

Internet et Outils (IO2)

Examen de session 1 (durée 2h)

16 May 2017

*Documents autorisés : trois feuilles A4 recto-verso manuscrites ou imprimées.
Les ordinateurs et les téléphones mobiles sont interdits.
Le barème est donné à titre indicatif.*

Lisez bien tout le sujet avant de vous mettre au travail. Certaines questions dépendent d'autres, mais vous pouvez les aborder même sans avoir terminé les questions précédentes.

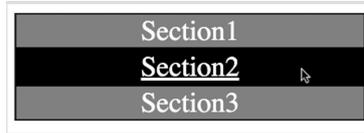
A. HTML et CSS

Écrire la feuille de style `menu.css` pour obtenir les effets décrits dans les deux questions ci-dessous sur le document HTML suivant, représentant un menu de navigation vertical :

```
<!DOCTYPE html>
<html>
<head>
  <title> Menu </title> <meta charset="utf-8">
  <link rel="stylesheet" href="menu.css" type="text/css">
</head>
<body>
  <nav>
    <ul id="menu">
      <li>Section1
        <ul>
          <li><a href="#">Paragraphe 1</a></li>
          <li><a href="#">Paragraphe 2</a></li>
        </ul>
      </li>
      <li>Section2
        <ul>
          <li><a href="#">Paragraphe 1</a></li>
          <li><a href="#">Paragraphe 2</a></li>
          <li><a href="#">Paragraphe 3</a></li>
        </ul>
      </li>
      <li>Section3
        <ul>
          <li><a href="#">Paragraphe 1</a></li>
          <li><a href="#">Paragraphe 2</a></li>
        </ul>
      </li>
    </ul>
  </nav>
</body>
</html>
```

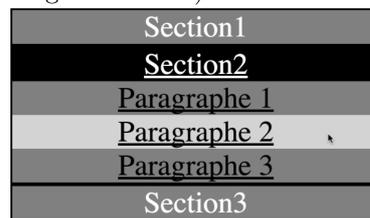
Question 1 — 3 points • les éléments `` imbriqués (c'est-à-dire contenus dans des éléments ``) ne sont initialement pas visibles.

- L'élément d'identifiant menu a une largeur de 200px ; son texte est blanc sur fond gris.
- Ses enfants (c'est à dire les sections) sont encadrés d'une bordure noire et ont texte centré (penser à éliminer le padding par défaut ainsi que la puce des listes). Quand la souris survole une section, elle prend couleur de fond noir et son texte devient souligné (cf. image ci-dessous).



Question 2 — 2 points Compléter la feuille de style `menu.css` pour faire en sorte que :

- Quand la souris survole une section, la liste imbriquée dans cette section devient visible, sur fond gris. Le texte des liens aux paragraphes est noir et souligné.
- Quand la souris survole un élément contenant un lien à paragraphe, celui-ci prend couleur de fond gris clair (cf. image ci-dessous).



B. Développement serveur et bases de données

Dans les réponses aux questions suivantes, veillez à utiliser les fonctions PHP pour factoriser votre code, pouvoir le réutiliser et le rendre lisible. Le but est d'écrire un mini-blog permettant l'ajout, l'édition et la suppression d'articles.

Attention : Toutes les pages du site doivent être attachée à la même feuille de style `monStyle.css` (qu'il n'est pas nécessaire d'écrire), et doivent comporter, en pied de page, l'adresse email du Webmaster.

- Question 3 — 8 points**
1. Écrire un script nommé `blog.php` contenant une suite de 5 articles, chacun avec un numéro identifiant unique, un titre, un contenu textuel, une date de création et de dernière modification. Chaque article est accompagné d'un bouton *supprimer* et un bouton *éditer*. Les scripts qui gèrent ces boutons ne sont pas à écrire tout de suite. Toutefois veillez à ajouter suffisamment d'information dans le/les formulaires, pour que leur soumission permette de retrouver le numéro identifiant de l'article associé. À la fin de la page, ajouter un bouton *nouveau*. Ne pas se soucier de la mise en page.
 2. Écrire un script PHP nommé `nouveau.php` qui gère le bouton *nouveau*. Ce script affiche un formulaire de saisie d'un nouvel article, terminant avec un bouton *enregistrer*. L'utilisateur doit pouvoir saisir seulement le titre et le contenu textuel de l'article. C'est-à-dire, dans le formulaire de saisie le numéro de l'article n'est pas affiché ; les deux dates (identiques et égales à la date courante) sont affichées dans des champs de texte, mais ne sont pas modifiables. Le code qui gère ce formulaire de saisie n'est pas à écrire tout de suite.

- Question 4 — 4 points**
1. Définir un schéma de base de données relationnelle qui permet de stocker des articles. Produire la commande SQL de `CREATE TABLE` correspondante. Veiller à bien introduire une contrainte de clef primaire.
 2. Réécrire en PHP la page `blog.php` (qui sera maintenant nommée `blog1.php`). Elle s'occupe toujours d'afficher la liste des articles selon le même format que dans la Question 1, mais ceux-là sont maintenant lus de la bases données, dans la table des articles.
 3. Écrire le script `supprimer.php` qui répond à l'action de suppression de chaque article. Le script efface l'article de la base de données et redirige vers la page principale (`blog1.php`) pour re-afficher la liste des articles mise à jour.

Question 5 — 3 points 1. Modifier le script `nouveau.php`, qui sera maintenant nommé `message.php`.

Indiquer les modifications nécessaires pour que `message.php` soit associé aussi bien au bouton *nouveau* qu'aux boutons *éditer* des différents articles. Si `message.php` est requis par un bouton *nouveau* il se comporte comme `nouveau.php`. S'il est requis par un des boutons *éditer*, il récupère l'article à éditer de la base de données et l'affiche dans le formulaire de saisie, pour modification. Dans les deux cas seulement le titre et le texte de l'article seront modifiables, comme à la Question 1.

2. Modifier `message.php`, qui sera maintenant nommé `message1.php` pour qu'il gère également l'enregistrement dans la base de données du nouveau message, ou bien du message édité. Dans `message1.php` l'action associée au formulaire de saisie est `message1.php`. Veiller à mettre à jour la date de dernière modification avant l'enregistrement s'il s'agit d'une opération de modification d'un article existant. Après les mises à jour (insertion ou modification) le script doit rediriger vers la page principale (`blog1.php`) pour re-afficher la liste des articles.

Question 6 — bonus 1. Modifier le schéma de la base de données pour représenter également les utilisateurs du site (avec nom, pseudo et mdp), ainsi que les utilisateurs auteurs des articles. Veuillez à bien concevoir le schéma pour éviter les redondances. Produire les commandes SQL de CREATE TABLE correspondantes. Veuillez à bien définir les contraintes de clef étrangère.

2. Introduire une page de login qui vérifie les informations de login dans la table des utilisateurs, dans la base de données. La page s'occupe de démarrer une session PHP en cas de succès. Veiller à stocker les bonnes informations dans les variables de session. Indiquer quelles modifications sont nécessaires dans les scripts `blog1.php` et `message1.php` pour obtenir que :

- l'accès à toutes les pages soit protégé avec les sessions ;
- l'auteur soit affiché avec chaque article ;
- l'utilisateur courant puisse voir tous les articles mais puisse supprimer/éditer uniquement ceux dont il est auteur ;
- un nouvel article, ainsi que un article modifié, soit toujours enregistré avec auteur égal à l'utilisateur modifiable dans le formulaire de saisie/modification d'un article).

3. Indiquer également les modifications nécessaires pour qu'un lien *déconnexion* soit affiché sur toutes les pages du site (exception faite de la page de login, bien évidemment). Écrire le script `deconnexion.php` pointé par ce lien, qui s'occupe de détruire la session courante, et rediriger vers la page principale.