Page interactive != Dynamique

Dans le répertoire du serveur Web  
Dossier :

**tmp** 🡪 dossier dans lequel le serveur va placer les fichiers qu’il utilise au cours de son fonctionnement. C’est son dossier de travail

**Logs** 🡪 Il contient les traces des events qui se déroulent au niveau du serveur

**Htdocs** 🡪 contient les ressources du serveur qui sont accessibles pour les clients. C’est le répertoire de publication

**Conf** 🡪 contient les fichiers de configuration du serveur. Un fichier de config est un fichier qui contient les paramètres de fonctionnement du serveur (ici par exemple le répertoire de publication, le port utilisé etc…). Il peut y avoir plusieurs fichiers de config, dans ce cas il y en aura un principal qui contiendra le workflow global et d’autres secondaires.

*httpd.conf*

S’il y a des params de config dans des fichiers secondaires, on observe la directive include

Les directives sont de la forme <Nom de la directive> + <Valeur de la directive>

Les directives secondaires se trouvent dans le dossier « extra » (le nom peut varier) à l’intérieur du dossier conf ; du coup si on cherche quelque chose qui ne figure pas dans httpd.conf, on sait où chercher (Directive = include conf/extra/…)  
A l’interieur du fichier http.conf il y a quelques modifications à faire

1. Servername 🡪 correspond au nom de domaine (en local) de la page
2. Listen 🡪 <pour la signification, lire la doc au-dessus de la directive>
3. LoadModule
4. LoadModule php7\_module
5. ServerAdmin
6. ServerName
7. DocumentRoot 🡪 Repertoire de publication
8. DirectoryIndex
9. ErrorLog
10. Alias 🡪 Un alias permet d’eviter de rentrer le chemin complet, c’est un raccourci. C’est de la forme « **Alias <valeur\_alias> <Chemin original>** »
11. DefaultType
12. Userdir 🡪 permet de créer des repertoires de publication personnels pour les utilisateurs du serveur

Une bonne solution est de créer un repertoire de publication pour chaque utilisateur de la machine.

*php.ini*

C’est le fichier de configuration de php

Goto Language Options et voir les directives

1. engine = On
2. short\_open\_tag = Off 🡪 (cette balise autorise le ‘?>’ ou non dans le code php)
3. precision = ??
4. max\_execution\_time = ??
5. max\_input\_time = ??
6. memory\_limit = ?? (= (memoire totale / 100)/2)
7. post\_max\_size = ?? (ROM)

On a 2 types d’expressions :

* Les expressions d’instruction
* Les expressions < ?>

PHP est un langage faiblement typé (declaration implicite).

Le « . » est l’operateur qui permet la concat

Expression = suite syntaxiquement correcte d’operateurs et d’operandes

VARIABLES D’ENVIRONNEMENT

Variable d’environnement = variable du S.E qui est accessible a n’importe quelle appli presente sur le système d’exploitation.

Path = chemin d’acces vers un repertoire

VARIABLES D’ENVIRONNEMENT EN PHP

Variables accessibles a tous les programmes en php et qui contiennent des infos sur l’environnement dans lequel se trouve le programme qu’on ecrit. Ces infos sont entre autres :

Le navigateur qui a envoye la requete d’execution du fichier -> HTTP\_USER\_AGENT

L’adresse du ?? -> REMOTE\_ADDR

Le nom du serveur -> SERVER\_NAME

Le numero du port -> SERVER\_PORT

Le nom du repertoire source -> DOCUMENT\_ROOT

Pour faire appel a ces variables, on utilise des variable superglobales (tableau associatif) (de 3facons)

1’ $0 = $\_SERVER[« REMOTE\_ADDR » ]

2’ $0 = getenv(“REMOTE\_ADDR”)

3’ $0 = $REMOTE\_ADDR; (Cette methode necessite une modif dans le ficiher de conf)

Un grand formulaire ou petit formulaire ne depend pas du nombre de champs, mais plutôt de la nature des donnees qui seront envoyees via elle. Donc **POST** est preconisee pour les ***ptits formulaires*** et **GET** pour les ***grands formulaires***.

|  |
| --- |
| Exemple : if ( *!*isset($\_REQUEST[‘nom’])) {> ?  Code formulaire html… // pour verifier si la var dans la condition est presente.  Code formulaire html…  < ?php } else {  Traitement PHP  }  ?> |

La function **isset($v)** -> renvoie vrai si la variable existe. **emty($v)** -> pour verifiere si la variable est vide