

PHP : Les formulaires

PHP permet de récupérer les données envoyées via un formulaire. Cet article couvre les bases du travail avec les formulaires en PHP.

Objectifs

- Comprendre comment fonctionne un formulaire HTML.
- Savoir quand utiliser **GET** ou **POST**.
- Utiliser les superglobales `$_GET` et `$_POST` pour récupérer les données dans PHP.
- Respecter les bonnes pratiques du protocole HTTP.

Structure d'un formulaire HTML

```
1 <form action="traitement.php" method="get">
2   <label>Votre nom :</label>
3   <input type="text" name="nom">
4   <input type="submit" value="Envoyer">
5 </form>
```

- **action** : page qui traitera les données (ex. : `traitement.php`)
- **method** : choix entre **GET** (lecture) ou **POST** (modification)

Différences entre GET et POST

| Critère | GET | POST |
|-----------------|--|-----------------------------------|
| Visibilité | Les données sont visibles dans l'URL | Les données sont invisibles |
| Taille | Limitée à l'URL (~2000 caractères) | Illimitée (selon serveur) |
| Sécurité | Moins sécurisé | Plus adapté aux données sensibles |
| Usage | Lire, filtrer, rechercher | Envoyer, modifier, supprimer |
| Effets attendus | Pas de modification des données | Peut modifier les données |

 **Rappel important** : une requête **GET** ne doit **jamais** modifier des données côté serveur. Elle est destinée à **récupérer** des informations.

Exemple avec GET (récupération)

HTML :

```
1 <form action="recherche.php" method="get">
2   <input type="text" name="q" placeholder="Recherche...">
```

```
3 | <input type="submit" value="Rechercher">
4 | </form>
```

PHP (recherche.php) :

```
<?php
1 | $terme = $_GET['q'];
2 | echo "Vous avez recherché : $terme";
?>
```

● Données visibles dans l'URL :
recherche.php?q=PHP



Exemple avec POST (envoi)

HTML :

```
1 | <form action="contact.php" method="post">
2 |   <input type="text" name="nom">
3 |   <textarea name="message"></textarea>
4 |   <input type="submit" value="Envoyer">
5 | </form>
```

PHP (contact.php) :

```
<?php
1 | $nom = $_POST['nom'];
2 | $message = $_POST['message'];
3 |
4 | echo "Merci $nom pour votre message :<br>";
5 | echo nl2br(htmlspecialchars($message));
?>
```

🔒 Les données sont transmises discrètement, sans apparaître dans l'URL.



Quand utiliser quoi ?

- **GET :**
 - Affichage d'un article par ID.
 - Filtres de recherche ou tri.
 - Lien partageable (ex. : `page.php?id=12`).
- **POST :**
 - Connexion.
 - Inscription.
 - Envoi de commentaires, formulaires, etc.



Exercices (à faire en local)

1. Crée un formulaire de recherche (`GET`) qui affiche le mot recherché.
2. Crée un formulaire de contact (`POST`) qui affiche le nom et le message.
3. Teste avec des caractères spéciaux (`<`, `>`, `&`) et protège les données avec `htmlspecialchars()`.