

Cours SGBD

TP – Introduction à HTML, CSS, PHP

15 novembre 2006

1 Introduction

1.1 Serveurs Web

Vous placerez les documents produits dans le sous-répertoire `public_html` de votre répertoire Unix principal. Si votre login est `toto`, vous pourrez accéder à un document `titi.html` via un serveur Web à l'URL :

- `http://miage.isi.u-psud.fr/~toto/titi.html` pour les MIAGE
- `http://s2.ie2.u-psud.fr/~toto/titi.html` pour les M1

1.2 Validation

Pour vérifier qu'il n'y a pas d'erreurs dans l'écriture de fichiers HTML ou CSS, en plus de regarder le résultat dans un ou plusieurs navigateurs, on pourra utiliser des *validateurs* qui permettent de vérifier la conformité d'un document. Les URL suivantes sont des liens vers des services de validation en ligne.

HTML : `http://validator.w3.org/`

CSS : `http://jigsaw.w3.org/css-validator/`

La validation est importante pour s'assurer que l'ensemble des clients Web (navigateurs graphiques, navigateurs textes, navigateurs auditifs, moteurs de recherche, outils expérimentaux de traitement de pages...) puisse comprendre le contenu HTML et éventuellement CSS. Cependant, pour des raisons de temps, nous ne pourrions pas vérifier que toutes les pages produites sont bien valides, mais il faut garder ça à l'esprit.

2 HTTP

1. Créer un fichier texte `titi.txt` comportant quelques mots, et le placer dans votre répertoire `public_html`. Ouvrez ce fichier dans un navigateur Web (au choix suivant ce qui est installé, mozilla, firefox, galeon, konqueror...) :
 - d'abord en accédant directement au fichier (Fichier/Ouvrir ou protocole `file` ://)
 - ensuite en accédant à ce fichier via le serveur Web (protocole `http` ://)
2. Faire un croquis de l'architecture client/serveur HTTP. Quels sont les fichiers qui sont échangés lorsqu'on demande `titi.txt` ?
3. Taper dans un terminal `telnet miage.isi.u-psud.fr 80` (pour les MIAGE) ou `telnet s2.ie2.u-psud.fr 80` (pour les M1). Une fois connecté, taper les lignes (si votre login est `toto`, et en remplaçant `miage.isi.u-psud.fr` par `s2.ie2.u-psud.fr` le cas échéant) :

```
GET /~toto/titi.txt HTTP/1.1
Host: miage.isi.u-psud.fr
```

Puis valider en tapant deux fois de suite sur la touche <Entrée>. Que se passe-t-il? À quoi cela correspond-il?

3 Première page HTML

1. Créer le document HTML élémentaire ci-dessous et afficher le dans un navigateur. Valider.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="fr">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <title>Page de test</title>
  </head>
  <body>
    <h1>Page de test</h1>
    <p>Ceci est une page de test.</p>
  </body>
</html>

```

Le document fourni est un *squelette* qui pourra servir de base à l'ensemble des pages HTML que vous écrirez.

- Dans le paragraphe (p), mettre le mot "page" en *emphase* (**em**, habituellement rendu par de l'italique) et le mot "test" en *forte emphase* (**strong**, habituellement rendu par du gras).
- Ajouter une *règle horizontale* (**hr**) à la fin de votre document, en dessous de laquelle on mettra un *titre de sous-section* (**h2**) "Toto".
- HTML permet également de spécifier certaines indications de *présentation* (même si on préférera CSS pour ce but). Ajouter un attribut `bgcolor="blue"` à l'élément `body`. Ajouter `<blink>Je clignote !</blink>` en-dessous du titre "Toto".
- Ajouter un lien vers `http://www.u-psud.fr/` au tout début du document.
- Reproduire en HTML le tableau ci-dessous (ne pas chercher pour l'instant à mettre de bordures, on verra comment faire avec CSS) :

Bli	Blo	Blu
123	456	789
234	721	946

- Ajouter à la fin du document un formulaire contenant les champs suivants :
 - Un champ texte nommé `x`
 - Une liste déroulante à 4 entrées :
 - bleu
 - rouge
 - vert
 - jaune
 - Un groupe de boutons radio Oui/Non
 - Un bouton de réinitialisation du formulaire
 - Un bouton de soumission du formulaire

Remarque : pour avoir du HTML correct, les champs du formulaire (`input`), ainsi que les étiquettes `label` et le reste du contenu ne doivent pas être mis directement dans la balise `form`, mais encapsulés dans des balises `div` (bloc), `p` (paragraphe) ou `fieldset` (ensemble de champs).

On positionnera la méthode du formulaire à `post`. L'attribut `action` du formulaire doit pointer vers un script prenant en charge le traitement des données du formulaires. Que se passe-t-il si on indique le nom du fichier `html` comme `action`? Nous verrons dans la partie 5 comment utiliser PHP pour traiter les données du formulaire.

4 Première feuille de style CSS

- Rajouter dans votre document HTML un *lien* (`link`) vers une feuille de style externe `style.css`. On mettra ensuite toutes les indications de mise en forme dans cette feuille de style.
- Remplacer l'attribut `bgcolor` de la page HTML par une règle CSS `body { background-color: blue; }`
- Remplacer l'élément `blink` par un `span class="blink"` et ajouter la règle CSS `.blink { text-decoration: blink; }`
- Ajouter les règles CSS suivantes pour ajouter une bordure à votre table :


```
table { border-collapse: collapse; }
td, th { border: solid 1px black; }
```
- Ajouter la règle suivante et regarder ce qui se passe quand on passe la souris sur le lien au début de votre page :


```
a:hover { background-color: red; }
```

5 Premiers scripts PHP

Remarque : les scripts PHP seront utilisés pour produire des pages HTML, et s'inscriront donc naturellement dans un squelette de page HTML.

5.1 Script élémentaire

1. Créer un fichier `toto.php` dans votre répertoire `public_html` et y mettre :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="fr">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <title>Page de test</title>
  </head>
  <body>
    <p><?php echo "Bonjour monde!"; ?></p>
  </body>
</html>
```

Tester ce script (en HTTP, on ne peut plus utiliser le protocole `file ://` maintenant qu'on a un comportement dynamique côté serveur).

2. Définir dans le code PHP de `toto.php` une variable `$x` initialisé à 10, et afficher à l'aide d'une boucle les nombres de 1 à `$x`.
3. Initialiser `$x` à `$_REQUEST['x']`, et indiquer `toto.php` comme action du formulaire du document HTML précédemment créé. Tester.

5.2 Table de multiplication

Créer un script PHP affichant la table de multiplication de 1 à 10 sous forme d'un tableau HTML. Ne pas oublier les en-têtes de ligne et colonne appropriés.

5.3 Tableaux associatifs

1. Les paramètres HTTP passés par la requête sont disponibles dans le tableau associatif PHP `$_REQUEST`. Créer un script `affiche_parametres.php` qui affiche le contenu du tableau associatif `$_REQUEST` sous forme d'un tableau HTML; on pourra utiliser l'instruction :
`foreach ($tableau_associatif as $cle => $valeur) { }`
2. Ajouter à la fin de ce script un formulaire avec quelques champs, dont on fera pointer l'action vers le script lui-même. Tester.
3. Faire en sorte de n'afficher l'ensemble des paramètres HTTP que si celui-ci n'est pas vide.