

PHP - SQL

ENSIIE FISA FC Bases du Web 2025-2026

Remarque : Pour démarrer un serveur PHP pour vous entraîner ou développer rapidement une page, utilisez la commande suivante (dans un environnement linux)

```
php -S localhost:8888
```

Vous pouvez ensuite vous connecter à l'adresse <https://localhost:8888>

Exercice 1 — *Page simple sur vos animaux favoris*

Créez une page web PHP `animals.php`.

- Définissez au début du fichier dans cette page un dictionnaire `$animals`, par exemple

```
$animals = array( "foo" => "bar", "bar" => "foo", );
```

 Ce dictionnaire doit contenir :
 - dans ses clefs les noms de vos 5 animaux favoris
 - dans ses valeurs un tableau contenant
 - le nom en latin de l'animal
 - un lien vers sa page wikipedia (ou la page du sujet le plus proche)
 - un booléen disant s'il est plutôt gros ou plutôt petit
 - un commentaire
- Démarrez ensuite dans ce fichier une page html. Ce fichier doit afficher :
 - une liste non ordonnée de tous les noms des 5 animaux. Chaque nom doit être dans une balise `a` pour faire un lien interne à la page (avec `href="#id`" où `id` est l'id d'un élément dans la page HTML.) Ce lien interne servira à aller plus bas dans la page pour avoir du détail sur l'animal correspondant.
 - un filet de séparation et un espace vide d'environ 500px
 - un tableau contenant 5 colonnes `nom`, `nom latin`, `wikipedia`, `taille?` `commentaire` Chaque ligne du tableau contient les 5 infos relatives chaque l'animal données dans le dictionnaire. La colonne `taille?` doit afficher `Gros` ou `Petit` selon le booléen dans le tableau.
 - un filet de séparation et un espace vide d'environ 500px
 - A chaque ligne doit être attribué un `id` qui doit être utilisé pour que chaque lien d'un animal de la liste non ordonnée pointe vers la ligne du tableau correspondant (donc si vous avez un animal `chat`, en cliquant sur ce lien dans la liste non ordonnée, la navigateur se déplace jusqu'à la ligne du tableau correspondant). Les espaces vides de la page servent à s'assurer que le déplacement fonctionne correctement.
- Créer une base de données (en utilisant le tutoriel correspondant)
- Dans cette base, créez une table `animals` avec 5 colonnes `name latin link size comment` avec une requête `CREATE TABLE`. La clef primaire sera la colonne `name`
- Avec une requête `INSERT`, insérez dans cette table vos 5 animaux avec les infos initialement entrées dans le tableau `$animals`.
- Copiez votre fichier PHP dans un nouveau fichier `animals_sql.php`
- Dans ce fichier, initialisez votre tableau `$animals` avec un tableau vide `$animals = array()`.
- Créez une connexion à votre base de données avec `pg_connect`.
- Effectuez une requête `SELECT` avec `pg_query` pour récupérer les infos des animaux dans la table `animals` et remplissez votre tableau `$animals` avec ces infos. Vérifiez que tout fonctionne bien sur votre page. Vous devriez avoir le même résultat que dans la première page.

Exercice 2 — *Formulaire*

1. Créez une base de données (en utilisant le tutoriel correspondant).
2. Dans cette base, créez une table `people` avec 3 colonnes `id` `name` `age` avec une requête `CREATE TABLE`. La clef primaire sera la colonne `id`.
3. Avec une requête `INSERT`, insérez dans cette table 3 lignes contenant le nom d'une personne et son age (fictifs).
4. Créez un fichier `age.php` dans lequel vous vous connectez à la base avec `pg_connect` et dans lequel vous effectuez une requête `SELECT` pour récupérer les infos de la table pour les stocker dans un tableau.
5. Démarrez ensuite dans ce fichier une page html. Ce fichier doit afficher :
 - un tableau avec 3 colonnes affichant les noms et ages des personnes de la table et, dans la 3e colonne, 2 boutons `Edit` et `Delete`
 - un formulaire contenant 3 `inputs` (de type `text`, `number` et `hidden`), des `labels` et un `button` pour soumettre. Le premier champs correspond au nom d'une personne et le second à l'age d'une personne. L'`input` de type `hidden` correspond à l'`id` de la personne dans la base de donnée.
6. Faites en sorte que, si on clique sur un des boutons `delete`, la page se recharge et la ligne correspondante disparaisse du tableau (et donc de la base de données). Utilisez pour cela une requête de type `GET` vers la même page, et une requête `DELETE`.
7. Faites en sorte que, si on clique sur un des boutons `edit`, la page se recharge et les `inputs` du formulaire sont préremplis avec le nom de la personne sur laquelle on a cliqué, son age et son id. Utilisez pour cela une requête de type `GET` vers la même page et une requête `SELECT` ou des paramètres de la requête `GET` au choix.
8. Faites en sorte que, si on valide le formulaire, les champs correspondant au nom et à l'age soient remplis pour que le formulaire envoie les infos.
9. Faites en sorte que, si on valide le formulaire, une requête `POST` vers la même page est transmise. Si l'`id` est rempli dans le formulaire `hidden`, alors il faut effectuer une requête `UPDATE` à la base de données pour mettre à jour la personne correspondante. Sinon, il faut effectuer une requête `INSERT` pour ajouter quelqu'un à la base.
10. Effectuez des vérifications en PHP avant d'effectuer les requêtes `UPDATE` ou `INSERT` pour vérifier que le nom transmis par le formulaire fait moins de 20 caractères, que l'age est un nombre et que l'`id` est un nombre. Utilisez ensuite les fonctions telles que `htmlspecialchars` pour éviter des injections HTML dans le code.
11. Si ce n'est pas fait, utilisez des requêtes préparées pour éviter des injections SQL.

Exercice 3 — *Chaud Froid avec session*

Créez une page PHP `chaudfroid.php`. En utilisant une variable de session, faites en sorte que, la première fois que la page est visitée, un nombre aléatoire entre 0 et 20 est choisi. Ce nombre ne doit pas être réinitialisé lorsque la page est rechargée. Faites ensuite une page avec un formulaire qui propose un jeu de chaud-froid pour retrouver le nombre.