

TP3

IHM – PHP

Ressources externes

En complément du cours, vous pouvez utiliser :

- Pour HTML et CSS :
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>
<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>
<https://www.w3schools.com/>
- Pour PHP :
<https://www.php.net/manual/en/>
<https://www.w3schools.com/php/default.asp>
- Pour valider le contenu :
<https://validator.w3.org/>

Information importante

Ce TP est à rendre de manière **individuelle** pour le **vendredi 21 avril**. L'espace de rendu sur Moodle fermera à 23h59. Vous mettrez dans un **dossier zip tous vos dossiers**.

Penser à valider chaque page HTML construite.

Mise en place du TP

Dans ce TP vous allez reprendre la structure du TD4 et le design de votre précédent site (TP précédent). Une partie de la note du TP sera sur la structure de votre site, le choix des noms des fichiers et les fonctions écrites. Vous devez donc avoir une structure **propre**.

Lors de ce TP vous allez devoir utiliser une base de données. Télécharger le fichier *pokemon.sql* sur moodle et importer le dans *phpMyAdmin*.

Observer la structure de la base de données, c'est-à-dire les différentes tables, noms des colonnes et relations. N'oubliez pas également de regarder le contenu, cela pourra vous aider par la suite.

Objectif du TP

Le but de ce TP est de réaliser un **Pokédex pour différent joueurs**. Un dresseur pourra se connecter à son compte sur le site internet pour visualiser les Pokémons attrapés ou s'il les a déjà observés. Une fonctionnalité pour mettre à jour le Pokédex sera également proposé. Si aucun dresseur n'est connecté, un Pokédex de base sera affiché. Bien évidemment, chaque internaute pourra accéder à une description plus poussée d'un Pokémon.

Exercice 1 : Connexion à une base de données

Au début de cet exercice, vous devez avoir une structure simple de votre site avec un fichier *index.php* contenant uniquement les parties communes de toutes vos futures pages web (l'entête, un début de barre de navigation et le pied de page). Adapter le design de votre site internet pour être cohérent par rapport au thème de ce TP.

La base de données doit être également déjà importée sur *PhpMyAdmin* et vous avez déjà pris connaissance de sa structure.

1. Modifier le fichier *config-bdd.php* situé dans le dossier *includes* de votre projet. Vous devez mettre à jour les paramètres de connexion à la nouvelle base de données.
2. Vérifier si vos deux fonctions génériques permettant de gérer la connexion et déconnexion à la base de données se situent bien dans votre projet. Elles doivent être placées dans le fichier *php/functions-DB.php*.
1. Tester les deux fonctions dans la page principale de votre site. Appeler la fonction *connectionDB()* et récupérer dans une variable le lien avec la BDD (flux). Cette fonction doit être insérée avant un affichage de données provenant de la base de données. Appeler la fonction *closeDB()* avec le bon paramètre d'entrée à la fin de votre fichier, lorsque vous n'avez plus besoin de réaliser de requêtes. Vous pouvez par exemple la placer avant la fermeture de la balise *body*.
2. Tester votre site internet en allant sur *index.php*. Vous ne devez avoir aucun message d'erreur si la connexion a été correctement réalisée. Pour rappel, les messages d'erreurs s'affichent puisque nous avons inséré la ligne suivante dans notre fichier *php* : `mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);`

Exercice 2 : Première requête

Maintenant que la connexion à la base de données est réalisée, nous allons écrire une première requête. Nous allons tout d'abord préparer des fonctions génériques si elles n'ont pas encore été écrites.

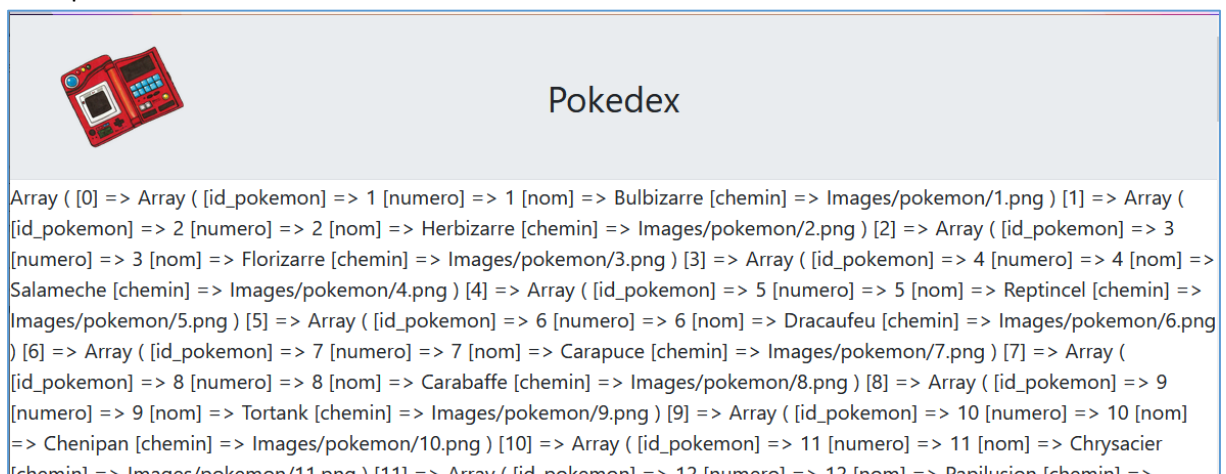
1. Dans le fichier *php/functions-DB.php*, compléter les deux fonctions *readDB()* et *writeDB()*. Observer les paramètres d'entrée et la variable de retour attendu.
2. Dans le fichier *index.php*, écrire une requête permettant de récupérer tous les numéros et noms des Pokémon. Afficher le résultat à l'aide de la fonction *print_r ()*. Pour mieux visualiser les résultats, vous pouvez encadrer cette fonction de la balise HTML suivante : `<pre> ... </pre>` (prettier). Observer le résultat obtenu.
3. La requête précédente nous a retourné un tableau associatif. A l'aide de boucles *foreach* afficher à l'aide de la fonction *echo* chaque élément de ce tableau associatif.
4. Cette requête ne vous sera pas utile dans la suite du TP, vous pouvez mettre le bout de code concerné en commentaire.

Exercice 3 : Un Pokédex

Dans cet exercice nous allons afficher un Pokédex qui va permettre d'afficher pour tous les internautes l'image, le numéro et le nom de tous les Pokémon de la base de données.

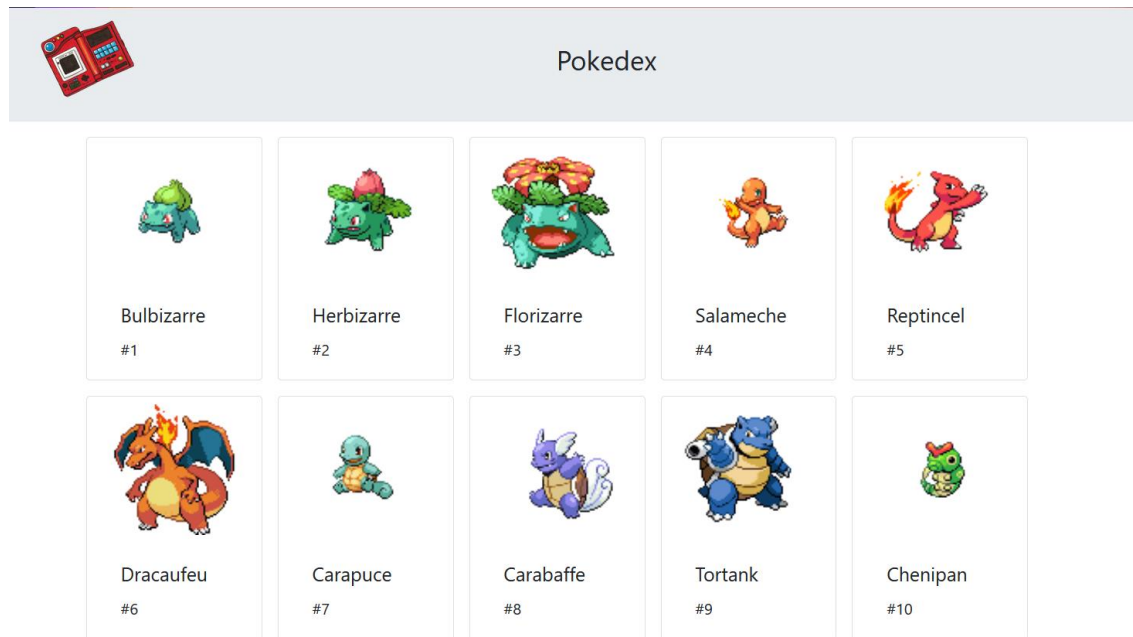
1. Créer un nouveau fichier *php/functions_query.php* qui contiendra toutes les fonctions permettant d'interroger la base de données.
2. Créer dans ce nouveau fichier une fonction : **getPokedex(\$mysqli)**
 Cette fonction va nous permettre de questionner la base de données et d'obtenir toutes les informations que nous devons afficher par la suite dans la page *index.php*. Cette **fonction retourne un tableau associatif** contenant pour chaque Pokémon : son id, son numéro, son nom et une image. Attention, sur moodle vous avez plusieurs dossiers contenant des images. Vous pouvez choisir le dossier que vous voulez.
 Pensez à utiliser la méthode **readDB(...)** du fichier *function_DB.php*.
Quelle requête SQL devez-vous utiliser ?

3. Testez votre fonction. Pour l'instant vous pouvez l'appeler directement dans *index.php* et afficher le résultat via l'appel de la fonction *print_r*.
 Vous pouvez obtenir le résultat suivant :



4. Créer un nouveau fichier *php/functions_structure.php*.
5. Nous allons maintenant afficher le Pokédex d'une manière plus agréable pour l'internaute. Créez une fonction **displayPokedex(\$pokedex)** dans le fichier *functions_structure.php*.
 Cette fonction prend en paramètre le tableau associatif contenant le résultat de la précédente requête SQL. La fonction va parcourir ce tableau associatif et afficher le contenu dans des balises HTML. Vous êtes libre de choisir la forme de votre Pokédex (cards, tableau, etc.).
6. Appelez la fonction **displayPokedex(\$pokedex)** dans le fichier *index.php*.

Voici un exemple du résultat attendu :



Exercice 4 : Un Pokémon

Dans cet exercice, nous souhaitons maintenant afficher le détail d'un Pokémon.

1. Créer une page **pokemon.php**.
2. Lorsque nous cliquons sur l'image d'un Pokémon, nous sommes redirigés vers la page **pokemon.php**. Ajouter sur l'image du Pokémon une balise pour réaliser un lien hypertexte. Le numéro (ou le nom) du Pokémon peut être passé en tant que paramètre de l'URL (GET).
3. Récupérer ce numéro au début de la page **pokemon.php** et l'afficher. Penser à vérifier si le paramètre existe bien dans la page. Si ce n'est pas le cas, vous pouvez réaliser au début de votre fichier une redirection vers le fichier **index.php** à l'aide de la fonction php `header()` :
<https://www.php.net/manual/en/function.header.php>
4. Nous souhaitons afficher sur cette page diverses informations :
 - le nom, le numéro et la description du Pokémon,
 - les types du Pokémon,
 - les images liées à ce Pokémon,
 - la liste de ses attaques (avec toutes leurs informations),
 - les évolutions possibles du Pokémon (dans un premier temps, juste le Pokémon suivant et en bonus la succession d'évolution).

Réalisez les requêtes SQL nécessaires pour obtenir ces informations. Affichez ensuite le résultat sur la page **pokemon.php**.

Veuillez respecter également le principe des deux fichiers **functions-query.php** et **functions-structure.php** : ajouter des fonctions dans le premier fichier pour interroger la base de données et ajouter des nouvelles fonctions dans le second fichier pour réaliser la structure HTML de la page web.

5. Tester les nouvelles fonctionnalités.

Exercice 5 : Connexion d'un dresseur

Dans cet exercice, nous allons gérer la connexion et déconnexion d'un dresseur Pokémon au site internet.

1. Créer la page **connection.php**. Cette page sera visible par un internaute, elle contiendra des balises HTML.
2. Créer les deux pages suivantes : **php/login.php** et **php/logout.php**. Ces deux pages contiendront uniquement du code PHP pour traiter l'identification du dresseur et sa déconnexion.
3. Modifier la barre de navigation afin d'accéder à la page **connection.php** (possibilité d'ajouter un item *connexion* par exemple).
4. Ajouter un formulaire HTML dans la page **connection.php** pour envoyer à la page **login.php** le *login* et le *mot de passe* du dresseur. Vous utiliserez la méthode *POST* pour transmettre les données.
5. Dans un premier temps, récupérer les informations du formulaire dans la page **login.php** lorsque le formulaire est envoyé. Les afficher à l'aide d'un *echo*. Quelle variable superglobale avez-vous utilisé ? Si cette partie fonctionne, vous pouvez mettre en commentaire ce code.
6. Dans le fichier **php/functions-query.php**, ajouter une fonction prenant en paramètre d'entrée un login et un mot de passe. Attention, cette fonction va interroger la base de données vous devez donc rajouter comme paramètre d'entrée le lien de connexion à la base de données.
Cette fonction va permettre de vérifier si un dresseur existe dans la base avec le login et le mot de passe passés en paramètre. Quelle requête pouvez-vous écrire ?
La fonction va retourner le résultat de la requête, c'est-à-dire un tableau associatif (celui-ci peut être vide).
7. Dans le fichier **php/login.php**, réaliser les étapes suivantes uniquement lorsque le formulaire de connexion est envoyé :
 - Ouvrir la connexion avec la base de données.
 - Vérifier si le login et le mot de passe correspondent à un dresseur.
 - Si ce n'est pas le cas (la requête retourne false ou un tableau vide), se rediriger vers la page de connexion à l'aide de la fonction *header*.
 - Si un dresseur est trouvé, utiliser **les variables de sessions** pour stocker le login du dresseur, son id et un booléen permettant de savoir facilement si le dresseur est connecté. Effectuer ensuite une redirection vers la page contenant le Pokédex.
 - N'oubliez pas de fermer la connexion avec la base de données avant les redirections.
8. Modifier les pages précédentes pour savoir si un dresseur est connecté sur celles-ci (utilisation des variables de session). Vous pouvez temporairement afficher un message sur ces pages pour savoir si un dresseur est connecté ou non.
9. Modifier la barre de navigation lorsqu'un dresseur est connecté. L'item *connexion* sera remplacé par l'item *déconnexion* (utilisation des variables de session et de structure de conditions). Lorsqu'un dresseur clique sur l'item *déconnexion*, la page **php/logout.php** sera exécutée.

10. Réaliser la déconnexion d'un dresseur dans la page **php/logout.php**. Vous devez **détruire les variables de sessions**, puis effectuer une redirection vers la page **index.php**.

Exercice 6 : Pokédex d'un dresseur

Lorsqu'un dresseur est connecté, le Pokédex affiche ses propres découvertes. Il va mettre en évidence les Pokémon vus et attrapés.

1. Dans **php/fonctions-query.php**, créer **une nouvelle fonction** permettant de retourner pour un dresseur donné, le nombre de fois qu'un Pokémon a été vu et attrapé (ce Pokémon est un paramètre d'entrée de la fonction).
Vous pouvez si vous le souhaitez afin d'éviter des requêtes multiples, ajouter une fonction **getPokedexDresseur** qui sera similaire à la fonction **getPokedex**, mais qui retournera en plus dans son tableau associatif les valeurs *nbVue* et *nbAttrape* pour chaque Pokémon de la base de données. Pour la requête, vous pouvez regarder les RIGHT ou LEFT JOIN.
2. Modifier la page du Pokédex pour le dresseur connecté. Après avoir appelé une des fonctions précédentes, modifier l'affichage :
 - Un Pokémon jamais observé peuvent avoir une image plus transparente que les autres (propriété CSS opacity).
 - Un Pokémon attrapé peut avoir une icône à côté de son image ou de son nom (une Pokéball par exemple).
3. Ajouter également ces informations dans le détail de chaque Pokémon (page **pokemon.php**).

Exercice 7 : Modification du Pokédex d'un dresseur

Vous proposerez dans cet exercice un formulaire pour permettre au dresseur connecté de modifier son Pokédex.

1. En respectant toujours la séparation des fichiers de votre site internet, créer sur une nouvelle page **php** votre formulaire permettant au dresseur de :
 - choisir son Pokémon (liste déroulante initialisé via une requête SQL),
 - modifier le nombre de vue,
 - modifier le nombre de Pokémon de cette espèce attrapée.
2. Modifier la barre de navigation afin de pouvoir accéder à cette page pour un dresseur connecté.
3. Récupérer dans une autre page **php** le résultat de ce formulaire et modifier la base de données. Attention au type de la requête (UPDATE ou INSERT INTO).

Exercice 8 : Modification supplémentaire (Bonus)

Nous voulons afficher sur le Pokédex les images des types des Pokémon. Implémenter cette fonctionnalité. Plusieurs solutions sont possibles : soit lier les images dans la base de données ou les charger en fonction du type du Pokémon.