

Introduction à PHP

Formulaires HTML et PHP, interactions avec le client

monnerat@u-pec.fr

29 mars 2026

IUT de Fontainebleau

Formulaire et traitement

- Passage des paramètres depuis un formulaire

- Transmission de paramètres

- Encodage des données

- Validation des données

Éléments HTML de formulaire

- Texte

- Cases à cocher

- Listes

- Upload de fichiers

Formulaire et traitement

Formulaire

Permet l'envoi de données du navigateur (client) au serveur (l'application) par http(s).

Pourquoi faire ?

- Créer un espace sécurisé par authentification.
- Uploader des fichiers.
- Administrer le contenu d'un site.
- Payer ses impôts
- Gérer un panier d'articles.
- Etc.

En HTML, `<form>` et `</form>` déclare le début et la fin d'un formulaire.

- L'attribut `action` indique l'url destinatrice des données saisies après soumission.

```
<form action="TraiterFormulaire.php">
  <!--
  le formulaire
  -->
  <input type="submit" value="Envoyer">
</form>
```

- La soumission des données (l'envoi) est déclenchée grâce à un élément `input` ou `button` de type `submit` (par un click), ou l'appuie sur la touche "Entrée".

Formulaire et traitement

Passage des paramètres depuis un formulaire

RFC 9110

Lorsqu'un formulaire est envoyé à un script php du serveur web, toutes ses données se retrouvent dans des superglobales. Il s'agit de tableaux associatifs accessibles dans tout le script. On utilisera :

- `$_GET []` ~> contient toutes les données envoyées via l'URL (en GET)
- `$_POST []` ~> contient les données envoyées via un formulaire en POST
- `$_REQUEST []` ~> est la fusion des deux premiers.

Récupération des données

C'est la valeur de l'attribut **name** des éléments html du formulaire qui sert de clés dans les tableaux précédents.

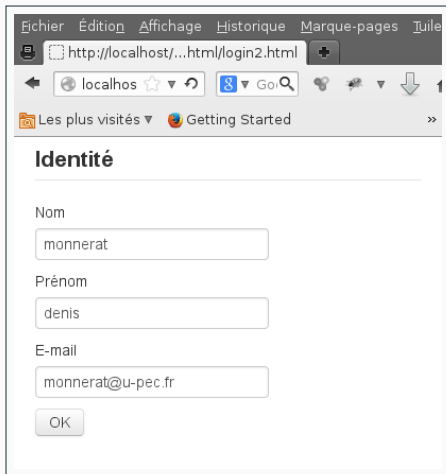
Le navigateur envoie une liste de paires (nom, valeur), que le serveur met à disposition du script php.

Méthode POST

La méthode post permet entre autres de transmettre des "valeurs" saisies par l'utilisateur à un script (php), dans la **corps** de la requête http depuis un formulaire html.

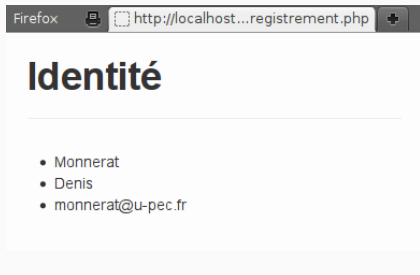
```
<form method="post" action="enregistrement.php">
  <legend><b>Identite</b></legend>
  <label >Nom :
    <input type="text" name="nom">
  </label>
  <label>Prenom
    <input type="text" name="prenom">
  </label>
  <label>Email :
    <input type="email" name="email">
  </label>
  <input type="submit" name="accepter" value="OK">
</form>
```


L'exemple affiche un formulaire simple dans le navigateur : trois champs de saisie de texte.



Lorsque l'utilisateur valide et envoie les données au serveur, les variables du formulaire sont passées au script php destination au moyen du tableau `_POST`.

```
// on recupere les valeurs  
// du formulaire  
$nom=$_POST['nom'];  
$prenom=$_POST['prenom'];  
$email=$_POST['email'];  
// on les traite  
echo "<li>$nom</li>";  
echo "<li>$prenom</li>";  
echo "<li>$email</li>";
```



Identité

- Monnerat
- Denis
- monnerat@u-pec.fr

Attention ! Il est important de "désinfecter" les données externes avant de les traiter et les afficher (ce qui n'est pas le cas dans cet exemple).

Remarque : la fonction `extract` permet de créer des variables à partir des clés d'un tableau associatif. chaque case du tableau génère une variable du nom de sa clé.

```
// on recupere les valeurs du formulaire  
extract($_POST);  
// on les traite  
echo "<li>$nom</li>";  
echo "<li>$prenom</li>";  
echo "<li>$email</li>";
```

Méthode GET

L'attribut `method` vaut `get` (valeur par défaut)

```
<form method="get" action="enregistrement.php">
  <fieldset>
    <legend><b>Identite</b></legend>
    <label>Nom :
      <input type="text" name="nom">
    </label>
    <!-- etc -->
  </fieldset>
  <input type="submit" name="accepter" value="OK">
</form>
```

Les données sont passées directement via l'URL (elles sont encodées).

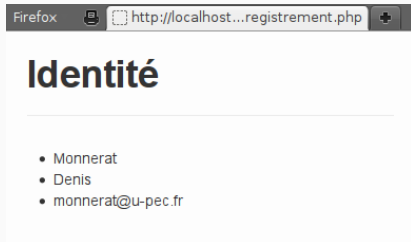
http://localhost/~denis/public_html/enregistrement.php?nom=Monnerat&prenom=Denis&email=monnerat%40u-pec.fr&accepter=OK

Méthode GET

enregistrement.php?nom=Monnerat&prenom=Denis&email=monnerat%40u-pec.fr&accepter=OK

Les variables se récupèrent à l'aide du tableau `_GET`

```
// on recupere les valeurs  
// du formulaire  
$nom=$_GET['nom'];  
$prenom=$_GET['prenom'];  
$email=$_GET['email'];  
// on les traite  
echo "<li>$nom</li>";  
echo "<li>$prenom</li>";  
echo "<li>$email</li>";
```



Formulaire et traitement

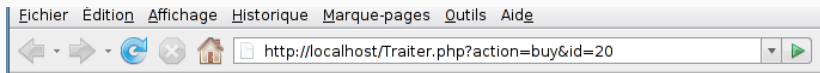
Transmission de paramètres

Passage de paramètres à un script

On peut passer des arguments à un script directement dans l'url, comme le fait l'envoi des données avec la méthode GET.

```
$id=3;  
echo "<a href='Traiter.php?action=buy&id=$id'>Acheter</a>";
```

ce qui donnera l'URL :



La syntaxe pour passer des arguments à une page est la suivante

```
mon_url_ami?var1=valeur1&var2=valeur2&var3=valeur3 ...
```

Les variables transmises au script sont disponibles dans le tableau `_GET`.

Exemple

```
if (isset($_GET['rubrique']))
    $rubrique = $_GET['rubrique'];
else
    $rubrique = "";
// $rubrique = $_GET['rubrique'] ?? ''; (php7)
switch($rubrique){
case "web" :
    include('web.php');
    break;
case "dev" :
    include('dev.php');
    break;
case "maths" :
    include('mathematique.php');
    break;
default :
    include ('accueil.php');
}
```

Exemple avec Google search

The image shows a browser window with a Google search for "IUT de Fontainebleau". The search results are displayed on the left side of the page, while a detailed knowledge panel for "IUT Sénart-Fontainebleau" is on the right. The browser's address bar shows the URL: https://www.google.fr/?gl=fr&rlz=1c1=mmMDV8F2D6ex8arFV1K_IBw5gws_rd=ssl#q=iut+de+fontainebleau. The search results include several links to the IUT's website, such as "IUT Sénart-Fontainebleau - Bienvenue à l'IUT" and "I.U.T. Fontainebleau - Ville de Fontainebleau". The knowledge panel on the right provides information about the institution, including its address (Route forestière Hurtaut, 77300 Fontainebleau), telephone number (01 60 74 68 00), and a map. It also features a "Recherches associées" section with links to other universities like "IUT d'Univ... de Mame-la..." and "Université Paris-Est Mame-la...".

Environ 206 000 résultats (0,48 secondes)

IUT Sénart-Fontainebleau - Bienvenue à l'IUT
www.iut5.fr
Bienvenue à l'IUT de Sénart Fontainebleau - découvrez nos IUT et nos licences professionnelles !
Vous avez consulté cette page de nombreuses fois. Date de la dernière visite : 05/03/16

Nos formations
Toute l'offre de formation de l'IUT Sénart Fontainebleau : DUT ...

Informatique
Informatique - Sénart Fontainebleau.
Version pdf ...

Contacts et organigramme
Contacts et organigramme. Version pdf, Imprimer la page ...

Carrières sociales
Le département Carrières sociales forme des armateurs ...

GEA
Gestion des entreprises et des administrations - Sénart ...

Autres résultats sur www.pec.fr

Frédéric GERVAIS | Maître de conférences, Département ...
fredericgervais.com/
Il y a 1 jour - Maître de conférences, Département Informatique, IUT Sénart Fontainebleau (UPEC)
Vous avez consulté cette page de nombreuses fois. Date de la dernière visite : 10/02/16

I.U.T. Fontainebleau - Ville de Fontainebleau
www.fontainebleau.fr/?iut-de-fontainebleau-deux
L'IUT Sénart Fontainebleau, composante de l'UPEC est installée sur 2 sites, l'un en plein cœur de la forêt de Fontainebleau, l'autre à proximité du centre de la ...

IUT de Sénart/Fontainebleau: Département Informatique
www.iut-fbleau.fr/
IUT de Sénart Fontainebleau. Bienvenue sur le site interne du département informatique. Retenue 2015 - Portraits d'anciens étudiants - Fiche de ...

Images correspondant à IUT de Fontainebleau
Signaler des images inappropriées

IUT Sénart-Fontainebleau
Site Web

Institut universitaire de technologie

Adresse : Route forestière Hurtaut, 77300 Fontainebleau
Téléphone : 01 60 74 68 00
[Suggérer une modification](#)

Avis
1 avis de Google

Recherches associées Voir d'autres éléments (plus de 15)

IUT/Univ... de Mame-la... Institut universitaire de technologie
Institut Université de Techn... Institut universitaire de technologie
Université Paris-Est Mame-la... Université
IUT de Crétel-VL... Institut universitaire de technologie
I.U.T. Paris Descartes Institut universitaire de technologie

Commentaires

https://www.google.fr/?gl=fr&rlz=1c1=mmMDV8F2D6ex8arFV1K_IBw5gws_rd=ssl#q=iut+de+fontainebleau < [3/4] Top

Exemple avec Google search

=cr&ei=rmMDV8f2D6ex8wfVIK_IBw&gws_rd=ssl#q=iut+de+fontainebleau

E Expert [W3](#) The W3C Markup Val...



Pages

Actualités

Vidéos

Plus ▾

Outils de recherche

(0,48 secondes)

Fontainebleau - Bienvenue à l'IUT

Fontainebleau - découvrez nos DUT et nos licences

page de nombreuses fois. Date de la dernière visite : 06/03/16

Informatique



Formulaire et traitement

Encodage des données

L'attribut `enctype` de la balise `form` contrôle l'encodage des données. Par défaut, les données issues d'un formulaire sont de "type" `application/x-www-form-urlencoded` (RFC RFC 1866). Il s'agit d'une seule chaîne de caractères de la forme

```
var1=valeur1&var2=valeur2
```

qui est url encodée :

- Les caractères non alphanumériques sont remplacés par `%xx`, (`xx` est le code ascii du caractère en hexadécimal)
- Les caractères `; / ? : @ & = + $,` sont réservés :
 - `?` : début de `QUERY_STRING`.
 - `&` : séparateur de champs.
 - `=` : séparation entre le champ et sa valeur.
- Les espaces sont remplacés par des `+`.
- Les caractères non ascii, par exemple `é`, et encodé en utf-8 `C3A9`, et url encodé `%C3%C9`.

URL	Statut	Domaine	Poids	Remot
POST enregistrer	200 OK	localhost	329 B	:::1]:80
<div style="display: flex; justify-content: space-between;"> En-têtes Post Réponse HTML </div>				
Paramètres		application/x-www-form-urlencoded		
accepter	OK			
email	monnerat@u-pec.fr			
nom	monnerat			
prenom	denis			
Source				
<pre>nom=monnerat&prenom=denis&email=monnerat%40u-pec.fr&accepter=OK</pre>				

Attention

Seules les données au format application/x-www-form-urlencoded sont disponibles dans `_GET` ou `_POST`.

Quelle méthode utiliser ?

Dans les spécifications du protocole HTTP(s), rfc 9110

<https://www.rfc-editor.org/rfc/rfc9110.html> ↗,

- GET est sûre et idempotente : la requête récupère une ressource (lecture) mais ne modifie pas la ressource. Une répétition laisse le serveur dans le "même état". Données dans l'url de la requête.

```
/search?q=chat
```

- POST n'est pas idempotente. La requête change l'état du serveur (création de ressources). Données dans le corps de la requête.

Pour résumer, GET doit être utilisée uniquement pour récupérer une ressource, POST pour en ajouter.

Évidemment, il s'agit de recommandations ...

1. Génération du premier formulaire.

1. Génération du premier formulaire.
2. Envoie des données.

1. Génération du premier formulaire.
2. Envoie des données.
3. Reception des données.

1. Génération du premier formulaire.
2. Envoie des données.
3. Reception des données.
4. Vérification de la validité des champs.

1. Génération du premier formulaire.
2. Envoie des données.
3. Reception des données.
4. Vérification de la validité des champs.
5. Si OK, on traite les données (écriture bd, etc.)

1. Génération du premier formulaire.
2. Envoie des données.
3. Reception des données.
4. Vérification de la validité des champs.
5. Si OK, on traite les données (écriture bd, etc.)
6. Sinon, on régénère le formulaire, en y insérant les données valides déjà saisies par l'utilisateur. (attribut `value`)

1. Génération du premier formulaire.
2. Envoie des données.
3. Reception des données.
4. Vérification de la validité des champs.
5. Si OK, on traite les données (écriture bd, etc.)
6. Sinon, on régénère le formulaire, en y insérant les données valides déjà saisies par l'utilisateur. (attribut `value`)

Règle d'or

Il ne faut jamais faire confiance aux données issues d'un formulaire. On reviendra de manière plus précise sur les bonnes pratiques en termes de sécurité.

Formulaire et traitement

Validation des données

```
mixed filter_var ( mixed $variable  
[, int $filter = FILTER_DEFAULT  
[, mixed $options ]] );
```

```
mixed filter_var_array ( array $data  
[, mixed $definition  
[, bool $add_empty = TRUE ]] );
```

↪ Validation et nettoyage d'une donnée.

↪ Utilisation de filtres prédéfinis.

On a des fonctions analogues qui travaillent directement sur des entrées "externes" (get, post, cookie, etc.) avec `filter_input` et `filter_input_array`.

Filtres de validation

FILTER_VALIDATE_

- BOOLEAN
- INT
- EMAIL
- FLOAT
- IP
- URL
- etc.

Filtres de nettoyage

FILTER_SANITIZE_

- NUMBER_INT
- NUMBER_FLOAT
- STRING
- URL
- SPECIAL_CHARS

```
$email = "john.doe@example.com";

// Remove all illegal characters from email
$email = filter_var($email, FILTER_SANITIZE_EMAIL);

// Validate e-mail
if (filter_var($email, FILTER_VALIDATE_EMAIL) === false) {
    echo("$email is a valid email address");
} else {
    echo("$email is not a valid email address");
}
```

```
<?php
// $_POST = ['pseudo' => 'Marc', 'passwd' => 'kjhksdnksn'];
filter_input(INPUT_POST, 'pseudo', FILTER_SANITIZE_STRING);
// Renvoie 'Marc'
filter_input(INPUT_POST, 'email', FILTER_VALIDATE_EMAIL);
// Non-trouvé, renvoie
// NULL sans provoquer de notice.
```

Sur tout un formulaire

```
<?php
$filter_def = [
    'login' => FILTER_SANITIZE_STRING,
    'passwd' => FILTER_UNSAFE_RAW,
    'passwd2' => FILTER_UNSAFE_RAW,
    'email' => [
        'filter' => FILTER_VALIDATE_EMAIL,
        'flags' => FILTER_NULL_ON_FAILURE
    ],
    'email2' => FILTER_UNSAFE_RAW,
    'dateDeNaissance' => [
        'filter' => FILTER_VALIDATE_REGEXP,
        'options' => [
            'regexp' => '#^\d{2}/\d{2}/\d{4}$#' // dd/MM/YYYY
        ],
        'flags' => FILTER_NULL_ON_FAILURE
    ]
];
```

```
<?php
$resultat = filter_input_array(INPUT_POST, $filter_def);

if (
    in_array(null, $resultat, true)
    || $resultat['email'] != $resultat['email2']
    || $resultat['passwd'] != $resultat['passwd2']
    ) {
    // Formulaire invalide
} else {
    // Formulaire OK
}
```

On peut créer ses propres filtres.

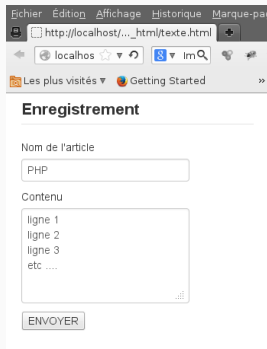
Éléments HTML de formulaire

Eléments HTML de formulaire

Texte

- Champ texte : créer avec la balise `<input>` et l'attribut `type text`.
- Zone de texte (zone de texte sur plusieurs lignes) : créer avec la balise `<textarea>`

```
<form method="post" action="enregistrement.php">
  <fieldset>
    <legend><b>Enregistrement</b></legend>
    <label>Titre</label>
    <input type="text" name="titre">
    <label>Contenu</label>
    <textarea rows="5" name="contenu"></textarea>
  </fieldset>
  <button type="submit" name="accepter">
    ENVOYER
  </button>
</form>
```



The screenshot shows a web browser window with the URL `http://localhost/..._html/texte.html`. The browser's address bar shows `localhost` and search engines like Bing and ImQ. Below the browser, there is a form titled "Enregistrement". The form has a legend "Enregistrement" and two main sections: "Nom de l'article" with a text input field containing "PHP", and "Contenu" with a multi-line text area containing "ligne 1", "ligne 2", "ligne 3", and "etc". At the bottom of the form is a button labeled "ENVOYER".


```
echo $_POST['titre']."<br />";  
echo nl2br($_POST['contenu']);
```

Notez bien la fonction `nl2br` qui permet de convertir les sauts de lignes de la zone de texte en balise `
` pour l'affichage HTML.

Remarques

- champ texte qui permet de saisir un mot de passe : Il s'agit toujours de la balise `input` avec l'attribut `password`
- champ texte caché Il s'agit toujours de la balise `input` avec l'attribut `hidden`

```
<input type="password" name="nom" value="">  
<input type="hidden" name="nom" value="">
```

HTML5 a rajouté de nouveaux type input

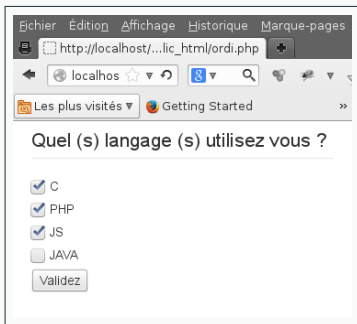
color	date	datetime
datetime-local	email	month
number	range	search
tel	time	url
	week	

Eléments HTML de formulaire

Cases à cocher

balise input avec l'attribut checkbox.

```
<form method="post" action="traitement.php">
  <legend> Quel (s) langage (s) utilisez vous ?</legend>
  <input type="checkbox" name="langages[]" value="C"> C
  <input type="checkbox" name="langages[]" value="PHP"> PHP
  <input type="checkbox" name="langages[]" value="JS"> JS
  <input type="checkbox" name="langages[]" value="JAVA"> JAVA
  <button type="submit">
    Validez
  </button>
</form>
```



```
$langages = $_POST['langages'];  
foreach ($langages as $unlangage)  
    echo ('La case ' . $unlangage . 'a ete cochee <br>');
```

balise input avec l'attribut radio : un seul choix possible.

```
<form action="reponse.php" methode=post>
  Qui etait Pythagore ?
  <input type="radio" name="question1" value=1>
  Un realisateur de film d'horreur.
  <input type="radio" name="question1" value=2>
  Un mathematicien grec.
  <input type="radio" name="question1" value=3>
  Un danseur mondain
  <input type=submit value=valider>
</form>
```

Il existe d'autres valeurs pour l'attribut type de la balise `input`

- Boutons
 - `reset` qui vide tous les champs d'un formulaire.
 - `submit` pour envoyer les données d'un formulaire.
 - `image` pour créer un bouton de validation graphique.

- Zone de données.
 - `file` pour soumettre un nom de fichier à uploader.
 - `hidden` pour envoyer une donnée cachée depuis un formulaire.

Eléments HTML de formulaire

Listes

Balise select

```
<form action="traiter.php" method=post>
  Quel est votre langage prefere ?
  <select name="langage">
    <option value=1>C</option>
    <option value=2>PHP</option>
    <option value=3>Bash</option>
    <option value=4>Assembleur Mips</option>
  </select>
  <input type=submit value=Valider>
</form>
```

Un seul choix possible. On peut néanmoins, en rajoutant `multiple` dans la balise, autorisé plusieurs choix.

Eléments HTML de formulaire

Upload de fichiers

```
<form enctype="multipart/form-data"
  action="fileupload.php"
  method="post">
  <input type="hidden" name="MAX_FILE_SIZE" value="100000">
  Transfere le fichier <input type="file" name="monfichier">
  <input type="submit">
</form>
```

Le tableau `_FILE` contient un tableau pour chaque fichier transféré (clé correspondant à l'attribut `name`) contenant :

- `tmp_name` : le nom (et chemin) du fichier temporaire sous lequel le fichier a été stocké.
- `name` : le nom qu'avait le fichier dans l'espace de l'utilisateur
- `size` : la taille du fichier en octets
- `type` : le "MIME type" du fichier (ex : `text/html`, `image/gif`, etc.)