

Veille internet:

1990 : le web 1.0 : constitué de pages web et d'hyperliens, les sites sont 'statiques'.

2000 : le web 2.0 : ou web social, participatif. l'ère de la connexion permanente entre les individus du monde entier (réseaux sociaux, plateforme vidéo en ligne, blogs, coopérative, wiki,...).

2010 : le web 3.0 : internet des objets, web sémantique, cloud, objets connectés, etc.

Le Web 3.0 donne accès à une masse d'informations qui sont réunies par les entreprises. Ces données Web sont collectées, stockées et analysées grâce à l'intelligence artificielle (IA).

<https://www.younicom.fr/web-4-0-changements-differences-web-3-0/>

2020-2030 : le Web 4.0 'L'Internet de l'ADN'! Certaines technologies présentes dans nos smartphones comme les enceintes connectées. Siri, Google Assistant, ChatGPT sont des prémices de la quatrième génération du web <https://www.nouvelobs.com/rue89/rue89-internet/20150307.RUE8170/le-web-4-0-sera-celui-du-genome-et-y-a-de-quoi-flipper.html>

La quatrième génération d'Internet sera ultra-intelligente, elle s'inscrira dans un contexte dans lequel l'IA, la réalité augmentée, l'apprentissage automatique et l'Internet des Objets (IoT) seront importantes et feront partie de notre quotidien.

Comment se caractérise le web 3.0 ?

L'expression Web 3.0 est utilisée par certains spécialistes évoquant le futur proche de l'actuel web 2.0. Selon ces analystes, la prochaine étape d'évolution du World Wide Web pourrait désigner soit l'Internet des objets, soit l'émergence du web sémantique. Pour certaines personnes, le web 3.0 est avant tout marqué par la personnalisation, rendue possible par le stockage et l'utilisation des données.

Depuis son apparition dans les années 1990, le web est passé de l'ère des pages internet reliées par des liens hypertextes à l'ère de la connexion permanente entre les individus du monde entier dans les années 2000. Les **réseaux sociaux** (Facebook, Twitter, Instagram), les plateformes de vidéos en ligne (YouTube), les blogs, la généralisation du smartphone connecté et la **technologie coopérative wiki** ont inspiré l'appellation web 2.0 à Dale Dougherty, cadre de l'entreprise O'Reilly Media en 2003.

Le web 3.0 dans la réalité

Depuis le début des années 2010, l'expression **web 3.0** est régulièrement évoquée, mais elle fait débat. Certains considèrent que c'est juste un terme marketing pour faire le buzz, d'autres pensent qu'il s'agit bien d'une nouvelle étape distincte des précédentes. Pour l'instant, c'est une expression qui est ancrée dans l'avenir, non dans le présent. Sur le plan technique, cependant, on parle de déjà d'**application 3.0** ou de **solution 3.0**, qui doit répondre à certains critères :

- **indépendante** : elle doit être également utilisable sur tous les supports (type d'écran, sortie d'imprimante etc.)
- **universelle** : elle ne doit être sous l'emprise d'aucun système d'exploitation (marque, fabricant de logiciel ou de matériel)
- **accessible à tous** en conformité avec les directives du W3C et ouvert à diverses bases de données.

Site statique, site dynamique, définitions :

Initialement, aux prémices du web, les sites étaient dits "statiques". C'est-à-dire que, pour chaque page du site qu'on voulait créer, on créait un document HTML. Il fallait donc un document HTML pour la page d'accueil, un autre pour la page 'à propos', etc. Cela devenait donc vite fastidieux quand il y avait des dizaines de pages à faire. En plus s'il y avait une modification à faire qui était commune à toutes les pages, il fallait reprendre toutes les pages, et les remettre 'en ligne' chez un hébergeur.

Les langages serveur

Comme les demandes ont évolué, on a eu besoin de sites beaucoup plus sophistiqués avec beaucoup plus de contenus, il y a eu besoin de développer des technologies et parmi ces technologies il y a eu des langages serveur (comme le PHP, le C#) qui ont permis de faciliter la conception de très gros sites. Facilité à l'échelle des développeurs, pas forcément à l'échelle de l'utilisateur final puisque le PHP et le C# sont des vrais langages de programmation côté 'serveur', qui demandent d'avoir des compétences en développement.

Le site non dynamique

Reprenons le cas d'un site non dynamique, lorsque vous tapez dans la barre d'adresse URL ou que vous cliquez sur un lien, vous demandez en fait à un serveur de vous afficher un document HTML qui est stocké à un endroit précis sur ce serveur. Le serveur obéit et vous renvoie ce fameux document que vous lui avez demandé. On est bien d'accord, le document existe et est enregistré sur le serveur.

Le site dynamique

Dans un site dynamique, le processus est différent, le fichier HTML n'existe pas encore. Il va être généré en fonction de paramètres que vous allez lui envoyer.

- Vous allez les lui envoyer soit de façon consciente, en remplissant un formulaire et en cliquant sur 'envoyer'. Par exemple, un formulaire de demande de devis, vous allez donner la quantité de produits désirée, le type de produit désiré et puis le serveur va recevoir ces paramètres et va traiter la demande.
- Vous pouvez aussi le faire de façon inconsciente, c'est-à-dire en cliquant sur un lien. Ce lien a des paramètres qui se mettent dans l'URL qui va prendre la forme de :
« mon-site.com?ID=279&autoplay=1 », le script a été fait pour comprendre ces paramètres, évidemment on n'envoie pas n'importe quels paramètres à n'importe quel script, et en fonction de ces paramètres il va créer un fichier HTML.

Cela se passe comme ceci, vous envoyez une demande avec des paramètres, consciemment ou inconsciemment, le serveur comprend ces paramètres, génère un fichier HTML puis vous le réadresse. Il s'affiche alors sur votre écran.

Donc pour qu'un site dynamique puisse exister on a besoin d'un langage appelé "langage serveur". Et les 'langages serveur' par excellence, sont les langages C# et PHP.

Le stockage de données

Le C# et le PHP ont besoin, à un moment donné, de stocker de l'information. Admettons, que vous ayez un utilisateur qui se connecte à votre site, il faudra lui permettre de s'authentifier, en créant une fonction pour qu'il puisse se créer un compte sur votre site, il faut stocker les informations de l'utilisateur quelque part. Dans ce cas de figure, on les enregistre sur une base de données. C'est pour ça que, pour générer un fichier, le C# et PHP fonctionnent de pair avec une base de données qui sera pour le C# du SQL server et pour le PHP, du MySQL.

***Un site statique** montre des informations telles que l'a conçu leur auteur, sans jamais changer en fonction de l'utilisateur qui visionne le site. Tous les utilisateurs voient le même contenu.

On entend par page statique, non pas une page sans mouvements ou sans animations, mais une page visible telle qu'elle a été conçue.

Ces pages peuvent présenter toute forme de contenu, animations images, musique, vidéo etc... mais elles sont toujours présentées de la même façon. Elles ne changent pas et c'est en ce sens qu'elles sont statiques.

***Un site dynamique** varie son contenu en fonction des données apportées ou demandées par l'utilisateur: classement des produits sur un site de vente, création d'un compte personnel sur un site social, ...Il nécessite une Base de données, qui sera interrogée par les langages de programmation Backend !

En opposition aux pages statiques, les pages dynamiques permettent de présenter les informations de différentes manières selon l'interaction avec le visiteur.

Les pages sont alors construites "à la volée" grâce à une programmation conçue par le web developer. Le contenu est issu d'une base de données en fonction de critères établis par l'internaute puis mis en page en temps réel.

C'est le cas par exemple d'un site e-commerce: présentation des articles par thèmes, couleurs, prix etc... C'est également le cas des blogs et des forums où les visiteurs peuvent participer au contenu du site. C'est aussi le cas d'un système de mises à jour, ou encore pour l'authentification/sécurité.

Comment choisir si un site nécessite du 'statique' ou du 'dynamique'?

Tout d'abord il convient de déterminer le rôle de la page.

Tout système d'interaction avec le visiteur nécessitera une partie dynamique.

Pour un système de mise à jour, il conviendra de réfléchir aux fréquences des mises à jour. Si le contenu ne change peu ou qu'une fois à l'année, il n'est pas nécessaire de passer par des pages dynamiques. Une intervention ponctuelle du webmaster peut très bien convenir.

Quels sont les avantages et les inconvénients des deux méthodes?

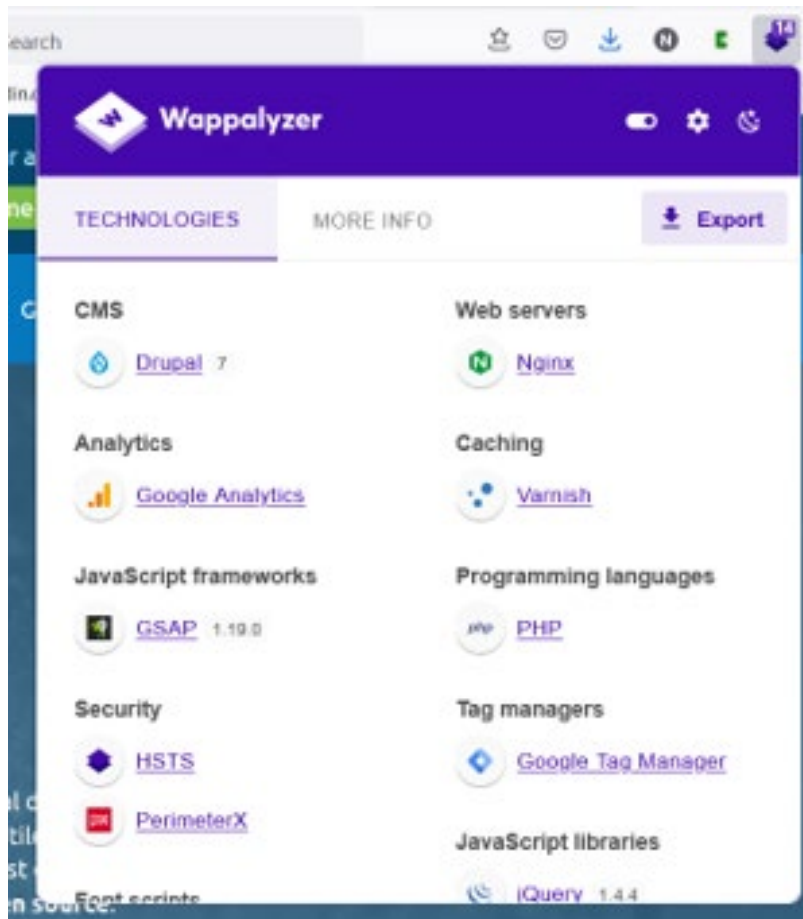
L'utilisation des pages dynamiques est plus souple pour l'évolution du site mais nécessite plus de temps de travail et des compétences poussées pour la programmation.

En opposition, les pages statiques sont plus "figées" mais plus simples à réaliser.

Conclusion

Pour résumer, le choix du type de site que l'on va créer dépend principalement de notre besoin, il est totalement inutile de créer ou développer un site web dynamique quand on n'a besoin que de pages simples qui ne changeront jamais (par exemple, un site informationnel d'une entreprise, où le site ne joue que le rôle de carte visite).

Truc & astuce : installer **WAPPALYZER** sur le pc, vous donne pour chaque site les logiciels, langage de programmation, cms et autres méthodes utilisés pour créer un site. Il suffit de voir dans la barre de navigation, à la suite de l'url, les logos et langages utilisés. <https://www.wappalyzer.com/>
Téléchargement pour Windows : <https://wappalyzer.fr.uptodown.com/windows/telecharger>
Par exemple , visitez le site du CMS Drupal : <https://www.drupal.org/> en haut à droite , cliquez sur Wappalyzer et il vous indique tous les logiciels, langages utilisés pour créer le site ou l'application.



FUTURE WEB APPLICATION DEVELOPER

Voir les concepts: web 1.0 VS web2.0 , site dynamique VS site statique, applications web

L'objectif est d'analyser au moins un site statique, et un site dynamique.

D'en comprendre la définition de chacun, de les analyser, de les comparer, d'en comprendre les différences, avantage/désavantage, les langages utilisés, les applications web proposées,...et à travers cela mener l'enquête sur le métier, les tâches du 'web developer' !

>Définir et analyser les applications dans le site dynamique , proposez des solutions pour améliorer les sites statiques et dynamiques.

>Lister les différents 'types' d'applications et fonctionnalités/web services, les langages/ logiciels/framework/outils/serveurs utilisés, ainsi que leurs utilités.

> Allez voir ce qu'est le métier du web développement, par exemple en allant voir des Fiches métier 'Web developer' 'Fullstack', 'frontend & Backend' et la liste et définition des applications pour le web.

D'après wikipédia :

Une application Web est un logiciel applicatif qui imite un site web et affiche son interface homme-machine dans un navigateur web. Le logiciel est hébergé par un serveur web transformé. Il est exécuté de façon partagée par les ordinateurs serveur et client (l'utilisateur). Cette technologie est apparue dans les années 1990.

En informatique, une **application web** (aussi appelée **web application**, de l'anglais) est une application manipulable directement en ligne grâce à un navigateur web et qui ne nécessite donc pas d'installation sur les machines clientes, contrairement aux applications mobiles¹. De la même manière que les sites web, une application web est généralement installée sur un serveur et se manipule en actionnant des widgets à l'aide d'un navigateur web, via un réseau informatique (Internet, intranet, réseau local, etc.). Exemples : Des messageries web, les systèmes de gestion de contenu, les wikis et les blogs sont des applications web.

Les moteurs de recherches, les logiciels de commerce électronique, les jeux en ligne, les logiciels de forum, les agrégateurs peuvent être sous forme d'application web.

Des appareils réseau tels que les routeurs sont parfois équipés d'une application web dans leur micrologiciel².

Les applications web font partie de l'évolution des usages et de la technologie du Web appelée Web 2.0.

Utilisations :

La technologie des applications web permet de nombreux usages. Les usages populaires sont : l'authentification, les moteurs de recherche, le webmail, le commerce électronique et les jeux en ligne.

>Un moteur de recherche est une application web qui recherche des documents.

>Un webmail est une application web pour recevoir et envoyer du courrier électronique.

>Un système de gestion de contenu est une application web qui présente des documents. La présentation des documents est similaire à celle d'un site web, cependant les documents sont générés par le logiciel lors de chaque demande. Le système de gestion de contenu effectue les traitements nécessaires à la mise en forme et la présentation des documents.

>Un weblog est un système de gestion de contenu où des éléments de contenu sont présentés dans l'ordre chronologique de leur date de création.

>Un wiki (de l'hawaïen *wikiwiki* qui signifie *vite*) est un système de gestion de contenu qui vise à simplifier la création *collaborative* des documents. Il autorise plusieurs personnes à effectuer des modifications simultanées, et est équipé d'espaces de discussion³.

>Un site web marchand est un système de gestion de contenu, où le contenu sont des annonces concernant des produits. Il est utilisé pour la vente par correspondance. Les visites et les opérations d'achat sont enregistrées à des fins de marketing. Les sites web marchands sont utilisés aussi bien pour la vente des produits d'une société que pour des ventes entre particuliers ou des ventes aux enchères et nécessite un paiement en ligne.

>Un jeu par navigateur est un jeu vidéo réalisé sous la forme d'une application web.

>Un logiciel de forum permet des discussions ouvertes entre des utilisateurs : un utilisateur écrit un message, et ce message peut être lu par tous les autres utilisateurs. Les logiciels de forums sont parfois réalisés sous forme d'application web.

>La messagerie instantanée (anglais *chat*) permet l'échange instantané de messages texte entre différents utilisateurs. Les logiciels de messagerie instantanée sont parfois réalisés sous forme d'application web. Les messages peuvent être transmis à un autre utilisateur du logiciel, ou un téléphone portable via le short message service (abrégié SMS).

>Google Maps est une application web qui permet de consulter des cartes géographiques du monde entier.

>Facebook est une application web qui permet à chaque utilisateur de se constituer un réseau social (amis, associés, personnes qui partagent les mêmes centres d'intérêt).
etc.

Un.e développeur.euse full stack est un.e informaticien.nne capable de réaliser des tâches à n'importe quel niveau technique de la pile des différentes couches qui constituent une application informatique.

De façon plus précise, il.elle est capable de :

>configurer l'infrastructure matérielle et le système d'exploitation ainsi que les dépendances entre les différents composants logiciels à utiliser ;

>concevoir, manipuler et interroger des bases de données ;

>concevoir le code de 'back-end' et les API d'accès à ces données sous forme de webservices, en utilisant des langages tels que c#, Python, PHP ;

>concevoir le code de 'front-end' qui est exécuté sur la plate-forme de l'utilisateur, le plus souvent un smartphone ou un navigateur Web, dans des langages tels que Javascript et ses bibliothèques,

HTML, CSS ;

>être capable de gérer et planifier un projet avec des méthodes de développement agiles telles que Scrum ou Kanban, pouvoir interagir avec le client et ses besoins, et documenter l'ensemble des parties du projet.

Un.e développeur full stack est ainsi familier.ère avec chacune de ces couches, même s'il.elle a souvent une affinité ou des compétences plus étendues dans l'une d'elles. Cette connaissance large lui permet de travailler sur un projet de A à Z sans avoir à recourir à d'autres développeurs pour les petits projets. Pour les gros projets, cela rend la vie plus facile aux autres développeurs du même projet, car il sait comment les différentes parties communiquent et s'articulent. Son caractère pluridisciplinaire permet également de faciliter les remplacements ponctuels ou durables.¹

Dernier petit rappel ☺ à mémoriser à vie !

Site statique vs site dynamique

Un **site statique** est généralement un ensemble de pages «pré formées» (fichiers HTML) stockées sur un serveur Web. Aucun traitement n'est effectué du côté du serveur, qui ne fait qu'envoyer un ensemble de fichiers (HTML, images) au navigateur de l'utilisateur final.

Le contenu d'un site dit statique est codé en dur dans les pages HTML, où l'on pourra trouver pêle-mêle du code HTML, du javascript, du flash... Autant dire que la mise à jour s'annonce délicate voire périlleuse pour qui n'a pas quelques notions de base en HTML. Certes, il existe des éditeurs HTML, il n'en reste pas moins que la maintenance d'un tel site est coûteuse en temps et peu flexible sur de gros sites; pour ajouter un lien de navigation, il faudra par exemple éditer plusieurs fichiers HTML du site. Pour des sites de quelques dizaines de page, le statique conviendra tout à fait.

Un **site dynamique** est généralement associé au couple PHP/MySQL, ou c#/SQL. Le contenu du site est désormais stocké dans une base de données (MySQL par exemple), il est donc dissocié de la «présentation», c'est à dire la partie du code en charge de la restitution des données dans le navigateur de l'utilisateur final.

Le serveur Web reçoit une requête HTTP émise par le visiteur, il interroge la base de données puis construit une page de résultat qu'il restitue au visiteur via le navigateur.

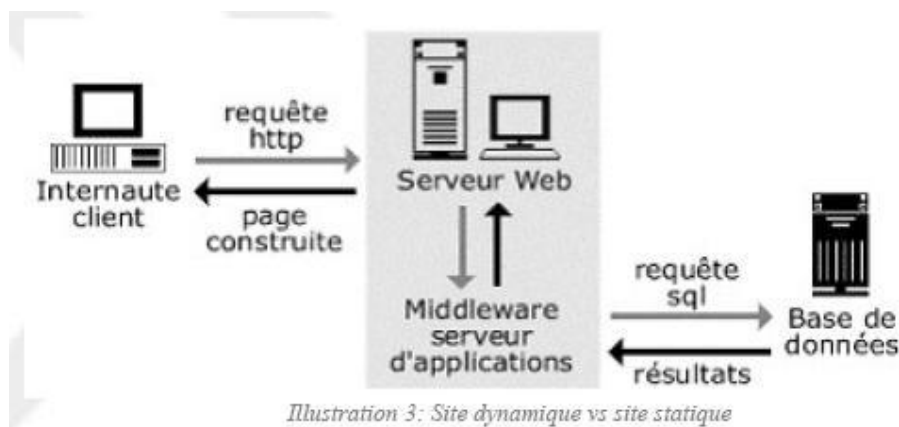


Illustration 3: Site dynamique vs site statique

La page restituée pourra être différente en fonction de l'utilisateur connecté sur le site (accès autorisé ou pas à certaines parties du site par exemple). Le contenu du site pourra être mis à jour dynamiquement; par exemple, pour une page listant les articles de votre site, vous n'aurez à pas ajouter manuellement un lien vers un article fraîchement créé. Tout ce traitement est effectué par le serveur Web.

I-D. PHP/MySQL

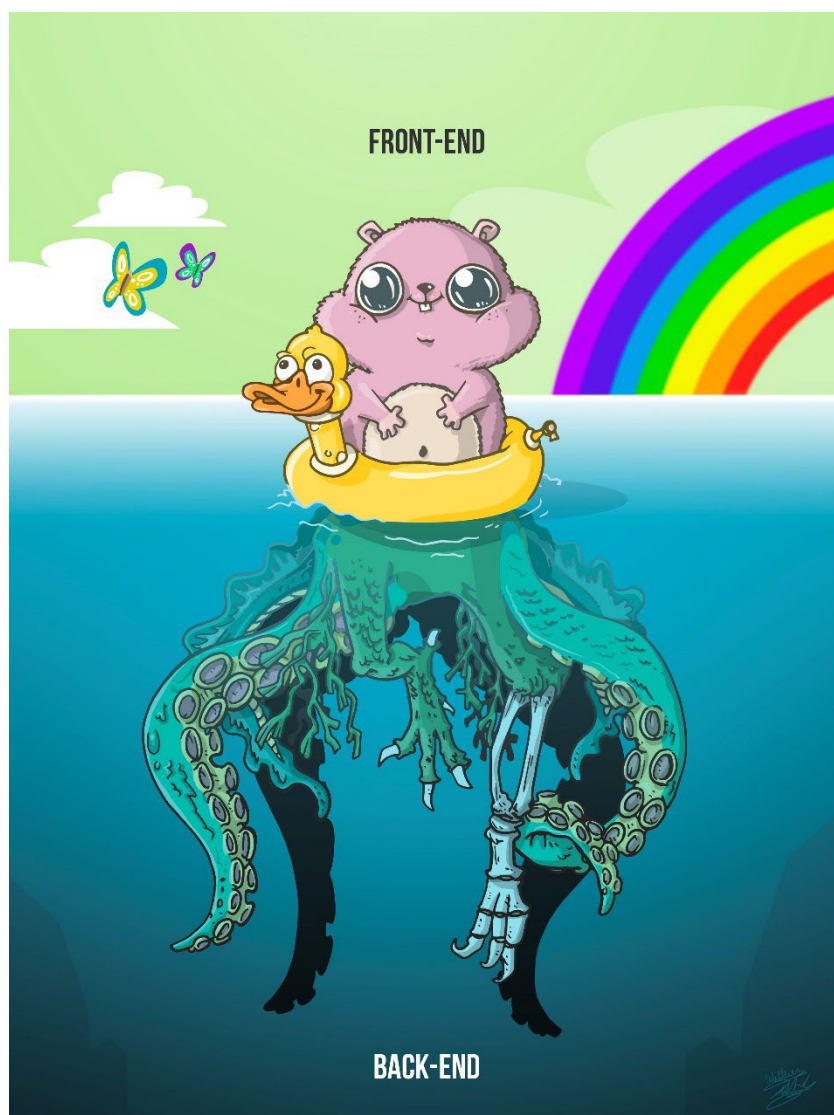
Prenons une simple tableau ; en première ligne on trouve ce qu'on appelle des champs (Nom, Prénom, Email...), les lignes suivantes s'appellent des enregistrements, le tout formant une table. Une base de données MySQL est une table ou plus souvent un ensemble de tables qui permettent de stocker des informations (ici une liste d'utilisateurs mais il peut s'agir de n'importe quel autre type d'information).

	A	B	C	D	E
1	Nom	Prénom	Email	Date de naissance	Pays
2	LOTTER	Quentin	qlotter@calembour.com	01/04/45	France
3	PARIZZI	Cecilia	cparizzi@calembour.com	13/02/75	Italie
4	EUVAL	Hector	heuval@calembour.com	25/09/65	Belgique

Microsoft SQL et MySQL fonctionnent sur le même principe à cette différence près que MySQL est gratuit. Le langage de programmation qui va permettre d'aller interroger cette table (on dit aussi effectuer une requête) ou modifier (en ajout, édition ou suppression) son contenu, via des morceaux de scripts intégrés à des pages web. Toute cette cuisine se fait directement sur le serveur qui héberge votre site.

Enfin, **scrutez les URLs pour reconnaître les langages utilisés**, par ex. belfius.be/connection.aspx ;

Site statique 'vitrine' d'Interface3.be (site statique) <https://www.interface3.be/> ; Site dynamique du réseau des anciennes d'Interface3 : <http://w4n.interface3.be/fr> -testez l'url:
http://w4n.interface3.be/fr?check_logged_in=1



WEB « 1.0 »



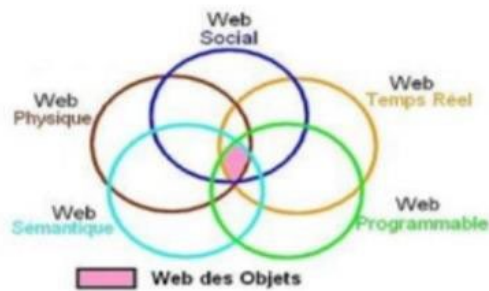
Relier les ordinateurs
1989, Tim Berners-Lee

WEB 2.0



Relier les hommes
2005, youtube

WEB 3.0



Relier les objets entre eux
et avec les hommes

