

L2, développement *Web*

(tous documents autorisés)

Présentation du sujet

On dispose d'un formulaire inclus dans une page Web déjà valide au sens XHTML Strict. Voici le code qui produit ce formulaire :

```
<form action="non.php">
  <p>
    <b>Combien de rendez-vous désirez-vous ? </b>
  </p><p>
    <span id="unite">N = </span>
    <input type="text" id="entree" />
    <input type="submit" id="bouton" value="valider" onclick="envoyer()" />
    <input type="submit" value="Vider" onclick="vider()" />
  </p><p>
    <textarea rows="10" cols="60" id="sortie" class="invisible">
  </textarea>
</p>
</form>
```

Un rendu possible de ce code dans un navigateur est affiché ci-dessous :



Combien de rendez-vous désirez-vous ?

N =

On remarquera que la zone de texte (`textarea`) dont l'identifiant est `sortie` n'est pas affichée, grâce au style `invisible`.

Question 1

Que faut-il modifier dans l'élément `form` pour qu'un clic sur chacun des deux boutons ou que le fait d'appuyer sur la touche "Entrée" dans la zone de saisie ne charge pas la page définie par l'attribut `action` mais au contraire exécute ce qu'il faut ? On rappelle que l'attribut `action` est obligatoire et qu'il faut garder `non.php` comme valeur de cet attribut, notamment en cas d'erreur *Javascript*.

Question 2

Donner le code Javascript de la fonction *vider()*. Cette fonction doit mettre la chaîne vide dans la zone de saisie du nombre N et rendre invisible la zone de texte nommée **sortie**.

Question 3

Donner le code Javascript de la fonction *envoyer()*. Cette fonction devra fournir les dates des prochains rendez-vous à partir de la date courante. On admettra qu'il y a en général deux ou trois rendez-vous, cinq tout au plus et que la fréquence des rendez-vous est mensuelle. Cette fonction affichera ces dates dans la zone **sortie** après l'avoir rendue visible grâce au style `visible`.

On pourra faire abstraction de la notion de jour dans la semaine. On supposera de plus que, dans le cas où l'utilisateur ne donne pas de nombre de rendez-vous, il s'agit d'une demande d'un seul rendez-vous.

On soignera les affichages en fonction du nombre de rendez-vous pris qu'on numérottera soigneusement.

Voici par exemple ce qui pourrait être affiché si au mardi 16 octobre 2012 l'utilisateur demande trois nouveaux rendez-vous (attention aux années!) :

Combien de rendez-vous désirez-vous ?

N =

Voici les dates des 3 prochains rendez-vous

1. : 16 Novembre 2012
2. : 16 Décembre 2012
3. : 16 Janvier 2013

Remarque. Pour que ce soit plus simple à programmer, il est conseillé d'utiliser un tableau Javascript nommé `tabMois` défini comme suit :

```
tabMois = new Array("(vide)","Janvier","Février","Mars","Avril",  
                    "Mai","Juin","Juillet","Août","Septembre","Octobre",  
                    "Novembre","Décembre") ;
```

Question 4

Comment pourrait-on prendre en compte le jour de la semaine? Par exemple le 15 octobre 2012 est un lundi mais le 15 décembre 2012 est un samedi or le rendez-vous ne peut avoir lieu que du lundi au vendredi.

A défaut de fournir le code Javascript, on proposera une solution algorithmique sachant qu'un rendez-vous ne peut pas être avancé. Pour l'exemple précédent, le rendez-vous mensuel qui suit le jeudi 15 novembre 2012 est donc le lundi 17 décembre (car le 15 décembre est un samedi, jour impossible).

Question 5

Répondez à la question suivante en dix lignes au minimum, sans limite de maximum. Votre réponse devra mettre en évidence votre culture du Web. Il est conseillé d'utiliser au moins trois mots de trois syllabes ou plus pour « transmettre un contenu rédactionnel fort ».

Que manque-t-il à XHTML au niveau des formulaires pour que la saisie des informations de type date, texte et numérique soit plus facile à gérer en Javascript ?