



## Module Web 05

# Mettre à jour une Base de données



## Les compétences visées

</>	Catégorie	Je suis capable de :	
P H P	La balise PHP	Ajouter un code PHP dans la balise <?php	
		Inclure un fichier PHP	
		Positionner la balise php dans le code HTML	
	Les fonctions	Créer des fonctions	
		Utiliser des fonctions	
	Les variables	Créer et utiliser des variables PHP (\$)	
		Utiliser des variables superglobales	
		Utiliser des tableaux associatifs	
	Les formulaires	Récupérer les données d'un formulaire	
		Créer et/ou modifier un formulaire html	
	La classe mysqli	Créer une connexion avec le serveur mysql	
		Exécuter une requête SQL	
		Traiter la réponse de la requête SQL	
M y S Q L	Ecrire une requête	Sélectionner une données dans une bdd	
		Insérer une donnée dans une bdd	
		Mettre à jour une donnée dans une bdd	
G I T	Le dépôt local	Créer un nouveau dépôt local (git init)	
		Ajouter des fichiers dans le dépôt local (git add)	
		Sauvegarder les modifications dans le dépôts (git commit)	
	Le dépôt distant	Créer un dépôt distant et se synchroniser avec (git remote)	
		Publier un dépôt local dans un dépôt distant (git push)	

## TD 1 - Analyse du formulaire d'inscription

### Objectif

Dans ce TD, nous allons analyser le code html d'un formulaire afin de le traiter avec le code PHP

```
<div id="inscription">
  <h1>Inscrivez-vous !</h1>
  <form id="form_inscription" method="post" action="rest.php">
    <label for="nom">Nom : </label>
    <input type="text" id="nom" name="nom_utilisateur">
    <label for="prenom">Prénom : </label>
    <input type="text" id="prenom" name="prenom_utilisateur">
    <label for="adresse_mail">Adresse mail : </label>
    <input type="email" id="adresse_mail" name="adresse_mail_utilisateur">
    <label for="naissance">Date de naissance : </label>
    <input type="date" id="naissance" name="naissance_utilisateur">
    <label for="pseudo">Pseudo : </label>
    <input type="text" id="pseudo" name="pseudo_utilisateur">
    <label for="mdp1">Mot de passe : </label>
    <input type="password" id="mdp1" name="mdp1_utilisateur" required>
    <label for="mdp2">Ressaisir le mot de passe : </label>
    <input type="password" id="mdp2" name="mdp2_utilisateur" required>
    <input type="submit" id="bouton_inscription" name="inscription" value="M'inscrire">
  </form>
</div>
```

### Analyse du formulaire

? En vous référant au code HTML, donner les différentes valeurs « name » attribuées aux « input » du formulaire d'inscription.

? Que signifie action= »rest.php« ?

? Selon le formulaire précédent, donner les différentes clés du tableau associatif \$\_POST qui seront générées.

? Donner le code permettant de créer les variables \$nom\_utilisateur, \$prenom\_utilisateur, \$naissance\_utilisateur, \$pseudo\_utilisateur, \$mdp1\_utilisateur, \$mdp2\_utilisateur et \$inscription, en utilisant une boucle **foreach** et le tableau associatif \$\_POST.

## Requête insertion

? Donner la requête permettant d'insérer un nouvel utilisateur en utilisant les variables précédentes

? Donner le code complet, voir Module MW04/Défi-5, permettant d'exécuter la requête précédente

## Défi 1 - Récupérer des données de formulaire

### Objectif

Récupérer et insérer les données du formulaire d'inscription

### Insérer un nouvel utilisateur

Dans le fichier **rest.php** faite les modifications afin de répondre aux étapes suivantes :

- </>** Ajouter la boucle foreach du TD-1 si la variable \$\_POST n'est pas vide.
- </>** Ajouter la condition qui permet de vérifier si la variable \$inscription existe.
- </>** Si elle existe faire la requête permettant d'insérer un nouvelle utilisateur. Attention pour insérer un utilisateur, il se servir de la requête insert du TD-1 et des lignes de code du module MW04 – Défi-5
- </>** Vérifier que l'utilisateur à bien été insérer dans la base de données en vous connectant à phpmyadmin

### Amélioration : pseudo unique

Le problème de la méthode précédente est que l'on ne vérifie pas si le pseudo est libre avant insertion. Pour améliorer l'étape d'insertion, nous allons effectuer une étape supplémentaire permettant de vérifier si le pseudo existe ou non dans la bdd.

- ?** Donner la reqête permettant de sélectionner le nom d'un utilisateur connaissant son pseudo.

- </>** Coder les étapes suivantes :

```
// Exécuter la requête précédente et récupérer la réponse de la BdD dans la variable $res (query)
// si le nombre de ligne du résultat $res est différent de zéro (num_rows)
//Alors on insérer un utilisateur (insert)
//Sinon on affiche un message d'erreur
```

## Défi 2 - Rediriger l'utilisateur vers la page d'accueil

### Objectif :

Une fois le traitement par le fichier rest.php est effectué, il faut renvoyer l'utilisateur vers la page d'accueil.

### Redirection

Pour rediriger vers un autre fichier php , on utilisera :

```
header('Location:chemin-vers-fichier.php');
```

#### Dans le cas d'une connexion

Après avoir effectué les étapes de connexion dans le fichier rest.php, il faut maintenant rediriger l'utilisateur vers la page d'accueil

**</>** Compléter la partie de connexion afin de renvoyer l'utilisateur vers la page d'accueil (index.php).

#### Dans le cas d'une inscription

**</>** Compléter la partie d'inscription afin de renvoyer l'utilisateur vers la page d'accueil (index.php).

### Afficher un message d'erreur

Lorsqu'il y a une erreur de connexion ou d'inscription, il faut afficher un message d'erreur à l'utilisateur. Pour cela le fichier rest.php doit envoyer cette information au fichier index.php.

#### Partie connexion

##### Modification du fichier rest.php4

**</>** Dans la partie du message d'erreur, il faut rediriger vers la page d'accueil en ajoutant un paramètre dans l'URL : « erreur=PseudoConnexion ». Modifier la fonction « header » afin d'ajouter la prise en compte de l'erreur par le fichier index.php comme dans l'exemple suivant :

```
header('Location:index.php?erreur=PseudoConnexion');
```

Ainsi si le fichier index.php voit dans la barre URL apparaître l'information « erreur=PseudoInscription » alors un message d'alerte permettra d'informer l'utilisateur qu'il y a une erreur d'inscription.

##### Modification du fichier index.php

Pour récupérer une information dans la barre URL, il faut utiliser la variable \$\_GET

**?** Donner le nom de la clé ainsi que la valeur associée stockée dans la variable \$\_GET

**</>** A la fin du fichier index.php ajouter le code permettant de vérifier si il y a l'information erreur dans la barre URL. Si elle vaut PseudoConnexion alors on affiche une alerte à l'utilisateur qu'il y a une erreur dans le formulaire de connexion.

#### Partie Inscription

##### Modification du fichier rest.php

**</>** Dans la partie du message d'erreur, il faut rediriger vers la page d'accueil en ajoutant un paramètre dans l'URL : « erreur=PseudoInscription ». Modifier la fonction « header »

##### Modification du fichier index.php

**</>** A la fin du fichier index.php ajouter le code permettant de vérifier si il y a l'information erreur dans la barre URL. Si elle vaut PseudoInscription alors on affiche une alerte à l'utilisateur que le pseudo est déjà utilisé.

## Défi 3 - Les cookies

### Objectif :

Garder une trace des données de navigation du client sur PC client.

### Création d'un cookie

? En cherchant sur internet, donnez la définition d'un cookie informatique.

? En vous du lien suivant, <https://www.php.net/manual/en/function.setcookie.php>, lister les différents paramètres à intégrer lors de la création d'un cookie

? Donner le code permettant de créer le cookie pseudo dont la valeur est login1, d'une durée de une heure accessible sur l'ensemble du site

? Quel tableau associatif est alors utilisé pour accéder aux cookies ?

? Donner le code permettant d'accéder au contenu du cookie pseudo

### Destruction d'un cookie

? Donner le code permettant de supprimer le cookie précédent

## Défi 4 - Utilisation des cookies

### Objectif

Ajouter des cookies afin de stocker des informations sur l'utilisateur pour lui proposer un service personnalisé.

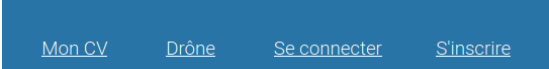

### Ajouter un cookie

#### Modification du fichier rest.php

##### Formulaire de connexion et inscription

- </> Dans la partie qui concerne le formulaire de connexion, ajouter le code permettant de créer un cookie après avoir validé la connexion et avant de renvoyer l'utilisateur vers le fichier index.php
- </> Faire de même pour la partie inscription

#### Modifier le fichier index.php

 Figure 1: Barre de navigation avant la connexion	 Figure 2: Barre de navigation après la connexion
---	--

Il faut ici modifier la barre de navigation lorsqu'une connexion est réalisée. Elle sera effective une fois que le cookie sera créé et tant qu'il sera présent.

- </> Dans la balise <nav> de votre index.php, compléter le code suivant

```
<a href=""> Se connecter </a>
<a href=""> S'inscrire </a>
```

afin de tenir compte de la présence ou non du cookie pseudo, en utilisant les balises

```
<a href="rest.php?profil"> Profil </a>
<a href="rest.php?deconnexion"> Se deconnecter </a>
```

Attention, il ne faudra pas oublier les balise <?php ..... ?> si vous désirez ajouter du code php dans votre fichier index.php

- </> Vérifier le bon fonctionnement de votre code en vous connectant à l'aide du formulaire de connexion

### Supprimer un cookie

Nous allons ici modifier le fichier rest.php afin de supprimer tous les cookies lorsque \$\_GET['deconnexion'] existe.

- </> Modifier le fichier rest.php afin de répondre à la problématique précédente. Astuce utiliser une boucle foreach sur la variable \$\_cookie
- </> Vérifier le résultat en vous connectant puis vous deconnectant. Vous devriez retrouver la barre de navigation d'origine.



## Défi 5 - Modifier son profil

### Objectif :

Afficher le profil de l'utilisateur dans un formulaire et lui permettre de mettre à jour les informations.

### Récupération des données

Modifier le fichier rest.php afin de créer des cookies stockant le nom, le prénom et l'adresse mail de l'utilisateur contenu dans le cookie 'pseudo'. Voici les étapes à écrire dans rest.php :

- 1 - Détecter si le terme profil est dans l'URL
- </> 2 - Ecrire une requête permettant de récupérer le nom, le prénom et l'email de l'utilisateur connaissant son pseudo
- 3 - Créer trois cookies contenant le nom, le prénom et l'email
- 4 - Rediriger l'utilisateur vers l'index.php en y ajoutant le terme profil

### Afficher le profil de l'utilisateur

Ici nous allons modifier le fichier index.php afin de modifier la section en fonction de la demande de l'utilisateur. Si l'utilisateur souhaite afficher le profil alors il faudra remplacer le contenu de la section par un formulaire contenant le nom, le prénom, l'email et le pseudo de l'utilisateur.

- </> Modifier le fichier index.php afin d'intégrer le formulaire figure 3. Il devra renvoyer vers le fichier rest.php et le bouton aura comme « name='miseajour' »
- </> Intégrer le formulaire dans une balise php avec la condition if(isset(\$\_GET['profil'])) ce qui affichera le formulaire lorsque l'utilisateur clique sur « Profil »
- </> Mettez le reste du code de la section dans un else afin de ne garder que le formulaire sur la page web

### Mettre à jour le profil

- </> Modifier le fichier rest.php afin de détecter le bouton « miseajour »
- Compléter le code précédent avec
- 1 - la requête mettant à jour (« UPDATE ») les données de l'utilisateur. Attention la mise à jour du pseudo ne peut se faire que si le pseudo n'est pas déjà utilisé.
- </> 2 - mettre à jour ensuite les cookies
- 3 - renvoyer l'utilisateur vers la page d'accueil.

Figure 3: Formulaire profil