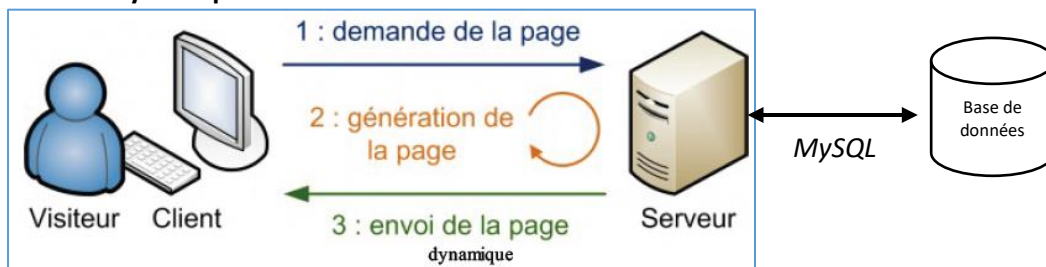


RAPPEL - MEMENTO

Les sites dynamiques :



Le **PHP** est un langage exécuté par le serveur.

Il permet de personnaliser la page en fonction du visiteur et/ou d'un contenu dans une base de données, d'effectuer des calculs, etc. Il génère une page HTML.

Wamp

- Apache (=serveur)
- Php (page dynamique)
- Mysql (base de données)

PHP

```
<?php
```

```
?>
```

```
// commentaires
```

```
/* longs commentaires */
```

```
echo "bonjour ";
```

```
include ("menu.php");
```

```
variable
```

```
echo $nomdelavariab
```

```
opérateur de base
```

```
+ - * / ( ) %
```

En-tête (header)

Menu

Corps
(partie centrale)

Pied de page (footer)

Conditionnelles

```
if ( )  
{  
  
} else {  
  
}
```

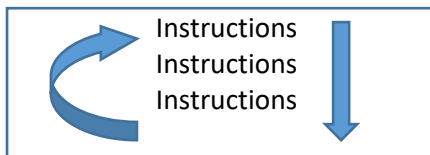
```
if ( )  
{  
  
} elseif ( )  
{  
  
} else {  
  
}
```

```
switch ( )  
{  
case 1 :  
case 2 :  
case 3 : ...  
break;  
default : ...  
}
```

```
<?
$moyenne=14;
if($moyenne<12){
    echo "refusé";
}elseif($moyenne<14){
    echo "satisfaction";
}elseif($moyenne<16){
    echo "distinction";
}elseif ($moyenne<18){
    echo "grande distinction";
}else{
    echo "la plus grande distinction";
}??>
```

```
<?
Switch($moyenne){
    case 1 :case 2 :case 3 :case 4 :case 5 :
    case 6 :case 7 :case 8 :case 9 :case 10 :
    case 11 : echo("refusé");
        break;
    case 12 :
    case 13 : echo("satisfaction");
        break;
    case 14 :
    case 15 : echo("distinction");
        break;
    case 16 :
    case 17 : echo("grande distinction");
        break;
    case 18 :
    case 19 :
    case 20 : echo("la plus grande distinction");
        break;
    default : echo("chiffre incorrect");
}
??>
```

Les répétitives



WHILE

```
while ($continuer_boucle == true)
{
// instructions à exécuter dans la boucle
}
```

FOR

```
for (compteur; condition; modification du compteur)
{
// instructions à exécuter dans la boucle
}
```

```
<?php
$i= 1;
while ($i <= 10)
{
echo $i ;
$i ++;      // $i=$i +1
}
?>
```

```
<?php
for ($i=1;$i<=10;$i++)
{
echo "$i <BR>";
}
?>
```

Les fonctions

→Éviter de répéter du code PHP

Les formulaires

Méthode POST

```
<form action="resultat.php" method="post">
Entrez votre prénom : <input type="text" name="prenom" />
<input type="submit" value="valider" />
</form>
```

Récupération de la valeur d'un champ de formulaire :

```
$_POST["prenom"]
```

Base de données

BD = Ensemble d'information organisées

SQL = Structured Query Language = Langage de requêtes dans une BD

Attribut, Enregistrements, relation (1 à N) (N à N)

Domaine = ensemble de valeur

Clé primaire / Clé étrangère

Présentation de PHPMysqlAdmin – MySQL

✕ Les instructions SQL ne sont pas sensibles à la casse

| | |
|--|--|
| Création d'une base de données | CREATE DATABASE |
| Création d'une table | CREATE TABLE (...) Champs et type de données Entiers INT / Flottants FLOAT / Chaîne VARCHAR TEXT /Date et Heure DATE DATETIME / Ensemble ENUM |
| Création des relations entre les tables <i>Clé primaire (exemple ISBN, Reg Nat, ADN ...)</i> | PRIMARY KEY |
| Attribut NOT NULL / UNIQUE Valeur par défaut DEFAULT INDEX → <u>recherches et les tris</u> | |
| Clé étrangère | CONSTRAINT fk FOREIGN KEY (fkey) REFERENCES TABLE (id) |
| Mise à jour d'une base de données | ALTER TABLE ... (modification) DROP TABLE ... (suppression) |

| | |
|-------------------------|---|
| Ajout de données | INSERT INTO TABLE (...) VALUES (...) <i>(attention à l'ordre des champs)</i> |
| Modification de données | UPDATE TABLE SET champ= "..." WHERE ID =... |
| Suppression de données | DELETE FROM TABLE WHERE ID =... |

Interrogation de base de données

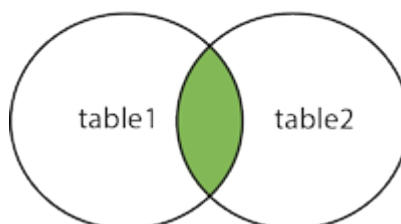
Requêtes de sélection

| | |
|-------------------------------------|--|
| Simple | SELECT attribut FROM table |
| Avec filtre | SELECT attribut FROM table WHERE attribut = " ... " |
| Avec tri | SELECT attribut FROM table ORDER BY attribut ASC |
| Suppression des doublons | SELECT DISTINCT attribut FROM table |
| Avec jointure | SELECT attribut FROM table1, table2 WHERE table1.attribut = table2.attribut |
| <hr/> | |
| Multiple | |
| Requêtes imbriquées | SELECT * FROM table1 WHERE ID=(SELECT ID FROM table2 WHERE attribut = " ... ") |
| Requêtes sélection multi-tables | SELECT * FROM table1,table2 WHERE table1.ID = table2.IDREF AND attribut = " ... " |
| Avec regroupement | SELECT ... FROM table GROUP BY attribut HAVING... |
| Champs calculés/ Fonctions de MySQL | MIN(x), MAX(x), AVG(x), SUM(x), COUNT(x)... |

Les jointures internes

INNER JOIN

INNER JOIN

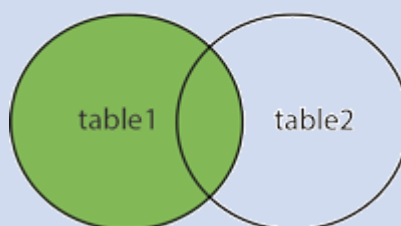


```
SELECT ...  
FROM table1  
INNER JOIN table2 ON table1.ID = table2.IDREF
```

Les jointures externes

LEFT OUTER JOIN

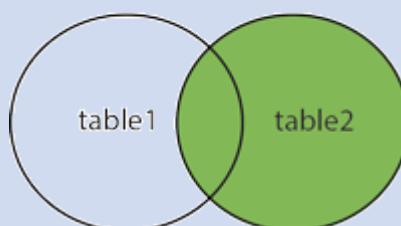
LEFT JOIN



```
SELECT ...  
FROM table1  
LEFT OUTER JOIN table2 ON table1.ID = table2.IDREF
```

RIGHT OUTER JOIN

RIGHT JOIN



```
SELECT ...  
FROM table1  
RIGHT OUTER JOIN table2 ON table1.ID = table2.IDREF
```

Requêtes ensemblistes

Union

```
SELECT attribut FROM table1  
UNION
```

Intersection

```
SELECT attribut FROM table2  
SELECT attribut FROM table1  
WHERE attribut IN (SELECT attribut FROM table2)
```

Différence

```
SELECT attribut FROM table1  
WHERE attribut NOT IN (SELECT attribut FROM table2)
```

Produit cartésien

```
SELECT * FROM table1,table2
```

PDO est une **classe PHP** destinée à permettre à PHP de communiquer avec un serveur de données. (PDO : Php Data Object)

PDO fait la distinction entre deux familles de requêtes :

⊕ **QUERY** - Pour récupérer une information (SELECT)

⊕ **EXEC** - Pour apporter des changements (INSERT, UPDATE, DELETE)

```
<?php
// Connexion au serveur
try {
    $dns = 'mysql:host=localhost;dbname=toto';
    $utilisateur = 'root';
    $motDePasse = '';
    $connection = new PDO( $dns, $utilisateur, $motDePasse );
} catch ( Exception $e ) {
    echo "Connexion à MySQL impossible : ", $e->getMessage();
    die();
}
?>
```

Affichage (SELECT)

```
// On envoie la requête
$select = $connection->query("SELECT * FROM t_contact");
/*Nous traitons les résultats en boucle et c'est lors de l'utilisation de fetch() que nous spécifions le format de
récupération pour le traitement.*/
while( $enregistrement = $select->fetch(PDO::FETCH_OBJ) )
{
    // Affichage des champs
    echo $enregistrement->Nom, ' ', $enregistrement->Prenom;
}
```

INSERTION(INSERT)

```
// Ajout de données
try {
    $connection->exec("INSERT INTO t_contact VALUES(, 'Térier', 'Alex')");
} catch ( Exception $e ) {
    echo "Une erreur est survenue lors de l'insertion ";
}
}
```