

**I. Introduction:**

**1. Définition d'un site web dynamique:**

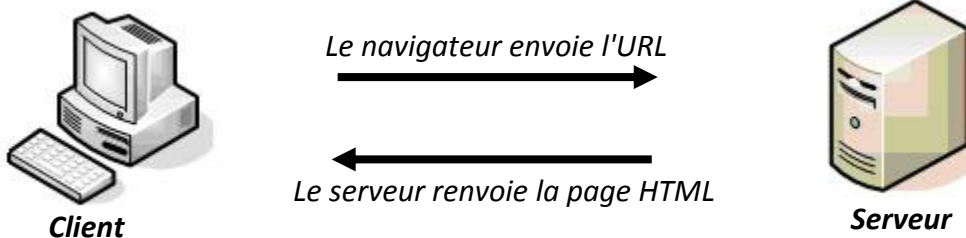
Un site Web dynamique est un site Web dont les pages peuvent être générées dynamiquement en fonction d'une demande de l'utilisateur.

Le contenu peut être obtenu en combinant l'utilisation d'un langage de scripts et une base de données.

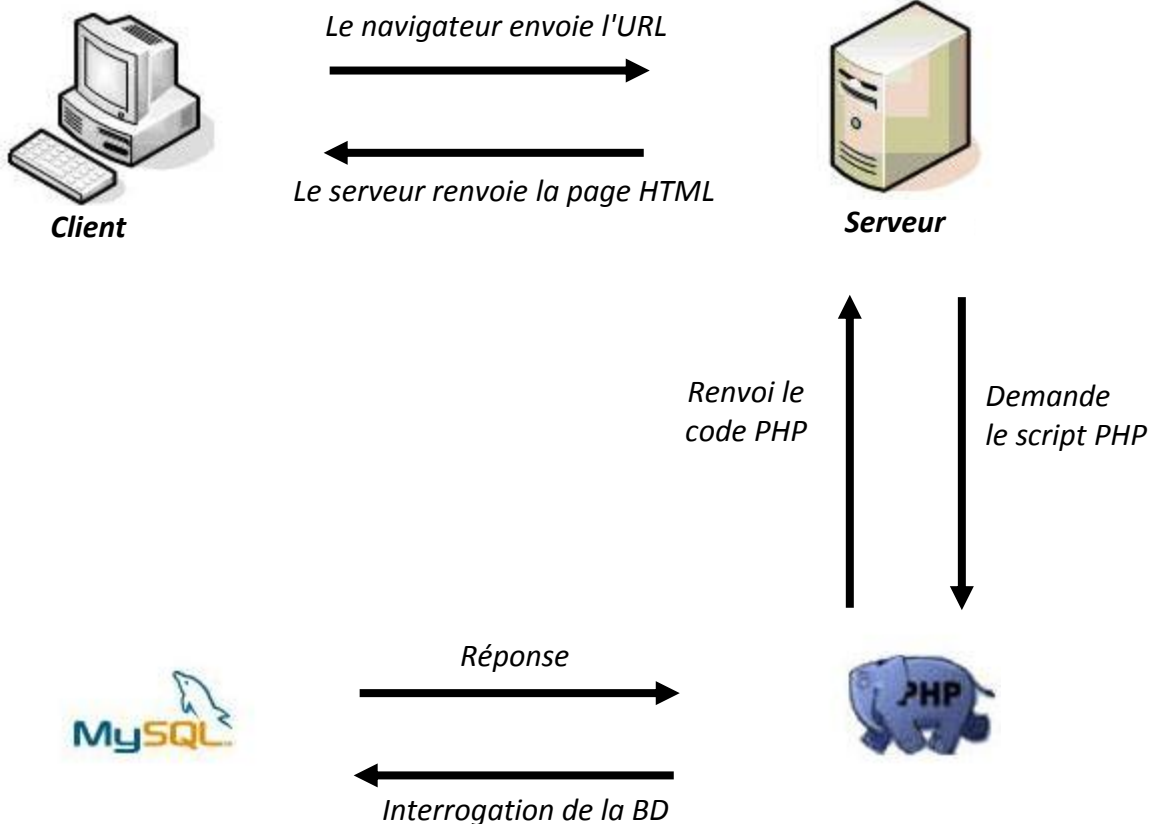
Dans le monde des logiciels libres, il s'agit généralement de PHP pour le langage script et MySQL pour la base de données.

**2. Principe de fonctionnement:**

Page Web Statique :



Page Web dynamique :



## **Description :**

- Le navigateur envoie l'adresse d'une page Web au serveur.
- Le serveur web cherche dans son arborescence le fichier correspondant.  
Si ce fichier porte l'extension d'une application PHP (.PHP, .PHP3...), le serveur transmet ce fichier à PHP.
- PHP analyse et exécute le code reçu (code PHP) qui se trouve entre les balises `<? et ?>`.  
Si ce code contient des requêtes vers une base de données MySQL, PHP envoie les requêtes SQL.
- La base de données renvoie les informations trouvées au script PHP qui va les exploiter.
- PHP continue l'exécution du code puis retourne le fichier dépourvu du code PHP au serveur web.
- Le serveur renvoie un fichier ne contenant plus de PHP, donc seulement du HTML au navigateur qui l'interprète et l'affiche.

## **II. Le langage PHP:**

### **1. Présentation:**

Le langage PHP a été créé en 1994 par Rasmus Lerdorf et qui signifiait à cette époque Personal Home Page.

Plusieurs améliorations ont été apportées à ce langage. Depuis sa version 3, ce langage s'appelle Hypertext Preprocessor.

Actuellement on est à la version 5.3.

A l'aide de PHP, on peut réaliser des forums, des compteurs de visiteurs, des livres d'or, ...

### **2. Environnement et développement :**

Pour développer un site web dynamique il faut installer un serveur web en local pour tester les scripts et un SGBD pour tester la connexion à la base de données.

Sous l'environnement Windows, on peut utiliser **EasyPHP** qui installe Apache, PHP, MySQL et aussi une interface d'administration appelée : phpMyAdmin.

### **3. Syntaxe de base du langage PHP :**

#### **Principe :**

Les scripts PHP sont généralement intégrés dans le code d'un document HTML :

- En HTML : `<? Code PHP ?>`
- En JavaScript : `<script language = "PHP"> Code PHP </script>`

#### **Activité : TP1**

#### **Constatation :**

Pour que le script soit interprété par le serveur, 4 conditions sont nécessaires:

- Le fichier contenant le code doit avoir l'extension **".php"**
- Le code PHP contenu dans le code HTML doit être délimité par les balises `<? et ?>`
- Le fichier php doit être enregistré dans le répertoire:

**C:\Program Files\EasyPHP\WWW**

- Le fichier php ne sera exécuté qu'à travers son adresse web.

#### **Remarque :**

Pour commenter une seule ligne on utilise `//` et pour commenter plusieurs lignes on utilise `/*` au début du bloc et `*/` à sa fin.

## 4. Les structures de données :

### a. Les constantes :

Pour définir une constante, on utilise la fonction `define()`.

#### Exemple :

Ecrire le code suivant :

```
<?
define("LSF", "Lycée Secondaire Feriana");
echo("Mon lycée est : ".LSF);
?>
```

L'opérateur "." permet la concaténation des contenus des objets.

### b. Les variables :

En PHP, les variables sont représentées par le signe \$ suivi par le nom de la variable.

Un variable est déclaré dès qu'on lui affecte une valeur (`$n = "5";`).

Pour manipuler les variables, on distingue les fonctions suivantes:

**settype** : permet de forcer la conversion du type d'une variable en utilisant la syntaxe suivante :

`int settype (string var, string type)`

**gettype** : retourne le type d'une variable : `string gettype(var);`

**isset** : permet de tester si une variable contient une valeur ou non.

#### Remarque :

La conversion de type se fait à l'aide de la fonction `settype` ou en utilisant la syntaxe suivante :

`$Nomvar = (Type) $ NomVar;`

Le type peut être Integer, double, string, array ou object.

#### Exemples :

`$var = 11;`

`$var = (string)$var;` --> var contient la chaîne "11".

On peut connaître le type d'une variable en utilisant l'une des fonctions suivantes :

`is_long()`, `is_double()`, `is_string()`, `is_array()` et `is_object()`.

### c. Les opérateurs :

Les opérateurs de calcul, de comparaison et logiques sont les mêmes que ceux déjà vu en JavaScript.

`$x = 5;`

`$x = $x*4` --> La variable x contient 20.

## 5. Les entrées/sorties:

### Les entrées en PHP :

Pour récupérer des données à partir d'une page PHP, deux solutions sont possibles :

1. Récupérer les informations saisies à travers les objets graphiques des formulaires :

On utilise l'une des deux syntaxes suivantes :

**`$_GET['nom_objet']` ou `$_POST['nom_objet']`**

La syntaxe à utiliser dépend de la méthode (**method**) de la balise `<FORM>` (`get` ou `post`).

**nom\_objet** : désigne la valeur de la propriété **name** de l'objet duquel on veut récupérer des données.

2. Récupérer des informations envoyées en paramètre à un autre fichier PHP selon la syntaxe suivante :

`http://adresse_url/nom_fichier.php? var1 = val2 & var2 = val2 ... & varn = valn`

Pour récupérer les informations au niveau du fichier PHP, on utilise la syntaxe :

`$var1 = $_GET['var1'] ... $varn = $_GET['varn']`

## Les sorties en PHP :

L'affichage se fait grâce à l'instruction `echo` qui permet d'afficher, des chaînes de caractères, le contenu des variables et des balises HTML.

### Exemple :

```
<?
$x = 5;
echo("<i><b>La valeur de x est : $x</b></i>")
?>
```

## 6. Les structures de contrôle : voir cours JavaScript

### a. Les structures conditionnelles:

### b. Les structures itératives:

## 7. Les types de données structurés en PHP :

### a. Les tableaux:

On distingue 2 types de tableaux:

- Les tableaux à indices de type entier.
- Les tableaux associatifs utilisent des indices de type chaînes de caractères.

### Remarques :

- Par défaut, l'indice du premier élément d'un tableau est 0.
- Il est possible de stocker des données de types différents dans un même tableau.

### L'initialisation d'un tableau :

Il y a plusieurs façons d'insérer des éléments dans un tableau :

```
$T[0] = "A";
$T = array ("A",1); --> $T[1] = "A";
```

### Le parcours d'un tableau :

Il est possible de déterminer l'élément actif d'un tableau à l'aide de la fonction **current()** et l'indice de l'élément actif à l'aide de la fonction **key()**.

Pour parcourir un tableau, on peut utiliser la boucle suivante:

```
While (list($key,$value)=each($nom_tab))
{
Echo "$value";
}
```

On distingue des autres fonctions pour parcourir un tableau :

fonction	description
<code>Reset ()</code>	Remet le pointeur interne de tableau au début
<code>next ()</code>	Avance le pointeur interne d'un tableau
<code>prev ()</code>	Recule le pointeur courant de tableau
<code>end ()</code>	Positionne le pointeur de tableau en fin de tableau
<code>Count ()/sizeof ()</code>	Compte le nombre d'éléments d'un tableau

### Le tri d'un tableau :

PHP dispose de plusieurs fonctions de trie :

Les fonctions **sort()** et **rsort()** permettent de trier un tableau par valeur croissante ou décroissante.

Les fonctions **ksort()** et **krsort()** permettent de trier un tableau par indice.

### b. Les chaînes de caractères:

Les fonctions de chaînes de caractères :

**strlen()** : retourne la longueur d'une chaîne --> `Int strlen (string $CH);`

**substr()** : permet de copier une chaîne à partir d'une autre --> `string substr (string $CH, int $d, int $t);`

**trim()**: efface les espaces blancs de début et de fin d'une chaîne --> `string trim(string $CH);`

**str\_replace()**: remplace toutes les occurrences d'une chaîne dans une autre.

--> `string str_replace(string CH1 , remplacement, string $CH):`

## 8. Les fonctions en PHP :

Une fonction peut être définie en utilisant la syntaxe suivante :

```
function nom_fonction ($var1, $var2, ...)
```

```
{  
  Instructions  
  [return $résultat;]  
}
```

## 9. Utilisation de PHP avec MySQL:

### a. Création d'une base de données:

- Lancer EasyPHP.
- Charger l'adresse `http://127.0.0.1/mysql/`
- Créer une nouvelle base de données nommée **"BD4SI"**.
- Créer une table **"Elèves"** ayant la structure suivante :
  - Numéro (entier, clé primaire)
  - Nom (chaîne de 20 caractères)
  - Classe (chaîne de 10 caractères)
  - Moyenne (réel)
- Insérer les enregistrements suivants dans la table :

Numéro	Nom	Classe	Moyenne
001	B. Ahmed	4SI3	11.52
002	Guesmi	4SI3	14.15
003	Mestouri	4SI1	10.25

### Remarque :

Pour créer une nouvelle base de données, on peut cliquer avec le bouton droit de la souris sur l'icône d'EasyPHP puis on choisit l'option "Administration".

Dans la fenêtre qui s'affiche, on clique sur l'icône "GESTION BDD".

### b. Les fonctions de gestion d'une BDD MySQL :

L'utilisation d'une BDD en MySQL nécessite 4 étapes :

#### Etape 1 : Connexion au serveur :

```
int mysql_connect (string hostname, string username , string password)
```

Cette fonction retourne un entier permettant de vérifier l'établissement de la connexion.

Par défaut, on écrit:

```
mysql_connect ("localhost", "root", "");
```

#### Exemple :

```
<?  
if(!mysql_connect('localhost','root','')){ Echo'Connexion Impossible'; exit();}  
else{ Echo'Connexion réussie<br>';}  
?>
```

L'instruction `Exit()`; peut être remplacé par `die('Connexion Impossible');`

## Etape 2 : Sélectionner une base:

**mysql\_select\_db (string database, resource link\_identifieur )**

Avec:

database : nom de la base de données MySQL.

link\_identifieur : identifiant de la connexion ouverte par mysql\_connect(). C'est un paramètre facultatif.

Cette fonction retourne vrai ou faux selon que l'opération réussit ou non.

Si on ne donne pas le paramètre link\_identifieur, la fonction utilise la dernière connexion ouverte.

**Exemple :** `mysql_select_db ("BD4SI")`

## Etape 3 : Exécution d'une requête sql:

**mysql\_query (string query)**

Avec:

query : nom de la requête ou chaîne de caractères contenant la requête

Cette fonction envoie au serveur MySQL une instruction SQL à exécuter.

## Etape 4 : Exploitation d'une requête SQL:

**Requête de sélection :** pour sélectionner (afficher) les résultats on utilise:

**mysql\_fetch\_array (int result, int result\_type)**

Cette fonction permet d'extraire les résultats sous forme d'un tableau associatif (indice non numériques)

Avec :

result\_type : paramètre facultative qui peut être :

- MYSQL\_NUM : dans un tableau à indices numériques uniquement.

- MYSQL\_ASSOC : dans un tableau à indices associatifs (alphanumériques) uniquement.

- MYSQL\_BOTH : dans un tableau à indices numériques et associatifs la fois.

**Remarque :**

- Si l'argument result\_type n'est pas été spécifié, MYSQL\_BOTH est considérée comme valeur par défaut.

- MySQL\_num\_rows(\$Resultat); retourne le nombre d'enregistrement qu'ont été retrouvés par la sélection.

**Exemple:**

```
<?
mysql_connect ("localhost", "root", "");
mysql_select_db ("BD4SI");
$Resultat = mysql_query ("select Nom, Moyenne From Elèves");
While ($tab = mysql_fetch_array ($Resultat))
{echo($tab["Nom"]."-----".$tab["Moyenne"]."<br>");
}
echo("Le nombre d'élèves est : ". MySQL_num_rows($Resultat));
?>
```

**Insertion de données dans une table :**

```
<?
mysql_connect ("localhost", "root", "");
mysql_select_db('BD4SI');
$requete="Insert into Elèves Values('05','Sami','4SI2','11.12')";
$Resultat=mysql_query($requete);
$erreur=mysql_error();
print($erreur);
?>
```

### Recherche d'un enregistrement :

```
<?
mysql_connect ("localhost", "root", "");
mysql_select_db ("BD4SI");
$requet = "select *From Elèves Where Moyenne >= 10";
$Resultat = mysql_query ($requet);
print(mysql_num_rows($Resultat).' élèves qui ont la moyenne.<br>');
while($tableau=mysql_fetch_array($Resultat))
{echo($tableau["Nom"].' ---> '.$tableau["Moyenne"].'<br>');
}mysql_close();
?>
```

**Remarque :** Pour fermer une base de données on utilise la commande Close().

### Application

On considère la base de données **Elèves** créée précédemment

Créer un script en PHP qui permet de tester la connexion à la base et afficher la liste des élèves ainsi que leurs moyennes et leurs numéros dans un tableau.

Créer un formulaire en PHP qui permet d'insérer d'autres élèves à la base.

#### La page Affichage.php :

```
<html><head><title>e</title></head>
<body>
<?
if($id = mysql_connect("localhost","root","") )
{
    if(mysql_select_db("BD4SI") )
        echo "Succès de connexion<br>";
    else
        die("Echec de connexion à la base.");
}
else die("Echec de connexion au serveur de base de données.");

$requete ="select * from Elèves ";
$resultat = mysql_query($requete);
?>
<table border = 1>
    <tr> <td>Nom</td> <td>Moyenne</td> </tr>
<?
while($ligne = mysql_fetch_array($resultat))
{
?>
    <tr>
        <td><? echo($ligne["Nom"]); ?></td>
        <td><? echo($ligne["Moyenne"]); ?></td>
    </tr>
<?
}
?>
</table>
</body></html>
```



### La page Inscription.html :

```
<html><head><title>Inscription</title></head>
<body>
<form id="form1" name="form1" method="post" action="insert.php">
  <p>Numéro: <input type="text" name="Numéro" /> </p>
  <p>Nom: <input type="text" name="Nom" /> </p>
  <p>Classe: <input type="text" name="Classe" /> </p>
  <p>Moyenne: <input type="text" name="Moyenne" /> </p>
  <p> <input type="submit" name="Submit" value="Envoyer" />
  <input type="reset" name="effacer" value="Effacer" /> </p>
</form>
<p><a href="Affichage.php">Liste des élèves </a></p></body></html>
```

### La page Insert.php :

```
<?
if(mysql_connect("localhost","root",""))
{
  if(mysql_select_db("BD4SI"))
    echo "Succès de connexion<br><br>";
  else die("Echec de connexion à la base."); }
else die("Echec de connexion au serveur de base de données.");

$Num=$_POST["Numéro"];
$Nom=$_POST["Nom"];
$Classe=$_POST["Classe"];
$Moy=$_POST["Moyenne"];

$requete ="insert into Elèves values ('$Num', '$Nom', '$Classe', '$Moy')";
if(mysql_query($requete)) echo'insertion correct'; else echo "Erreur d'insertion";
?>
```