

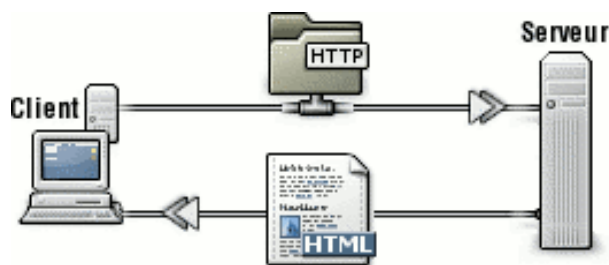
Concepts de base : sites statiques et dynamiques

Allez sur www.lesoir.be, un site "dynamique". Le contenu affiché est du HTML. Si on attend quelques minutes et on recharge la page, les contenus HTML affichés seront probablement différents (ex: section "Le fil info"). Le contenu de la page **est du HTML généré par un serveur au moment de charger la page**, il s'agit d'un site dynamique.

A priori, on ne sait pas si ce contenu HTML a été généré avec PHP, ASP, Ruby, Python... mais dans tous les cas on est sûr que le contenu qu'on visualise a été créé en utilisant un de ces langages de serveur.

1. Pages statiques

L'installation basique d'un serveur web sur un ordinateur (ex: Apache) permet à un navigateur (Client) de lui demander une page HTML.



Le processus est le suivant :

1. Le client (en utilisant un navigateur) **demande** la page HTML (ou PHP) au serveur (apache)
Ex: taper une adresse dans la barre de navigation (ou on navigue avec le navigateur)
2. Le serveur **cherche** la page HTML dans son disque (ex: dossier xampp\htdocs)
3. Le serveur **envoie une copie de la page HTML au client**
4. Le **navigateur** du client **interprète le HTML** de la page et **affiche** le résultat dans l'écran

Important : dans notre cas, l'ordinateur qui joue le rôle de client est le même qui joue le rôle de serveur !

Pour démarrer ce processus il suffira d'écrire l'adresse de la page dans la barre d'adresses du navigateur du client (<http://nomserveur/dossier/nomPage.html>).

Pour réaliser des exemples pratiques vous devez avoir un serveur web installé sur votre ordinateur. Allez sur la section **Téléchargement et installation de XAMPP** plus bas dans ce document avant de commencer l'exemple.

Exemple pratique :

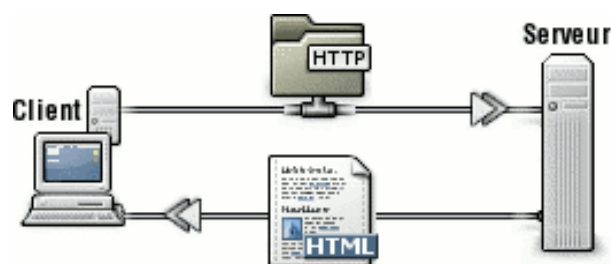
1. Créez le dossier C:\xampp\htdocs\monSiteStatique et une page maPage1.html contenant le code suivant :

```
<html>
<body>
Bonjour je suis une page statique
</body>
</html>
```

2. Tapez l'adresse dans le navigateur :

<http://localhost/monSiteStatique/maPage1.html>

Le serveur Apache cherchera cette page html dans votre disque dur pour l'envoyer ensuite au client (votre navigateur) :



Important : Au lieu du nom du serveur on peut utiliser son adresse IP

<http://127.0.0.1/monSiteStatique/maPage1.html>

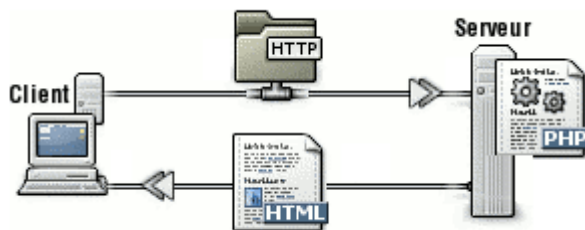
127.0.0.1 est l'adresse qu'identifie notre propre ordinateur. Elle est définie dans le fichier **hosts** de notre système d'exploitation.

2. Pages dynamiques (sans BD)

A différence des pages statiques, **une page dynamique peut contenir tant du langage serveur (ex: PHP, ASP...) comme du HTML ou d'autres langages de client (ex: Javascript, VB Script, CSS).**

Une page php doit avoir l'extension **.php** (pas .html!), ce qui indique au serveur qu'il doit lancer le **module PHP** (une sorte de "plugin" du serveur qui est en soi l'interprète) pour interpréter son contenu PHP.

Rappelez-vous que dans le cas d'une page statique (html) le serveur doit uniquement envoyer une copie de la page html qui se trouve dans son disque dur. Dans le cas des pages dynamiques il faut rajouter quelques opérations :



1. Le client demande la page **PHP** (pas HTML!) au serveur
2. Le serveur cherche la page **PHP** dans son disque
3. **Le module de PHP parcourt la page.** S'il trouve de balises de php, il interprète **le code et génère le code HTML correspondant**
4. Le serveur envoie la page HTML générée au client
5. Le client affiche le HTML reçu dans le navigateur

Exemple pratique (installez XAMPP d'abord)

1. Le client demande une page .php qui se trouve dans le disque dur du serveur

Ex: `http://localhost/monsiteDynamique/maPage1.php`

2. Le serveur cherche la page PHP dans son disque, contenant:

```
<html>
<body>
Bonjour je suis une page dynamique.
<br>

<?php
    echo "L'heure actuelle est: <br>";
    echo date("h i s");
?>

</body>
</html>
```

3. Le module PHP génère le code HTML correspondant au code PHP

Le module PHP cherchera et interprètera le contenu en langage PHP, qui se trouve toujours entre les **balises PHP** `<?php` et `?>`

```
<?php
    echo "L'heure actuelle est: <br>";
    echo date("h i s");
?>
```

Le code PHP peut accomplir un éventail de tâches différentes dans le serveur: générer du code HTML, accéder à une base de données, créer un fichier PDF, écrire dans le disque dur du serveur... il s'agit d'un langage complet de programmation et il n'y a pas de limites.

Concrètement, le code PHP de cette page sert uniquement à créer le code HTML pour afficher un message de texte et l'heure actuelle du serveur ("fonction" "date" qu'on étudiera plus tard). Le HTML généré à partir du code précédant est:

L'heure actuelle est:
03 16 22pm

Une fois le navigateur reçoit la page HTML, il affiche:

```
<html>
<body>
Bonjour je suis une page dynamique.
<br>
L'heure actuelle est: <br>03 16 22pm
</body>
</html>
```

Observez qu'une partie de cette page provient du code HTML originale et une autre partie provient du HTML généré à partir du code PHP de la page.

4. Le serveur envoie la page HTML générée au client

Une fois la page HTML est créée, le serveur l'envoie au client.

5. Le client (navigateur) affiche le HTML reçu

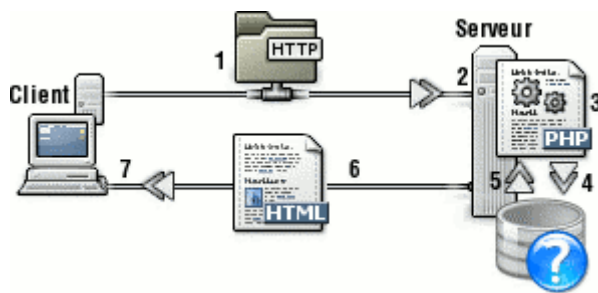
Le client a qu'à afficher du code HTML...

Bonjour je suis une page dynamique.
L'heure actuelle est:
03 24 34pm

3. Pages dynamiques (avec BD)

On peut créer un site dynamique sans utiliser une BD (on vient de le faire, même si c'était très simple!), mais la plupart d'applications web existantes font usage d'une base de données.

Un site d'internet peut avoir une partie statique en HTML/CSS et une partie dynamique **dont le contenu provient d'une base de données**. La partie dynamique changera en fonction du contenu qu'on récupère de la base de données (ex: afficher les vols entre deux villes dans un certain intervalle de dates). Le fonctionnement d'une page dynamique qui interagit avec une BD est le suivant:



Ex: recherche sur le site de Wikipedia

1. L'utilisateur charge une page (HTML ou PHP) contenant un formulaire
2. L'utilisateur saisit les termes de recherche dans la boîte "Rechercher" du formulaire et clique sur le bouton de recherche
3. Le **client (navigateur) demande le chargement d'une page PHP** en **envoie au serveur le contenu du formulaire** (les termes de recherche) en utilisant le protocole http. Le serveur cherche la page **PHP** dans son disque
4. Le **module PHP interprète le code PHP de la page**. A différence de l'exemple précédant, cette page PHP **fait une requête au serveur de bases de données MySQL** (pour rechercher les termes obtenus du formulaire).
5. Le **serveur BD renvoie le résultat de la recherche au module PHP du serveur web**
6. Le **module PHP continue à interpréter la page et génère le code HTML correspondant**
7. Le serveur envoie la page HTML générée au client
8. Le client affiche le HTML reçu dans le navigateur

Un autre exemple basé sur le même principe : recherche de vols sur Brussels Airlines

1. On va sur le site de Brussels Airlines
2. On remplit le formulaire de recherche et on clique sur "Recherche" et notre navigateur envoie le contenu du formulaire (ville départ, arrivé, dates etc...) au serveur. On demande au serveur de charger une page PHP ("action" du formulaire) qui utilisera ces données pour chercher dans une BD de vols
3. Le serveur cherche la page PHP indiquée dans "action"
4. Le module PHP interprète le code de la page, qui contiendra forcément une requête à la base de données, basée sur les données reçues du formulaire
5. Le serveur de BD cherche les vols dans la BD et renvoie le résultat (une liste de vols qui atteignent les critères)
6. Le module PHP continue à interpréter la page et génère le code HTML correspondant au résultat
7. Le serveur envoie la page HTML générée au client contenant la liste de vols
8. Le client affiche le HTML reçu dans le navigateur

Installation de XAMPP

Nous avons besoin d'au moins trois éléments pour pouvoir réaliser des sites dynamiques complets en PHP:

1. Serveur Web → ex : Apache
2. Module PHP (interprète du code PHP)
3. Serveur de BD → ex : MySQL

Dans notre cas, le serveur web et le serveur de BD se trouveront dans la même machine. Le but est de simuler un environnement serveur sur notre ordinateur de bureau. **Notre ordinateur sera à la fois le client et le serveur.** Normalement on devrait installer chaque élément séparément et les configurer pour qu'ils travaillent ensemble (ex: serveur web + module PHP), ce qui n'est pas toujours évident.

Heureusement il y a eu de personnes qui ont consacré leur temps et leur effort à rassembler ces éléments dans un seul package qu'on peut installer facilement et qui nous permet de développer directement nos sites dynamiques.

Nous allons travailler avec un de packages le plus connus : **XAMPP server** (), mais il y en a plein d'autres tels que XAMPP ou LAMPP (cherchez List of Apache–MySQL–PHP packages sur Wikipedia). Cet package contient les trois éléments et la configuration de base est presque immédiate.

1. Allez sur le site et téléchargez la version qui vous convienne :
<https://www.apachefriends.org/download.html>
2. Faites double clic et suivez la procédure d'installation

Une fois le serveur est installé vous aurez accès au panel de control de XAMPP. Vous allez utiliser ce panel principalement pour démarrer et arrêter les serveurs Apache et MySql, mais on peut réaliser plein d'autres actions.

On peut configurer Apache, PHP et MySql en modifiant les fichiers httpd.conf, php.ini et my.ini respectivement.

Vous devez créer vos pages html ou php dans le dossier **c:/xampp/htdocs**

Attention: si on utilise Skype, il faut changer une option par défaut (en Skype). Dans Outils → Options de connexion → Connexion, et décocher la case "utiliser les ports 80 et 443 comme alternative". Puis enregistrer et quitter Skype pour que les modifications soient prises en charge et réessayer de redémarrer XAMPP. Ce problème vient du fait que le port 80 est utilisé par XAMPP (service http) mais il est déjà utilisé par Skype. Le port 80 à l'avantage de ne pas être bloqué par les pare-feu car il est normalement dédié au serveur web et reste donc ouvert au travers du pare-feu pour permettre de se connecter à internet. Seul les 1025 premiers ports ont un rôle définit. Les autres applications se servent d'autres ports.

