

## I) Introduction :

### 1) Définition d'un site web dynamique :

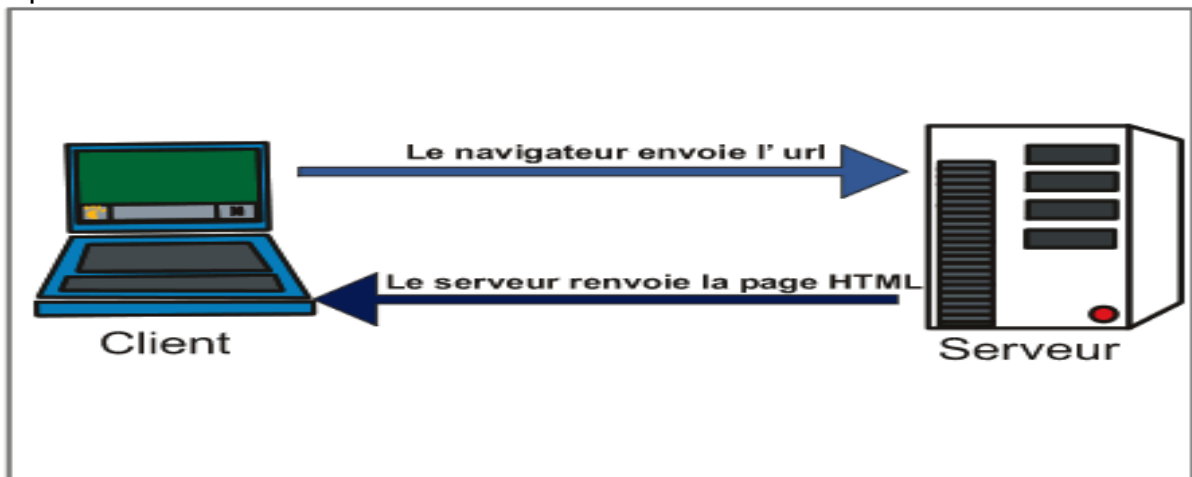
Le principe d'une page dynamique est d'être construite en fonction de critères spécifiques. La présentation et le contenu affichés peuvent ainsi être personnalisés de manière interactive, en fonction du contenu de la .....des internautes, des langues, etc...

Alors que les pages statiques font appel au ....., langage de description de données, les pages dynamiques sont mises en oeuvre grâce à un langage de programmation.

Un site dynamique est plus exigeant pour un ..... qu'un site ..... En effet, alors que pour un site statique le serveur se contente d'afficher les pages, pour un site dynamique, il doit effectuer différents traitements et accéder à ..... Pour chaque demande de l'utilisateur, il y a une interrogation au ..... qui construit la page

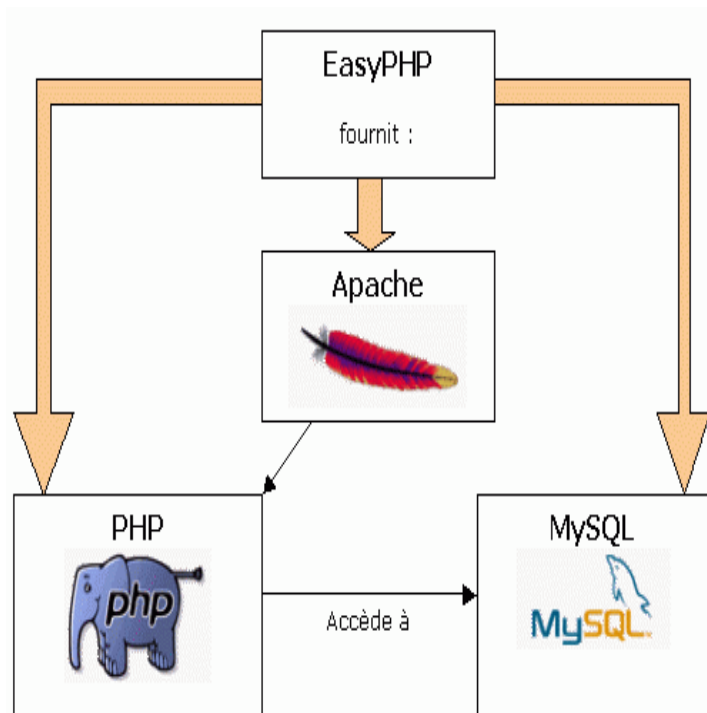
### 2) Principe de fonctionnement :

Voici, en simplifiant, ce qu'il se passe lorsque vous consultez une page..... Dite statique :



- Votre navigateur envoie l'adresse ..... que vous avez tapée
  - Le serveur web est un "ordinateur" présent sur l'Internet et qui héberge la page que vous demandez. Sur ce serveur on trouve ....., logiciel apte à traiter les requêtes ..... que vous envoyez lorsque vous demandez une page web. Apache va donc chercher le fichier demandé dans son arborescence et renvoie à votre navigateur la page .....
  - Votre navigateur interprète les différents langages se trouvant dans ce fichier (HTML, JavaScript, CSS, etc ...) et affiche la page.
- Maintenant, voyons ce qui se passe lorsque votre page HTML contient du code ..... :

- Votre navigateur envoie l'adresse que vous avez tapée Le serveur web cherche dans son arborescence si le fichier existe, et si celui-ci porte une extension reconnue comme une application PHP (.PHP, .PHP3, .PHP4 par exemple). Si c'est le cas, le serveur web transmet ce fichier à PHP.
- PHP pars le fichier, c'est-à-dire qu'il va analyser et exécuter le code PHP qui se trouve entre les balises <? et ?>. Si ce code contient des requêtes vers ....., PHP envoie la requête SQL. La base de données renvoie les informations voulues au script qui peut les exploiter (pour les afficher par exemple).



- PHP continue de parser la page, puis retourne le fichier dépourvu du code PHP au serveur web.

- Le serveur web renvoie donc un fichier ne contenant plus de PHP, donc seulement du HTML au navigateur qui l'interprète et l'affiche.

## II) Le langage PHP:

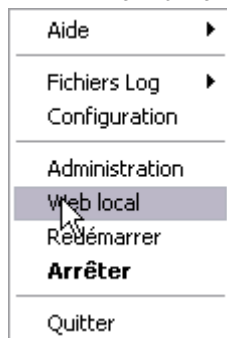
**1) Présentation &: Environnement de développement:**

**2) Syntaxe de base de PHP:**

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

- Le fichier contenant le code doit .....  
.....
- Le code PHP contenu dans le code HTML doit être délimité par des **balises** du type .....  
.....

- Le fichier php doit être .....  
.....
- Le fichier php ne sera exécuté qu'à travers son adresse web (utiliser l'option web local)



### 3) Structures de données: activité cercle/ âge


#### a) Les constantes:


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


#### b) Les variables:

- Php n'impose pas de déclaration explicite des variables avant de les utiliser.
- Pour déclarer une variable, il suffit de la précéder par le signe **.....** et lui affecter .....  
.....

- Pour manipuler les variables, on distingue les fonctions suivantes :

 **settype**: permet de forcer la conversion du type de la variable  
settype(\$dc,"integer") ou \$classe=(integer)\$classe

 **gettype**: permet de connaître le type de la variable

 **isset**: permet de tester si une variable est définie

**c ) Les opérateurs : permettent de modifier mathématiquement la valeur d'une variable**

Opérateur	Dénomination	Effet	Exemple	Résultat (pour x=7)
+	opérateur d'addition	Ajoute deux valeurs	$x+3$	...
-	opérateur de soustraction	Soustrait deux valeurs	$x-3$	....
*	opérateur de multiplication	Multiplie deux valeurs	$x*3$	...
/	plus: opérateur de division	Divise deux valeurs	$x/3$	....
=	opérateur d'affectation	Affecte une valeur à une variable	$x=3$	
%	opérateur modulo	Donne le reste de la division entière entre 2 nombres	$x\%3$	...

• **Les opérateurs de comparaison**

Opérateur	Dénomination	Exemple	Résultat
=	opérateur d'égalité	$x==3$	Retourne 1 si \$x est égal à 3, sinon 0
<	opérateur d'infériorité stricte	$x<3$	Retourne 1 si \$x est inférieur à 3, sinon 0
<=	opérateur d'infériorité	$x<=3$	Retourne 1 si \$x est inférieur ou égale à 3, sinon 0
>	opérateur de supériorité stricte	$x>3$	Retourne 1 si \$x est supérieur à 3, sinon 0
>=	opérateur de supériorité	$x>=3$	Retourne 1 si \$x est supérieur ou égal à 3, sinon 0
!=	opérateur de différence	$x!=3$	Retourne 1 si \$x est différent de 3, sinon 0

• **Les opérateurs logiques (booléens)**

Opérateur	Dénomination	Effet	Syntaxe
ou OR	OU logique	Vérifie qu'une des conditions est réalisée	((condition1)  condition2))
&& ou AND	ET logique	Vérifie que toutes les conditions sont réalisées	((condition1)&&condition2))
XOR	OU exclusif	Vérifie qu'une et une seule des conditions est réalisée	((condition1)XOR(condition2))
!	NON logique	Inverse l'état d'une variable booléenne (retourne la valeur 1 si la variable vaut 0, 0 si elle vaut 1)	(!condition)



**Les entrées/sorties: Pour entrer des données à une page php vous pouvez:**

- Récupérer les documents depuis un formulaire en utilisant les tableaux prédéfinis ..... ou ..... Suivant la méthode utilisée lors de la déclaration du formulaire
- Avec la méthode GET, les données à envoyer au serveur sont écrites directement dans l'URL. Dans la fenêtre de votre navigateur, cela ressemble à ceci :  
[www.example.com/register.php?firstname=peter&name=miller&age=55&gender=male](http://www.example.com/register.php?firstname=peter&name=miller&age=55&gender=male)  
Toutes les informations saisies par l'utilisateur (les paramètres dits URL) sont transmises aussi librement que l'URL elle-même.
- La méthode POST écrit les paramètres URL dans la requête HTTP pour le serveur. Les paramètres ne sont donc pas visibles pour les utilisateurs et la portée des requêtes POST est illimitée.

	GET	POST
Visibilité	Visible pour l'utilisateur dans le champ d'adresse	Invisible pour l'utilisateur
Marque-page et historique de navigation	Les paramètres de l'URL sont stockés en même temps que l'URL.	L'URL est enregistrée sans paramètres URL.
Cache et fichier log du serveur	Les paramètres de l'URL sont stockés sans chiffrement	Les paramètres de l'URL ne sont pas enregistrés automatiquement.
Comportement lors de l'actualisation du navigateur / Bouton « précédent »	Les paramètres de l'URL ne sont pas envoyés à nouveau.	Le navigateur avertit que les données du formulaire doivent être renvoyées.
Type de données	Caractères ASCII uniquement.	Caractères ASCII mais également données binaires.
Longueur des données	Limitée - longueur maximale de l'URL à 2 048 caractères.	Illimitée.

✚ **Pour afficher le contenu** d'une variable ou un message, on peut utiliser la méthode ".....", ou utiliser un objet d'un formulaire

```
<input type="text" name="np" value=<? ..... (....); ?>>
```

## d ) Les structures de

contrôle:

### i. Les structures

conditionnelles:

#### ✚ L'instruction if ... else

```
if (condition réalisée) {  
  Liste d'instructions  
}else {  
  Autre série d'instructions  
}
```

#### ✚ L'instruction switch

```
switch (Variable) {  
  
case Valeur1:  
  Liste d'instructions  
break;  
case Valeur2:  
  Liste d'instructions  
break;  
case Valeurs...:
```

```
Liste d'instructions  
Break;  
Default:  
Liste d'instructions  
Break;  
}
```

### ii. Les structures itératives:

#### ✚ La boucle for:

```
For (compteur; condition; modification du compteur)  
{  
  Liste d'instructions}
```

#### ✚ L'instruction while:

```
While (condition réalisée)  
{  
  Liste d'instructions  
}
```

#### ✚ L'instruction do..While:

```
Do {  
  Liste d'instructions  
} While (condition réalisée)
```

## e ) Les types structurés : chaînes de caractères

1. **La fonction strlen** Elle retourne la longueur d'une chaîne.

**Syntaxe : int [strlen](#) (string str)**

2. **La fonction strpos**

Elle recherche la première occurrence d'un caractère dans une chaîne et retourne la position numérique de la première occurrence de ch1 dans la chaîne ch à partir d'une position donnée.

**Syntaxe : int [strpos](#) (string ch1, string ch, int position )**

3. **La fonction trim**

Elle efface les espaces de début et de fin de chaîne

**Syntaxe : string trim (string str, " ")**

4. **La fonction strcmp**

Elle compare en binaire des chaînes. Elle retourne < 0 si str1 est plus petit que str2 ; > 0 si str1 est plus grand que str2, et 0 s'ils sont égaux.

**Syntaxe : int [strcmp](#) (string str1, string str2)**

5. **La fonction strrev**

Elle inverse l'ordre des caractères d'une chaîne.

**Syntaxe : string strrev (string ch)**

6. **La fonction strtolower**

Elle convertit en minuscule tous les caractères d'une chaîne

**Syntaxe : string strtolower (string ch)**

7. **La fonction strtoupper**

Elle convertit en majuscule tous les caractères d'une chaîne

**Syntaxe : string strtoupper (string ch)**

8. **La fonction str\_replace**

Remplace toutes les occurrences d'une chaîne par une autre.

**Syntaxe string str\_replace (string search, string replace, string subject)**

[str\\_replace\(\)](#) remplace toutes les occurrences de search dans subject par la chaîne replace.

9. **La fonction substr** Elle extrait une partie d'une chaîne donnée

**Syntaxe : string [substr](#) (string ch, int pos, int nbre )**

10. **La concaténation** La concaténation des chaînes de caractères se fait avec l'opérateur • (Point)

### i. Les tableaux :

❖ Les tableaux indicés sont composés de plusieurs éléments dont chacun est repéré par une valeur numérique unique. La première cellule d'un tableau commence toujours par l'indice zéro (0).

❖ Les tableaux associatifs permettent de ranger des données par rapport à une valeur quelconque, telle qu'une chaîne de caractères ou un nombre.

- Il est également possible d'utiliser la fonction **"array"** afin de créer et de remplir un tableau

#### Exemple :

```
$jour = array("Dimanche" => 1, "Lundi" => 2, "Mardi" => 3, "Mercredi" => 4, "Jeudi" => 5, "Vendredi" => 6, "Samedi" => 7);
```

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

```
While (..... (.....,.....)= each  
(.....))  
{      Echo "$value";  
}
```

Ou bien

```
foreach (..... as .....)  
{      Echo ("$value.<br>");};
```

O bien

```
foreach ($ventes as $key=>$value)  
{      Echo ("$value.<br>");};
```

## MySQL

### 1-\ Création d'une base de données avec PHP Myadmin

1 – lancer EasyPHP		
2 – lancer PHPmyadmin	<p>Cliquer avec le bouton droit de la souris sur l'icône  et choisir administration</p> <p>Choisir Gestion BDD</p> 	
Créer une base de données	 <p>Ecrire le nom de la BDD Cliquer sur "Créer"</p>	
Créer une table	 <p>Nom de la table</p> <p>Nombre de champs dans la table</p> <p>Cliquer sur "Exécuter"</p>	

**Créer les champs**

Serveur: localhost Base de données: Lycée Table: eleve

Champ	Type	Taille/Valeurs	Interclassement	Attributs	Null	Défaut**	Extra	
	VARCHAR				not null			
	VARCHAR				not null		auto_increment	
	TINYINT				not null			
	TEXT				not null			
	DATE				not null			
	SMALLINT				not null			
	MEDIUMINT				not null			
	INT				not null			
	BIGINT				not null			
	FLOAT							
	DOUBLE							
	DECIMAL							
	DATETIME							
	TIMESTAMP							

Commentaires sur le champ

Saisir les noms des champs

Sélectionner le type de données

Type de la table: défaut

Interclassement: défaut

ajouter un entier si le type est "varchar"

Ajouter la valeur par défaut si elle existe

choisir "auto\_increment" si le champ avance automatiquement

Clé primaire

Sauvegarder

En fin cliquer sur "Sauvegarder"

## Etapes

- 1-Saisir le nom pour Créer une BD (interclassement)
  - 2-Créer des tables de type .....et insérer les champs leur types tailles et spécificité :.....
  - 3-Remplissez les tables par des enregistrements à travers la rubrique .....
  - 4-visualiser les enregistrements à travers rubrique .....
  - 5-modifier la table à travers la rubrique .....
- \*on peut accéder encore une fois pour modifier les données
- \*\*la base sera enregistré dans le dossier easy php /mysql /data

## 2-\ Se connecter à une base MySQL avec PHP

- `int mysql_connect (string hostname, string username , string password)` Avec :  
hostname : nom de l'hôte ou du site (chaîne de caractères)  
username : nom de connexion à la base (=login) (chaîne de caractères)  
password : mot de passe associé au login (chaîne de caractères)
- Par défaut avec easyphp, on écrit:  
**`mysql_connect ("localhost", "root", "") or die("erreur de connexion au serveur");;`**  
Pour fermer à la fin `mysql_close ();`

## 3-\ Sélection d'une base

`int mysql_select_db (string database, resource link_identifieur )` Avec:  
database : nom de la base de données MySQL qui contient les tables  
link\_identifieur : identifiant de la connexion ouverte par `mysql_connect()`. C'est un paramètre facultatif  
**Exemple `mysql_select_db("lycee") or die("erreur de connexion au serveur");;`**

## 4/ exécution d'un requête

`mysql_query (string query)` Avec:  
query : nom de la requête ou chaîne de caractères contenant la requête  
Exemple `:$requete="select*from eleve";`  
`$resultat=mysql_query($requete);`

## 5/exploitation des résultats d'une requête

**Une requête de sélection, il faut afficher les résultats en utilisant:**

`array mysql_fetch_row (int result)/array mysql_fetch_array (int result, int result_type)`

<b>une ligne ou un enregistrement</b>	<pre>&lt;? for (\$i=0;\$i&lt;mysql_num_rows(\$resultat);\$i++) { \$enreg=mysql_fetch_row(\$resultat); ?&gt; &lt;td&gt;&lt;? echo \$enreg[0];?&gt;&lt;/td&gt; &lt;td&gt;&lt;? echo \$enreg[1];?&gt;&lt;/td&gt;</pre>
<b>un tableau</b>	<pre>while(\$enreg=mysql_fetch_array(\$resultat,MYSQL_ASSOC)) { ?&gt;&lt;td&gt;&lt;? echo \$enreg["num_ordre"];?&gt;&lt;/td&gt; &lt;td&gt;&lt;? echo \$enreg["nom_prenom"];?&gt;&lt;/td&gt;&lt;?&gt;&lt;/table&gt;</pre>

## **Autres Fonctions utilisées**

### **1- `int mysql_num_rows (int result)`**

Retourne le nombre d'enregistrements qui ont été retournés par la sélection.

### **2- `int mysql_insert_id (int link_identifiant )`**

Retourne le dernier identifiant généré par un champ de type AUTO\_INCREMENTED.

Exemple : Suite à une requête d'insertion, on veut afficher le numéro auto incrémenté :

```
<?php echo "Votre numéro d'identifiant est : " .mysql_insert_id() ;?>
```

### **3- `int mysql_affected_rows (int link_identifiant )`**

Cette fonction permet de fixer le nombre de lignes insérées, mises à jour ou supprimées par la précédente requête SQL (INSERT, DELETE, REPLACE ou UPDATE) envoyée au serveur.Exemple :

Suite à une commande "UPDATE", on voudrait savoir combien de lignes ont été modifiées :

```
<?php  
echo mysql_affected_rows() ." enregistrement(s) modifiés";  
?>
```