

Pages Web dynamiques

I- Introduction :

1- Site Web dynamique :

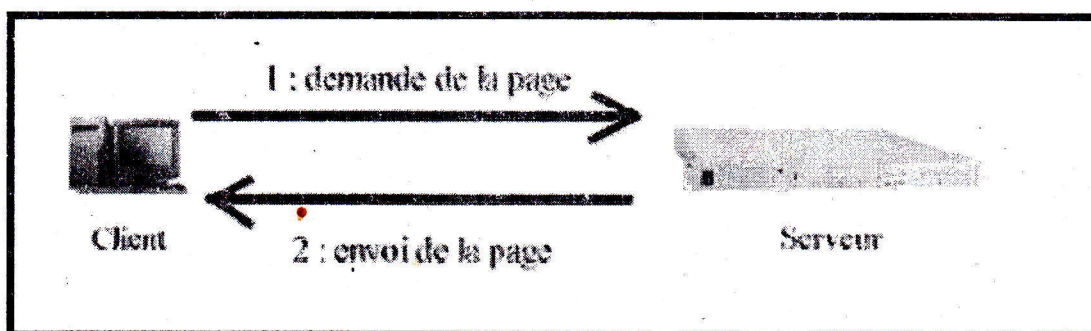
Un site Web dynamique est un site dont les pages peuvent être générées à « la volée » dynamiquement en fonction d'une demande d'un utilisateur.

Le contenu de la présentation des pages peuvent être gérés indépendamment. Ainsi, on définit un modèle de présentation dans lequel peuvent être « dynamiquement » un contenu extrait le plus souvent d'une base de données.

La construction de sites dynamique repose sur des technologies de scripts côté serveur telles que le Php, Asp,..., il s'agit souvent de Php pour le langage de script et MySQL pour la base de données.

2- Principe de fonctionnement :

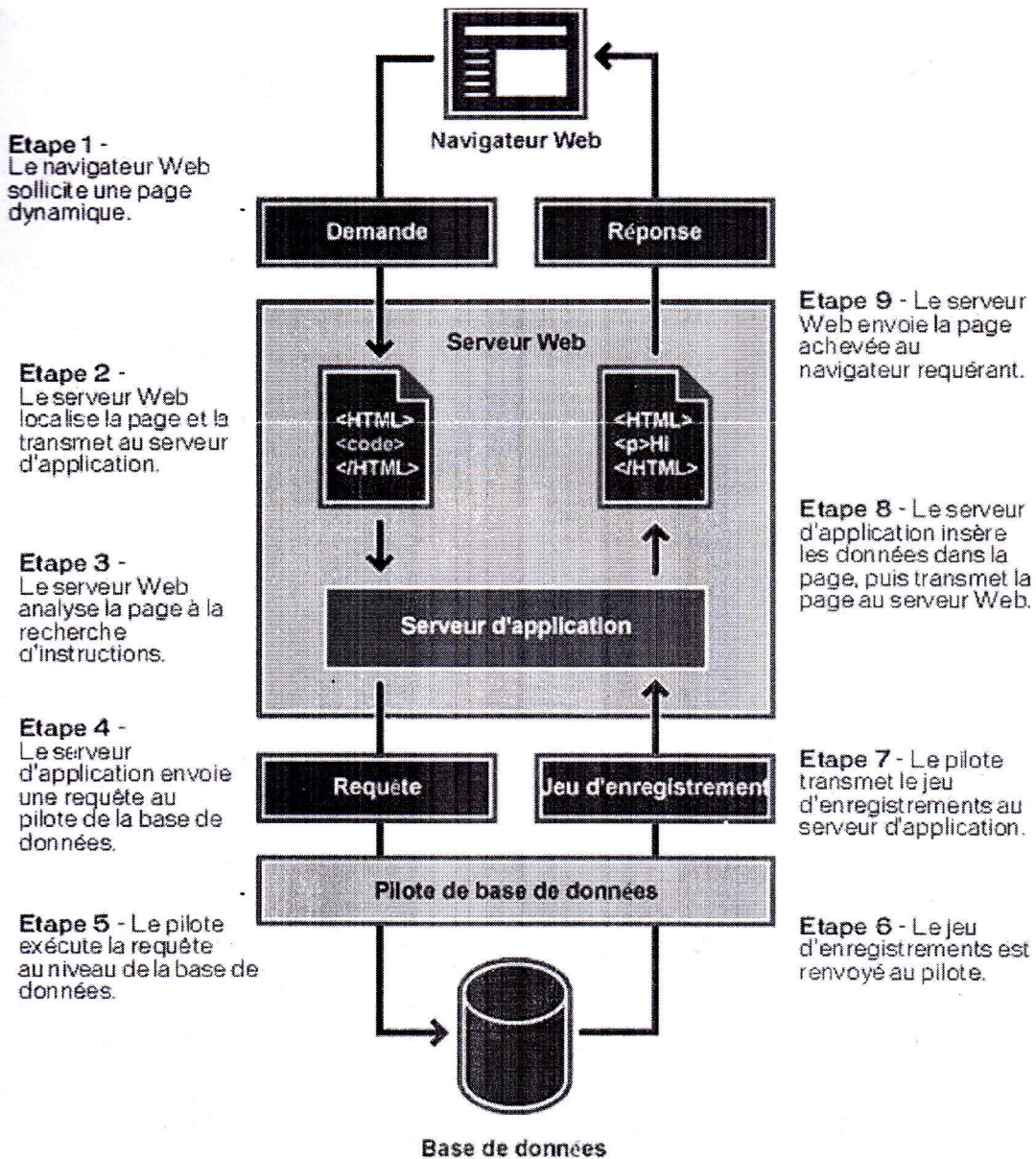
Le schéma suivant présente ce qui se passe lorsque vous consultez une page html dite statique.



Il y a 2 étapes :

1. Le client (c'est vous) demande à voir une page web. Il va donc faire une demande au serveur : "S'il te plaît, envoie-moi la page vacances.html".
2. Le serveur lui répond en lui envoyant la page vacances.html : "Tiens, voici la page que tu m'as demandée".

Maintenant, voyons ce qui se passe lorsque votre page html contient du code Php :



II- Le langage PHP :

1- Présentation :

- PHP est un langage interprété (un langage de script) exécuté du côté serveur et non du côté client (un script écrit en Javascript).

2- Environnement de développement :

Pour le développement d'un site web dynamique, il faut installer en local un serveur Web, qui servira à tester les scripts développés et un SGBD pour tester la connexion à la base de données utilisée et tester les requêtes de manipulation des données de la base. Pour les programmeurs Windows, il existe un utilitaire très pratique (EasyPhp) qui installera

Apache,Php, Mysql et aussi PhpMyAdmin(interface pour gérer des bases de données MySQL).

Activité1 page 121

- L'option configuration : permet de configurer les divers composants d'Easyphp
- L'option Web local permet d'accéder à la racine du serveur Web(contenu de c :\Program Files\EasyPhp\www)
- Les options arrêter et redémarrer permettent respectivement d'arrêter et de relancer les serveurs
- L'option fichiersLog permet d'ouvrir les différents fichiers de configuration de PHP, MySql, Apache et EasyPhp
- L'option quitter permet d'arrêter les différents serveurs puis quittés EasyPhp.

3-Syntaxe de base du langage Php :

Activité 2 Page 122

* Pour que le script soit interpréter par le serveur quatre conditions sont nécessaires :

- Le fichier contenant le code php doit avoir l'extension php
- Le script php doit être contenu dans le code HTML doit être délimité par les balises< ? et ?>
- Le script php doit être enregistré dans le répertoire d'hébergement du serveur apache (c:\Program Files\easyphp\www)
- Le fichier Php ne sera exécuté qu'à travers son adresse Web(http://localhost/... Ou http://nommachine/.... Ou http://127.0.0.1/.....).

* Pour commenter une seule ligne on utilise une double barre oblique. Pour commenter

plusieurs lignes on utilise(/*) au début du bloc du commentaire(/*) à la fin.

4- Les structures des données :

a- Les constantes :

Activité3 page 123

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

b- Les variables

Activité 4 pages 123 et 124

Déclaration de variables :

En Php les variables sont représenter par u signe « \$ » suivi du nom de la variable.
Une variable est déclarée dès que vous lui affectez une valeur

Conversion du type :

1^{ère} méthode :

- Avec la fonction **settype**(string var, string type), renvoie TRUE en cas de succès, FALSE sinon.

- Le type peut être : INTEGER, STRING, ARRAY, DOUBLE ou OBJECT.

2^{ème} méthode :

En précédant les variables à convertir par des clauses type

Exemple :

- \$var=4.8 ;// \$var est un double

- \$var=(int)\$var ;//c'est maintenant un entier(valeur 4)

- \$var=(double)\$var ;//c'est de nouveau un double(valeur 4.0)

-\$var_chaine=(string) \$var_chaine est une chaîne(valeur « 4 »)

Fonction de manipulation de variables :

- string **gettype**(var) :Retourne le type de la variable Php var. Ce type peut être : integer, string, array, object, class, unknown type(type inconnu)
- int **isset**(var) :retourne le résultat TRUE si la variable var possède une valeur, FALSE sinon.

Exemple :

\$prénom = "Ali" ;

echo (isset(\$prénom)) ;//TRUE

c- Les opérateurs :

Voir tableaux pages 125 et 126

5- Les structures de contrôle :

a- Les structures conditionnelles :

L'instruction if	Les structures de branchement
<pre>if(condition) {expression vrai} else {expression faux}</pre>	<pre>Switch(expression) { case resultat1 :traitement1 ;break ; case resultat2: traitement2;break; default: traitement; }</pre>

b- Les structures itératives:

La boucle while	La boucle do...while	La boucle for
<pre>While(condition) { //instructions }</pre>	<pre>do { Action; } while (condition);</pre>	<pre>for(expr1;expr2;expr3) { instructions }</pre>
<p>Exemple</p> <pre>\$i=1; while (\$i <= 10) { echo "- \$i -"; \$i++; }</pre>	<p>Exemple</p> <pre>\$i=1; do { echo " \$i "; \$i++; } while (\$i <= 10)</pre>	<p>Exemple:</p> <pre>for (\$i = 1; \$i <= 10; \$i++) { echo "- \$i -"; }</pre> <p>La première expression (expr1) est évaluée (exécutée), quoi qu'il arrive au début de la boucle.</p> <p>Au début de chaque itération, l'expression expr2 est évaluée. Si l'évaluation vaut TRUE, la boucle</p>

		continue et l'instruction est exécutée. Si l'évaluation vaut FALSE , l'exécution de la boucle s'arrête. A la fin de chaque itération, l'expression <code>expr3</code> est évaluée (exécutée).
--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

6- Les entrées/sortiesActivité :

- Lancer easyphp
- Créer deux fichiers texte avec l'extension php (tpphp5.php et tpphp6.php) sous le répertoire tpphp (G1 ou G2)

Code de tpphp5.php

```
<html>
<head>
</head>
<body>
<form action="tpphp6.php" method="post" name="tester">
<table>
<tr>
<td>NOM:</td> <td><input name="nom" type="text" size="20" /> </td>
</tr>
<tr>
<td>PRENOM:</td> <td><input name="prenom" type="text" size="20" /> </td>
</tr>
</table>
<input name="envoyer" type="submit" value="Envoyer" />
</form>
</body>
</html>
```

Code de tpphp6.php

```
<html>
<body>
</body>
<head>
<?
  if(isset($_POST["nom"]))
  {$nom=$_POST["nom"];}
  else{ $nom="";}
  if(isset($_POST["prenom"]))
  {$prenom=$_POST["prenom"];}
  else
  {$prenom="";}
  echo("$nom<br>$prenom");
?>
</head>
</html>
```

Constatations :

Les entées en php :

Récupérer des informations saisies à travers les objets graphiques des formulaires : on utilise l'une des deux syntaxes suivantes :

\$nom_variable=\$_GET["nom objet"] : si on utilise la propriété method= "get" dans la formulaire

\$nom_variable=\$_POST["nom objet"] : si on utilise la propriété method= "post" dans la formulaire

Les sorties en php

L'affichage se fait grâce à l'instruction **echo** qui permet d'afficher :

- Des chaînes de caractères
Exemple : echo("test1") ;
- Des chaînes de caractères avec le contenu de variable
Exemple : echo("la valeur de x :\$x") ;
- Des chaînes de caractères, des variables et des balises HTML
Exemple : echo("la valeur de x :\$x") ;

7- Les types de données structurées en php :**Activité 7 page 132****a- Les chaînes de caractères**

Une chaînes de caractères est un ensemble de caractères entre guillemets

Exemple : \$variable="PHP " ;

Les fonctions de chaînes de caractères :

Nom	Description	Exemples
string substr (string source, int start , int length)	retourne le segment de string défini par start et length	<?php echo substr ('abcdef', 1); // bcdéf echo substr ('abcdef', 1, 3); // bcd <? <i>Copy</i>
string trim (string str)	retourne la chaîne str, après avoir supprimé les espaces blancs en début et fin de chaîne.	<? echo trim('23 /11 / 2008 .'); // 23 /11 / 2008 . >
int strlen (string string)	Retourne la taille de la chaîne string.	<?php \$str = 'abcdef'; echo strlen (\$str); // 6 \$str = ' ab cd ' ; echo strlen (\$str); // 7 > <i>length</i>
string implode (string glue, array pieces)	retourne une chaîne contenant la représentation en chaîne de caractères de tous les éléments du tableau pieces , dans le même ordre, avec la chaîne glue , placée entre deux éléments.	<?php \$array = array('nom', 'email', 'telephone'); \$comma_separated = implode ("", \$array); echo \$comma_separated; // nom,email,telephone >
array explode (string delimiter, string string)	retourne un tableau de chaînes. Ce sont les sous-chaînes extraites de string, en utilisant le séparateur delimiter .	<?php \$pizza = "pièce1 pièce2 pièce3 pièce4 pièce5 pièce6"; \$pieces = explode (" ", \$pizza); echo \$pieces[0]; // pièce1 echo \$pieces[1]; // pièce2 >
string	retourne une chaîne dont toutes	\$voyelles = array("a", "e", "i", "o", "u", "A", "E", "I",

str_replace (search, replace, subject)	les occurrences de search dans subject ont été remplacées par replace.	"O", "U"); echo \$consonnes = str_replace (\$voyelles, "", "Bonjour le monde"); // Génère : Bnjr l mnd
-----------------------------------------------	------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------

8- les fonctions en php :

Activité 8 page 133

Une fonction peut être définie en utilisant la syntaxe suivante :

```
function Nom_De_La_Fonction(argument1, argument2, ...)
```

```
{
    Corps de ma fonction
[ return $valeur_de_retour ;]
}
```

9-les tableaux

Il existe 2 types de tableaux, les tableaux nominatifs et les tableaux associatifs

1- Les tableaux nominatifs (à indice) de types entier par défaut (L'indice du première élément commence par 0)

2- Les tableaux associatifs, qui utilisent des indices de types chaînes de caractères

Remarque

- Il est possible de stocker des éléments différents dans un même tableau.
- La déclaration d'un tableau se fait de la même manière que la déclaration d'une variable avec un indice se trouvant entre [et].

* Création et initialisation d'un tableau

- `$tableau[0] = "p";` ou `$tab[]="p"` // on crée un tableau, et sa première valeur est "p "

- La déclaration et l'initialisation d'un tableau peuvent également se faire par l'intermédiaire de la fonction 'array()'. Cette fonction permet de préciser les indices ainsi que les valeurs du tableau (à l'aide de l'opérateur =>).

Exemple :

```
$tab=array("p",2);
```

```
$vente_hebdo=array("lundi"=>7,"mardi"=>5,"jeudi"=>9,"vendredi"=>2);
```

*Parcours d'un tableau

Tout tableau possède un pointeur interne qui conserve l'indice et la valeur de l'élément actif . il est possible de déterminer la valeur de l'élément actif à l'aide de la fonction **current()** et l'indice de l'élément actif à l'aide de la fonction **key()**

Exemple :

```
$c=key($vente_hebdo);
```

```
$v=current($vente_hebdo);
```

```
echo("l'élément courant $c est=$v<br>");
```

les deux fonctions each() et list() peuvent être utilisées conjointement afin de parcourir un tableau :

Exemple :

```
while(list($indice,$value)=each($vente_hebdo))
```

```
{
echo"<br> les valeurs du $indice:$value unités<br>";
```

```
}
```

C'est-à-dire j' affecte à \$indice l'indice de l'élément courant et j' affecte à \$value la valeur de l'élément courant.

Voir tableau page 131

Le tri de tableau :

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

Les fonction **ksort**(nom du tableau) et **krsort**(nom du tableau) permettent de trier un tableau par indice.


10- Utiliser MySQL avec Php :

a- Création d'une base de données avec PhpMyAdmin :

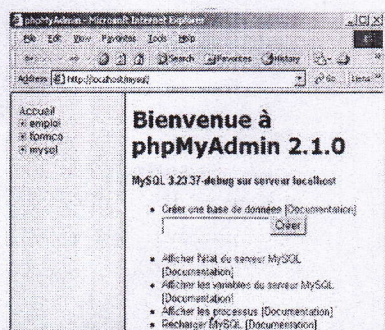
Activité 9 page 134

Création d'une base MySQL

La création de la base ainsi que celles des tables sera faite à l'aide de **PhpmyAdmin**. Pour démarrer ce programme, faire **clic droit sur l'icône de EasyPhp**, puis choisir **"Configuration"** et **"PhpmyAdmin"**.



- Création d'une base de données MySQL



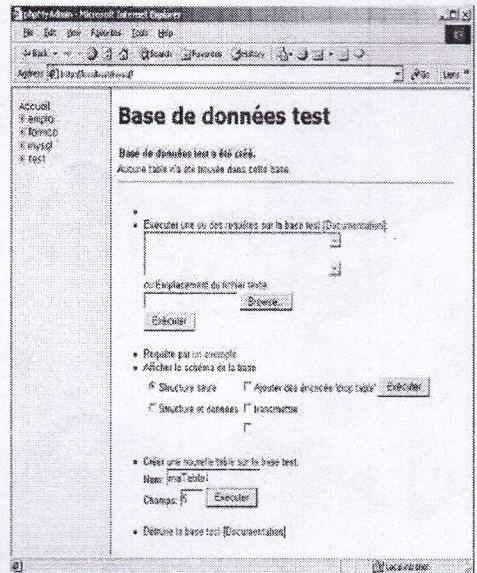
Le navigateur s'ouvre proposant :

- de créer une nouvelle base de données
- ou de choisir parmi les bases de données déjà créées (liste dans le cadre de gauche)

Lorsque la base est créée (ici, elle s'appelle "test"), la création d'une table est possible :

Dans la rubrique, **"créer une nouvelle table sur la base test"**, donner le **"nom"** de la table et le nombre de **"champs"** qu'elle contient. Cette valeur peut être modifiée ultérieurement.

"Exécuter" la commande



Base de données test - table maTable1

Champ	Type	Longueur	Attributs	Null	Défaut	Extra	Primaire	Index	Unique
id	INT			not null		auto_increment	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
nom	CHAR.			not null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
prenom	CHAR			not null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
adr	CHAR	100		null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ville	CHAR			not null	Bordeaux		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Commentaires sur la table:

Table en format:

[Documentation]

On y retrouve les 5 lignes qui vont décrire les champs de la table, elles-mêmes constituées de différentes colonnes.

- **Champ** = nom du champ
- **Type** = type de la donnée (numérique, alphanumérique, date, ...)
- **Longueur** = longueur de la données (adr sur 100 caractères)
- **Attributs** = type de valeurs (binaire, non signé, ...)
- **Null** = la valeur peut être nulle ou pas
- **Défaut** = valeur par défaut
- **Extra** = numérotation automatique (auto_increment)
- **Primaire** = clé de la table (automatiquement indexée et unique)
- **Index** = lors d'une recherche selon ce champ, permet un accès plus rapide aux enregistrements
- **Unique** = les doublons de ce champ sont interdits dans l'ensemble de la table

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

Activité 10 page 135 :

L'utilisation de MySQL avec Php s'effectue en quatre étapes :

1^{ère} étape :

- Connexion au serveur de données : utiliser le syntaxe suivant :

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

Par défaut on utilise les paramètres suivants :

hostname="localhost"

username="root"

password=""

* La fonction mysql_connect() retourne un entier permettant de vérifier l'établissement de la connexion.

2^{ème} étape :

Sélection de la base

```
int mysql_select_db(string database_name,[int link_identifier])
```

Le parameters database_name est obligatoire, le paramètre link_identifier est facultatif. La fonction retourne true ou false selon que l'opération réussit ou non.

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

```
int mysql_query(string query)
```

Envoie au serveur mysql une instruction SQL à executer

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

Requête de sélection

A la suite d'une requête de sélection, les données sont mises en mémoire.

Pour pouvoir les exploiter, Php gère un pointeur de résultat, c'est à dire qu'il repère un enregistrement parmi les autres et lorsqu'on veut le lire un, c'est celui qui pointé qui sera retourné et le pointeur est déplacé vers l'enregistrement suivant.

La fonction de lecture du résultat est :

```
array mysql_fetch_array(int resultat, int resultat_type): extrait la ligne sous forme d'un tableau associatif.
```

Le paramètre result_type et facultatif. Il peut prendre les valeurs suivantes :

- MYSQL_NUM : Le tableau ne contient que des indices numériques
- MYSQL_ASSOC : Le tableau ne contient que des indices associatifs
- MYSQL_BOTH : Le tableau contient à la fois des indices numériques et des indices associatifs.

Remarques :

- Si l'argument result_type n'a pas spécifié, MYSQL_BOTH est considérée comme valeur par défaut de cet argument.

- int mysql_num_rows(int result) :retourne le nombre d'enregistrement qui ont été retournés par la sélection.

A apprendre:

```
mysql_connect("localhost","root","");// Connexion au serveur de données
```

```
mysql_select_db("bd");// Sélection de la base
```

```
$requete="select * from élèves";//Exploitation du requête
```

```
$resultat=mysql_query($requete);//résultat de requête enregistrer dans la variable $resultat
```

- L'insertion, la modification et la suppression des données dans les tables se font à travers des requêtes SQL utilisées en paramètres de la fonction mysql_query, à savoir respectivement des requêtes de types INSERT, UPDATE et DELETE.

Exemple :**Insérer des enregistrements dans la table élèves**

num_ordre:	<input type="text"/>
nom prénom:	<input type="text"/>
classe:	<input type="text"/>
moy_ann:	<input type="text"/>
<input type="button" value="Enregstrer"/> <input type="button" value="Annuler"/>	

Code de la page insertion.html

```

<html>
<head>
<title>Formulaire d'insertion</title>
</head>
<body>
<h1> Enregistrer</h1>
<form name="f1" method="post" action="inserer.php">
<table border=2>
<tr>
<td>num_ordre:</td>
<td>
<input type="text" name="num_ordre" maxlength="4">
</td>
</tr>
<tr>
<td>nom_prénom:</td>
<td>
<input type="text" name="nom_prénom">
</td>
</tr>
<tr>
<td>classe:
</td>
<td><input type="text" name="classe">
</td>
</tr>
<tr>
<td>moy_ann:</td>
<td>
<input type="text" name="moy_ann">
</td>
</tr>
</table>
<input type="submit" value="Enregistrer">
<input type="reset" value="Annuler">
</form>
</body>
</html>

```

Code de la page inserer.php

```

<html>
<head></head>
<body>
<?
$num_ordre=$_POST["num_ordre"];
$nom_prénom=$_POST["nom_prénom"];
$classe=$_POST["classe"];
$moy_ann=$_POST["moy_ann"];
mysql_connect("localhost","root","");
mysql_select_db("bd");
$requete="INSERT INTO élèves VALUES ('$num_ordre','$nom_prénom','$classe','$moy_ann')";
$resultat=mysql_query($requete);
Mysql_close();
?>
</body>
</html>

```

Code du page consultaion.php

```
<html>
<head></head>
<body>
<?
mysql_connect("localhost","root","");
mysql_select_db("bd");
$requete="select * from élève";
$r=mysql_query($requete);

if($r==NULL){
echo"<center><font size=5 color='green'>lexecution de la requete a echoué!!!</font></center>";
}
else
{
echo"<center><font size=5 >voila la liste des eleves</font></center>";
echo"<table border =1 width=80% align='center'>";
echo"<tr><th>CODE</th><th>NON PRENOM</th><th>CLASSE</th><th>MOYENNE
ANNUELLE</th></tr>";

while($t=mysql_fetch_array($r)){
echo"<tr align='center'>";
echo"<td>".$t["num_ordre"]."</td>";
echo"<td>".$t["nom_prénom"]."</td>";
echo"<td>".$t["classe"]."</td>";
echo"<td>".$t["moy_ann"]."</td>";
echo"</tr>";
}
echo"</table>";
}

Mysql_close();
?>
</body>
</html>
```

Code de la page Modification.html

```

<html>
<head>
<title>Formulaire de Modification</title>
</head>
<body>
<h1> Modifier</h1>
<form name="f1" method="post" action="modifier.php">
<table border=2>
<tr>
<td>num_ordre:</td>
<td>
<input type="text" name="num_ordre" maxlength="4">
</td>
</tr>
<tr>
<td>nom_prénom:</td>
<td>
<input type="text" name="nom_prénom">
</td>
</tr>
<tr>
<td>classe:
</td>
<td><input type="text" name="classe">
</td>
</tr>
<tr>
<td>moy_ann:</td>
<td>
<input type="text" name="moy_ann">
</td>
</tr>
</table>
<input type="submit" value="Modifier">
<input type="reset" value="Annuler">
</form>
</body>
</html>

```

Code de la page modifier.php

```

<html>
<head></head>
<body>
<?
$num_ordre = $_POST["num_ordre"];
mysql_connect("localhost","root","");
mysql_select_db("bd");
$requete="select * from élève where num_ordre=$num_ordre";
$r=mysql_query($requete);
$n=mysql_num_rows($r);

if($n==0){
echo"<center><font size=5 color='red'>le code de l'élève n' existe pas dans le base</font></center>";
}
else{
$requete="update élève set
nom_prénom='".$_POST["nom_prénom"]."',classe='".$_POST["classe"]."',moy_ann='".$_POST["moy_ann"]."'
where num_ordre =$num_ordre";
$r=mysql_query($requete);
echo"<center><font size=5 color='green'>la mise à jour a etais effecté avec succès</font></center>";
}
Mysql_close();

```

```
?>
</body>
</html>>
```

Code de la page recherche.html

```
<html>
<head>
<title>Formulaire de recherche</title>
</head>
<body>
<h1> choisir le numero d'ordre de l'élève objet de la recherche</h1>
<form name="f1" method="post" action="chercher.php">
<table border=2>
<tr>
<td>num_ordre:</td>
<td>
<input type="text" name="num_ordre" maxlength="4">
</td>
</tr>

</table>
<input type="submit" value="chercher">
<input type="reset" value="Annuler">
</form>
</body>
</html>
```

Code de la page chercher.php

```
<html>
<head></head>
<body>
<?
$num_ordre=$_POST["num_ordre"];
mysql_connect("localhost","root","");
mysql_select_db("bd");
$requete="select * from élève where num_ordre=$num_ordre";
$r=mysql_query($requete);
$n=mysql_num_rows($r);
if($n==0){
echo"<center><font size=5 color='red'>le code de l'élève n'existe pas dans le base</font></center>";
}
else{
echo"<center><font size=5 color='green'>la recherche a etais effectué avec succès</font></center>";
echo"<center><font size=5 >voila la liste des élèves</font></center>";
echo"<table border =1 width=80% align='center'>";
echo"<tr><th>CODE</th><th>NON PRENOM</th><th>CLASSE</th><th>MOYENNE
ANUELLE</th></tr>";
while($t=mysql_fetch_array($r)){
echo"<tr align='center'>";
echo"<td>".${t["num_ordre"]}."</td>";
echo"<td>".${t["nom_prénom"]}."</td>";
echo"<td>".${t["classe"]}."</td>";
echo"<td>".${t["moy_ann"]}."</td>";
echo"</tr>";
}
echo"</table>";
}
}
```

```
Mysql_close();
?>
</body>
</html>
```

Code de la page suppression.html

```
<html>
<head>
<title>Formulaire de suppression</title>
</head>
<body>
<h1> choisir le numero d'ordre de l'eleve à supprimer</h1>
<form name="f1" method="post" action="supprimer.php">
<table border=2>
<tr>
<td>num_ordre:</td>
<td>
<input type="text" name="num_ordre" maxlength="4">
</td>
</tr>

</table>
<input type="submit" value="supprimer">
<input type="reset" value="Annuler">
</form>
</body>
</html>
```

Code de la page supprimer.php

```
<html>
<head></head>
<body>
<?
mysql_connect("localhost","root","");
mysql_select_db("bd");
$requete="select * from élève where num_ordre=".$_POST["num_ordre"].""";
$r=mysql_query($requete);
$n=mysql_num_rows($r);
if($n==0){
echo"<center><font size=5 color='red'>le code de l' élève n' existe pas dans la base</font></center>";
}
else{
$requete="delete from élève where num_ordre=".$_POST["num_ordre"].""";
$r=mysql_query($requete);

echo"<center><font size=5 color='green'>la suppression a etais effecté avec succès</font></center>";

}

Mysql_close();
?>
</body>
</html>
```