

# TP noté Programmation web

ENSIIE - 2014/2015

Documents autorisés : 2 feuilles A4 recto verso.

Les exercices sont indépendants les uns des autres.

À la fin du sujet, vous pourrez trouver des rappels de syntaxe HTML, CSS, DTD, les fichiers PHP et SQL, ainsi que deux maquettes d'application.

## Partie 1 - Création d'une application web

L'objectif est de développer une application permettant aux élèves de l'ENSIIE de faire leur choix de deuxième langue vivante.

### 1 - Modélisation des traitements (UML)

L'application sera composée de deux pages : une page de récapitulatif des choix effectués par les élèves, qui sera consultable par la scolarité de l'école ; et une page d'inscription, qui sera utilisée par les élèves pour indiquer leur choix.

1 ) Dans le diagramme de cas d'utilisation, quels sont les acteurs et les relations éventuelles entre ces acteurs ?

2 ) Dans le diagramme de cas d'utilisation, quels sont les cas d'utilisation et les relations éventuelles entre ces cas ?

3 ) Donnez le nom du diagramme UML permettant de détailler le déroulement temporel d'un cas d'utilisation.

### 2 - HTML

Les squelettes des deux pages principales ont été créés : ce sont les fichiers `inscription.php`, qui permettra à un élève de s'inscrire à un cours, et `affichage.php`, qui affichera l'ensemble des inscriptions pour la scolarité. Une base de données `inscriptions` a également été créée, qui contient les relations suivantes :

```
cours(idcours, langue, niveau, enseignant)
```

```
eleve(idleve, nomeleve, prenomeleve, langue2 => cours.idcours)
```

Voir l'annexe D pour la description complète de la base de données.

Attention, le respect de la syntaxe HTML stricte sera pris en compte dans la notation.

La page `inscription.php` contient un formulaire d'inscription permettant aux élèves d'effectuer leur choix de langue vivante 2 parmi la liste des cours disponibles.

4 ) Ajoutez un paragraphe contenant le texte d'accueil "Bienvenue sur la page d'inscription" sur la page `inscription.php` sous le titre de la page.

5 ) Complétez le formulaire d'inscription afin qu'il comporte les éléments suivants (voir les commentaires comprenant le numéro de la question dans le code php) :

- deux champs de texte permettant de rentrer le prénom et le nom de famille de l'élève ;
- un champ caché permettant de savoir si le formulaire a été envoyé ;
- un bouton de validation affichant **M'inscrire !**.

6 ) Le remplissage de certains champs doit être vérifié ; notamment le nom et prénom doivent être obligatoirement remplis. Cette vérification peut se faire côté client ou côté serveur. Quels sont les langages utilisés dans chacun des cas et quelles sont les différences entre les deux possibilités ?

7 ) Que permet de sécuriser la fonction `securiser($string)` du fichier `util.php` ?

8 ) Ajoutez un lien hypertexte vers la page `inscription.php` après l'affichage des messages de retour (afin que l'élève puisse revenir à la page d'accueil de l'inscription).

### 3 - CSS

Le fichier `enssie.css` contiendra la définition des styles pour les pages de l'application.

9 ) Changez la taille des titres de premier niveau pour qu'ils s'affichent en 20px.

10 ) Changez la couleur du texte des paragraphes de classe `alert` pour qu'il s'affiche en rouge.

11 ) Ajoutez trois points d'exclamation après les paragraphes de classe `alert`.

### 4 - PHP

La page `affichage.php` affiche la liste des cours de langue, et pour chaque cours, la liste des élèves inscrits.

12 ) La liste des élèves inscrits pour chacun des cours de langue est ici obtenue par une seule requête SQL. Quelle était l'autre solution possible pour obtenir cette liste ; expliquez les différences entre les deux solutions.

13 ) Complétez la page `affichage.php` afin qu'elle affiche la liste des cours de langue avec la langue, le niveau et l'enseignant, et pour chaque cours, la liste des inscrits dans ce cours avec leur prénom et leur nom.

14 ) Faites appel à la classe `alert` définie dans le CSS pour afficher des messages d'erreur dans les cas où la connexion à la base de données ne s'est pas faite, et où il n'y a pas encore d'inscriptions.

## Partie 2 - XML

Le fichier `brochure.xml` contient (une toute petite partie) de la brochure des enseignements de l'ENSIIE, et devra ensuite être affiché sous plusieurs formats (site web, brochure papier...).

15 ) Créez la DTD `brochure.dtd` correspondant à ce document.

Rappel : pour vérifier que le document XML est conforme à la DTD, vous pouvez utiliser la commande suivante : `xmllint --noout --dtdvalid fichierdtd.dtd fichierxml.xml`

16 ) Quel serait l'avantage d'une description par un schéma XML pour les attributs `id` ou `coeff` ?

Le fichier `musique.xsd` correspond au schéma d'un catalogue de musique XML.

17 ) Créez un fichier XML validé par ce schéma, contenant au moins 3 entrées et permettant au maximum de tester les différents cas possibles.

Rappel : pour vérifier que le document XML est conforme à la DTD, vous pouvez utiliser la commande suivante : `xmllint --noout --schema fichierschema.xsd fichierxml.xml`

## A Quelques éléments de syntaxe HTML

<code>&lt;h1&gt;...&lt;/h1&gt;</code>	titres de niveau n= 1 à 6
<code>&lt;p&gt;&lt;/p&gt;</code>	paragraphe
<code>&lt;a href="url"&gt;...&lt;/a&gt;</code>	lien
<code>&lt;div&gt;...&lt;/div&gt;</code>	section d'un document
<code>&lt;br/&gt;</code>	retour à la ligne
<code>&lt;img src="url"&gt;...&lt;/a&gt;</code>	image
<code>&lt;ul&gt;...&lt;/ul&gt;</code>	liste non ordonnée (ol ordonnée)
<code>&lt;li&gt;...&lt;/li&gt;</code>	élément d'une liste
<code>&lt;table&gt;...&lt;/table&gt;</code>	tableau
<code>&lt;tr&gt;...&lt;/tr&gt;</code>	ligne d'un tableau
<code>&lt;td&gt;...&lt;/td&gt;</code>	cellule d'un tableau
<code>&lt;form&gt;...&lt;/form&gt;</code>	formulaire
<code>&lt;input type="xx"&gt;...&lt;/input&gt;</code>	élément d'un formulaire (type=text,password,submit,checkbox,hidden...)
<code>&lt;select&gt;...&lt;/select&gt;</code>	liste déroulante
<code>&lt;option&gt;...&lt;/option&gt;</code>	élément d'une liste déroulante
<code>&lt;textarea&gt;...&lt;/textarea&gt;</code>	zone de texte de plusieurs lignes

## B Quelques éléments de CSS

sélecteurs:

<code>#exemple</code>	éléments d'identifiant exemple
<code>.exemple</code>	éléments de classe exemple

valeurs:

<code>font-size</code>	taille des caractères
<code>color</code>	couleur du texte
<code>font-style</code>	style du texte (normal, italic, oblique)
<code>text-align</code>	alignement du texte (center, right, justify, left)
<code>background-color</code>	couleur de fond de la page

<code>:exemple</code>	pseudo classe (:active, :hover, :first-child...)
<code>:: exemple</code>	pseudo élément (::after, ::before, ::first-letter, ::selection...)

## C Syntaxe DTD

`<!ELEMENT nomelement (modele)>`

avec:

- types: EMPTY      élément vide  
          (PCDATA) chaîne de caractères  
          éléments fils
- connecteurs: ,      séquence  
                  |      choix
- indicateurs d'occurrence: (rien)    un seul  
                                  ?      0 ou un  
                                  \*      0 ou plus  
                                  +      1 ou plus

`<!ATTLIST nomelement nomattribut type mode [defaut]>`

avec:

- type : CDATA, ID, IDREF, (a|b|c)
- mode: #REQUIRED, #IMPLIED, #FIXED

## D Base de données inscriptions

archive\_eleves/create\_fill\_bd\_lv2.sql

```
-- #####
-- # Application : SQL script
-- # File       : create_fill_bd_lv2.sql
-- # Revision   : mars 2015
-- # Author     : Anne-Laure Ligozat
-- # Function    : structure et alimentation de la base de donnees
--               inscriptions
-- #####

DROP TABLE IF EXISTS cours, eleve CASCADE;

CREATE TABLE cours(
  idcours INTEGER PRIMARY KEY,
  langue VARCHAR(20),
  niveau VARCHAR(20),
  enseignant VARCHAR(30)
);

CREATE TABLE eleve(
  ideleve SERIAL PRIMARY KEY, /* serial = entier avec incrémentation automatique
  */
  nomeleve VARCHAR(30),
  prenomeleve VARCHAR(20),
  langue2 INTEGER,
  CONSTRAINT fk_langue2 FOREIGN KEY(langue2) REFERENCES cours(idcours)
);

INSERT INTO cours(idcours, langue, niveau, enseignant) VALUES (1, 'italien',
  'débutant', 'Paulo Mozza');
INSERT INTO cours(idcours, langue, niveau, enseignant) VALUES (2, 'japonais',
  'avancé', 'Ryo Sakai');
INSERT INTO cours(idcours, langue, niveau, enseignant) VALUES (3, 'suédois',
  'débutant', 'Första Sista');

INSERT INTO eleve(nomeleve, prenomeleve, langue2) VALUES ('Kent', 'Clark', 3);
INSERT INTO eleve(nomeleve, prenomeleve, langue2) VALUES ('Wayne', 'Bruce', 2);
INSERT INTO eleve(nomeleve, prenomeleve, langue2) VALUES ('Parker', 'Peter', 3);
INSERT INTO eleve(nomeleve, prenomeleve, langue2) VALUES ('Prince', 'Diana',
  NULL);
```

## E Fichier inscription.php

archive\_eleves/inscription.php

```
<?php
include("util.php");
entete("Inscription d'un étudiant");

// si le formulaire a été envoyé
if(isset($_POST['insc'])){
  if(empty($_POST['prenom'])
    || empty($_POST['nomfamille']))
```

```

)
{
    print "<p class='alert'>Merci de remplir votre prénom et votre nom de
        famille</p>";
}
else{
    if(inscription($_POST['prenom'], $_POST['nomfamille'], $_POST['choix'])){
        print "<p>Inscription effectuée !</p>";
    }
    else{
        print "<p class='alert'>Problème rencontré lors de la requête...</p>";
    }
}
}
/* Q8: ajouter un lien vers la page inscription.php */
}
else{
?>

<!-- Q4: ajouter un paragraphe d'accueil ici -->

<form method="post" action="inscription.php">
    <!-- Q5 a: ajouter un champ caché de nom insc permettant de savoir si le
        formulaire a été envoyé -->
    <p>Veuillez indiquer vos prénom et nom</p>
<!-- Q5 b: ajouter deux champs pour entrer les prénom et nom de famille de l'
    élève, qui seront contenus dans les variables prenom et nomfamille -->

    <p><select name="choix">
    <?php
    if( $db=connexionbd() ){
        $requete="SELECT * FROM cours ORDER BY langue ASC,niveau DESC";
        $result = pg_query($db,$requete);
        while( $tuple_courant=pg_fetch_assoc($result) ){
    print "<option value=\"".$tuple_courant['idcours']."\">".$tuple_courant['
        langue'].\" \".$tuple_courant['niveau'].\"</option>";
        }
    fermeturebd($db);
    }
?>
</select></p>

    <!-- Q5 c: ajouter un bouton de validation -->
    </form>

<?php
}
pied();
?>

```

## F Fichier affichage.php

archive\_eleves/affichage.php

```

<?php
include("util.php");
entete("État des inscriptions");

if( $db=connexionbd()){
    $requete="SELECT * FROM cours JOIN eleve ON (langue2=idcours) ORDER BY
        idcours";
    $result = pg_query($db,$requete);

```

```

if(pg_num_rows($result)>0){
    $tuple_courant = /* Q13 a : récupérer le résultat de la requête dans le
        tableau associatif $tuple_courant*/

    // boucle sur les cours
    while($tuple_courant){
        $cours_courant = $tuple_courant['idcours'];
        print "<h2>Cours de ".$tuple_courant['langue']." niveau ".$tuple_courant['
            niveau']." avec ".$tuple_courant['enseignant']."</h2>\n<ul>\n";
        $nb_inscrits=0;

        // boucle sur les eleves inscrits au cours
        while($tuple_courant
            && $cours_courant == $tuple_courant['idcours']){
            /* Q13 b: afficher un élément de liste contenant le prénom de l'élève suivi
                de son nom de famille */
            $tuple_courant = /* Q13 c : récupérer le résultat de la requête dans
                le tableau associatif $tuple_courant*/
            $nb_inscrits++;
        }
        print "</ul>\n";
        print "<p>Nombre d'inscrits: $nb_inscrits</p>";
    }
}
else{
    /* Q14 a: afficher un message d'erreur */
}
fermeturebd();
}
else{
    /* Q14 b: afficher un message d'erreur */
}
}

pied();
?>

```

## G Fichier util.php

archive\_eleves/util.php

```

<?php

function entete($titre){
    print "<!DOCTYPE html>\n";
    print "<html>\n";
    print "<head>\n";
    print "<meta charset=\"utf-8\"/>\n";
    print "<title>$titre</title>\n";
    print "<link rel=\"stylesheet\" href=\"ensiee.css\"/>";
    print "</head>\n";
    print "<body>\n";
    print "<h1>$titre</h1>\n";
}

function pied(){
    print "</body>\n";
    print "</html>\n";
}

function connexionbd(){

```

```

return pg_connect("host=pgsql dbname=inscriptions user=login password=password
");
}

function fermeturebd(){
    pg_close();
}

function inscription($prenom, $nomfamille, $choix){
    if($bd=connexionbd()){
        $requete="INSERT INTO eleve(nomeleve, prenomeleve, langue2) values ('".
            securiser($nomfamille)."', '".securiser($prenom)."', $choix)";
        if(pg_query($bd,$requete)){
            fermeturebd();
            return 1;
        }
        else{
            fermeturebd();
            return 0;
        }
    }
    else{
        return -1;
    }
}

function securiser($string){
    return addslashes($string, '%_');
}

?>

```

## H Maquettes

**État des inscriptions**

**Cours de japonais niveau avancé avec Ryo Sakai**

- Bruce Wayne

Nombre d'inscrits: 1

**Cours de suédois niveau débutant avec Första Sista**

- Clark Kent
- Peter Parker

Nombre d'inscrits: 2

FIGURE 1 – Affichage des cours

**Inscription d'un étudiant**

Merci de remplir le formulaire ci-dessous

Veillez indiquer vos prénom et nom

prénom:  nom de famille:

italien débutant

FIGURE 2 – Inscription aux cours