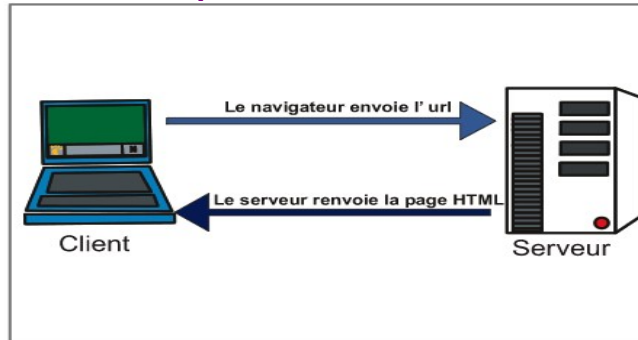


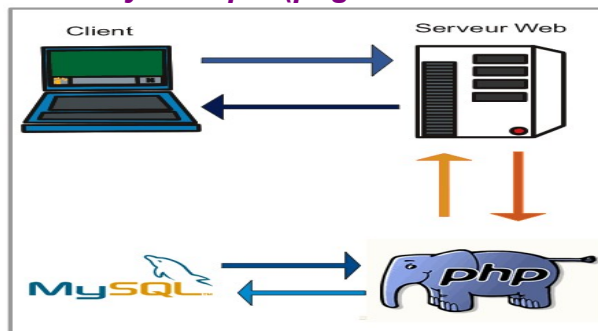
I. Introduction

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 **Hypertexte 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
- ❖ 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).

II.3. Syntaxe de base du langage Php :

Activité 1 : Premier code php

1- Lancez EasyPhp

2- Créer un répertoire sous le nom « 4SI » sous le répertoire racine du serveur web apache (c:\program files\EasyPHP\www)

3- Créer un fichier texte nommé « act1 » avec l'extension php et saisir le code suivant :

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

4- Ouvrez le navigateur avec l'adresse : <http://localhost/4SI/act1.php>

Remarques :

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

II.4. Les structures de données :

a) Les constantes

Act2.php

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

Remarques:

- Pour définir une constante, on fait appel à la fonction define();
- L'opérateur « . » permet la concaténation des contenus des objets.

b) Les variables

Act3.php	Act4.php
<pre><html><head><title>getType et setType</title></head> <body> <?php setType(\$toto, "integer") ; echo getType(\$toto) ; //Renvoie integer echo getType(\$toto); \$i=2+2; echo getType(\$i);//Renvoie integer \$i="j"j'aime le chiffre ".\$i; echo getType(\$i) ;//renvoie string ?> </body></html></pre>	<pre><html><head><title>isSet et unSet</title></head> <body> <?php echo isSet(\$toto);//Renvoie FALSE \$toto=""; echo isSet(\$toto) ;//Renvoie TRUE unSet(\$toto) ; echo isSet(\$toto);//Renvoie FALSE ?> </body></html></pre>

Remarques:

- ❖ Déclaration de variables : Les variables sont représentées par un signe dollar « \$ » suivi du nom de la variable ;
- ❖ Fonctions utilisées avec les variables :
 - setType() : affecte un type à une variable.
 - getType() : retourne le type de la variable.
 - isSet() : Détermine si une variable existe (a été initialisée).
 - unSet() : Détruit les variables.
- ❖ On peut convertir le type d'une variable en l'a précédant par des clauses (type)

Exemple : \$x=12.5

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

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 :

Act5 :

1. créer un fichier nommé « formulaire.html » et saisir le code suivant :

```
<html>
<body>
<center><h1>Inscription</h1></center><br>
<form name= "f1" action= "enregistrer.php" method="POST">
Nom:<input type="text" name="nom"><br>
Prenom:<input type="text" name="prenom"><br>
<center><input type="submit" value="valider"><input type="reset">
</form>
</body>
</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 :

- \$variable=\$_GET['variable']
- \$variable=\$_POST['variable']

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

❖ **Les sorties :** l’affichage se fait grace à l’instruction « echo »

Exemple : echo ("La valeur de x est : \$x);

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");?>>

II.6. Les structures de contrôle :

a) Les structures conditionnelles :

L’instruction IF :	Les structures de branchement :
<pre>If(condition) { instructions1 } Else { instructions2 }</pre>	<pre>Switch (expression) { Case resultat1: instructions 1 Break; Case resultat2: instructions 2 Break; ... Default : instructions default }</pre>

b) Les structures itératives :

La boucle while:	La boucle do..while	La boucle for :
<pre>While (condition) { Instructions }</pre>	<pre>Do { Instructions } while (condition)</pre>	<pre>For (expression1;expression2;expression3) { Instructions }</pre>

II.7. Les types de données structurées en php :

a) Les tableaux

En php, deux types de tableaux sont utilisés :

- Tableau à indices de types entier (l'indice du premier élément du tableau est 0) ;
- Tableau associatif : qui utilise des indices de type chaînes de caractères.

Avec php, il est possible de stocker des éléments de types différents dans un même tableau.

Act6 : Créer un fichier nommé « Tableau.php » est saisir le code suivant :

```
<?
//initialisation du tableau
$tab=array("c","d","a","e","b");
//accès aux elements du tableau
echo("<br>l'element d'indice 1 est : ".$tab[1]."<br>");
echo("-----<br>");
list($cle,$valeur)=each($tab); //each() renvoi une paire clef-valeur du tableau donné et passe
au suivant
echo("l'element courant $cle; est=$valeur;<br>");
echo("element suivant est :<br>");
$cle=key($tab); //retourne l'indice de l'element courant du tableau
$valeur=current($tab); //retourne la valeur l'element courant du tableau
echo("$cle:$valeur;<br>");
echo("-----<br>");
//afficher les elements d'un tableau
reset($tab); //remet le pointeur au debut du tableau
while(list($cle,$valeur)=each($tab))
{
echo("<br>$cle:$valeur");
}
echo("<br>-----<br>");
//tri par valeur
reset($tab);
rsort($tab); //tri le tableau par valeurs croissantes
while(list($cle,$valeur)=each($tab))
{
echo("<br>$cle:$valeur");
}
echo("<br>-----<br>");
//tri par cle
reset($tab);
krsort($tab); //tri le tableau par indices décroissants
while(list($cle,$valeur)=each($tab))
{
echo("<br>$cle:$valeur");
}
echo("<br>-----<br>");
?>
```

Constatations :

❖ **Initialisation d'un tableau :**

Exemple1 : \$tab[0]= 2 ; \$tab[1]="A" ;...

Exemple2 : \$tab=array(2, "A", ...);

Exemple3 : tableau associatif

\$nom_tableau=array("lundi"=>2,"mardi"=>"A",...);

❖ **Parcours d'un tableau :**

Tout tableau possède un pointeur interne qui conserve l'indice et la valeur de l'élément actif. Pour déterminer la valeur de l'élément actif on utilise la fonction **current()**, et l'indice de l'élément actif on utilise la fonction **key()**.

Exemple :

\$cle=key(\$tab);

\$valeur=current(\$tab);

Echo("l'élément courant \$cle = \$valeur");

Les deux fonctions each() et list() sont utilisées conjointement afin de parcourir un tableau :

- each(\$nom_tableau) : retourne la paire cle/valeur courante du tableau et avance le pointeur du tableau ;
- la fonction list(\$clé,\$valeur) : construit un tableau temporaire à partir des variables scalaires passées en argument.
- La fonction reset(\$nom_tableau) : remet le pointeur interne du tableau au début

Exemple :

```
While(list($cle,$valeur)=each($tab))
```

```
{
```

```
Echo("<br>$cle:$valeur");
```

```
}
```

❖ **Le tri d'un tableau :**

- tri par valeur : on utilise les fonctions sort() et rsort() en ordre croissant et décroissant ;
- tri par clé : on utilise les fonctions ksort() et krsort() en ordre croissant et décroissant.

b) Les chaînes de caractères

Activité : créer un fichier nommé « chaine.php » et saisir le code suivant :

```
<?
//traitement de chaînes de caractères
echo(substr("tunisie",-2)); //retourne les deux derniers caractères de la chaîne
echo("<br>");
echo(substr("tunisie",-5,3)); //retourne les 3 caractères de la chaîne à partir de la position -5
echo("<br>");
echo(trim(" exemple chaîne ")); //retourne une chaîne sans espaces
echo("<br>");
$chaîne="nom|prenom|adresse";
$champs=explode("|",$chaîne); //Retourne un tableau de chaînes, chacune d'elle étant une
sous-chaîne du paramètre string extraite en utilisant le séparateur delimiter .
$compteur=0;
while($compteur<sizeof($champs))
```

```
{
echo $champs[$compteur];
echo("<br>");
$compteur++;
}
echo("<br>");
$chaine="tout est rouge";
$chaine=str_replace("rouge","bleu",$chaine); //remplace une chaine par une autre
echo $chaine;
?>
```

Constatations :

Les fonctions de chaîne de caractères :

- ❖ **substr(chaine source, debut, taille)** : retourne une partie de la chaîne source depuis debut et de longueur taille (lorsque une taille négative est spécifiée, la chaîne renvoyée se terminera à cette distance de la fin de la chaîne source).
- ❖ **Trim(chaine)** : retire les espaces blancs de début et de fin de chaîne et retourne la chaîne nettoyée.
- ❖ **Strlen(chaine)** : retourne la longueur de la chaîne.
- ❖ **Implode(séparateur, nom_tableau)** : retourne une chaîne constituée de tous les éléments du tableau séparées par le séparateur.
- ❖ **Explode(séparateur, chaine)** : retourne un tableau qui contient les éléments de la chaîne séparée par le séparateur.
- ❖ **Str_replace(modèle, remplacement, chaine)** : remplace toutes les occurrences de modèle par remplacement dans la chaîne

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 ;]
}
```

Exemple :

```
<?
function carree($nombre)
{
$c=$nombre*$nombre ;
return $c ;
}
Echo ("le carree de 3 est :".carree(3)) ; //affiche la valeur 9
?>
```

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 « inscription »
2. Créer une table « personne » ayant la structure suivante :
 - Num_CIN (chaîne de caractères, clé primaire)

- Nom (chaîne de caractères)
- Prenom (chaîne de caractères)
- Ville (chaîne de caractères)
- Email (chaîne de caractères)
- HTML (booléen)
- Javascript (booléen)
- PHP (booléen)

b) Les fonctions pour la gestion d'une base de données MySQL

L'utilisation de MySQL avec PHP s'effectue en 4 étapes :

- Connexion au serveur de données ;
- Sélection de la base de données ;
- Exécution de la requête ;
- Exploitation des résultats de la requête.

1^{er} étape : Connexion au serveur de données :

mysql_connect (nom_machine, nom_utilisateur, mot_de_passe)

Pour se connecter, il faut définir l'adresse du serveur ainsi que le nom d'utilisateur et le mot de passe. La fonction retourne un entier permettant de vérifier l'établissement de la connexion.

Exemple :

```
$machine="localhost";
```

```
$utilisateur="root";
```

```
$mot_de_passe="";
```

```
mysql_connect($machine,$utilisateur,$mot_de_passe) or die("Impossible de se connecter au serveur web<br>");
```

2^{ème} étape : Sélection de la base de données :

Mysql_select_db (nom_base)

La fonction retourne true ou false selon que l'opération réussit ou non.

Exemple :

```
$base_de_donnee ="inscription";
```

```
mysql_select_db($base_de_donnee) or die("Impossible de se connecter à la base de données inscription<br>");
```

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

Mysql_query(string query)

Envoie au serveur mysql une instruction SQL à exécuter.

Exemple :

```
$resultat=mysql_query($requete) or die("<br>Erreur : Insertion de données échouée !!!");
```

```
echo("<br>enregistrement effectue avec succes<br>");
```

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

➤ **Insertion de données**

Exemple :

```
$requete="INSERT INTO personne
```

```
(Num_CIN,Nom,Prenom,Ville,Email,HTML,javascript,PHP)
```

```
VALUES ('$z1','$z2','$z3','$z4','$z5','$z6','$z7','$z8');";
```

➤ **Requête de sélection**

Exemple : Affichage des enregistrements de la table « personne »
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) ;