PHP : Les formulaires

L2 MPCIE - UE Développement Web

David Lesaint

david.lesaint@univ-angers.fr

Janvier 2019

Rappel sur les formulaires HTML5

Les formulaires

- Base de l'interactivité des sites Web.
- Echanges avec serveur par le protocole HTTP en utilisant méthode `GET` ou, de préférence, `POST`.
  - Envoi de données.
  - Déclenchement d'une requête dans une BDD.
  - Création de page dynamique en réponse.

Composants de formulaires

- Saisie de texte et mots de passe.
- Choix par boutons radio, cases à cocher, listes de sélection.
- Envoi de fichiers ou données cachées.

Création de formulaire

Attributs de l'élément `<form>`

`action="f.php"`

- Désigne le fichier de traitement des données saisies.
- Chemin absolu ou relatif (de préférence), ou URL.
- `action="<?= $_SERVER["PHP_SELF"] ?>"` si le fichier de traitement contient le formulaire.
- Possibilité d'utiliser d'autres protocoles (`mailto:a@b.c`).

Création de formulaire

Attributs de l'élément `<form>`

`method="get"` ou `method="post"`

- Méthode HTTP d'envoi des données.
- HTTP GET par défaut.
  - Données en clair dans l'URI : le *query-string*

    `http://www.abc.fr/f.php?x=bleu&y=12`
    - Ajouté à l'URL avec `?`
    - Affectations de champs séparées par `&`
    - Caractères accentués, symboles de ponctuation . . . encodés en hexadécimal.
  - Limitée à ~2000 caractères.
- HTTP POST recommandée.
  - Encapsule les données dans le corps de la requête.
  - Données de tout type.

Création de formulaire

Attributs de l'élément `<form>`

`name="monformulaire"`

- Utile essentiellement pour JavaScript.

`enctype="type-encodage"`

- `"multipart/form-data"` pour le transfert de fichiers.
- `"application/x-www-form-urlencoded"` par défaut.

```
<form method="post" action="f.php"
enctype="application/x-www-form-urlencoded">
...
</form>
```

L'élément `<input/>`

L'attribut `type` permet de distinguer différents types de champs

- Texte en clair et mot de passe.
- Email et numéro de téléphone.
- Nombre et date.
- Choix unique et choix multiple.
- Soumission et réinitialisation.
- Fichier et données cachées.

L'attribut `name` est obligatoire

- Identifie chaque champ côté serveur.
- Ne pas confondre avec l'attribut `id` utilisé pour CSS et manipulation JS via le DOM.

Champ texte mono-ligne en clair

Attributs de l'élément `<input type="text"/>`

`size="nombre"`

- Largeur de la zone affichée en #caractères.

`maxlength="nombre"`

- Nombre maximum de caractères à saisir.

`value="texte"`

- Texte affiché et transmis par défaut.

```
<input type="text" name="ville" size="30"
maxlength="40" value="Angers"/>
```

Ergonomie : effacement au clic de la valeur par défaut via JS

```
<input ... onclick="this.value=”"/>
```

Autres champs texte

Adresse e-mail : `<input type="email"/>`
N° de téléphone : `<input type="tel">`

Validation par regex avec l'attribut `pattern`.

Mots de passe : `<input type="password"/>`

Caractères saisis affichés par des *

Nombres : `<input type="number"/>`

Encadrement forcé avec attributs `min` et `max`.
Obligation de saisie avec `required`.

Dates : `<input type="date"/>`

Champ texte au format `AAA-MM-JJ` avec interface selon navigateur.
Encadrement avec attributs `min` et `max`.
Variantes : `time` (heure), `month`, `week`, `datetime-local` (date et heure), `datetime` (date, heure et fuseau horaire).

Boutons radio

Attributs de l'élément `<input type="radio"/>`

`checked="checked"`

- Bouton coché par défaut.

`value`

- Indispensable comme pour champs texte.

Utiliser la même valeur d'attribut `name` entre boutons pour imposer un choix exclusif

```
<label>Homme</label><input type="radio"
name="genre" value="h"/>
<label>Femme</label><input type="radio"
name="genre" value="f"/>
```

Cases à cocher

Elément `<input type="checkbox"/>`

Utiliser la même valeur d'attribut `name` concaténée avec `[]` pour permettre un choix multiple :

```
<input type="checkbox" name="lang[]"
value="php"/>
<input type="checkbox" name="lang[]"
value="javascript"/>
```

Boutons de soumission et RAZ

Elément `<input type="submit"/>`

Indispensable d'avoir au moins un bouton de soumission.
Attribut `value`

- Le texte affiché du bouton.
- Permet d'effectuer différentes tâches de traitement selon la valeur du bouton pressé.

Elément `<input type="reset"/>`

Réinitialise les champs à leurs valeurs par défaut ($\neq$ effacement).

Fichiers et données masquées

Elément `<input type="file"/>`

Même apparence qu'un champ texte et incorpore un sélecteur de fichier sur le poste client.

Attribut `accept`

- Définit le ou les types de fichiers acceptés.

Attribut `multiple`

- Possibilité de sélectionner et téléverser plusieurs fichiers.

```
<input type="file" name="image" accept=".jpeg,
.jpg, .png" multiple>
```

Elément `<input type="hidden"/>`

Permet d'envoyer des données invisibles pour l'utilisateur (e.g. jeton de sécurité, identifiant de transaction, taille maximum de fichier téléversable).

Zones de texte multilignes

Attributs de l'élément `<textarea>`

`cols`

- Nombre de colonnes de la zone de texte.

`rows`

- Nombre de lignes de la zone de texte.

Liste déroulante

Elément `<select>`

Sélection simple ou multiple d'options définies chacune par un sous-élément `<option>`.

Attribut `size`

- Nombre d'options visibles simultanément.

Attribut `multiple="multiple"`

- Autorise la sélection multiple (avec touche Ctrl).

Attributs de l'élément `<option>`

`value`

- La valeur transmise si l'option est sélectionnée.

`selected="selected"`

- Option présélectionnée par défaut.

exemple6-1.html

Récupération des données de formulaire

Différents cas de figure :

- Les valeurs uniques.
- Les valeurs multiples.
- Les fichiers
- Les boutons d'envoi multiples.

Récupération de valeurs uniques

Contenues dans le tableau superglobal `$_POST` (ou `$_GET`)

exemple6-2.php

Vérification avec `isset()` sur champ texte et boutons radio.
Tous deux `NULL` initialement au chargement de la page.
Si champ non saisi et boutons non cochés à la soumission :

- La valeur du champ est la chaîne vide : vérification plus fine avec `empty()`.
- La valeur des boutons radio reste `NULL`.

Récupération de valeurs multiples

Exemples

La valeur de l'attribut `name` doit se terminer par `[]` :

- Cases à cocher de même attribut `name`.
- Liste de sélection multiple avec attribut `multiple`.

Récupération des valeurs via `$_POST` qui est alors un tableau bi-dimensionel.

exemple6-5.html, exemple6-5.php

Récupération de fichiers

`<input type="file" name="f".../>`

Le téléversement renomme le fichier et le place dans un répertoire tampon sur le serveur.
Récupération via tableau associatif `$_FILES["f"]`.
Eléments de `$_FILES["f"]` :

- `"name"` : le nom d'origine.
- `"type"` : le type du fichier.
- `"size"` : la taille du fichier.
- `"tmp_name"` : le nom du fichier sur le serveur.
- `"error"` : code d'erreur en cas d'échec du téléchargement.

`move_uploaded_file(string src, string dest)` permet de renommer et déplacer le fichier.

exemple6-6.php

Maintien de l'état du formulaire

Motivations

Ne pas avoir à faire ressaisir l'intégralité du formulaire en cas de saisie invalide.

- Si le script de traitement contient le formulaire, ce dernier sera réaffiché dans son état initial.

Approche 1 (exemple6-3.php)

Intégrer dans un champ texte la dernière valeur saisie :

```
...<?php if(isset($_POST["k"])) echo $_POST["k"] ?>...
```

Intégrer dans chaque bouton coché l'attribut `checked` :

```
...<?php if(isset($_POST["k"]) && $_POST["k"]=="v") echo
"checked =\"checked\"" ?>...
```

Maintien de l'état du formulaire

Approche 2 (exemple6-5.html, exemple6-5.php)

- Un fichier HTML contient le formulaire dont l'action pointe sur un fichier PHP.
- Le fichier PHP redirige vers le fichier HTML par JavaScript.

```
echo "<script>alert(...);window.history.back();</script>";
```

Approche 3 (form.html / post-redirect.php / post.php)

- Un fichier HTML contient le formulaire dont l'action pointe sur un fichier PHP.
- Le fichier PHP redirige vers un second fichier PHP avec la méthode `header` : redirection temporaire HTTP avec transfert des valeurs de `$_POST` (code 307).

```
header("Location: $url",true,307);
```