# PARTIE 1 :

▪ **Web** *(World Wide Web)* : Système interconnecté de pages web publiques accessibles à travers Internet. Une page web elle-même est un document dit "HyperText" écrit en HTML (*HyperText Markup Language*). Elle est reliée à d'autres pages web par le biais d'hyperlinks, éléments cliquables présents à l'intérieur de la page web, menant d'un document HyperText à un autre. Enfin, la page web est accessible indépendamment de toute autre en utilisant son adresse propre, dite URL (Uniform Resource Locator).

▪ **Internet** : Vaste réseau informatique public de terminaux (ordinateurs, smartphones, etc.) qui communiquent les uns avec les autres - ces terminaux pouvant constituer leur propre réseau, on parle de réseau de réseaux. L'information circule à travers ces réseaux par le biais de protocoles standardisés (suite TCP/IP). Il existe plusieurs applications de l'internet, par exemple le Web, le courrier électronique, le partage de fichiers en peer-to-peer…

▪ **W3C** *(World Wide Web Consortium)* : Organisme à but non lucratif permettant de définir les standards pour les technologies liées au Web. La dernière révision majeure du HTML, HTML5, par exemple, a été réalisée par le W3C. Cette organisation se compose d'acteurs majeurs de l'industrie informatique, notamment des universités, ou des entreprises comme Google, Mozilla, Apple.

---

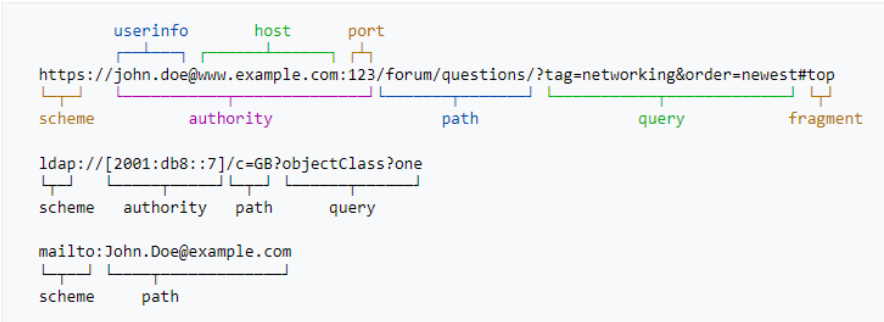
# PARTIE 2 :

▪ **URI** *(Uniform Resource Identifier)* : Courte séquence de caractères utilisés pour identifier une ressource présente sur le Web. Les URL et les URN (voir après) sont tous deux un type d'URI : tous les URL/URN sont des URI, mais l'inverse n'est pas nécessairement vrai.

Un URI contient jusqu'à cinq parties, dont seules les deux premières sont obligatoires :

`scheme :// authority path ? query # fragment`
→ Schéma : protocole utilisé
→ Autorité : identifie le domaine
→ Chemin : chemin d'accès à la ressource
→ Requête : représente une action de requête
→ Fragment : désigne un aspect partiel (réducteur) d'une ressource.

*Exemples :*

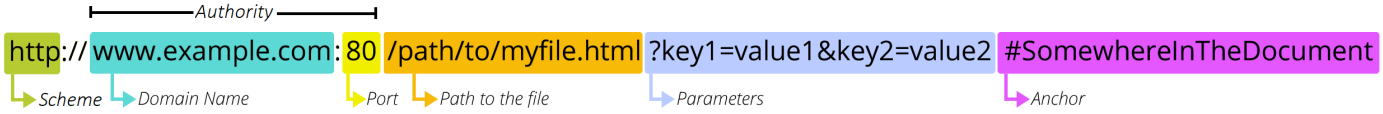



▪ **URL** *(Uniform Resource Locator)* : Adresse d'une ressource donnée sur le Web, utilisée par les navigateurs pour accéder à ladite-ressource :

`scheme :// host : port / path ? query # fragment`
→ Hébergeur : composé du sous-domaine, du domaine et du domaine de premier niveau, pointe à l'emplacement du serveur hébergeant la ressource.

→ Port : définit le type de service requis par l'utilisateur. Le port défaut pour les serveurs HTTP est le port 80, celui des serveurs HTTPS est le port 443. Il est commun de ne pas voir le port apparaître dans l'URL d'une page Web, puisque ces deux ports, standardisés, sont rarement indiqués par les navigateurs.

*Example:*




- **URN** *(Uniform Resource Name)* : Identifiant unique et statique d'une ressource du Web. Contrairement à une URL, un URN ne peut pas être changée et permet donc de tracker l'existence (ou la non-existence) d'une ressource spécifique indépendamment de sa location ou de son accessibilité par Internet. Format :

`urn:<NAMESPACE-IDENTIFIER>:<NAMESPACE-SPECIFIC-STRING>`

*Exemple :*

`urn:isbn:0-486-27557-4` mène à une édition spécifique de Roméo et Juliette.

---

# PARTIE 3 :

- **DNS** *(Domain Name System) :* service qui traduit les noms de domaine Internet en adresse IP. Puisque chaque matériel réseau connecté à Internet est accessible par son adresse IP et non par son nom, le navigateur doit procéder à une résolution de nom du domaine, pour traduire le nom de domaine en son IP, afin d'accéder au serveur du site Web en question. Le serveur DNS, présent sur un ordinateur sous forme d'IP, est généralement fourni et configuré automatiquement par le FAI (Fournisseur d'Accès Internet).

- **OSI** *(Open System Interconnexion)* : Norme de communication proposée par l'ISO (Organisation Internationale de Normalisation), divisée en sept couches.

- **TCP/IP** *(Transmission Control Protocol / Internet Protocol)* : ensemble des règles standardisées qui permettent à des ordinateurs de transférer des données sur un réseau, comme Internet. Dans cet ensemble, TCP et IP sont deux protocoles distincts. Le modèle TCP/IP décompose l'information en paquet (pour une meilleure fiabilité et gestion des erreurs) selon une procédure en quatre couches (accès réseau, internet, transport, application). Les données sont désassemblées en passant à travers ces quatre couches côté destinateur, et sont recomposées en y passant dans l'autre sens côté destinataire.

Ce protocole est souvent utilisé de pair avec d'autres protocoles Internet, par exemple HTTP (navigateurs/site web), FTP (transfert de fichier) ou SMTP (courriels).

- **HTTP[S]** *(HyperText Transfer Protocol [Secure])* : Protocole de transfert de documents HyperText / Hypermédias, par exemple écrits en HTML. Ce protocole est situé sur la couche la plus haute du Modèle OSI (application) et utilise typiquement le protocole TCP comme couche de transport. C'est un protocole de type client-serveur : la requête est initiée au serveur par le destinataire (navigateur ou API par exemple), qui lui renvoie ou non la ressource en question.

La différence entre HTTP et HTTPS réside dans la couche de chiffrement supplémentaire offerte par le HTTPS, sécurisant davantage les données transmises en maintenant leur confidentialité et leur intégrité. Le passage à "niveau" passe par l'acquisition et l'installation d'un certificat SSL/TLS auprès d'une autorité de certification reconnue.

- **FTP** *(File Transfer Protocol)* : Protocole de transfert de ficher. De la même manière que le HTTP, le protocole FTP fonctionne sur le modèle client-serveur, utilise le protocole TCP/IP et se trouve sur la plus haute couche du modèle OSI. Ce protocole a pour objectifs de permettre un partage efficace de fichiers entre machines distantes et de permettre une indépendance aux systèmes de fichier client/serveur.

Également comme le protocole HTTP, le protocole FTP peut être renforcé en FTPS, en passant par la même procédure.

- **Architecture Client-Serveur** : mode de communication entre plusieurs composants d'un réseau. Un composant de ce réseau est soit client, soit serveur. Le serveur est passif : il attend une requête du client et est supposé lui renvoyer l'objet de la requête, souvent appelé "service". Cette architecture est particulièrement commune et se retrouve par exemple dans les systèmes de messagerie : un client de messagerie envoie une requête au serveur de messagerie pour avoir accès et afficher les mails stockés sur le serveur.

---

# PARTIE 4 :

- **Navigateur Web** : Application logicielle utilisée pour accéder à des sites Web. Lorsque l'utilisateur envoie une requête pour recevoir une page Web, le navigateur récupère les fichiers depuis un serveur Web et en affiche le contenu. Un navigateur a au minimum la charge d'afficher la partie HTML de la page web, mais il est attendu aujourd'hui qu'il soit capable d'en afficher correctement le contenu selon les scripts (généralement suivant la norme ECMAScript, par exemple Javascript), et le style (fichier CSS (Cascading Style Sheets) par exemple).

- **Moteur de recherche** : système logiciel permettant de retourner les résultats d'une recherche de l'utilisateur sur le Web. Généralement, un moteur de recherche est disponible à l'accès à travers un navigateur Web, les plus connus étant Google, Bing, Duck Duck Go, etc. Un moteur de recherche est composé de trois parties :
  → Exploitation : le moteur de recherche explore systématiquement le Web en suivant récursivement tous les hyperliens qu'il trouve et récupère les ressources jugées intéressantes.
  → Indexation : Indexe chacune de ces pages Web en utilisant les termes qu'elle contient jugés pertinents et représentatifs de son contenu. Les termes en question sont jugés selon leur "poids", ou leur importance / prévalence dans le document.
  → Recherche : Restitue les résultats de la requête au moteur par l'utilisateur (seule partie visible pour l'utilisateur). De manière très simplifiée, le moteur de recherche affiche les résultats triés par l'algorithme par ordre de pertinence opposée : les résultats qui semblent le plus correspondre aux termes utilisés dans la requête sont retournés en premier.

---

# PARTIE 5 :

- **Référencement** : Le référencement regroupe plusieurs méthodes qui permettent d'indexer les contenus de sites web ou encore Internet de manière générale.

Il y a deux stratégies différentes de référencement qui se complètent :
→ référencement naturel/organique (SEO)
→ référencement payant/liens sponsorisés (SEA)

Le référencement va analyser et déterminer ce qui vaut le coup d'être oui ou non indexé.

- **SERP** *(Search Engine Result Page)* : Réponse à la requête d'un utilisateur sur un moteur de recherche. Les résultats affichés avec la SERP sont faits pour correspondre au mieux aux besoins de l'utilisateur, en affichant sur la page web du moteur de recherche les résultats les plus pertinents. On y retrouvera l'adresse URL, la description brève du contenu, un aperçu, la date de la dernière indexation…

- **SEM** *(Search Engine Marketing)* : Utilise différents outils, techniques et stratégies pour optimiser la visibilité des sites web et pages à travers les moteurs de recherche. Il utilise les deux types de référencement : SEO et SEA.

- **SEO** *(Search Engine Optimisation)* : type de référencement dit "organique", le moteur de recherche utilise un algorithme pour trouver quels sont les meilleurs sites web selon la requête. Contrairement au SEA il assure un positionnement à long terme dans les résultats de recherche, même s'il peut prendre plus de temps à avoir de la visibilité ou juste ne pas en avoir.

- **SEA** *(Search Engine Advertising)*: Référencement payant. Exemple : Google Ads.

Afin que son site ou sa page web soit affichée avec plus de visibilité, il y a un système d'achat aux enchères de mots clés. C'est une satisfaction immédiate, les résultats finissent généralement en haut de la page du moteur de recherche pour avoir une visibilité maximale. Cependant, vu que c'est un prix à payer, s'il n'y a plus de contribution financière, le site web peut finir par être moins référencé sur le long terme, contrairement au SEO.

- **SMO** *(Social Media Optimisation)* : Réunit un ensemble de méthodes qui ont pour but d'augmenter le nombre de visiteurs sur une page web.

C'est un référencement social, où le site web sera présenté dans les médias ou les réseaux sociaux. Il s'additionne au SEO pour améliorer la popularité du site web.

---

## PARTIE 6 :

- **Responsive Design** : Correspond à une variété de pratiques permettant aux pages web de modifier leur mise en page et apparence de s'adapter selon le type d'appareil, les différentes tailles d'écran, leur résolution… Il se différencie de type d'adaptation statique, liquide ou adaptatif pour les sites web.

C'est Ethan Marcotte qui en 2010 a nommé le concept de Responsive Design, réunissant trois techniques :
→ L'utilisation de grilles fluides
→ L'implémentation d'images fluides
→ Les media queries

---

## PARTIE 7 : → Schéma ←

- **Jeu de caractères** : Connu sous le nom de "charset" en anglais, c'est une notion essentielle afin de préserver une interopérabilité. C'est un système d'encodage qui permet aux machines de savoir comment reconnaître un caractère (lettres, nombres, ponctuation, espace…), que l'on utilise pour telle ou telle raison, comme le support de langues occidentales sur les ordinateurs.

A l'heure actuelle, l'Unicode (UTF-8) est le plus reconnu pour le support de langue universel.

- **Encodage** : Représentation de la correspondance entre les octets et le texte. → Schéma ←

---

# PARTIE 8 :

- **Library** : Ensemble de fonctions et de sous-procédés codés en un certain langage de programmation, supposé faciliter la création logicielle dans ce même langage et améliorer son efficacité. Une bibliothèque n'est pas exécutable, elle représente davantage un ensemble de fonctions à portée de main que l'on peut appeler individuellement, quand on et où l'on veut tant qu'elle est importée au début du code.

Par exemple, la bibliothèque standard de Go permet d'utiliser un certain nombre de fonctions utilitaires, par exemple celles du package Strings, qui offre la possibilité d'une manipulation simple des chaînes de caractères.

- **Framework** : Ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture. Un framework représente un squelette, un "blueprint" de programme que le développeur peut adapter selon ses besoins. Par exemple, Django (python), VueJS (JavaScript) et Laravel (PHP) sont tous trois des frameworks Web vastement utilisés aujourd'hui.

La différence entre une bibliothèque et un framework est que le développeur peut utiliser des parties de bibliothèques sans réellement avoir à suivre de structure pour coordonner le tout, la bibliothèque s'adapte au code ; de l'autre côté, un framework est une entité squelette plus ou moins rigide : le code doit s'y adapter pour que le tout fonctionne correctement.

- **API** *(Application Programming Interface)* : solution informatique qui permet à des applications de communiquer entre elles et de s'échanger mutuellement des services ou des données sans nécessairement avoir besoin d'accéder à l'application entière ou connaître les détails de sa mise en œuvre. De manière plus poussée, il s'agit d'un ensemble de classes, méthodes, fonctions et variables qui permettent d'accéder facilement aux services d'un logiciel.

Il existe différents types d'API : ouverte à tous les développeurs (ex. Twitter, Discord, API du gouvernement), fermée (utilisation réservée généralement à une entreprise et ses partenaires), et partenaire (sélectivement rendue disponible à des utilisateurs triés et autorisés).

Les API utilisent des protocoles et des architectures qui peuvent différer. Les architectures communes sont les API REST, RPC et SOAP.

---

# PARTIE 9 :

- **CMS** *(Content Management System)* : système de gestion de contenu. C'est un logiciel en ligne permettant à quelqu'un de publier, organiser, changer ou retirer divers types de contenus : articles, pages web, boutiques en ligne. Tout cela en n'ayant pas besoin de connaissances particulières en informatique.

Exemples : Wordpress, Joomla, Shopify.

Avantages du CMS :
→ Personnalisation avancée
→ SEO
→ Gain de temps
→ Interactivité
→ Responsivité

---

# PARTIE 10 :

- **Veille Technologique** : Consiste à se tenir informé sur les nouvelles technologies et leur commercialisation, les inventions des concurrents ainsi que les nouvelles technologies de manière générale, dans le but d'être le meilleur de son domaine. Activité particulièrement importante pour les entreprises, qui peuvent y obtenir un avantage sur la concurrence ou diminuer les coûts de production.

- **Flux RSS** *(Really Simple Syndication)* : Fichier XML disponible sur un site contenant des informations sur les derniers contenus publiés, leur date de publication, l'URL pour les consulter et soit un aperçu du contenu, soit le contenu complet en HTML.

- **Agrégateur de Flux RSS** : Application web, logiciel ou module intégré à un logiciel qui permet d'afficher, sur une page web personnalisable, les nouveautés des sites sur lesquels on fait de la veille informationnelle ou technologique.

- **Atom** *(Atom Syndication Format)* : Format de document fondé sur XML conçu pour la syndication, défini pour compléter et remplacer RSS jugé insuffisant, notamment sur les plans de l'internationalisation et la modularité du flux. Par exemple, Atom tend à préciser le type de contenu des balises HTML, là où RSS 2.0 ne le fait pas.

---

# PARTIE 11 :

- **Editeur de code** : éditeur de texte, avec la simple différence que certaines fonctionnalités sont rajoutées pour simplifier l'écriture et la lecture du code par un programmeur. Ces fonctionnalités peuvent être le surlignage de la syntaxe, la fermeture automatique des parenthèses, crochets et accolades, et le remplissage automatique.

*Exemples :* Atom, Sublime Text, Brackets, VS Code, Vim.

- **IDE** : Plus complet qu'un éditeur de code, un IDE possède, en plus de l'éditeur lui-même, un compilateur ou un interpréteur de code, et un débugger.

*Exemples* : Eclipse, IntelliJ IDEA, Visual Studio.

---

# PARTIE 12 :

- **Langage de programmation** : Langage informatique utilisé par un programmeur pour « communiquer » avec un ordinateur et lui faire passer des instructions, réunies en un programme supposé réaliser une tâche donnée.

Un langage de programmation peut être dit de bas-niveau, ou de haut-niveau : un langage bas-niveau est un langage très peu abstrait, dit « proche » de la machine et donc du binaire. Ils permettent de pleinement interagir avec la partie matérielle de l'ordinateur (mémoire, processeur, etc.). Un langage de haut-niveau est nettement moins dépendant du matériel : il crée généralement des programmes portables et non liés à un ordinateur ou à une puce ; de manière générale, le programmeur n'a pas ou peu besoin de s'occuper des ressources matérielles de l'ordinateur. Ces niveaux ne sont pas stricts : si l'Assembleur est considéré de bas-niveau, et Javascript de haut-niveau par exemple, certains langages comme le C++ se situent d'une certaine manière au « milieu » du spectre.

Enfin, un langage de programmation peut être compilé ou interprété : un langage compilé nécessitera de passer par un compilateur pour fonctionner (ex. C), tandis qu'un langage interprété pourra fonctionner directement sur n'importe quelle machine (ex. Python).

- **Orienté Objet** : La programmation orientée objet est un paradigme se basant sur la création d'objet et leur interaction entre eux. Un objet est composé de données, les attributs, et de fonction, les méthodes, afin de manipuler ces données. Une voiture ou une personne peuvent être interprétés comme des objets, la voiture aurait l'attribut « kilométrage », qui augmenterait lorsque sa méthode « rouler » serait appelée.

*Exemples :* Python, C++, Java, Javascript.

- **Procédural** : La programmation procédurale est un paradigme qui se fonde sur des appels procéduraux. Une procédure est un bout de code réutilisable à différents endroits de celui-ci. La possibilité de réutiliser une procédure permet de réduire considérablement la taille du code et améliorer la compréhension de son exécution. De plus, la maintenabilité est grandement améliorée.

La programmation procédurale est soumise aux effets de bord en théorie, mais la modularité palie cela, rendant les variables fermées à certains modules. Grâce à cela, on peut imagine le principe de bibliothèques, permettant d'importer du code étranger, sous seule condition que le code se base uniquement sur des variables passées en paramètres.

*Exemples* : C, Pascal, Basic, COBOL.

- **Séquentiel** : La programmation séquentielle est un paradigme de programmation. L'exécution d'un programme séquentiel se base toujours sur les mêmes instructions. Ce paradigme s'oppose directement à celui de la programmation événementielle.

*Exemple* : Calcul de paie.

- **Evènementiel** : La programmation événementielle est un paradigme de programmation. Il se base sur des événements et change les instructions du code selon certains états de variables ou d'événements (comme le changement de valeur d'une variable, un clic de souris, une saisie clavier). La majorité des langages de haut niveau permettent la programmation événementielle.

*Exemple* : Algorithme de création de la gamme de Pythagore.

---

# PARTIE 13 :

- **HTML** : permet de définir et structurer le contenu sur le web. "HyperText" fait référence aux liens qui connectent des pages web entre elles, dans le même site ou d'un site à un autre. Le "Markup" (balises) est utilisé pour annoter des textes, des images... pour l'affichage dans le navigateur web.

- **CSS** : s'occupe du côté apparence/esthétique/présentation/mise en forme d'une page web. Il s'associe au HTML pour le styliser sélectivement.

- **Javascript** : Le JavaScript est un langage utilisant différents paradigmes de programmation. C'est un langage script léger et orienté objet, on l'associe au HTML et au CSS pour le côté fonctionnel sur les pages web avec ses mécanismes.