

PRODUCTION ELECTRONIQUE AVANCEE

LE LANGUAGE PHP

Objectifs

- Créer des pages web dynamiques en utilisant le langage Php ;

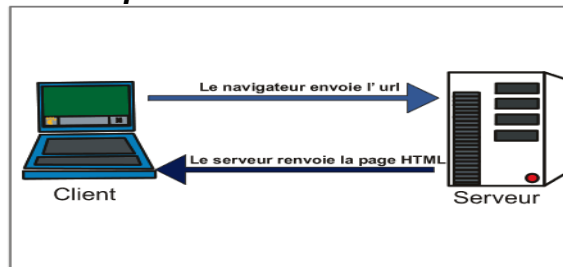
I. Introduction

I.1. Qu'est ce qu'un site web dynamique :

C'est un site dont les pages peuvent être générées dynamiquement en fonction d'une demande d'un utilisateur. La construction de sites dynamiques repose sur des technologies de script coté serveur telles que PHP, ASP, etc. Dans le monde de logiciels libres, il s'agit souvent de PHP pour le langage de script et MySQL pour la base de données.

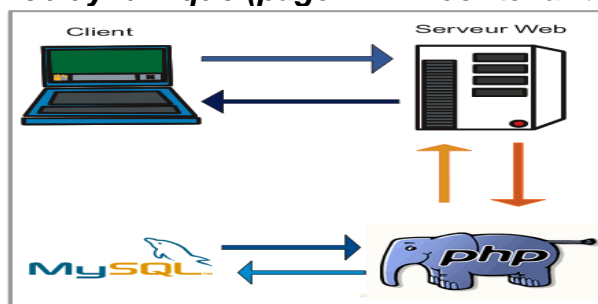
I.2. Principe de fonctionnement :

a) Cas d'un site web statique



- Le navigateur envoie l'URL tapée par l'utilisateur ;
- Le serveur web va chercher dans son arborescence la page demandée en utilisant un logiciel serveur web (apache par exemple) capable de traiter les requêtes http et renvoi la page HTML au navigateur ;
- Le navigateur interprète les différents langages se trouvant dans le fichier (HTML, JavaScript, etc.) et affiche la page.

b) Cas d'un site web dynamique (page HTML contenant du code PHP)



- Le navigateur envoie l'URL tapée ;
- Le serveur web cherche dans son arborescence si le fichier existe, et porte une extension reconnue comme une application PHP. Si c'est le cas, il transmet ce fichier à PHP.
- PHP parse le fichier (analyse et exécute le code PHP) puis retourne le fichier dépourvu du code PHP au serveur web.
- Le serveur web renvoie un fichier ne contenant plus de PHP, donc seulement du HTML au navigateur qui l'interprète et l'affiche.

II. Le langage Php

II.1. Présentation :

PHP signifie **Hypertext Preprocessor** : C'est un langage dit de script qui s'inclut dans du HTML et qui est traité côté serveur et non côté client conçu pour des applications Internet et gérer la communication avec les bases de données.

II.2. Environnement de développement :

Pour le développement d'un site web dynamique, il faut installer en local un serveur Web qui sert pour tester les scripts et un SGBD pour tester la connexion à la base de données utilisée et tester les requêtes de manipulation des données de la base. Il existe un utilitaire pratique : EasyPhP qui installe le serveur Web Apache, Php, MySQL et phpMyAdmin (interface gratuite pour la gestion des bases de données MySQL).

Activité 1 : Découvrir l'interface de EasyPhp.

II.3. Syntaxe de base du langage Php :

Activité 2 : Premier code php

1- Lancez EasyPhp

2- Créer un répertoire sous le nom « 4SI1G1/G2 » sous le répertoire racine du serveur web apache (c:\program files\EasyPHP1-8\www)

3- Créer un fichier texte nommé « tp1 » avec l'extension php et saisir le code suivant :

```
<html>
<head>
<title>Premiere page en php</title>
</head>
<body>
<?
echo("<b>Bienvenue au langage php</b><br>");
//echo permet d'afficher un message
?>
</body>
</html>
```

4- Ouvrez le fichier

Remarques :

- Le code php doit être délimité par les balises
- Le fichier php doit être enregistré dans le répertoire d'hébergement du serveur apache (c:\program files\EasyPHP1-8\www) ;
- Le fichier n'est exécuté qu'à travers son adresse web : (<http://localhost/>... Ou http://nom_machine/... Ou <http://127.0.0.1/>...);
- Pour commenter une ligne : (*//*), plusieurs lignes (*/*...*/*).

II.4. Les structures de données :**a) Les constantes**

- ❖ **Activité** : Créer un fichier texte avec l'extension php « tp2.php » et saisir le code suivant :

```
<html>
<head>
<title>Les constantes</title>
</head>
<body>
<?
define("PI","3.14");
echo("La valeur de PI est :".PI);
?>
</body>
</html>
```

- ❖ **Constatation:**

.....

.....

.....

b) Les variables

- ❖ **Activité** : Créer un fichier texte avec l'extension php « tp3.php » et saisir le code suivant :

```
<html>
<head>
<title>Les variables</title>
</head>
</body>
<?
//utilisation des variables
$libelle="Disque dur" ;
$PU=120.500 ;
$description= "" ;
echo (gettype($libelle). "<br> ".gettype($PU). "<br>") ;
//Conversion de types
settype($PU, "integer") ;
echo ("le type de la variable PU est : ".gettype($PU). " est le contenu de PU est : ".$PU.
"<br>") ;
$PU = (string)$PU ;
echo ("le type de la variable PU est : ".gettype($PU). " est le contenu de PU est : ".$PU.
"<br>") ;
//test d'existence de variables
echo ("test1 : ".isset($x). "<br>");
echo ("test2 : ".isset($PU). "<br>") ;
?>
</body>
</html>
```

✚ **Constatation:**

❖ Déclaration de variables :

-
- Une variable est automatiquement déclarée dès que vous lui attribuez une valeur ;

❖ Conversion de types :

- Avec la fonction « **settype** »

Syntaxe :

Exemple :

- En précédant la variable à convertir par des clauses (type)

Exemple : \$x=12.5

\$x=(integer)\$x // c'est un entier qui vaut 12

❖ Fonctions de manipulation de variables :

- **string gettype(var)** :
- **int isset (var)** :

c) **Les opérateurs**

- ❖ Les opérateurs de calculs : + ; - ; * ; / ; = (affectation) ; % (opérateur modulo) ;
- ❖ Les opérateurs de comparaison : = ; < ; > ; <= ; >= ; != (différence) ;
- ❖ Les opérateurs logiques : || ou OR ; && ou AND ; XOR ; ! (non logique).

II.5. Les entrées/sorties :

Activité :

1. Ouvrir le fichier nommé « formulaire.html » :
2. Créer un fichier nommé « enregistrer.php » et saisir le code suivant :

```
<?
Echo("<center><b>Merci pour l'enregistrement</b></center><br>");
$nom=$_POST['nom'];
$prenom=$_POST['prenom'];
Echo("Bienvenue Mr : ".$nom. " ".$prenom);
?>
```

3. Modifier la méthode « POST » par la méthode « GET » dans le fichier « formulaire.html ».
4. Modifier le code de la page « enregistrer.php » par le code suivant :

```
<?
Echo("<center><b>Merci pour l'enregistrement</b></center><br>");
$nom=$_GET['nom'];
$prenom=$_GET['prenom'];
Echo("Bienvenue Mr : ".$nom. " ".$prenom);
?>
```

Constatations :

- ❖ **Les entrées** : Pour récupérer les données à partir des informations saisies à travers les objets des formulaires :

-
-

Dépend de la valeur de la propriété METHOD de la balise FORM (GET ou POST).

NB :

.....

.....

- ❖ **Les sorties** : l'affichage se fait grâce à l'instruction « echo »

Exemple :

NB : il est possible d'afficher des informations à travers les objets graphiques d'un formulaire :

Exemple : <input type="text" name="np" value=<?echo("\$np");?>>

Constatations :

- ✓ En php, deux types de tableaux sont utilisés :
 - Tableau à indices de type.....
 - Tableau associatif : qui utilise des indices de type
- ✓ Il est possible de stocker des éléments de dans le tableau .
- ✓ L'initialisation d'un tableau
 - Pour les tableaux a indice de type entier , on utilise la syntaxe suivant :
 - Syntaxe** :
 - Exemple** :.....
 - Pour les tableaux associatif ,on utilise la syntaxe suivant
 - Syntaxe** :
 - Exemple** :.....
- ✓ Parcours d'un tableau associatif :
 - Tout tableau possède un pointeur interne qui conserve l'indice et la valeur de l'élément actif
 - Pour déterminer l'indice et la valeur de la case courante on peut utiliser :
 - La fonction **Key()** ;
 -
 - syntaxe** :
 - exemple** :.....
 - La fonction **current()**;
 -
 - syntaxe** :
 - exemple** :.....
 - Pour parcourir un tableau
 - Les deux fonctions **each()** et **list()** sont utilisées conjointement afin de parcourir un tableau.
 - syntaxe** :
 - exemple** :.....
- ❖ Le tri d'un tableau :
 - tri par clé : on utilise les fonctions ksort() et rsort() en ordre croissant et décroissant.
 - syntaxe** :
 - exemple** :.....
 - tri par valeur : on utilise les fonctions sort() et rsort() en ordre croissant et décroissant ;
 - syntaxe** :
 - exemple** :.....
- ❖ la fonction sizeof(\$nomtableau) retourne la taille du tableau
- ❖ les autres fonctions : voir livre page 131

b) Les chaînes de caractères

❖ **Activité** : compléter le tableau suivant :

Nom	Syntaxe	rôle	exemple
Chr	\$var=chr(x) ;		\$v=chr(65); echo (\$v) ;.....
Ord	\$var=ord(c) ;		\$v=ord("a"); echo (\$v) ;.....
strlen	\$var=strlen(chaine) ;		\$v=strlen("bac"); echo (\$v) ;.....
substr	\$v=substr(chaine, debut, nbc) ;		\$v=substr("informatique",2,5); ; echo (\$v) ;.....
trim	\$v=trim(chaine) ;		\$v=trim(" info "); echo (\$v) ;.....
strcmp	\$v=strcmp(ch1,ch2);		\$v=strcmp("info","pascal"); echo (\$v) ;.....
strtolower	\$v=strtolower(chaine);		\$v=strtolower("PASCAL"); echo (\$v) ;.....
str_replace	\$v=str_replace(s_ch1,s_ch2,\$ch)		\$chaine="tout est rouge"; \$ch=str_replace("rouge","bleu",\$chaine); Echo(\$chaine);.....
implode	\$v=implode(chaine,nomtableau);		\$t=array(1,2,3); \$v= implode("",\$t); echo (\$v) ;
explode	\$tableau=explode(\$ch1,\$ch2);		\$t="a*b*c"; \$v= explode("",\$t); echo (\$v[2]) ;

II.8. Les fonctions en php :

Une fonction est définie par la syntaxe suivante :

```

Function nom_fonction($argument1,$argument2,...,$argument_n)
{
Corps de la fonction
[return $valeur_retour ;]
}

```

❖ **Activité :** créer un fichier nommé « fonction.php » et saisir le code suivant :

```

<html><head><title>Les fonctions</title>
</head>
<body>
<?
function carree($nombre)
{
$c=$nombre*$nombre ;
return $c ;
}
Echo ("le carree de 3 est :".carree(3)) ;
?>
</body></html>

```

II.9. Utilisation de MySQL avec PHP :**a) Création d'une base de données avec MySQL**

Activité :

1. Lancer EasyPHP puis démarrer PhpMyadmin et créer la base de données « gestion des voitures »
2. Créer une table « voiture » ayant la structure suivante :
 - mat_voiture (chaîne de caractères, clé primaire)
 - marque (chaîne de caractères)
 - puissance(entier)
 - nbre_km (chaîne de caractères)
- 3 . ajouter le champs suivant : prix (entier)
4. ajouter les lignes suivantes a la table voiture.

mat_voiture	marque	puissance	nbre_km	prix
152 Tu 52	Golf	120	133	120000
100 Tu 1425	Opel	140	1500	7800
90 Tu 471	Fiat	150	1450	5600

3^{ème} étape : Exécution d'une requête SQL :

Envoie au serveur mysql une requête SQL à exécuter.

Syntaxe :

Exemple :

➤ **Remarque**

On peut vérifier si l'exécution de la requête est effectuée ou non par l'une de ces deux fonctions :

- ✓ **or die("message") :**
- ✓ **mysql_affected_rows() :**

4^{ème} étape : Exploitation d'une requête SQL :

➤ **Insertion de données**

Syntaxe :

Exemple :

➤ **Modification de données**

Activité : créer un fichier « modifier.php » qui permet de recevoir les information envoyer par le fichier « modification.html » puis de modifier le nombre des kilométrage d'une voiture donne.

❖ **Réponse :**

❖ **Constatations :**

Syntaxe :

❖ **Constatations :**✓ **Syntaxe de la requête de sélection :**

.....
.....

- Chaque ligne retournée par une requête de sélection est enregistré dans la mémoire.
 - Pour extraire une ligne de la requête vers un tableau associatif il faut utiliser la fonction **mysql_fetch_array(\$requete_resultat)**

✓ **Syntaxe :**

.....
.....

Remarque :

Lorsque la requête retourne plusieurs lignes on doit utiliser la boucle while pour extraire les lignes une par une.

Exemple :

.....
.....

- La fonction **mysql_num_rows(\$resultat)** permet d'afficher le nombre total des lignes retournées par la requête .

Exemple :

.....
.....

- La ligne sélectionnée est stockée dans un tableau associatif avec les indices qui représentent le nom des champs dans la table. **\$nom_tableau["nom_champs"]**

.....

Créer un fichier nommé « affichage.php » et saisir le code suivant :

```
<html>
<body>
<?
mysql_connect("localhost","root","");
mysql_select_db("inscription");
$requete="select * from personne";
$resultat=mysql_query($requete);
echo mysql_num_rows($resultat);
?>
<table border="1">
<tr>
<td>numero cin</td>
<td>nom</td>
<td>prenom</td>
<td>ville</td>
<td>Email</td>
<td>HTML</td>
<td>javascript</td>
<td>PHP</td>
</tr>
<?
while($i=mysql_fetch_array($resultat))
{
```



```
?>
<tr>
<td><? echo $i["Num_CIN"];?></td>
<td><? echo $i["Nom"];?></td>
<td><? echo $i["Prenom"];?></td>
<td><? echo $i["Ville"];?></td>
<td><? echo $i["Email"];?></td>
<td><? echo $i["PHP"];?></td>
<td><? echo $i["javascript"];?></td>
<td><? echo $i["PHP"];?></td>
</tr>
<?
}
?>
</table>
</body>
</html>
```

A la suite d'une requête de sélection, les données sont mises en mémoire. Pour pouvoir les exploiter, php gère un pointeur de résultat qui permet de repérer un enregistrement parmi les autres et après lecture le pointeur se déplace vers l'enregistrement suivant.

La fonction de lecture est : `Mysql_fetch_array(resultat, resultat_type)`

Exemple :

```
$i=mysql_fetch_array($resultat)
```

Le paramètre `resultat_type` est facultatif. Il peut prendre les valeurs suivantes :

- `MYSQL_NUM` : le tableau contient des indices numériques ;
- `MYSQL_ASSOC` : le tableau contient des indices associatifs ;
- `MYSQL_BOTH` : le tableau contient à la fois des indices numériques et associatifs.

Remarques :

- Si l'argument `resultat_type` n'est pas indiqué, `MYSQL_BOTH` est considéré comme valeur par défaut de cet argument.
- `Mysql_num_rows(resultat)` retourne le nombre d'enregistrements qui ont été retourné par la sélection.

➤ **Recherche de lignes**

La recherche de lignes se fait à travers la requête SQL suivante :

```
SELECT liste_nom_colonnes FROM nom_table WHERE condition;
```

Application: Créer une page nommée « recherche.html » dans le même répertoire contenant un formulaire permettant de saisir le numéro de CIN d'une personne, en cliquant sur le bouton valider on fait appel à la page « resultat_recherche.php » qui affiche les données concernant cette personne.

- Code de la page « recherche.html » :

```
<html><body>
<h1><center>Veuillez saisir votre CIN</center></h1><br>
<form name="f2" action="resultat_recherche.php" method="GET">
Donner le numéro de la CIN<input type="text" name="z1"><br>
<input type="submit" value="valider" name="v1">
<input type="reset" value="annuler" name="r1">
</form></body></html>
```

- Code de la page « resultat_recherche.php » :

```

<html>
<body>
<?
$z1=$_GET['z1'];
mysql_connect("localhost","root","") or die("impossible de se connecter au serveur<br>");
mysql_select_db("inscription") or die ("impossible de trouver la base de données<br>");
echo("connexion au serveur et a la base reussit<br>");
$requete="select * from personne where Num_CIN='$z1'";
$resultat=mysql_query($requete) or die ("impossible d'exécuter la requête<br>");
$Num=mysql_num_rows($resultat);
if($Num==0)
echo (" aucune personne ne correspond à cette CIN<br>");
else
{
?>
<table border="1">
<tr>
<td>numero cin</td>
<td>nom</td>
<td>prenom</td>
<td>ville</td>
<td>Email</td>
<td>HTML</td>
<td>javascript</td>
<td>PHP</td>
</tr>
<?
while($i=mysql_fetch_array($resultat))
{
?>
<tr>
<td><? echo $i["Num_CIN"];?></td>
<td><? echo $i["Nom"];?></td>
<td><? echo $i["Prenom"];?></td>
<td><? echo $i["Ville"];?></td>
<td><? echo $i["Email"];?></td>
<td><? echo $i["HTML"];?></td>
<td><? echo $i["javascript"];?></td>
<td><? echo $i["PHP"];?></td>
</tr>
<?
}
?>
</table>
<?
}
?>
</body>
</html>

```

➤ **La suppression de lignes**

La suppression de lignes se fait à travers la requête SQL suivante :

```
DELETE FROM nom_table WHERE condition;
```

Application: Créer une page nommée « suppression.html » dans le même répertoire contenant un formulaire permettant de saisir le numéro de CIN d'une personne, en cliquant sur le bouton valider on fait appel à la page « resultat_suppression.php » qui permet de supprimer les données concernant cette personne.

- Code de la page « suppression.html » :

```
<html>
<body>
<h1><center>Veuillez saisir la CIN correspondante</center></h1><br>
<form name="f2" action="resultat_suppression.php" method="GET">
Donner le numéro de la CIN<input type="text" name="z1"><br>
<input type="submit" value="valider" name="v1">
<input type="reset" value="annuler" name="r1">
</form>
</body>
</html>
```

- Code de la page « resultat_suppression.php » :

```
<html>
<body>
<?
$z1=$_GET['z1'];
mysql_connect("localhost","root","") or die("impossible de se connecter au serveur<br>");
mysql_select_db("inscription") or die ("impossible de trouver la base de données<br>");
echo("connexion au serveur et a la base réussit<br>");
$req="select * from personne where Num_CIN='$z1'";
$res=mysql_query($req);
$i=mysql_num_rows($res);
if($i>0)
{
$requete="delete from personne where Num_CIN='$z1'";
$resultat=mysql_query($requete);
echo ("suppression effectuée");
}
else
echo("suppression non effectuée, personne inexistante");
?>
</body>
</html>
```

➤ **Modification de données**

La modification de données se fait à travers la requête SQL suivante :

```
Update nom_table SET nom_colonne1 = valeur, nom_colonne2=valeur, ... WHERE (Conditions) ;
```

Application:

1. Créer une page nommée « modif_personne.html » dans le même répertoire contenant un formulaire permettant de saisir le numéro de CIN de la personne à modifier ses données, en cliquant sur le bouton valider on fait appel à la page « form_modification.php » qui présente un formulaire identique au formulaire d'inscription dans lequel on affiche les données de la personne à modifier ses données.

- Code de la page « modif_personne.html » :

```
<html>
<body>
<h1><center>Veuillez saisir la CIN</center></h1><br>
<form name="f2" action="form_modification.php" method="GET">
Donner le numéro de la CIN<input type="text" name="z1"><br>
<input type="submit" value="valider" name="v1">
<input type="reset" value="annuler" name="r1">
</form>
</body>
</html>
```

- Code de la page « form_modification.php » :

```
<html>
<body>
<?
$z1=$_GET['z1'];
mysql_connect("localhost","root","") or die("impossible de se connecter au serveur<br>");
mysql_select_db("inscription") or die ("impossible de trouver la base de données<br>");
echo("connexion au serveur et a la base reussit<br>");
$requete="select * from personne where Num_CIN='$z1';";
$resultat=mysql_query($requete) or die ("impossible d'executer la requete<br>");
$Num=mysql_num_rows($resultat);
if($Num==0)
echo (" aucune personne ne correspond à votre critère de recherche<br>");
else
while($i=mysql_fetch_array($resultat))
{
?>
<b>Affichage du resultat</b><br>
<form name="f3" action="modifier.php" method="GET">
Num_CIN:<input type="text" name="a" value=<? echo $i["Num_CIN"];?>><br>
Nom:<input type="text" name="b" value=<? echo $i["Nom"];?>><br>
Prenom:<input type="text" name="c" value=<? echo $i["Prenom"];?>><br>
Ville:<input type="text" name="d" value=<? echo $i["Ville"];?>><br>
Email:<input type="text" name="e" value=<? echo $i["Email"];?>><br>
HTML:<input type="text" name="f" value=<?echo $i["HTML"];?>><br>
javascript:<input type="text" name="g" value=<? echo $i["javascript"];?>><br>
PHP:<input type="text" name="h" value=<? echo $i["PHP"];?>><br>
Modification des modules:<br>
<input type="checkbox" name="i" value="1">HTML<br>
<input type="checkbox" name="j" value="1">javascript<br>
<input type="checkbox" name="k" value="1">PHP<br>
<input type="submit" value="modifier">
<?
}
?>
</form>
</body>
</html>
```

2. Après modification des données et en cliquant sur le bouton modifier de la page «form_modification.php » on fait appel à la page « modifier.php » qui va appliquer les modifications dans la base de données.

- Code de la page « modifier.php » :

```
<?
$a=$_GET['a'];
$b=$_GET['b'];
$c=$_GET['c'];
$d=$_GET['d'];
$e=$_GET['e'];
if(isset($_GET['i']))
$i=true;
else
$i=false;
if(isset($_GET['j']))
$j=true;
else
$j=false;
if(isset($_GET['k']))
$k=true;
else
$k=false;
$machine="localhost";
$utilisateur="root";
$mot_de_passe="";
$base_de_donnee ="inscription";
mysql_connect($machine,$utilisateur,$mot_de_passe) or die("Impossible de se connecter
au serveur web<br>");
mysql_select_db($base_de_donnee) or die("Impossible de se connecter à la base de
données formation<br>");
echo("Connexion au serveur web et à la base de données réussite<br>");
$requete="update personne set
Nom='$b',Prenom='$c',Ville='$d',Email='$e',HTML='$i',javascript='$j',PHP='$k' where
Num_CIN='$a';";
$resultat=mysql_query($requete) or die("<br>Erreur : modification echouée !!!");
echo("<br>modification effectuée avec succès<br>");
?>
```