

## Technologies Web 1

### ASI3 - Devoir machine

Durée : 2 heures 30

Documents de cours annotés

Vous trouverez dans `/opt/files/` une archive `ExamenTechnoweb1.tar.gz` contenant les annexes nécessaires à la réalisation de l'examen, les transparents et exemples du cours ainsi que les corrections des TDM. Vos compte-rendus de TDM seront copiés automatiquement dans `/tmp`, quelques temps après votre connexion. Vous avez aussi accès à la phpdoc dans le répertoire `/usr/share/doc/php-doc/`.

Afin d'accéder à 5 sites web (<http://www.w3schools.com/>, <https://api.jquery.com/>, <http://php.net/>, <http://www.xmlvalidation.com/> et <http://validator.w3.org/>), vous devez modifier la configuration du proxy. Pour Firefox, dans Edition/Préférences/Avancé/Réseau/Connexion/Paramètres, sélectionnez Configuration manuelle du proxy et saisissez `asi-examen.insa-rouen.fr` et le port 80. Cochez l'utilisation du serveur proxy pour tous les protocoles.

À la fin de l'examen, nettoyez votre `home` de façon à n'y conserver que vos fichiers exercice : votre travail sera récupéré sous la forme d'une copie de votre `home`. L'architecture à respecter est : sur votre **Bureau** un répertoire par exercice (**Exercice1**, **Exercice2**, ...), avec dans chaque répertoire autant de sous-répertoires **fonctionnels** que de questions répondues pour l'exercice correspondant (Q1, Q2, ...). Quand vous passez d'une question à une autre, il est donc conseillé de sauvegarder votre production dans le bon répertoire avant de passer à la question suivante. Je vous remercie de ne pas prévoir de répertoire lorsque vous ne répondez pas à une question.

**Toute archive ne respectant pas les contraintes ci-dessus ainsi que tout home volumineux (i.e. contenant des fichiers non nécessaires à la correction) seront sanctionnés d'un malus.**

### Exercice 1 : XML, DTD, CSS (4pts)

Vous êtes en charge de la mise en place d'un dictionnaire en ligne, style wikipédia, contenant un ensemble de fiches décrivant des concepts. Une fiche contient un concept (un mot à définir), une définition et la liste des contributeurs à la fiche correspondante.

1. Écrivez la DTD décrivant un dictionnaire, i.e. contenant une liste de fiches (`dictionnaire.dtd`).
2. Réalisez le fichier XML `dictionnaire.xml` correspondant à l'exemple ci-dessous.  
NB : les éléments de mise en forme, de ponctuation ainsi que les textes "Dictionnaire" et "Contributeurs" ne font évidemment pas partie des données du fichier.
3. Écrivez une CSS (`styleDictionnaire.css`) afin de mettre en forme le fichier précédent.
4. Modifiez DTD et fichier XML afin de pouvoir lier un concept à plusieurs autres concepts proches.

#### Dictionnaire

**chat** : *Petit animal fourbe à fourrure*

Contributeurs :

- Bob Leponge
- Samantha Lo

**ASI** : *Chouette département de l'INSA*

Contributeurs :

- Alexandre Pauchet

### Exercice 2 : HTML, Javascript, jQuery (5pts)

Pour cet exercice, vous trouverez dans l'archive fournie la bibliothèque `libs/jquery-2.1.4.min.js` nécessaire au fonctionnement de jQuery, ainsi qu'une page HTML vierge à compléter avec sa CSS correspondante.

1. Créez une page HTML (`index.html`) contenant un formulaire identique à celui présenté ci-dessous. Pour l'affichage, la CSS fournie ne doit pas être modifiée.
2. Ajoutez le code Javascript/jQuery nécessaire pour ouvrir une pop-up de validation lorsque l'utilisateur demande à effacer le formulaire.

3. Ajoutez le code Javascript/jQuery nécessaire à la validation du formulaire, lors de sa soumission :
  - un message "Vide !" est ajouté pour chaque champ laissé vierge ;
  - un message "Email !" est ajouté si l'email saisi n'est pas correct (de type *xxx@domaine.extension*) ;
  - un message "Erreur !" est ajouté si la 2nde saisie du mot de passe n'est pas identique à la première. Chaque message est affiché en rouge à côté du champ correspondant, lui-même encadré de rouge. Un exemple pour chaque type de message est présenté ci-dessous.
4. Ajoutez le code Javascript/jQuery nécessaire pour n'activer le champ de 2nde saisie de mot de passe que lorsque le premier contient au moins un caractère. Il sera bien entendu inactif au chargement initial.

Formulaire d'inscription	
Login :	<input type="text" value="Saisissez un login"/>
Email :	<input type="text" value="Saisissez un email"/>
Mot de passe :	<input type="password" value="Saisissez un mot de passe"/>
2nde saisie :	<input type="password" value="Re-saisissez votre mot de passe"/>
<input type="button" value="S'inscrire"/> <input type="button" value="Effacer"/>	

  

Formulaire d'inscription	
Login :	<input type="text" value="Saisissez un login"/> Vide !
Email :	<input type="text" value="mauvais email"/> Email !
Mot de passe :	<input type="password" value="*****"/>
2nde saisie :	<input type="password" value="**"/> Erreur !
<input type="button" value="S'inscrire"/> <input type="button" value="Effacer"/>	

## Serveur LASP

Un serveur Apache Http avec le module PHP activé est lancé sur chacune des machines et configuré pour fonctionner avec les répertoires utilisateur. Vous pouvez y accéder par une requête HTTP sur **localhost:80**. Les fichiers log sont redirigés vers /tmp. Pensez à donner les droits d'accès corrects (**chmod o+x /home/user** et **chmod o+rx /home/user/public\_html**). Les 2 parties (Exercice 3 et Exercice 4) sont indépendantes.

### Exercice 3 : SimpleXML (5pts)

Vous trouverez dans l'archive fournie un fichier **publications.xml** à exploiter.

1. Réalisez un script **writeGet.php** prenant en paramètre GET 'index', correspondant à la *index*ème publication dans **publications.xml**. Ce script affichera l'article demandé à l'aide de simpleXML.
2. Réalisez un script **listeGet.php** affichant une liste déroulante et un bouton d'envoi. La liste sera construite dynamiquement à l'aide des titres de **publications.xml**. Le bouton d'envoi appellera le script **writeGet.php** afin d'afficher la publication sélectionnée.
3. Réalisez des scripts **listePost.php** et **writePost.php** identiques aux précédents mais en POST.
4. Réalisez un script **index.php** identique à **listePost.php** qui traitera lui-même le formulaire en affichant en dessous la publication sélectionnée.

### Exercice 4 : SQLite et gestion d'une session (6pts)

Vous trouverez dans l'archive fournie un répertoire **data** contenant 2 fichiers textes ainsi qu'un script **creationBD.php** qui vous permettra de créer la base de données nécessaires à la réalisation de cet exercice.

1. Complétez **getDefinition.php** afin d'afficher sous le formulaire la définition du terme saisi.
2. Complétez **fichePost.php** afin d'afficher sous le formulaire le contenu du fichier du terme saisi.
3. Complétez **ajoutTerme.php** afin de permettre l'ajout d'un nouveau terme et de sa définition à la BD. Aucune fiche ne sera saisie et il n'est pas demandé de vérifier si le terme est déjà présent dans la BD.
4. Modifiez **getDefinition.php**, **fichePost.php** et **ajoutTerme.php** de façon à ce qu'un terme ne puisse être ajouté que par un utilisateur connecté. Il ne s'agit ici pas d'authentification, mais seulement d'utiliser un lien dynamique valant "Se connecter / Se déconnecter" en haut de la page afin de permettre la connexion et la déconnexion d'un utilisateur. Lors d'une connexion/déconnexion, afin de revenir vers la page appelante, vous pourrez utiliser la variable super globale **\$\_SERVER['HTTP\_REFERER']**.