
To-do List Exercice

To-do List

Exercice

Qu'est qu'une To-do List ?

Une Todolist est un outil d'organisation personnelle qui permet de lister les tâches à accomplir, de suivre leur avancement et de les marquer comme étant terminées lorsqu'elles sont accomplies.

Les to-do lists sont couramment **utilisées pour gérer** :

- les tâches quotidiennes,
- les projets professionnels ou personnels,
- les listes de courses,
- les objectifs à long terme
- ...

Les to-do lists offrent **plusieurs avantages** :

- Amélioration de la productivité
- Réduction du stress
- Suivi des progrès
- Meilleure organisation

To-do List

Exercice

Les fonctionnalités de notre Todolist

Dans cet exercice, nous allons créer une Todolist en PHP et SQL avec les fonctionnalités suivantes :

- **Liste des tâches** : la liste des tâches s'affiche avec une case à cocher à côté de chaque tâche, indiquant si elles sont accomplies ou non.
- **Tâche accomplie** : en cochant la case à côté d'une tâche l'utilisateur indique qu'elle est terminée, plus il clique sur le bouton "Enregistrer" pour sauvegarder la modification.
- **Nouvelle tâche** : en saisissant du texte dans le champ prévu à cet effet et en cliquant sur le bouton "Ajouter", l'utilisateur voit sa tâche ajouter à la liste.

To-do List

Exercice

Organisation du projet de To-do list

Avant de commencer à coder, prenez le temps de réfléchir aux fonctionnalités et aux étapes nécessaires pour réaliser votre Todolist.

Pour **organiser vos idées**, vous pouvez, par exemple, créer des schémas, des diagrammes, et même une liste de tâches !

Il est également conseillé de **noter et de faire des schémas** représentants les éléments de la page et les fonctionnalités :

- listes de tâches,
- cases à cocher,
- formulaire d'ajout.



To-do List

Exercice

Organisation du code

Voici quelques conseils pour bien organiser votre code :

- Utilisez des commentaires pour faciliter la compréhension et le suivi de votre projet.
- Organisez et regroupez votre code par fonctionnalités quand c'est possible en ajoutant des sauts de ligne.
- Respectez l'indentation du code en utilisant la touche TAB pour mettre en retrait le code dans les boucles, les conditions, ...
- Utilisez des noms de variables clairs et explicites.
- Testez le résultat de votre code dans le navigateur à chaque étape pour trouver et corriger les erreurs plus facilement.



To-do List

Exercice

Configuration de la base de données

Pour stocker les tâches de notre to-do list, nous avons besoin d'une base de données.

Suivez les sous-étapes suivantes pour configurer cette base :

1. Connectez-vous à votre serveur MySQL à l'aide d'un outil comme **phpMyAdmin**.
2. Créez une **nouvelle base** de données et nommez-la "todolist".
3. Créez une **table** nommée "tache" avec les colonnes suivantes :
 - id (INT, clé primaire, auto-incrément)
 - titre (VARCHAR, 255 caractères)
 - fait (INT, valeur par défaut 0)
4. Ajouter des **données** de test dans la table.

To-do List

Exercice

Connexion à la base de données

Une fois la base et la table configurées, créez un fichier PHP pour interagir avec la base de données :

1. **Créez un dossier** dédié à votre projet Todolist dans C:\xampp\htdocs. Et **créez un fichier** "index.php" à la racine de celui-ci.
2. Ajoutez les variables pour les **paramètres de connexion** à la base de données.
3. Utilisez la fonction `mysqli_connect()` pour **établir la connexion**.
4. **Vérifiez si la connexion a réussi** en utilisant la fonction `mysqli_connect_error()`.

To-do List

Exercice

Afficher les tâches

Affichons maintenant les tâches stockées dans la base de données à l'aide des étapes suivantes :

1. Dans le code PHP, **créez une requête** SQL SELECT pour récupérer toutes les tâches.
2. **Exécutez la requête** à l'aide de la fonction `mysqli_query()`.
3. **Parcourez les résultats** à l'aide d'une boucle foreach.
4. Dans la boucle, **affichez chaque tâche** avec une case à cocher à côté.

Conseil : Pour vérifier si la requête a fonctionné, avant de réaliser l'étape 3, nous pouvons afficher ce qu'elle contient via la fonction `print_r()`.

To-do List

Exercice

Ajouter une nouvelles tâches

Créons maintenant un formulaire pour ajouter de nouvelles tâches à la Todolist.

1. Dans le fichier PHP, **créez un formulaire** HTML avec un champ texte et un bouton de soumission "Ajouter".
2. Dans le code PHP, **vérifiez si le bouton "Ajouter"** a été cliqué en utilisant la fonction isset().
3. **Créez une requête** SQL INSERT pour insérer la nouvelle tâche dans la base de données.
4. **Exécutez la requête** à l'aide de la fonction mysqli_query().

Rappel : Tester le code dans son navigateur à chaque étape permettra de trouver et corriger les erreurs plus facilement.



To-do List

Exercice

Marquer une tâche comme accomplie

Pour marquer une tâche comme accomplie, nous allons mettre à jour la colonne "fait" dans la base de données.

- **Ajoutez un bouton** "Enregistrer" dans le formulaire contenant les tâches.
- Dans le code PHP, **vérifiez si le bouton** "Enregistrer" a été cliqué en utilisant la fonction isset().
- Parcourez les résultats de la base de données et **mettez à jour** la colonne "fait" en fonction des cases à cocher sélectionnées.

Rappel : Tester le code dans son navigateur à chaque étape permettra de trouver et corriger les erreurs plus facilement.



To-do List

Exercice

Tester votre développement

Maintenant que votre Todolist est fonctionnelle, il est important de la tester pour vous assurer de sa qualité et de sa sécurité. Voici quelques tests à effectuer :

Vérifiez que la saisie de **longues chaînes de texte** dans les formulaires ne provoque pas de problèmes d'affichage ou de fonctionnement.

Testez l'insertion de **caractères spéciaux** et de **balises HTML** dans les champs de texte pour vous assurer que l'application gère correctement ces entrées et évite les injections de code malveillant.

Assurez-vous que les champs obligatoires sont correctement validés et que les messages d'erreur s'affichent lorsque les utilisateurs soumettent des données invalides ou incomplètes.

Vérifiez que l'application ne plante pas ou n'effectue pas d'actions indésirables lorsque les utilisateurs **cliquent rapidement** et à plusieurs reprises sur les boutons.



To-do List

Exercice

Evolution fonctionnelles possibles

Voici quelques évolutions possibles pour améliorer notre Todolist :

- supprimer la tâches,
- ajouter des dates d'échéance,
- ajouter des priorités aux tâches,
- ajouter des catégories aux tâches,
- ajouter un système de filtres et de tri,
- ...toutes autres fonctionnalités qui vous paraissent utiles.