
Les formulaires en PHP



Méthodes d'envoi d'un formulaire

En PHP, les méthodes GET et POST sont deux moyens différents d'envoyer des données d'un formulaire HTML vers un script PHP pour être traitées.

- GET : envoie les données du formulaire en ajoutant les paramètres directement à l'URL. Les données sont visibles dans l'URL -> non protégées
- POST : envoie les données du formulaire de manière plus sécurisée, car elles ne sont pas visibles dans l'URL

Méthode GET

1. URL de base

<http://www.example.com/example.php>

2. Ajout des paramètres

Les paramètres sont ajoutés à l'URL après un point d'interrogation "?".

Chaque paramètre est composé d'une clé et d'une valeur, séparées par un signe égal "=".

Par exemple, si vous voulez ajouter un paramètre nommé nom avec la valeur john, l'URL deviendra :

<http://www.example.com/example.php?nom=john>

<http://127.0.0.1/maPage.php?nom=ROCHER&email=exemple@contact.fr&telephone=0567453423&message=Bonjour>

Remarque : on peut ajouter plusieurs paramètres à la suite

Méthode GET

3. Récupération des paramètres en GET

// Vérifiez d'abord si les paramètres existent

```
if (isset($_GET['user']) && isset($_GET['age'])) {  
  
    $nom = $_GET['nom']; // Récupère le paramètre 'nom'  
    $age = $_GET['age']; // Récupère le paramètre 'age'  
  
    echo "Nom d'utilisateur: " . $nom . "<br>";  
    echo "Âge: " . $age;  
  
}  
else {  
  
    echo "Les paramètres nécessaires ne sont pas fournis."  
  
}
```

Méthode GET

Exemple

1. Création d'un formulaire en HTML avec une méthode GET :

```
<form action="formulaireConfirmation.php" method="get">
  <div>
    <label for="nom">Nom</label>
    <input type="text" name="nom" id="nom" required>
  </div>
  <div>
    <label for="telephone">Téléphone</label>
    <input type="tel" name="telephone" id="telephone" required>
  </div>
  <div>
    <label for="email">E-mail</label>
    <input type="email" name="email" id="email" required>
  </div>
  <div>
    <label for="message">Message</label>
    <textarea name="message" id="message"></textarea>
  </div>
  <div>
    <input type="submit" value="Envoyer le message">
  </div>
</form>
```



Méthode GET

Exemple

2. Soumission d'un formulaire

Lorsque ce formulaire est soumis, les données seront disponibles dans le script PHP "formulaireConfirmation.php" via la superglobale `$_GET`. Vous pouvez les récupérer comme ceci :

```
Nom : <?php echo $_GET['nom']; ?><br />  
E-mail: <?php echo $_GET['email']; ?><br />  
Téléphone : <?php echo $_GET['telephone']; ?><br />  
Message : <?php echo $_GET['message']; ?><br />
```

Méthode POST

Pas de paramètres dans l'URL

1. Contrairement à GET, qui envoie des données via l'URL, POST transmet les données dans le corps de la requête HTTP.

Cela signifie que les données ne sont pas visibles dans l'URL, ce qui est plus sécurisé pour transmettre des informations sensibles comme les mots de passe.

2. Pas de Limitation de Taille : POST n'a pas de limitation de taille pour la quantité de données que vous pouvez envoyer

3. Utilisation en HTML : Dans un formulaire HTML, vous spécifiez la méthode POST en utilisant l'attribut `method="post"` dans la balise `<form>`. Par exemple :

```
<form action="submit.php" method="post">  
    <!-- éléments du formulaire ici -->  
</form>
```

4. Récupération des données en POST

1. Création d'un formulaire en HTML avec une méthode POST :

Méthode POST

Exemple

```
<form action="formulaireConfirmation.php" method="post">
  <p>
    <!-- recupere le Nom grace au input -->
    Nom :<input id="nom" name="nom" type="text" />
    <!-- recupere le Prénom grace au input -->
    Prénom : <input id="prenom" name="prenom" type="text" />
    <br>
    <!-- permet d'effacer les formulaires -->
    <input type="reset" value="Effacer" />
    <!-- permet d'envoyer les informations du formulaire au
    reponse.php-->
    <input type="submit" value="Envoyer" />
  </p>
</form>
```


Méthode POST

Exemple

2. Récupération des paramètres en POST

Lorsque ce formulaire est soumis, les données seront disponibles dans le script PHP "formulaireConfirmation.php" via la superglobale `$_POST`. Vous pouvez les récupérer comme ceci :

```
<p>Dans le formulaire précédent, vous avez fourni les  
    informations suivantes :  
</p>  
<?php  
    echo 'Prénom : ' . $_POST["prenom"] . '<br>';  
    echo 'Email : ' . $_POST["mail"] . '<br>';  
?>
```



Méthodes :

`$_POST`

`$_GET`

`$_REQUEST`

- le tableau associatif **`$_POST`** contient les données reçues via le corps de la requête HTTP,
- le tableau associatif **`$_GET`** contient les données reçues via l'url,
- le tableau associatif **`$_REQUEST`** contient l'ensemble des données reçues en GET et en POST.

Remarque : en cas de conflit de nom entre une donnée reçue en POST et en GET, seule la donnée reçue en POST est conservée.



Soumission

= Bouton type Submit

```
<!-- permet d'envoyer les informations du formulaire au  
reponse.php-->  
<input type="submit" value="Envoyer" />
```

Détection d'une soumission

```
if (isset($_POST['nom'])){\n    //le formulaire est soumis\n}\nelse {\n    //le formulaire n'a pas été soumis\n}
```



Types de champs

Type d'input	Définition
text	Définit un champ de saisie monoligne qui accepte du texte
number	Définit un champ qui accepte un nombre décimal
email	Crée un champ qui permet de renseigner une adresse mail
date	Permet à l'utilisateur d'envoyer une date
password	Créer un champ de saisie monoligne dont la valeur va être cachée
checkbox	Permet de créer une case à cocher. L'utilisateur peut cocher une ou plusieurs cases d'un coup
radio	Permet de créer un bouton radio. Par définition, un seul bouton radio peut être coché dans un ensemble
url	Crée un champ qui accepte une URL
file	Permet à un utilisateur de télécharger un fichier
submit	Crée un bouton d'envoi des données du formulaire

Type "Text"

= texte court

- Création HTML :

```
<label for="nomUtilisateur">Nom d'utilisateur :</label>
<input type="text" id="nomUtilisateur" name="nomUtilisateur">
```

- Récupération des données en POST :

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nomUtilisateur = $_POST['nomUtilisateur'];
}
```

Remarque : `$_SERVER["REQUEST_METHOD"]` => sert à vérifier la méthode de récupération utilisée (GET ou POST).

Textarea

= zone de texte

- Création HTML :

```
<textarea id="message" name="message" rows="4" cols="50">
Votre message ici...
</textarea>
```

- Récupération des données en POST :

```
if (!empty($_POST['message'])) {
    $message = $_POST['message'];
    // Traitez le message ici
    echo "Message reçu : " . htmlspecialchars($message);
}
else {
    echo "Aucun message reçu.";
}
```

Remarque : Fonction php htmlspecialchars() => Convertit les caractères spéciaux en entités HTML

Type "Radio"

= radio bouton

- Création HTML :

```
<label><input type="radio" name="genre" value="homme"> Homme</label>  
<label><input type="radio" name="genre" value="femme"> Femme</label>
```

- Récupération des données en POST :

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


Type "Checkbox"

= cases à cocher

- Création HTML :

```
<div>
  <input type="checkbox" id="sport" name="sport"
value="sport">
  <label for="sport">Sport</label>
</div>
<div>
  <input type="checkbox" id="lecture" name="lecture"
value="lecture">
  <label for="lecture">Lecture</label>
</div>
```

- Récupération des données en POST :

```
if (isset($_POST['sport'])) {
    echo htmlspecialchars($_POST['sport']) . "<br>";
}

if (isset($_POST['lecture'])) {
    echo htmlspecialchars($_POST['lecture']) . "<br>";
}
```

Remarque : Chaque `if` vérifie si une case à cocher spécifique a été cochée avec `isset()`.

Type "File"

= fichier

Différentes informations disponibles sur le fichier transmis :

`$_FILES["fichier"]["name"]` => nom et adresse originels du fichier sur le disque de l'utilisateur

`$_FILES["fichier"]["type"]` => type mime du fichier

`$_FILES["fichier"]["size"]` => taille du fichier en octets

`$_FILES["fichier"]["tmp_name"]` => nom et adresse du fichier temporaire stocké sur le serveur

`$_FILES["fichier"]["error"]` => code erreur associé au téléchargement

Type "File"

= fichier

- Création HTML :

```
<form action="envoi.php" methode="POST" enctype="multipart/form-data"
>
    <div>
        <input type="file" name="fichier" size="40">
        <input type="submit" value="Envoyer">
    </div>
</form>
```

L'attribut `enctype="multipart/form-data"` est nécessaire dans la balise `<form>` pour l'envoi de fichiers.

- Récupération des données en POST grâce à la superglobale `$_FILES[]` :

```
if (isset($_FILES["fichier"]) && $_FILES["fichier"]["error"] == 0) {
    $fichier = $_FILES["fichier"];

    // Déplacer le fichier téléchargé vers un nouvel emplacement
    $destination = "uploads/" . $fichier["name"];
    if (move_uploaded_file($fichier["tmp_name"], $destination)) {
        echo "Le fichier a été téléchargé avec succès.";
    }
    else {
        echo "Erreur lors du téléchargement du fichier.";
    }
}
```



Fonctions PHP utiles lors de la validation des données

- `move_uploaded_file()` => déplace un fichier téléchargé à une nouvelle localisation
- `htmlspecialchars()` => Convertit les caractères spéciaux en entités HTML
- `htmlentities()` => Convertit tous les caractères éligibles en entités HTML
- `strip_tags()` => Supprime les balises HTML et PHP d'une chaîne

```
$nom = htmlentities($_POST['nom']);
```

```
$texte_sans_tag_html = strip_tags($_POST['texthtml']);
```

Fonctions PHP utiles lors de la validation des données

- `intval()` => Retourne la valeur numérique entière

```
$nombre = intval($_POST['nombre']);
```

- `floatval()` => Convertit une chaîne en nombre à virgule flottante

```
$taux = floatval($_POST['taux']);
```

- `filter_var()` => Filtre une variable avec un filtre spécifique

Exemple avec un champ de type “password” :

```
// Récupération du mot de passe depuis $_POST
```

```
$passwordInitial = $_POST['password'];
```

```
// Nettoyage du mot de passe
```

```
$passwordClean = filter_var($passwordInitial, FILTER_SANITIZE_STRING);
```

Remarque : `FILTER_SANITIZE_STRING` est utilisé pour supprimer les balises et encoder ou supprimer les caractères spéciaux.

- `empty()` => Détermine si une variable est vide