

PHP

Passez à la vitesse supérieure avec PHP

Plan du cours

- Introduction à PHP
 - C'est quoi ?
 - Utilisation dans le développement web
- Architecture Client / Server
- Communication entre Client et Serveur
- Fonctionnement de site web statique et dynamique
- Environnement de développement
 - Téléchargement et installation d'un serveur (XAMPP, WAMP, MAMP,...)
- Premier pas avec PHP
 - Structure d'une page PHP
 - Affichage de texte
 - Commentaires

Plan du cours

- Variables et types de données
 - Déclaration de variable
 - Types de données de base
 - Opération sur les données
- Les structures conditionnelles
 - if, else, elseif, switch
- Les structure itératives
 - for, while, do while
- Inclusions de fichiers
 - include
 - require
- Les fonctions

Plan du cours

- Les tableaux
 - Tableaux indicés
 - Tableau associatifs
 - Parcours et affichage des éléments d'un tableau
- Formulaires HTML et Traitement PHP
 - Création des formulaire HTML
 - Soumission de formulaire et traitement des données
 - Validation des données côté serveur
- Interaction avec une Base de données CRUD
 - Requête d'insertion
 - Requête de lecture
 - Requête de mise à jour
 - Requête de de suppression

Plan du cours

- Sessions et Cookies
 - Utilisation des sessions pour maintenir l'état des utilisateurs
 - Gestion des cookies
- Gestion de fichiers
 - Lecture
 - Ecriture
- Hébergement d'un site

Formulaires HTML et traitement PHP

Les formulaires en PHP sont utilisés pour collecter des données saisies par les utilisateurs sur une page web et les traiter côté serveur. Une fois qu'un utilisateur soumet un formulaire, les données sont envoyées au serveur où on pourra les récupérer et les traiter à l'aide de PHP.

Création d'un formulaire PHP

Créez un formulaire HTML dans lequel les utilisateurs saisissent les données.

```
<form action="traitement.php" method="POST">
    <label for="nom">Nom :</label>
    <input type="text" name="nom"><br><br>

    <label for="email">Email :</label>
    <input type="email" name="email"><br><br>

    <label for="message">Message :</label><br>
    <textarea id="message" name="message" rows="5" cols="30"></textarea><br><br>

    <input name="envoyer" type="submit" value="Envoyer">
</form>
```

Création d'un formulaire PHP

Attributs `action` `method` et `name`:

- **action** : Spécifie l'URL vers laquelle les données du formulaire seront envoyées pour traitement.
- **method** : Définit la méthode HTTP utilisée pour envoyer les données
- **name** : est utilisé pour donner un nom aux champs de formulaire, ce qui permet de les identifier et de récupérer les données saisies par les utilisateurs lorsque le formulaire est soumis.

Soumission de formulaire et traitement des données

Traitement des données en PHP :

Créez le fichier traitement.php pour traiter les données soumises par le formulaire. Vous pouvez récupérer les données à l'aide du tableau superglobal \$_POST.

```
if (isset($_POST["envoyer"])) {  
    $nom = $_POST["nom"];  
    $email = $_POST["email"];  
    $message = $_POST["message"];  
  
    // Traitement des données, par exemple : enregistrement en base de données, etc.  
}
```

Validation des données

La validation des données côté serveur est une étape essentielle lors du traitement des données soumises par les utilisateurs via un formulaire. Elle vise à garantir que les données saisies sont valides, sécurisées et conformes à vos attentes avant de les utiliser dans votre application.

Vérification de la présence des données :

s'assurer que les champs obligatoires sont remplis avant de procéder à toute autre validation.

Validation des données

Validation du format des données :

Vérifier que les données saisies correspondent au format attendu (par exemple, une adresse e-mail valide, un numéro de téléphone correct, etc.)

Validation de la longueur et des limites :

S'assurer que les données saisies ne dépassent pas une certaine longueur ou ne dépassent pas les limites autorisées.

Validation des données

Validation des types de données :

Vérifier que les données saisies correspondent aux types de données attendus (par exemple, un nombre entier, une chaîne de caractères, etc.).

Filtrage et échappement des données :

Utilisez des fonctions d'échappement appropriées, comme `htmlspecialchars()`, pour prévenir les attaques de type XSS (Cross-Site Scripting) en neutralisant les caractères spéciaux.

Téléchargement de fichiers d'image

Pour permettre aux utilisateurs de télécharger des fichiers images depuis leur navigateur, vous pouvez créer un formulaire HTML avec un champ de type "file". Une fois le fichier téléchargé, vous pouvez utiliser les superglobales `$_FILES` pour manipuler les informations du fichier téléchargé. Il faut aussi ajouter l'attribut `enctype` au niveau de la balise `form`.

```
<form method="post" enctype="multipart/form-data" >
    <input type="file" name="image">
    <button name="envoyer">valider</button>
</form>
```