

IT 103 - Programmation Web  
TP4 - Formulaire - Javascript - PHP

## 1 Formulaire de demande de brochures

Vous allez créer un formulaire de demande de brochures pour des voitures.

1. Faites une page avec un formulaire permettant à un client de demander la brochure d'une voiture. Le formulaire contient un bouton "envoyer" pour soumettre la demande, et un bouton "reset" pour "remettre à zéro" (vide) toutes les cases du formulaire.
2. Les informations fournies doivent aussi inclure le genre (M., Mme, Mlle), un nom, un prénom, une adresse (rue, code postal et ville), un e-mail, un numéro de téléphone, une voiture (à partir d'une liste déroulante), un moyen de recevoir la brochure (postal et/ou email) et une zone de texte libre pour taper un commentaire éventuel.
3. Limitez le nombre maximum de caractères qui peuvent être tapés pour certains champs : 5 caractères pour le code postal, 10 caractères pour le numéro de téléphone, etc.
4. Tous les paramètres fortement contraints, c'est-à-dire qui ne peuvent avoir que peu de valeurs différentes possibles ne doivent pas être des champs avec des valeurs arbitraires mais des boutons radios, cases à cocher, listes de sélection, etc.
5. Pour tous les paramètres qui sont entrés comme des chaînes de caractères mais qui en fait sont des nombres (code postal, numéro de téléphone), l'entrée au clavier ne doit autoriser que les caractères qui sont des chiffres et interdire tous les autres caractères. Utilisez pour ça un bout de code JavaScript attaché à l'événement `onkeypress` pour l'entrée du champ, par exemple avec le modèle de code suivant :

Pour Internet Explorer :

```
<input ... onkeypress="
  if (event.keyCode ...) // si le code (ascii) de la touche enfoncée ...
    event.returnValue = false; // ignorer cet événement">
```

Pour Mozilla (Gecko) :

```
<input ... onkeypress="
  if (event.which ...) // si le code (ascii) de la touche enfoncée ...
    return false; // ignorer cet événement">
```

6. Certains paramètres du formulaire sont entrés comme des chaînes de caractères non contraintes et sont donc de valeur arbitraire. Pourtant, ils ne sont pas tous quelconques : 5 chiffres exactement pour le code postal, 10 chiffres exactement pour le numéro de téléphone, présence de @ dans l'adresse e-mail, etc. Ces paramètres doivent être vérifiés avec du code JavaScript avant d'être envoyés sur le serveur : si un des paramètres est incorrect, une fenêtre d'alerte JavaScript doit être poppée pour indiquer la nature du problème. Les données ne sont alors pas soumises. Vous pouvez pour cela utiliser le pattern de code suivant :

```
<script>
function validate(form)
  if (form.nomDeParamètre.value >= ...)
  ...
```

```

        return false; // Données invalides, on le les envoie pas
    else
        return true; // Validation OK, on peut envoyer les données

</script>
<form ... onsubmit="return valide(this);">

```

Notez que certains paramètres sont optionnels et peuvent être laissés vides (numéro de téléphone et zone de commentaire). Tous les autres paramètres doivent être non vides. Le fait de ne pas soumettre les données (réponse `false` à `onsubmit`) quand elles sont incorrectes (vides ou de format invalide) permet à l'utilisateur de les corriger sans avoir à les retaper et évite une communication inutile avec le serveur.

7. Si l'utilisateur clique sur "valider", une nouvelle page s'affiche qui dit "Votre demande a été transmise." et qui propose de revenir à la page d'accueil. Nous traiterons les données du formulaire dans les prochains TPs. Vous pouvez vous inspirer du formulaire situé à l'adresse : <http://www.enseirb.fr/~veille/Enseirb/IT103/tp/ressources/formulaire.pdf>

## 2 Récapitulatif

1. Lorsque tous les renseignements sont corrects, ils doivent être rassemblés et mis en forme (Utilisez CSS) dans une page générée automatiquement et où figure trois boutons : "annuler" (revenir à la page d'accueil), "modifier" (revenir au formulaire, sans perte des données déjà fournies), "valider" (envoi de la demande).
2. Si l'utilisateur clique sur le bouton "annuler", faites popper une fenêtre de confirmation JavaScript demandant à l'utilisateur s'il veut vraiment abandonner la demande. Si l'utilisateur confirme, vous devez revenir à la page d'accueil. Si l'utilisateur ne confirme pas, vous devez revenir à la page qui rassemble les renseignements. Si l'utilisateur clique sur "modifier", le formulaire lui est à nouveau proposé avec les champs déjà préremplis, les boutons déjà appuyés, tels qu'il les avait déjà fournis.
3. Si l'utilisateur clique sur "valider", une nouvelle page s'affiche qui dit "Votre demande a été transmise." et qui propose de revenir à la page d'accueil.

## 3 Structuration du site à l'aide de CSS/PHP

1. En s'inspirant de l'exemple donné en cours, utilisez des styles CSS pour avoir un menu qui soit à gauche de la page principale et qui regroupe toutes les pages créées jusqu'à précédent (accueil, demande de brochure, fiche pratique...).
2. Utilisez la fonction PHP `include()` ; pour inclure chaque page du site dans une seule et unique page (`index.php`) contenant la structure du site.  
Pour cela vous utiliserez le passage d'un argument dans l'url de la page `index.php` (méthode GET) pour indiquer la page à inclure.  
Ex : `index.php?page=accueil`

## 4 Génération d'une liste déroulante

1. A partir du fichier XML créé dans le TP précédent, créez un fichier XSLT pour générer la liste déroulante des voitures (liste de marque/modèle/version).
2. Modifiez le formulaire précédemment créé pour y intégrer la liste déroulante ainsi générée. Pour cela, vous utiliserez les fonctions Javascript présentes dans le fichier disponible à l'adresse suivante :

<http://www.enseirb.fr/~veille/Enseirb/IT103/tp/ressources/xslt.js>

Pour l'utilisation des fonctions, vous vous inspirerez du fichier disponible à l'adresse suivante :

<http://www.enseirb.fr/~veille/Enseirb/IT103/tp/ressources/test.html>

3. De la même manière, intégrez la fiche voiture créée à partir du fichier XSLT du TP précédent pour l'afficher dans le récapitulatif.