

Cours PHP

Objectifs

Maîtriser la syntaxe et les instructions de base en PHP.

Conserver et échanger des informations avec un internaute.

Sauvegarder, utiliser et exploiter pleinement une base de données en faisant communiquer les technologies PHP/SQL.

Créer un site web « 2en1 » avec une partie cliente (front) et une interface de gestion (back).

Développement de fonctionnalités. (CRUD orienté e-commerce)

1- A quoi sert le PHP ? Qu'est-ce que c'est ?

Le PHP (Hypertext Preprocessor) est un langage de programmation (fonctionnel) à l'inverse des langages Html et Css qui sont des langages de conception (structure et mise en forme de page web). Avec PHP, nous pouvons faire tous types de site et il n'y a plus de limite : moteur de recherche (type google), réseaux sociaux (type facebook), plateforme multimédia (type YouTube), site d'informations (type Wikipédia), boutique e-commerce (type amazon), petites annonces (type leboncoin), forum, blog, espace membre, etc. Pour mieux comprendre, prenons le cas d'un formulaire : 1 formulaire est déclaré en html, il est mis en forme en css (html + css = conception), mais si nous voulons récupérer et exploiter les saisies postées par l'internaute nous aurons besoin du PHP qui est un langage permettant d'effectuer des traitements (et c'est bien là le principe, si nous faisons un formulaire c'est pour récupérer des données !).

Autrement dit, sans PHP nous ne pourrions pas faire fonctionner un formulaire. Ensuite, si nous voulons sauvegarder ces données dans une base, nous aurons besoin du langage SQL. D'autre part, le PHP permet de créer des sites web dynamiques ! Et c'est très utile !

2- Qu'est-ce qu'un site web statique ?

Un site statique est généralement réalisé avec les langages Html et Css. La problématique c'est que ces sites ne sont pas pratiques à mettre à jour (exemple de mise à jour : ajouter une image, modifier du texte, ajouter un nouveau lien comme rubrique de menu, créer une nouvelle page).

Vous me direz peut-être que c'est simple et rapide pour vous d'adapter votre code-source puisque vous êtes à l'aise en Html et Css mais ce n'est pas le cas de tous et encore moins des clients qui achèteront les sites web (ils sont généralement peu techniques et veulent pouvoir effectuer des réglages rapidement et efficacement).

Le principal inconvénient d'un site statique c'est qu'il faut quelques connaissances techniques car nous sommes obligés de repasser par le code-source de chaque fichier pour effectuer des modifications.

Gardez bien à l'esprit qu'un site est modifié en permanence ! Même quand vous pensez qu'il est "terminé" il ne l'est pas vraiment. Il faut donc trouver des solutions pour faciliter les modifications et les mises à jour : c'est là que le site dynamique (propulsé par PHP) rentre en jeu!

3- Qu'est-ce qu'un site web dynamique ?

Un site dynamique est un site dont les informations proviennent d'une base de données (pour cela il faudra mélanger le langage php et le langage sql ensemble).

Attention!

il ne faut pas confondre un site dynamique avec un site de mouvements ou d'animations et couleurs clinquantes dans la page web en JS ou autres.

Pour obtenir un site dynamique, nous utiliserons le langage PHP, qui lui-même accueillera le langage SQL afin d'obtenir des informations contenues dans la base de données. Si nous arrivons à récupérer des données (provenant d'une base de données) sur notre site, nous pourrions aussi récupérer ces mêmes données pour un deuxième site.

Pourquoi 2 sites ? quel en serait l'intérêt ?

- Le premier site sera destiné au grand public pour la consultation des informations par les internautes.

un Front (aussi appelé, **front-office** ou **front-end**)

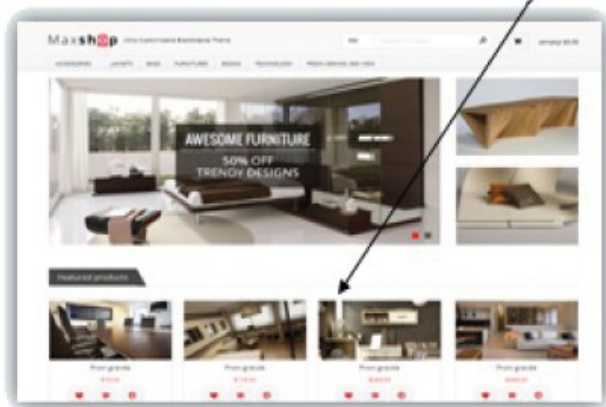
- Le deuxième site sera protégé par authentification et destiné à la mise à jour et modification de contenu par l'administrateur (propriétaire du site). un Back (aussi appelé, **back-office**, **back-end** ou interface de gestion).

Nous appelons donc « un site dynamique » un site dont les informations sont contenues dans une base de données et qui possède 2 interfaces : un premier site FRONT (pour l'affichage du site normal), un deuxième site BACK (pour la gestion et assurer les réglages du premier site). L'avantage d'un site dynamique c'est que les modifications seront beaucoup plus faciles et ne devront pas obligatoirement être réalisées dans le code source d'un fichier par une personne connaissant le code et ayant des compétences techniques. Les modifications pourront être effectuées dans une interface de gestion backOffice simple et intuitive, et par tous ! Voici un schéma illustrant cette situation :

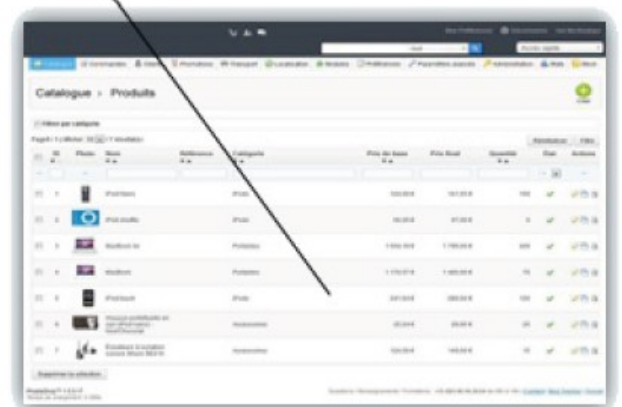
Base de données

ID	Nom	Prénom	Date de naissance	Sexe	Adresse	Code postal	Ville	Pays	Telephone	Email	Statut
1	John	Doe	1980-01-01	M	123 Main St	10001	New York	USA	212-555-1234	john.doe@example.com	Actif
2	Jane	Doe	1985-02-15	F	456 Main St	10001	New York	USA	212-555-5678	jane.doe@example.com	Actif
3	John	Smith	1975-03-20	M	789 Main St	10001	New York	USA	212-555-9012	john.smith@example.com	Actif
4	Jane	Smith	1982-04-10	F	101 Main St	10001	New York	USA	212-555-3456	jane.smith@example.com	Actif
5	John	Johnson	1978-05-05	M	202 Main St	10001	New York	USA	212-555-7890	john.johnson@example.com	Actif
6	Jane	Johnson	1983-06-12	F	303 Main St	10001	New York	USA	212-555-2345	jane.johnson@example.com	Actif
7	John	Williams	1979-07-18	M	404 Main St	10001	New York	USA	212-555-6789	john.williams@example.com	Actif
8	Jane	Williams	1984-08-22	F	505 Main St	10001	New York	USA	212-555-0123	jane.williams@example.com	Actif
9	John	Brown	1976-09-03	M	606 Main St	10001	New York	USA	212-555-4567	john.brown@example.com	Actif
10	Jane	Brown	1981-10-07	F	707 Main St	10001	New York	USA	212-555-8901	jane.brown@example.com	Actif

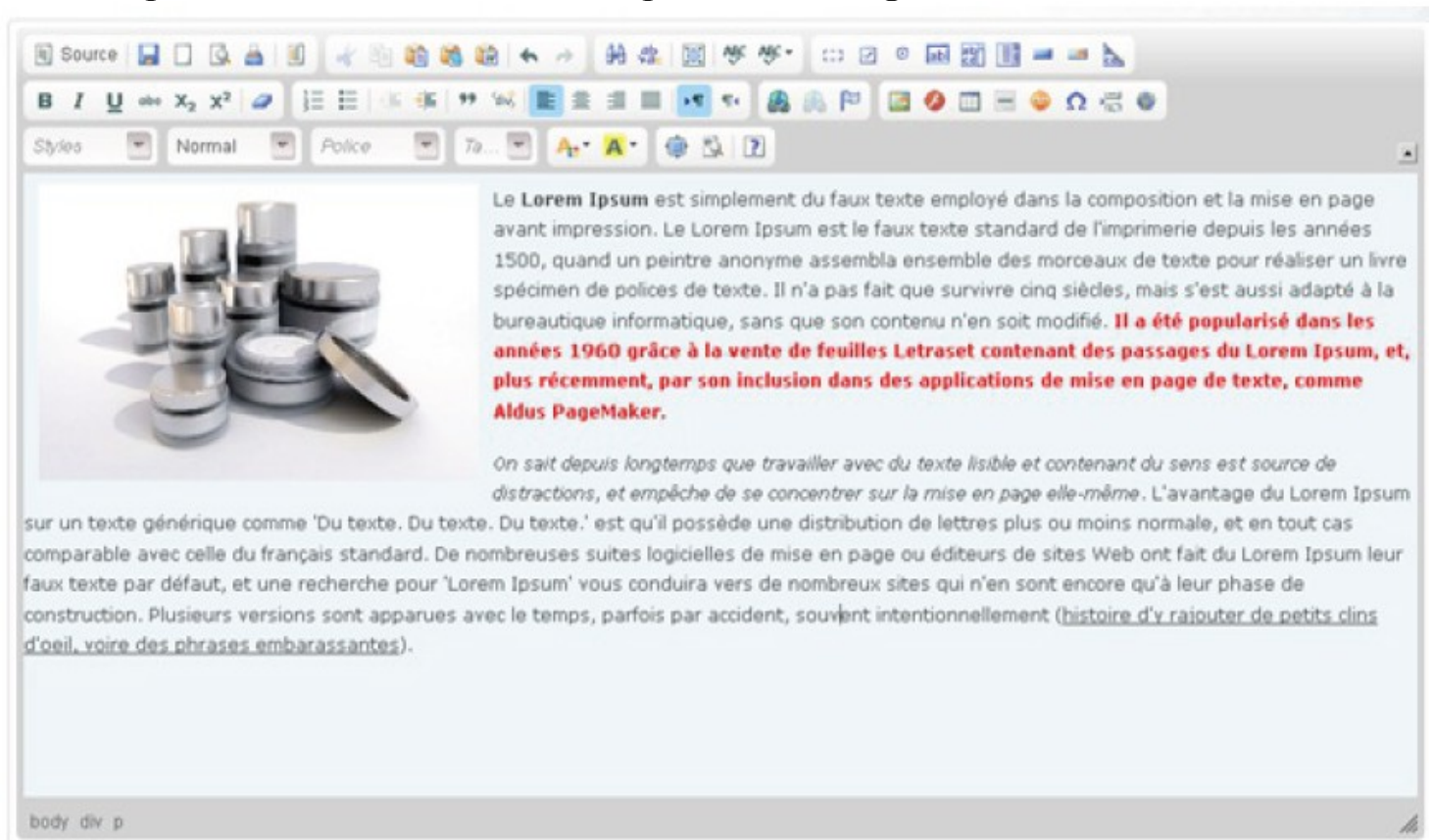
FRONT-OFFICE



BACK-OFFICE



Dans une interface de gestion (BackOffice), nous retrouvons la plupart du temps un éditeur avec des icônes permettant de gérer les contenus :



Par exemple l'icône permettant de créer un lien ajoutera automatiquement les balises `<a>` `` dans le code. Sans que le développeur n'ait besoin de l'écrire. Dans la plupart des cas, il n'y a pas qu'un seul site web mais deux sites web (un front et un back). C'est ce qu'on appelle un site web dynamique.

Qu'est-ce qu'un site dynamique ?

Un site dynamique est un site modifiable sans avoir besoin de retourner dans le code-source obligatoirement. Les sites web dynamiques sont majoritairement élaborés avec l'aide de php (d'autres langages de programmation peuvent être utilisés)

Le FRONT représente la partie que nous consultons en tant qu'internaute (pour l'affichage et la consultations des informations). Le BACK réservé aux administrateurs du site (gérants) permettant les mises à jour des informations. Si nous voulions faire la liste des sites dynamiques incluant du php et du sql, la liste serait longue voire interminable, puisque tous les sites (ou presque) sont concernés.

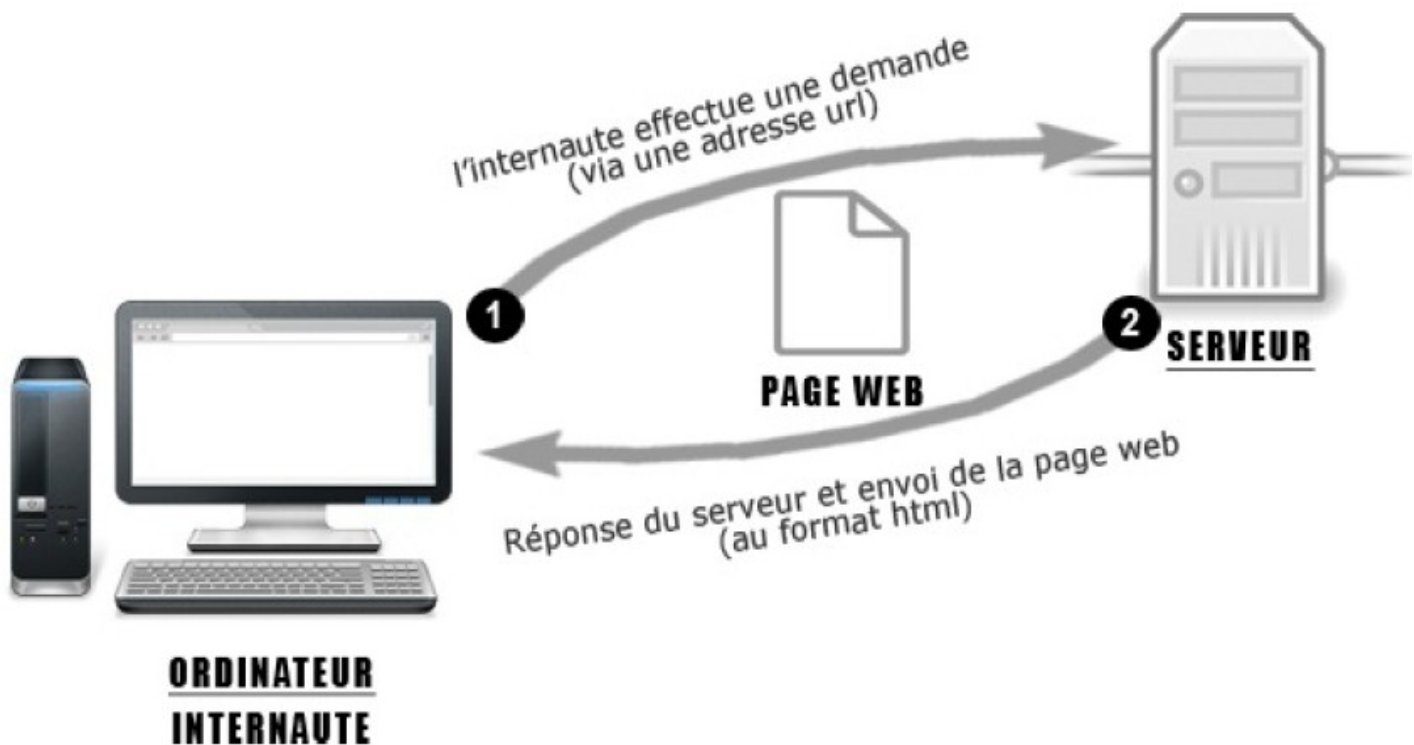
Prenons la chose à l'envers, quels sont les sites statiques (et donc non dynamiques) et n'incluant ni de Php ni de Sql ?

Des sites n'étant jamais mis à jour (assez rare), certains sites développés par les développeurs eux même pour se présenter (et encore ce n'est pas une majorité). Contrairement aux idées reçues, la plupart du temps même un site vitrine est un site dynamique ! Les clients demandent un site vitrine car ils veulent se présenter mais aussi pouvoir effectuer des modifications via une interface de gestion BackOffice !

Sinon, sans connaissance technique, comment feraient-ils pour modifier les textes, les images et créer de nouvelles pages ? Retenez le bien : La plupart du temps un site vitrine est un site dynamique ! Maintenant vous le savez, PHP est incontournable ! PHP est actuellement utilisé en version 8 sous forme de code procédural ou orienté objet.

En résumé: Pour construire votre page web vous aurez besoin de html et Css (partie structure et mise en forme). Pour la rendre fonctionnelle et dynamique vous aurez besoin de PHP et SQL (pour effectuer des sauvegardes et mener des opérations sur la base de données). JavaScript permettra d'assurer la programmation côté client et viendra compléter les autres langages.

4- Architecture Client / Serveur avec un site statique.



1. - L'internaute effectue une demande (via une adresse URL).
2. - Le serveur reçoit la demande et trouve le fichier (page web) intégrant uniquement du code Html et Css, il renvoie le contenu demandé à l'internaute. C'est aussi simple que ça. Les sites statiques n'utilisent que du code html et css.

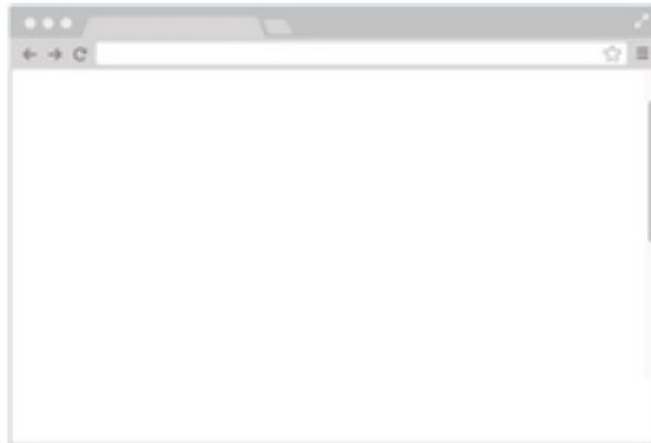
5- Architecture Client / Serveur avec un site dynamique

Comment fonctionne le Php ?

Lorsqu'une page web s'affiche sur votre écran, elle respecte un processus particulier décrit ci-dessous :

- 1- Un internaute souhaite accéder à une page web, il utilise un navigateur par lequel il sollicite une adresse url.
- 2- Cette adresse url pointe vers une ressource (fichiers contenant texte, image, code-source, etc.) hébergée par un serveur (emplacement de stockage applicatif).
- 3- Il arrive parfois que le serveur ait besoin de consulter une base de données pour intégrer des informations dans le contenu de la page web (requête SQL).

1 - navigateur de l'internaute :

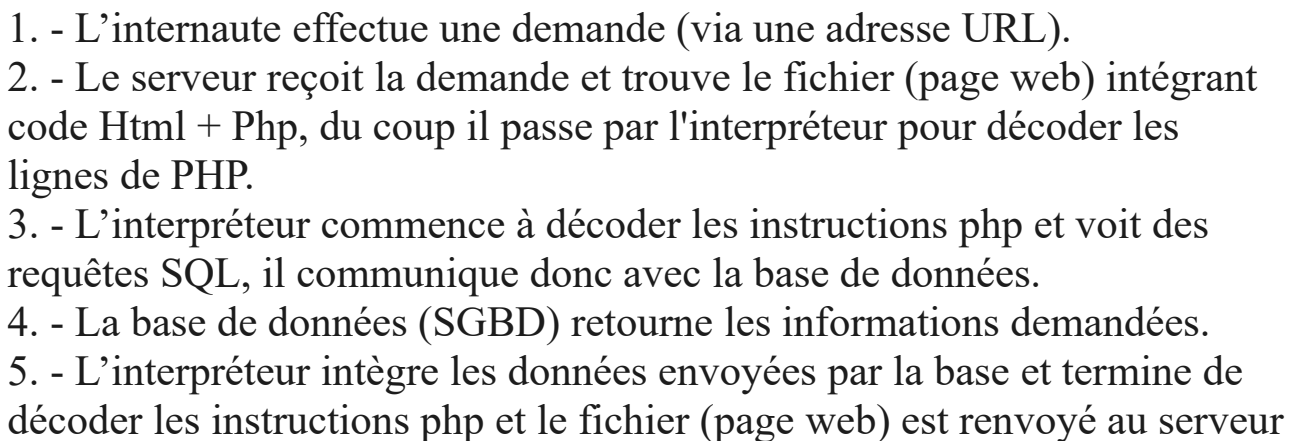


2 - serveur stockant la ressource :



avoir	3	j'avais eu	tu avais eu	il avait eu	<i>elle avait eu</i>	nous avions eu	vous aviez eu	ils avaient eu	<i>elles avaient eu</i>
avoir	3	j'aurais eu	tu aurais eu	il aurait eu	<i>elle aurait eu</i>	nous aurions eu	vous auriez eu	ils auraient eu	<i>elles auraient eu</i>
être	3	j'étais	tu es	il est	<i>elle est</i>	nous sommes	vous êtes	ils sont	<i>elles sont</i>
être	3	j'e serai	tu seras	il sera	<i>elle sera</i>	nous serons	vous serez	ils seront	<i>elles seront</i>
être	3	j'étais	tu étais	il était	<i>elle était</i>	nous étions	vous étiez	ils étaient	<i>elles étaient</i>
être	3	j'ai été	tu as été	il a été	<i>elle a été</i>	nous avons été	vous avez été	ils ont été	<i>elles ont été</i>
être	3					soyons	soyez		
être	3	j'e fus	tu fus	il fut	<i>elle fut</i>			ils furent	<i>elles furent</i>
être	3	j'e serais	tu serais	il serait	<i>elle serait</i>	nous serions	vous seriez	ils seraient	<i>elles seraient</i>
être	3	que tu sois	que tu sois	qu'il soit	<i>qu'elle soit</i>	que nous soyions	que vous soyez	qu'ils soient	<i>qu'elles soient</i>
être	3	j'avais été	tu avais été	il avait été	<i>elle avait été</i>	nous avions été	vous aviez été	ils avaient été	<i>elles avaient été</i>
être	3	j'aurais été	tu aurais été	il aurait été	<i>elle aurait été</i>	nous aurions été	vous auriez été	ils auraient été	<i>elles auraient été</i>
chanter	1	j'ai chanté	tu chantes	il chante	<i>elle chante</i>	nous chantons	vous chantez	ils chantent	<i>elles chantent</i>
chanter	1	j'ai chanté	tu chanteras	il chantera	<i>elle chantera</i>	nous chanterons	vous chanterez	ils chanteront	<i>elles chanteront</i>
chanter	1	j'ai chanté	tu chantaies	il chantait	<i>elle chantait</i>	nous chantions	vous chantiez	ils chantaient	<i>elles chantaient</i>
chanter	1	j'ai chanté	tu as chanté	il a chanté	<i>elle a chanté</i>	nous avons chanté	vous avez chanté	ils ont chanté	<i>elles ont chanté</i>
chanter	1		chante			chantons	chantez		
chanter	1	j'ai chanté	tu chantas	il chanta	<i>elle chanta</i>			ils chanteront	<i>elles chanteront</i>
chanter	1	j'ai chanté	tu chanterais	il chanterait	<i>elle chanterait</i>	nous chanterions	vous chanteriez	ils chanteraient	<i>elles chanteraient</i>
chanter	1	j'ai chanté	tu chantes	qu'il chante	<i>qu'elle chante</i>	que nous chantions	que vous chantiez	qu'ils chantent	<i>qu'elles chantent</i>
chanter	1	j'ai chanté	tu avais chanté	il avait chanté	<i>elle avait chanté</i>	nous avions chanté	vous aviez chanté	ils avaient chanté	<i>elles avaient chanté</i>
chanter	1	j'ai chanté	tu auras chanté	il aura chanté	<i>elle aura chanté</i>	nous aurons chanté	vous aurez chanté	ils auront chanté	<i>elles auront chanté</i>
finir	2	j'ai fini	tu finis	il finit	<i>elle finit</i>	nous finissons	vous finissez	ils finissent	<i>elles finissent</i>
finir	2	j'ai fini	tu finiras	il finira	<i>elle finira</i>	nous finirons	vous finirez	ils finiront	<i>elles finiront</i>
finir	2	j'ai fini	tu finissais	il finissait	<i>elle finissait</i>	nous finissions	vous finissiez	ils finissaient	<i>elles finissaient</i>
finir	2	j'ai fini	tu as fini	il a fini	<i>elle a fini</i>	nous avons fini	vous avez fini	ils ont fini	<i>elles ont fini</i>
finir	2		finis			finissons	finissez		
finir	2	j'ai fini	tu finis	il finit	<i>elle finit</i>			ils finiraient	<i>elles finiraient</i>
finir	2	j'ai fini	tu finirais	il finirait	<i>elle finirait</i>	nous finirions	vous finiriez	ils finiraient	<i>elles finiraient</i>
finir	2	que je finisse	que tu finisses	qu'il finisse	<i>qu'elle finisse</i>	que nous finissions	que vous finissiez	qu'ils finissent	<i>qu'elles finissent</i>

page web incluant du code php + html



au format 100% html (puisque l'interpréteur a transformé toutes les instructions Php en résultat Html).

6. - Le serveur renvoie la page Html au navigateur, sur l'ordinateur de l'internaute.

Comprenez-vous pourquoi vous ne verrez jamais 1 ligne de code Php dans le code-source d'une page web ?

En effet, même si la page web contient du code php, ce code est décodé par l'interpréteur de manière à ne renvoyer que du html au navigateur de l'internaute. De toute façon, le navigateur ne pourrait pas comprendre des lignes PHP, ce n'est pas son rôle, il doit simplement afficher la page web et pour ça il a uniquement besoin de garder le code Html et Css.

Si du code Php est tout de même passé au navigateur, il le considérera comme du texte et ne l'exécutera pas, du coup votre programmation ne fonctionnera pas. C'est aussi la raison pour laquelle vous ne pourrez jamais créer un fichier php sur votre bureau, ou le lancer directement et manuellement dans le navigateur, sans aucun serveur web ou interpréteur. En effet, il est impératif que votre fichier contenant le code php respecte tout ce cycle (schéma au dessus : interprétation et exécution, échange avec la base de données éventuellement, etc.).

Afin de faire fonctionner notre code Php nous utilisons le serveur web WAMP (qui inclut tout ce dont nous avons besoin) :

Windows - système d'exploitation.

Apache - serveur web (HTTP).

Mysql - SGBD (Système de Gestion de Base de Données).

Php - Interpréteur Php.

Bon à savoir :

- **PHP** est un langage exécuté côté serveur, ce ne sera donc pas une exécution par l'ordinateur ou le navigateur de l'internaute.
- **PHP** est un langage interprété, ses instructions sont traitées séquentiellement par le serveur (pas de compilation)
- **PHP** est open source, tout le monde peut l'utiliser et même vendre une de ces créations à l'aide du langage PHP.

Les sites dynamiques utilisent plusieurs langages : html, css, php, sql, javascript.

6- Les langages de programmation

Y'a t'il d'autres langages de programmation différents de Php pour construire un site web dynamique ?

Oui, d'autres langages exécutés côté serveur existent pour créer un site web dynamique : ASP (Active Server Page, développé par Microsoft), JSP (Java Server Pages, développé par Oracle), CGI (Common Gateway Interface) , etc.

Nous nous concentrerons d'abord sur Php dans le cadre de ce cours.

7- Qu'est-ce qu'un serveur ?

Un serveur est généralement un ordinateur qui héberge un site web, mis à disposition par un hébergeur (hébergeur = une société de service informatique). Cet ordinateur reste toujours allumé, c'est ce qui permet une consultation 24h/24 et 7j/7.

8- PHP procédural ou PHP Orienté Objet ?

Deux techniques de programmation sont souvent utilisées :

L'Approche Procédurale (le code est écrit séquentiellement).

L'Approche Orientée Objet (le code est encapsulé dans des méthodes de classes et fonctionne via l'interaction d'objets).

Avant de s'intéresser aux avantages et inconvénients de ces deux techniques, sachez que certaines notions de bases sont forcément apprises en approche procédurale.

L'approche orientée objet est souvent privilégiée par les entreprises car cela permet d'encourager le travail collaboratif et simplifie grandement les mises à jour. (surtout pour les projets de grande envergure, développés sur plusieurs mois et par plusieurs personnes).

Techniquement, nous ne pouvons pas aller plus loin en terme de finalité et de rendu fonctionnel avec l'orienté objet par rapport au procédural. C'est avant tout une méthodologie de travail (et non pas une évolution pour créer davantage).

9- Développement : from scratch ? cms ? Framework ?

A ce jour, il y a 3 grands moyens de créer un site web :

- À la main (from scratch, en partant de 0),
- avec l'appui d'un CMS,
- avec l'aide d'un FRAMEWORK,

Lorsqu'on utilise un CMS ou un FRAMEWORK, le code peut être procédural ou orienté objet (et parfois le code reprend un peu des deux techniques).

Avant de s'intéresser à chacun de ces moyens, il est indispensable de bien connaître et maîtriser le PHP (dans sa forme basique).