



COURS

Formulaire simple

```
<form action="" method="get">
  <label for="rue">Rue</label>
  <input type="text" name="rue" id="rue">
  <label for="code-postal">Code Postal</label>
  <input type="text" name="code-postal" id="code-postal">
  <label for="ville">Ville</label>
  <input type="text" name="ville" id="ville">
  <input type="submit" value="Envoyer">
</form>
```

form.php



Rue

Code Postal

Ville

Envoyer

Bouton de **soumission**
du formulaire



Quelques explications

Script PHP qui va **traiter** le formulaire lors de la **soumission**

Méthode HTTP utilisée lors de la **soumission**

```
<form action="" method="get">  
...  
</form>
```



```
<label for="rue">Rue</label>  
<input type="text" name="rue" id="rue">
```

Nom du champs dans le script PHP lors de la **soumission**

```
<form action="" method="get">  
...  
</form>
```



Méthode **GET**



```
<form action="" method="post">  
...  
</form>
```



Méthode **POST**





A la **soumission** du formulaire, le **script** déclaré dans l'attribut **action** sera **exécuté**

```
<form action="" method="post">  
...  
</form>
```



Ici c'est **form.php**



A la **soumission** du formulaire, le **script form.php** sera appelé avec la **méthode HTTP POST**



Tester si le script **form.php** est appelé avec la **méthode HTTP POST**

```
if ($_SERVER["REQUEST_METHOD"] === "POST") {  
  
}
```



Récupérer les valeurs saisies dans le formulaire

```
if ($_SERVER["REQUEST_METHOD"] === "POST") {  
    $rue = $_POST["rue"];  
    $codePostal = $_POST["code-postal"];  
    $ville = $_POST["ville"];  
}
```

\$_POST

```
<form action="" method="get">  
    <label for="rue">Rue</label>  
    <input type="text" name="rue" id="rue">  
    <label for="code-postal">Code Postal</label>  
    <input type="text" name="code-postal" id="code-postal">  
    <label for="ville">Ville</label>  
    <input type="text" name="ville" id="ville">  
    <input type="submit" value="Envoyer">  
</form>
```




Réaliser un traitement avec les valeurs saisies dans le formulaire

```
if ($_SERVER["REQUEST_METHOD"] === "POST") {  
    $rue = $_POST["rue"];  
    $codePostal = $_POST["code-postal"];  
    $ville = $_POST["ville"];  
  
    // Exemple de traitement des données  
    echo "Insertion $codePostal $codePostal $ville dans la base de données";  
}
```



Valider les données saisies par l'utilisateur

- ▶ Certains **champs** sont **obligatoires**
- ▶ Certains **champs** doivent **respecter** certaines **contraintes**





Valider les données saisies par l'utilisateur

► **Valider côté client**



► **Valider côté serveur**





Valider côté client



```
<label for="rue">Rue</label>  
<input type="text" name="rue" id="rue" required>
```

Le champs est **obligatoire**

Rue

Code P

Ville

Envoyer



! Veuillez renseigner ce champ.



Valider côté client



```
<label for="rue">Rue</label>
```

```
<input type="text" name="rue" id="rue" required>
```



**Facilement
désactivable !**



Valider côté serveur



► A faire **OBLIGATOIREMENT !**

► Démarche dans le code



Tester les **champs** nécessitant une **validation**



Afficher pour chaque **champs** les **erreurs de validation**



Permettre à l'utilisateur de ne pas avoir à ressaisir les **données valides** en cas **d'erreur**



Validation côté serveur





Rediriger en cas de **succès**



NE JAMAIS rester sur la page affichant le formulaire en cas de succès



Rediriger l'utilisateur vers une **autre page**

```
header("Location: ...");
```

Nom du script vers lequel **rediriger** l'utilisateur

Upload de fichiers



Envoyer un fichier vers le serveur afin de le stocker



Utiliser un formulaire



Choisir le fichier sur le poste client



A la soumission du formulaire, le fichier est uploadé (envoyé) sur le serveur afin d'y être stocké



COMMENT stocker le fichier sur le serveur ?



2 manières de faire



Stocker le fichier dans la base de données



Stocker le fichier dans le système de fichiers (dans un dossier sur le serveur)



Stocker le fichier dans le **système de fichiers** (dans un **dossier** sur le **serveur**)

NOM

▶ **Enregistrer le contenu du fichier** dans le **système de fichiers** (dans un **dossier** sur le **serveur**)

▶ **Sauvegarder le nom du fichier** dans la **base de données**



Sélectionner le fichier à uploader côté client

```
<form action="" method="post" enctype="multipart/form-data">  
...  
<label for="photo">Photo*</label>  
<input type="file" name="photo" id="photo">  
  
<input type="submit" value="Envoyer">  
</form>
```

Rue*

Code Postal*

Ville*

Photo*

Choisir un fichier Aucun fichier choisi

Envoyer



Récupérer le fichier uploadé côté serveur



Utiliser le tableau associatif `$_FILES` permettant de récupérer les fichiers uploadés avec la méthode POST

```
// Récupération du fichier  
print_r($_FILES);
```

**TRY
THIS!**

STRUCTURE

`$_FILES['userfile']['name']`

Le nom original du fichier, tel que sur la machine du client web.

`$_FILES['userfile']['type']`

Le type MIME du fichier, si le navigateur a fourni cette information. Par exemple, cela pourra être "image/gif". Ce type mime n'est cependant pas vérifié du côté de PHP et, donc, ne prend pas sa valeur pour se synchroniser.

`$_FILES['userfile']['size']`

La taille, en octets, du fichier téléchargé.

`$_FILES['userfile']['tmp_name']`

Le nom temporaire du fichier qui sera chargé sur la machine serveur.

`$_FILES['userfile']['error']`

Le code d'erreur associé au téléchargement de fichier.



Récupérer les infos liées au fichier uploadé

```
$nomFichier = $_FILES["photo"]['name'];  
$typeFichier = $_FILES["photo"]['type'];  
$tmpFichier = $_FILES["photo"]['tmp_name'];  
$tailleFichier = $_FILES["photo"]['size'];
```



Enregistrer le contenu du fichier dans le système de fichiers

```
// Test si le fichier (la photo) a été soumis
if (empty($nomFichier)) {
    // Test si le fichier est une image
    if (str_contains($typeFichier, "image")) {
        // Déplacement du fichier dans un dossier sur le serveur
        move_uploaded_file($tmpFichier, "../images/$nomFichier");
    } else {
        $erreurs['photo'] = "Le fichier n'est pas une image";
    }
} else {
    $erreurs['photo'] = "La photo est obligatoire";
}
```




Sauvegarder le nom du fichier dans la base de données

```
$nomFichier = $_FILES["photo"]['name'];
```

Contient le **nom** du **fichier** à **sauvegarder**



Il suffit d'**enregistrer** cette **information** avec les autres données saisies dans le formulaire (**SQL INSERT**)



Amélioration n°1

Côté client



Définir les types de fichier qui devraient être **acceptés**

```
<label for="photo">Photo*</label>  
<input type="file" name="photo" id="photo" accept="image/png, image/jpeg, image/jpg">
```



Types **MIME**





Amélioration n°2

Côté serveur



Définir une taille MAX de fichier



En octets

```
if (str_contains($typeFichier, "image")) {  
    // test si la taille du fichier est supérieure à 1Mo  
    if ($tailleFichier > 1000000) {  
        $erreurs['photo'] = "La taille du fichier est trop importante";  
    } else {  
        // Déplacement du fichier  
        move_uploaded_file($tmpFichier, "../images/$nomFichier");  
    }  
} else {  
    $erreurs['photo'] = "Seules les images sont acceptées";  
}
```



Valeur codée
en dur





Amélioration n°3



Générer un nom de fichier unique



```
$nomFichier = $_FILES["photo"]['name'];
```



► Contient le **nom** du **fichier** à **sauvegarder**

► 2 utilisateurs peuvent **uploader** un **fichier** portant le **même nom**



Amélioration n° 4



Autoriser UNIQUEMENT certaines extensions de fichiers

```
if (str_contains($typeFichier, "image")) {
    // ...
}
```



Test si le fichier est une image



Autoriser UNIQUEMENT les fichiers dont l'extension est png ou jpeg (jpg)