



Intégration de PHP dans HTML

PHP dans HTML c'est quoi ?

Les formulaires HTML sont une des bases de l'interaction utilisateur sur le web. Ils permettent à un visiteur de votre site de saisir des données et de les envoyer à un script sur le serveur, généralement pour être traitées ou enregistrées. En utilisant PHP, vous pouvez facilement récupérer ces données et les utiliser de nombreuses manières différentes.

Méthode POST

Voici comment créer un formulaire HTML simple en utilisant la méthode POST :

```
<form action="traitement.php" method="post">
  <label for="nom">Nom :</label><br>
  <input type="text" id="nom" name="nom"><br>
  <label for="email">Email :</label><br>
  <input type="email" id="email" name="email"><br><br>
  <input type="submit" value="Envoyer">
</form>
```

Pour traiter ces données en PHP, vous pouvez utiliser la superglobal `$_POST`, qui est un tableau associatif contenant les valeurs envoyées par le formulaire. Par exemple, pour récupérer la valeur du champ "nom" dans le script `traitement.php`, vous pouvez utiliser la syntaxe suivante :

```
$nom = $_POST['nom'];
```

Méthode GET

Il existe également une autre méthode d'envoi de formulaire, appelée GET. Elle fonctionne de manière similaire, mais les données sont envoyées dans l'URL et peuvent être vues par l'utilisateur. Pour utiliser la méthode GET, vous devez simplement changer la valeur de l'attribut `method` dans votre formulaire HTML :

```
<form action="traitement.php" method="get">
  <label for="nom">Nom :</label><br>
  <input type="text" id="nom" name="nom"><br>
  <label for="email">Email :</label><br>
  <input type="email" id="email" name="email"><br><br>
  <input type="submit" value="Envoyer">
</form>
```

Méthode GET

Pour récupérer les données envoyées en utilisant la méthode GET, vous pouvez utiliser la superglobal `$_GET` de la même manière que vous utiliseriez `$_POST`.

Introduction à l'intégration de PHP et HTML

Il est fréquent de vouloir intégrer du code PHP dans du code HTML et vice versa, afin de créer des pages web dynamiques.

```
<h1>Bienvenue sur mon site !</h1>
```

```
<?php
```

```
echo "Il est " . date("H:i:s") . " et nous sommes le " . date("d/m/Y") . ".";
```

```
?>
```

```
<p>Voici un nombre aléatoire : <?php echo rand(); ?></p>
```

Vous pouvez également intégrer du code HTML dans du code PHP en utilisant la fonction echo :

```
<?php
```

```
echo '</ul>';
```

```
echo "<p>Merci de votre visite !</p>";
```

```
?>
```

Exemple PHP dans HTML

Génération de tableaux HTML à partir de données en PHP

```
<?php
$donnees = [
    ['nom' => 'John', 'age' => 30],
    ['nom' => 'Jane', 'age' => 25],
    ['nom' => 'Bob', 'age' => 35],
];

echo '<table>';
echo '<tr>';
echo '<th>Nom</th>';
echo '<th>Age</th>';
echo '</tr>';

foreach ($donnees as $ligne) {
    echo '<tr>';
    echo "&<td>{$ligne['nom']}</td>";
    echo "<td>{$ligne['age']}</td>";
    echo '</tr>';
}

echo '</table>';
```

Introduction aux expressions régulières en PHP

Les expressions régulières sont des chaînes de caractères qui permettent de vérifier si une autre chaîne de caractères correspond à un certain format. Elles sont très utiles pour vérifier si des données entrées par un utilisateur sont valides, comme une adresse email ou un numéro de téléphone.

Qu'est-ce qu'une expression régulière et comment l'utiliser en PHP ?

Une expression régulière est une chaîne de caractères qui décrit un motif à rechercher dans une autre chaîne de caractères.

Voici comment utiliser une expression régulière en PHP :

```
$expression = '/^[a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\.[a-zA-Z]{2,4}$/';  
$email = 'john@example.com';  
  
if (preg_match($expression, $email)) {  
    echo "L'adresse email est valide.";  
} else {  
    echo "L'adresse email n'est pas valide.";  
}
```

Exercice

Réaliser grâce à une expression régulière en PHP une vérification du format d'un numéro de téléphone

Correction

```
$expression = '/^[0-9]{10}$/';  
$telephone = '0123456789';  
  
if (preg_match($expression, $telephone)) {  
    echo "Le numéro de téléphone est valide.";  
} else {  
    echo "Le numéro de téléphone n'est pas valide.";  
}
```