

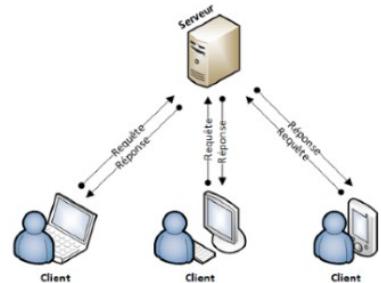
## Comprendre le web

### 1) Internet vs Web

- **Internet** : infrastructure mondiale qui permet aux ordinateurs de communiquer entre eux en suivant des règles de communication bien définies.
  - Exemples de services utilisant Internet mais pas le Web : email, VPN, messagerie instantanée (ex. WhatsApp).
- **Web** : service utilisant Internet pour afficher des pages et des applications.
  - Exemples de services utilisant le Web : Wikipédia, YouTube, Amazon.

### 2) Les acteurs du Web

- **Client** : programme qui affiche les pages web (navigateur : Chrome, Firefox).
- **Serveur** : Ordinateur qui héberge les pages qu'il envoie au client.



### 3) Le fonctionnement du Web

Comprendre le fonctionnement du web revient à comprendre l'interaction client-serveur.

- **L'utilisateur** tape une adresse (URL) dans son navigateur  
ex : <https://www.wikipedia.com>
- Le **navigateur** envoie une requête HTTP au serveur web.
- Le **serveur** renvoie une réponse HTTP contenant la page web écrite en HTML, CSS, JS
- Le **navigateur**, capable de lire le HTML, CSS et JS, affiche le contenu de la page à l'écran

HTTP : est le protocole utilisé pour permettre au navigateur et au serveur de communiquer ensemble. HTTPS est un protocole HTTP sécurisé.

- Taper <https://www.wikipedia.com> et identifier la réponse et les fichiers HTML, css et js envoyés.
- Sur la page, Faire un clic droit, Inspecter (ou F12).
  - Cliquer sur l'onglet Network (ou Réseau).
  - Recharger la page (F5) et observer.
  - Vous pouvez sauvegarder ces fichiers sur votre ordinateur : clic droit, Enregistrer sous.

### 4) Les langages du web

Les langages interprétés par le client (le navigateur)

- **HTML (HyperText Markup Language)** : définit la structure et le contenu d'une page web.
- **CSS (Cascading Style Sheets)** : contrôle la mise en forme et le style visuel d'une page web.
- **JavaScript (JS)** : ajoute de l'interactivité et du dynamisme aux pages web.



Il existe une multitude de langages interprétés par le serveur : **PHP, Python, Java**, etc.



## 5) Comprendre les URLs

Tapez L'URL : [https://fr.wikipedia.org/wiki/Uniform\\_Resource\\_Locator](https://fr.wikipedia.org/wiki/Uniform_Resource_Locator) et analysez-la.

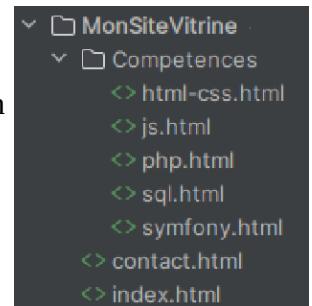


## 6) Sites web statiques vs dynamiques

**Statique** : Le serveur renvoie au client une page HTML enregistrée dans un dossier du serveur.

On accède à :

- la page d'accueil (index.html) : [www.monsitevitrine.fr](http://www.monsitevitrine.fr)
- la page contact : [www.monsitevitrine.fr/contact.html](http://www.monsitevitrine.fr/contact.html)
- la page PHP : [www.monsitevitrine.fr/competences/php.html](http://www.monsitevitrine.fr/competences/php.html)



**Dynamique** : les pages HTML sont fabriquées par le serveur et renvoyées au navigateur.

→ Exemple d'un client qui demande à sa banque la page affichant le solde de son compte.



1 : le client (le navigateur de M. Dupont) demande la page affichant le solde de son compte : [www.mabanque.fr/monsolde](http://www.mabanque.fr/monsolde)

2 : le serveur reçoit la demande, identifie l'utilisateur (M. Dupont).

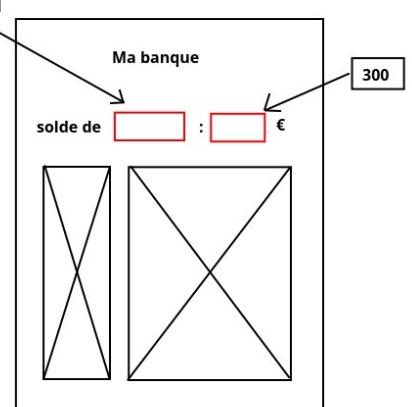
Le serveur demande à la base de données (qui contient le solde de tous ses clients) de lui envoyer uniquement le solde de M. Dupont.

3 : la base de données renvoie l'information « 300 » au serveur.

Le serveur « fabrique » une page HTML en y injectant les 300 € dans un modèle (template) de page HTML.

4 : le serveur renvoie la page HTML au client.

identifiant	client	solde
1	Dupont	300
2	Hamoui	110
3	Martin	25



:

C'est au **développeur frontend** de créer les templates (HTML, CSS et JS).

Alors que c'est au **développeur back-end** de créer des programmes qui fabriquent des pages HTML à partir du template. Il utilise pour cela un langage back-end comme PHP, Python, Java etc.

## **7) Les métiers du web**

- **Développeur web** : Personne qui crée des sites et applications web.
  - **Dev frontend** : implémente l'interface utilisateur en utilisant HTML, CSS, JavaScript en s'assurant que l'application est accessible et performante côté client.
  - **Dev backend** : conçoit et développe la logique côté serveur.
  - **Dev fullstack** : dev frontend & dev backend
- **DevOps** : gère l'infrastructure et automatisé les déploiements.
- **SysAdmin** : installe, configure et maintient les serveurs et le réseau.