “pseudo-classe”

Ils sélectionnent les éléments donnés mais uniquement dans un certain état

```
a:hover {  
  color: blue;  
}
```

Déclaration

C'est une règle simple comme `color: red;` qui détermine les propriétés de l'élément que l'on veut mettre en forme.

Propriétés

Les différentes façons dont on peut mettre en forme un élément HTML (dans ce cas, `color` est une propriété des éléments 'p'). En CSS, vous choisissez les différentes propriétés que vous voulez utiliser dans une règle CSS.

Valeur de la propriété

À droite de la propriété, après les deux points, on a la valeur de la propriété. Celle-ci permet de choisir une mise en forme parmi d'autres pour une propriété donnée (par exemple, il y a d'autres couleurs que `red` pour la propriété `color`).

Modèle Client-Server

Deux ordinateurs en réseau peuvent s'échanger des données. Dans la plupart des cas ces échanges ne sont pas “symétriques” : en effet un ordinateur A va souvent se contenter de demander des ressources (fichiers contenant du texte, photos, vidéos, sons...) à un ordinateur B. L'ordinateur B va lui se contenter de fournir des ressources à tous les ordinateurs qui lui en feront la demande. On dira alors que l'ordinateur A (celui qui demande des ressources) est un **client** alors que l'ordinateur B (celui qui fournit les ressources) sera qualifié de **serveur**.

Serveur

N'importe quel type d'ordinateur peut jouer le rôle de serveur, mais dans le monde professionnel les serveurs sont des machines spécialisées conçues pour fonctionner 24h sur 24h. Ils peuvent aussi avoir une grosse capacité de stockage afin de stocker un grand nombre de ressources (vidéos, sons,...). Une seule machine peut servir de nombreuses applications.



Afin assurer une continuité de service, dans les sociétés, plusieurs serveurs assurent exactement le même rôle (on parle de redondance). Vous vous doutez bien que Google ne possède pas qu'un seul serveur, en effet, en moyenne, chaque seconde, c'est environ 65000 clients qui se connectent aux serveurs du moteur de recherche de Google.

Aucun serveur, même extrêmement performant, ne serait capable de répondre à toutes ces requêtes. Google, Amazon, Facebook ou Netflix possèdent un très grand nombre de serveurs afin de pouvoir satisfaire les demandes des utilisateurs en permanence. Ces entreprises possèdent d'immenses salles contenant chacune des centaines ou des milliers de serveurs (ces serveurs sont rangés dans des armoires appelées "baie serveur").



Souvent les serveurs sont spécialisés dans certaines tâches, par exemple, les serveurs qui envoient aux clients des pages au format HTML sont appelés "serveur web".

Statique, dynamique

Il y a quelques années, le web était dit « statique » : le concepteur de site web écrivait son code HTML et ce code était simplement envoyé par le serveur web au client. Les personnes qui consultaient le site avaient toutes le droit à la même page, le web était purement « consultatif ».

Les choses ont ensuite évolué : les serveurs sont aujourd'hui capables de générer eux-mêmes du code HTML. Les résultats qui s'afficheront à l'écran dépendront donc des demandes effectuées par l'utilisateur du site : le web est devenu *dynamique*.

JavaScript :

HTML et CSS sont accompagnés de JavaScript. JavaScript (aucun rapport avec Java) est : • le seul langage que les navigateurs (chrome, firefox etc.) puissent exécuter

JavaScript remonte aux années 90 et a longtemps été perçu comme un accessoire. Il est maintenant le langage préféré des développeurs et le langage le plus utilisé dans le monde. JavaScript permet de faire faire les gros calculs (d'un jeu vidéo, d'une animation) sur la machine DU CLIENT et économise beaucoup de ressources pour le serveur. Sans JavaScript, pas de réseaux sociaux, pas d'internet moderne, aucun serveur ne pourrait faire autant de calculs.

Quelques particularités de JavaScript :

- Exécuté dans le navigateur, côté client. Par opposition aux calculs réalisés côté serveur
- Javascript est un langage "orienté objet" comme presque tous les langages "majeurs" (à l'exception notable du langage C, le "parent" des langages modernes.)
- Javascript permet, depuis peu, d'être exécuté côté serveur grâce à Node.js
- Javascript réagit à des "événements". Un *événement* est une action qui se produit dans le système et auquel javascript peut répondre :

Côté client, JavaScript est exécuté par le navigateur. Il est très limité. Il ne peut pas accéder aux fichiers de la machine (sinon imaginez le désastre : vous accédez à un site qui efface vos fichiers sans vous prévenir !). On charge un fichier JS depuis une page web avec la balise