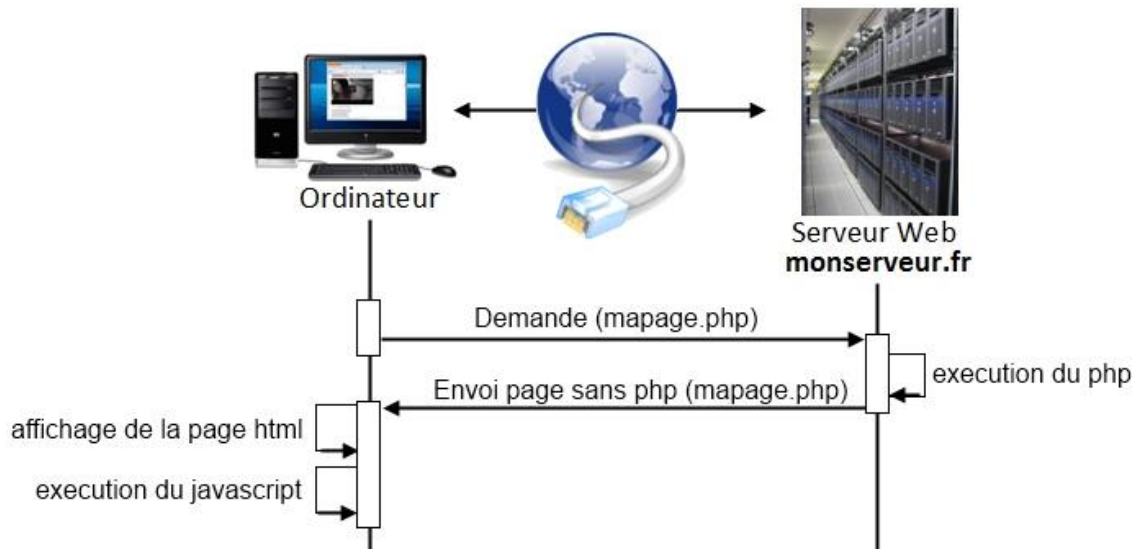


Fonctionnement global



Pour pouvoir communiquer entre eux, l'ordinateur client et le serveur ont besoin d'un logiciel appelé navigateur pour le client et serveur web pour le serveur.

Le navigateur seul ne pouvant pas interpréter le langage PHP nous utiliserons le logiciel WampServer.

Ce logiciel est composé entre autre de :

- Php my admin, permettant la création et la gestion de bases de données ;
- Un dossier « www » permettant de stocker notre site web. Il sera lu par le navigateur à l'adresse « localhost ».

Le PHP à quoi ça sert ?

Le PHP permet par exemple:

- De créer un site web qui interagit avec une base de donnée. (en lecture et écriture)
- De permettre à l'utilisateur d'interagir avec le site web. (accéder à une page spécifique, choix du design etc.)
- De permettre au site web de lire ou d'écrire dans un fichier (Texte, XML, CSV etc.)

En combinant ces 3 utilisations, on peut faire une infinité de choses (recherche par mot clé dans une Foire aux Questions, choix de la langue du site, jeu vidéo etc.)

Comment s'en servir ?

Pour que le serveur interprète le PHP il faut lui préciser que le fichier est un fichier contenant ce langage de programmation. Celui-ci est donc nommé sous la forme « monfichier.php ».

Concernant le code PHP, il est inséré dans le code HTML entre les caractères « <?php » et « ?> ».

On peut également afficher du HTML sans quitter le PHP en utilisant « echo »

Exemple : `echo "<h1>Bonjour Monsieur</h1>";`

Comme beaucoup de langage de programmation, une ligne de code PHP fini toujours par le caractère « ; ».

Pour stocker une valeur numérique (le temps du programme) pour pouvoir faire des calculs, l'incrémenter etc. on utilise des variables. On peut également y stocker du texte.

Les variables

Dans le langage PHP, on utilise une variable sans avoir déclaré son type auparavant.

Exemple : `$maVariable1 = 10 ;`

Ou : `$maVariable2 = "Bonjour";`

On peut de la même façon les afficher grâce à « écho ».

Exemple : `$monNom = "PARADE" ;`

`echo "Bonjour Monsieur $monNom" ;`

Comment interagir entre le programme côté serveur et le client ?

Pour que le client puisse communiquer avec le serveur, on utilise généralement un formulaire de saisie.

Exemple ->

login	
mdp	
Se connecter	

Les formulaires

Les formulaires sont créés en HTML avec la balise <form> (voir index).

Il existe deux méthodes pour envoyer les données d'un formulaire :

- La méthode GET affiche les variables en clair dans la barre d'adresse.
Exemple : <http://www.monsite.fr/?page=1>
Faire un lien <a> en renseignant les variables dans l'adresse permet de ne pas passer par le formulaire.
- La méthode POST permet d'envoyer des variables de façon plus discrète.

Réception d'un formulaire

Méthode GET :

`$_GET['nom']` est utilisé comme une variable pour utilisé la valeur de 'nom' envoyé dans le formulaire.

Méthode POST :

La méthode POST est similaire à la méthode GET : `$_POST['nom']`.

Exemple :

```
if(isset($_GET['envoyerEntreprise']))
{
    //si le bouton "envoyerEntreprise" a été validé

    echo "<font color=\"red\">".$_GET['entreprise']</font>";
    //afficher le nom de l'entreprise en rouge
}
```

Interagir avec une Base De Données

Se connecter :

```
$pdo_options[PDO::ATTR_ERRMODE] = PDO::ERRMODE_EXCEPTION;
$dbdd = new PDO('mysql:host=localhost;dbname=simu_ventilo',
                'root', '', $pdo_options);
```

Il existe deux méthodes en PHP pour envoyer une requête SQL :

- La fonction `exec()`
- La fonction `prepare()` puis `execute()`

Lire dans une BDD :

```
$reponse = $bdd->prepare("
    SELECT DAY(mp_date) AS jour FROM marche_partielle
    WHERE MONTH(mp_date) = ? AND YEAR(mp_date) = ? AND mp_idPiece = ?
");
$reponse->execute(array($mois, $annee, $_GET['idPiece']));
```

Ecrire dans une BDD :

```
$bdd->exec("
    INSERT INTO marche_complete(mc_date, mc_idPiece)
    VALUES ('".$annee."-".$mois."-".$i."', '".$_GET['idPiece']. "')
");
```

Effacer des données :

```
$bdd->exec("
    DELETE FROM marche_complete
    WHERE mc_idPiece='".$_GET['idPiece']."'
");
```