**PHP**

// Lire les donnees de la base de donnee

/\* $requete = $bdd->query('SELECT \* FROM users WHERE prenom = "Happy" ORDER BY id DESC');

\*/

//Ajouter un User avec : query() ou exec()

/\* $requete = $bdd->query('INSERT INTO users(prenom, nom, serie\_preferee)

                        VALUES ("Mark", "Zukecberg", "The FaceBook")');

 \*/

//Modification d un User

/\* $requete = $bdd->exec('UPDATE users

                       SET serie\_preferee ="Mission Impossible"

                       WHERE prenom ="Mark"'); \*/

/\* //Supprimer un user

$requete = $bdd->exec('DELETE

                      FROM users

                      WHERE prenom="Mark"'); \*/

//Afficher une erreur

// $requete = $bdd->exec('INSERT INTO users(prenom, nom, serie\_preferee)

//                        VALUE("Jean", "Polnaref", "JBA")')

//                        or die(print\_r($bdd->errorInfo())) ;

// \*\*\*\*\*\*\* Joindre jobs table et formation\_users \*\*\*\*\*\*\*\*\*

//1. Enregistrement

/\* $requete = $bdd->exec('INSERT INTO jobs(id\_user, metier)

                       VALUES(4, "Personnage")'); \*/

/\* //2. Les jointure interne avec WHERE:

$requete = $bdd->query('SELECT prenom, nom, serie\_preferee, metier

                        FROM users, jobs

                        WHERE users.id = jobs.id\_user'); \*/

//2. Les jointure interne avec JOIN:

/\* $requete = $bdd->query('SELECT prenom, nom, serie\_preferee, metier

                        FROM users

                        INNER JOIN jobs

                        ON users.id = jobs.id\_user'); \*/

//\*\* Les Allias se font comme suite:

//\*\*     - prenom.users : signifie prenom de la table users

//\*\*     - users AS u (ou tout simplement) users u : cree un allias

//\*\*     - u.prenom : allias de users de la colonne prenom.

//Le probleme des injections:

//$prenom = '" OR 1=1#';

//$prenom ="leonel";

//\*\*\*\*\* 1. Solution : Pour proteger il faut utiliser htmlspecialchars(....)

/\* $prenom = htmlspecialchars($prenom);

$requete = $bdd->query('SELECT prenom, nom, serie\_preferee, metier

                        FROM users

                        INNER JOIN jobs

                        ON users.id = jobs.id\_user

                        WHERE prenom ="'.$prenom.'"'); \*/

//\*\*\*\*\* 2. Solution: il faut maintenant utiliser prepare() et ensuite execute(array(...))

//\* On fait egalement une jointure externe grace a LEFT ou RIGHT \*\*\*/

//  $requete = $bdd->prepare('SELECT prenom, nom, u.serie\_preferee, j.metier

//                           FROM users u

//                           LEFT JOIN jobs j /\*jointure externe a gauche\*/

//                           ON u.id=j.id\_user

//                           WHERE prenom = ?/\*prepare la donnee\*/

//                           ');

// $requete->execute(array($prenom));//\*\*\*On execute la requete preparee

//\*\*Encryptage des donnees:

//\*\* - sh1()

//\*\* - md5() + grain de sel(chaine aléatoire)

* **Récapitulation sur les Variables et tableau**
* Variable: **$variable;**
* Tableau:
* $tab = **[1, 2, 3, .....];**
* $liste = **array(val1, val2,...);**
* $dict = **[cle1=>val1, cle2=>val2];**
* foreach(**$tab as** **$elemt**){

echo **$elemt[0];**

}

* foreach(**$dict as $cle=>$valeur**){....}
* **print\_r($tab)** : affiche rapidement un tableau
* **array\_key\_exists('cle', $tab)** -> bool

Indique si le **tab** contient la **cle**

* **in\_array('valeur', $tab)** ->bool

Indique si le **tab** contient la **valeur**

* **array\_search('valeur',$tab)**->$cle

Renvoi la cle correspondant

* **Mode d’écriture et de lecture dans un fichier :**
* **Lire dans un fichier:**
* **Ecriture dans un fichier :**
* **Connection a la base de données :**
* **Création de l’objet PDO :**

* **Faire une requête SQL :**
* **Récupérer le résultat de la requête :**

**$donnees** est un array qui contient champ par champ(**colonne**) les valeurs de la première entrée. **Exple :** $donnees[‘colonne1’] pour accéder a tous les entrées de la colonne 1.

* **Directive : LIMIT début, fin**

**Exple : LIMIT 0, 20**

* **Ordre de Methode SQL : "SFWOL"**
* SELECT
* FROM
* WHERE
* ORDER BY
* LIMITE
* **Requêtes préparées :**

Le but de ce type de requête a été introduite pour pallier aux problèmes de  **d injection SQL** lie a la fonction  **query(…)**  lorsqu’ on utilise des variables dans des requêtes.

* Pour se faire on utilise la fonction  **prepare(…)** et a la place des variables on met des marqueurs pour valeurs et ensuite on utilise la fonction **execute(…)** qui prend la valeur des variables.
* On peut aussi se servir des **marqueurs nominatifs** **:variable** au lieu des simples **?**

**Fig. Utilisation des marqueurs nominatifs 🡪 :possesseur et :prixmax**

* **Traquer les Erreurs :**

On peut déterminer exactement l’erreur dans une requête avec cette ligne de code ajouter lors de la création de l’objet **PDO**

* **Ajouter des donnees dans une table : INSERT INTO table**
* On se sert de la fonction **$pdo->exec(…)** chaque fois qu’ on fait des modifications dans une base de données.
* On peut également se servir des requêtes **prepare(…)**
* **Faire la mise a jour dans une table : UPDATE table SET**
* Avec methode **exec(…)**
* Avec methode **prepare(…)**
* **Supprimer des donnees dans la table : DELETE FROM table**
* **Les sessions :**
* **Declaration de session**

Se déclare au du début avant le HTML

* **Fermeture de session**
* **Les cookies :**

Les cookies doivent être place avant le HTML un peu comme les **session.**  Ils sont accessibles via la super global  **$\_COOKIE[‘nom\_cookie’]**

Il doit prendre les valeurs suivant :

* **Nom du cookie**
* **La valeur du cookie**
* **La date d’expiration du cookie sous forme de timestamp(** il s’agit du nombre de seconde écoulée depuis le 1ier janvier 1970**)**

**Exple :**

* **Sécurisation des cookies avec le mode http Only**

Ceci permet de réduire les  **faille XSS**  sur le site dans le cas où on n’a pas utilisé **htmlspecialchars.** Ainsi il rendra les cookies inaccessible avec javascript sur tous les navigateurs.

Ici **true**  veut dire qu’ on active le **mode http Only**

* **PHP\_MVC**

La structure d’un projet en design **MVC** est la suivante :

* **Un dossier models** va appeler(**inclure**) toutes les classes nécessaire a notre page. Et ces classes seront créés dans le dossier **\_classes**
* **Un dossier controllers**  qui contiendra tous fichiers pour le traitement PHP.
* **Un dossier views**  qui aura les pages html destinées a l’utilisateur.
* **Un dossier assets**  qui contiendra :
  + **Un dossier images**  pour les images
  + **Un dossier js**  pour stocker les scripts javascript
  + **Un dossier styles**  pour contenir le style css
* **Un dossier \_config**  qui contiendra les fichiers de configuration de la base de données et de configuration
* **Un dossier \_fonction**  qui contiendra toutes nos fonctions
* **Un dossier \_classes**  qui contiendra nos classes PHP
* **Un dossier \_script**  qui contiendra des fichiers PHP qui nous permettrons des faire des traitements sur une même page( Exple : requête AJAX)
* **Un fichier index.php**  c est le point d’accès ou toutes les requêtes arriverons.

**Exple: fichier index.php**

* **ucfirst(string) :** met la premiere lettre de la string en majuscule.
* **dossier \_config**: composer de
* fichier de configuration : **config.php**

**NB :** il sera disposé ainsi : les erreur, les sessions et les constantes

* Au niveau des **constante** on a les **paths :**
* **1 ligne :** **PATH\_REQUIRE**(**peut être utilise pour la fonction include ou required**) est une constante de chemin. Il prend le nom du dossier avec son chemin jusqu’ a la racine. Ensuite on fait **substr(…)** pour retirer le **index.php** . Ainsi on obtient tous les chemins des fichiers. Destiner pour le PHP.
* **2 ligne :** **PATH**(**va être utiliser pour les attribut html comme src ou href**) qui est utilisé avec **PHP\_SELF**  c est enfaite **URL** du site sans le nom du domaine. Ensuite grâce a substr(…) on retire **index.php** pour obtenir le  **lien courant du site.**  Destiner pour le HTML
* Au niveau des **constante** on a les **info du site**

Ce deuxième groupe contient les  **informations d’un site web.**

* Au niveau des **constante** on a les **data Facebook et base de donnees**

Ce 3 groupe permet l’affichage des donnees Facebook et les infos à la base de donnees.

* Au niveau des erreurs :
* Au niveau de session on aura :
* fichier de configuration de la base de données : **db.php**
* Ensuite on inclure le fichier **config.php et db.php** dans index.php
* **Dans le dossier \_functions**  on cree un fichier  **functions.php**  qui contiendra toutes les fonction qu on utilisera dans notre programme.
* Ensuite on inclure ce fichier  **functions.php**  dans  **index.php**
* En PHP place **exit ;**  dans un programme permet de ne plus exécuter la suite du programme.
* **Configuration du .htaccess :**  ceci va nous permettre d configurer le serveur APACHE. Cela va nous permettre de redesigner url ( **pour ne pas avoir des long url incompréhensible)**

Dans ce fichier on aura :

* **DirectoryIndex index.php** ceci pour spécifier que le dossier de base sera index.php
* **RewriteEngine on** nous permet d’activer **URL Rewriting**(permet de transformer une URL compliquée ou peu intelligible en une URL bien lisible et plus compréhensible pour l’internaute et les moteurs de recherche à la fois)
* **RewriteRule ^([a-zA-Z0-9\_\-]\*)?/?$ index.php?page=$1&%{QUERY\_STRING} :** ici on dit notre url commencera (**^**) soit avec l’un des caractères et se terminera(**$**) également par les même , cela sera présent soit une fois ou pas (**?**). Tout cela se groupe sera place dans **page=$1**, **%{QUERY\_STRING}** va considérer le restant des paramètres en get.