

Technologie Internet

Remarque : tous les fichiers cités sont disponibles sur le web à l'URL

`http://www.info.univ-angers.fr/pub/gh/internet/ti2005.zip`

Une copie locale est également disponible en

`K:\coursGH\internet\ti2005.zip`

Vous zipperez tous vos fichiers-solution en une archive `.zip` avec vos initiales. Vous indiquerez sur votre copie le nom de cette archive (la liste des fichiers à fournir est indiquée en fin de sujet).

1. Html et styles

Le document `xmp.htm` est-il correct au sens des marqueurs de HTML 3? de XHTML? de XML? Que peut-on lui reprocher?

Indiquer quelle commande ou quel logiciel vous utilisez pour tester ce fichier et indiquer le cas échéant les incorrections de ce fichier.

Après avoir lu les deux pages `p1.htm` et `p2.htm` proposez un feuille de style `produit.css` qui permet de faciliter les mises en forme communes. Mettez ensuite dans `n_p1.htm` une copie de `p1.htm` qui utilise ce style.

2. Javascript

Lorsqu'une page Web référence un site, elle utilise le marqueur <A HREF. On veut ici automatiser l'affichage d'une liste de sites à l'aide d'une liste de définition (marqueur <DL>), chaque site étant défini par son adresse et un commentaire. Par exemple pour le site <http://www.google.fr> avec le commentaire Le meilleur moteur de recherches, on voudrait produire la partie de page Web suivante :

```
<dt><a href="http://www.google.fr">http://www.google.fr</a></dt>  
<dd>Le meilleur moteur de recherches</dd>
```

Ecrire une fonction Javascript `ecritSite(cmt,adr)` qui met en forme ses deux paramètres à l'aide des balises indiquées. Vous insérerez ensuite cette fonction dans la page `listeSites.htm` que vous complétez afin d'afficher le site de Google précédemment cité ainsi que le site

<http://www.info.univ-angers.fr/>

affecté du commentaire

Département Informatique - Université Angers.

Pour les plus fort(e)s, indiquez comment on peut insérer des commentaires avec les symboles " et ' comme pour

- Département Informatique de l'Université d'Angers
- sans doute "le meilleur" moteur de recherches

3. Php et MySql

La table `stagesMass` de la base *mySql* nommée `test` contient une copie des stages de maitrise Mass effectués ces dernières années. On voudrait en extraire les noms et téléphones des personnes ainsi que leur statut (stagiaire ou maitre de stage) pour constituer un annuaire.

Ecrire un fichier `creann.mysql` exécutable en ligne de commande sous *Deneb* qui crée ou recrée une table *mySql* nommé `annu_XX` où `XX` sont vos initiales. Cette table ne contiendra que trois champs : `Nom`, `Statut` et `Tel`.

Ecrire ensuite un fichier `rempann.mysql` exécutable en ligne de commande sous *Deneb* qui remplit l'annuaire avec les renseignements demandés. On ne retiendra que les enregistrements pour lesquels les trois champs sont tous renseignés.

A la réflexion, on se demande s'il ne suffirait pas d'afficher l'annuaire en ligne dans une page *Web*. Donner un fichier `affann.php` qui à partir de la base des stages fournit le nom, le statut et le numéro de téléphone des personnes. Là encore, on n'affichera que les enregistrements pour lesquels les trois champs sont tous renseignés.

Discuter ensuite quelle solution (affichage en ligne ou extraction dans une autre base) vous parait la meilleure dans le cadre d'une utilisation ponctuelle (la mise à jour se faisant une fois par an) et dans le cadre d'une utilisation hebdomadaire (la base des stages correspondant alors à une grande entreprise avec de nombreux stagiaires pour des stages courts).

4. Applet Java

Ecrire une page web dans le fichier `examj.htm` qui utilise une applet java nommée `prepref` que vous définirez via le fichier `prepref.java`. L'applet affiche deux champs-texte avec leurs libellés et un bouton. Les deux champs-texte correspondent respectivement à un site et à un commentaire comme dans l'exercice 2. Lorsqu'on clique sur le bouton, l'applet affiche le texte HTML qui correspond aux parties `dt` et `dd` de l'exercice 2 de façon à ce que l'utilisateur puisse faire un copier-coller de ce texte

Ainsi, à la suite de la validation de la saisie

```
Site                                     (label)

+-----+                               (champ-texte)
+                                     +
+  http://www.google.fr  +
+                                     +
+-----+

Commentaire                             (label)

+-----+                               (champ-texte)
+                                     +
+  Le meilleur moteur de recherches  +
+                                     +
+-----+

      +====+
      | Ok |                               (bouton)
      +====+
```

On devra voir :

VOICI CE QUE VOUS POUVEZ RECOPIER

```
<dt><a href="http://www.google.fr">http://www.google.fr</a></dt>
<dd>Le meilleur moteur de recherches</dd>
```

5. Xml

Un stage de maitrise MASS met principalement en jeu les informations suivantes :

- année, titre et catégorie de stage (info, stat, gp, grh...),
- nom, prénom, adresse, téléphone et mail du stagiaire,
- nom, adresse et téléphone de l'entreprise,
- nom, prénom, téléphone et mail du maitre de stage.

Pour simplifier l'exercice, on ne gardera comme informations obligatoires que

- année, titre et catégorie de stage,
- nom et le prénom du stagiaire,
- nom et adresse de l'entreprise,
- nom et téléphone du maitre de stage.

Ecrire un document XML complet et correct (vous indiquerez quelle commande vous avez utilisée pour le tester) nommé `stage.xml` qui structure ces informations. On le remplira avec les données suivantes contenues dans le fichier `ti2005.stg` (stage fictif) :

```
2035 Etude sur la commercialité intramédicale (gp)
DUPON Jean
CHU Belle-Beille, 3 rue Pomme 49000 Angers
Mr ZOUAVE, 0241024100
```

Vous pouvez choisir les balises que vous voulez, soit avec des éléments texte soit avec des couples attribut/valeur sachant que comme d'habitude on veut être capable au minimum d'interroger la liste des stages pour obtenir les stages par année, par catégorie...

S'il vous reste du temps, écrivez la grammaire correspondante (au choix : DTD ou Schéma) ; on pourra la faire construire automatiquement par logiciel ou par une page Web — indiquez alors comment vous l'avez obtenue.

Liste des fichiers à fournir

Votre archive .zip devra contenir au minimum les fichiers suivants :

produit.css, n_p1.htm (exercice 1),

listeSites.htm (exercice 2),

creann.mysql, rempann.mysql, affann.php(exercice 3),

examj.htm, prepref.java, prepref.class(exercice 4),

stage.xml (exercice 5).

Vous pouvez y adjoindre **stage.dtd** ou **stage.xsd** pour l'exercice 5 ainsi que tout fichier-réponse de votre choix. Vous pouvez également utiliser la copie fournie pour rédiger vos remarques et arguments.

Compte tenu de la durée de l'épreuve et du travail demandé, il n'est peut être pas judicieux de vouloir tout tester sur ordinateur.

Esquisse de Solution

1. Html et styles

Le document `xmp.htm` s'affiche correctement avec un navigateur comme *Netscape* ou *Internet Explorer*. Toutefois, il contient de petites erreurs. Voici par exemple ce que dit `weblint` :

ligne 5
The HTML spec. recommends the TITLE be no longer than 64 characters.

ligne 18 :
empty container element <P>.

ligne 23 :
value for attribute HREF (`http://www.info.univ-angers.fr/pub/gh/`)
of element A should be quoted
(i.e. `HREF="http://www.info.univ-angers.fr/pub/gh/"`)

ligne 23 :
IMG does not have ALT text defined.

ligne 29 :
empty container element <P>.

Commençons par gérer les petites erreurs sans conséquence.

L'erreur en ligne 5 peut se corriger en raccourcissant le titre au texte "Page à tester".

On peut sans doute supprimer les balises <P> des lignes 18 et 29 pour ne plus avoir de paragraphes vides.

Au niveau de la ligne 23, il faut impérativement ajouter des guillemets pour l'URL ; pour l'image, on peut indiquer en mode ALT qu'il s'agit du retour à la page principale soit le texte correct :

```
<a href="http://www.info.univ-angers.fr/pub/gh/">  
</a>
```

On trouvera la page corrigée et donc correcte pour HTML 3 à l'URL

<http://www.info.univ-angers.fr/pub/gh/internet/xmp2.htm>

Si maintenant on se préoccupe de HTML 4 et de XHTML, à l'aide du site

<http://validator.w3.org/>

on voit apparaître une erreur grave : il n'y a pas de balise DOCTYPE comme première balise. Si on met

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

comme nouvelle première ligne du document, alors le document est valide pour la spécification "HTML 4.01 Transitional". On pourra le vérifier en vérifiant l'URL

<http://www.info.univ-angers.fr/pub/gh/internet/xmp3.htm>

Par contre si on veut un "vrai" document HTML 4, alors il reste du travail à faire comme on peut s'en rendre compte en vérifiant l'URL

<http://www.info.univ-angers.fr/pub/gh/internet/xmp4.htm>

Enfin, il est clair que xmp.htm n'est pas un document *XHTML* car sinon il commencerait par

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 ...
```

Et ce n'est pas non plus un document *XML* car de nombreuses balises ne sont pas fermées, comme par exemple les `<p>`. De plus les "entités" comme ` ` et `à` ne sont pas définies, pas plus que le codage (qui est sans doute ici ISO-8859 comme pour tout "bon français qui se respecte"). A ce propos, signalons qu'il est certainement peu cohérent de coder avec des accents comme dans "pensé" et de coder avec des entités comme dans "après".

Si l'on compare les deux fichiers `p1.htm` et `p2.htm`, on voit que les produits sont indiqués en rouge et en gras et que les arguments de vente sont en bleu et en gras. Un style doit donc certainement définir ces deux mises en forme.

Une première solution consiste à utiliser des balises nommées `rougeG` et `bleuG`. On définit ces balises dans le fichier `produita.css` qui contient

```
rougeG { font-weight : bold ; color: #990000 ; }
bleuG  { font-weight : bold ; color: blue ;    }
```

Il faut alors indiquer dans l'en-tête (avant `<BODY>`) du document *HTML* qu'on veut utiliser ces styles, soit l'instruction

```
<link rel="stylesheet" type="text/css" href="produita.css">
```

et il faut modifier en conséquence le corps du texte, soit les lignes

```
<rougeG>AAA</rougeG>
...
<bleuG>moins de 10 euros</bleuG>
```

au lieu des lignes

```
<font color="#880000"><b>AAA</b></font>
...
<font color="#000088"><b>moins de 10 euros</b></font>
```

On trouvera ce fichier à l'adresse

http://www.info.univ-angers.fr/pub/gh/n_p1a.htm

Toutefois cette solution n'est pas satisfaisante car la gestion des styles par Netscape ne supporte pas cette utilisation de balises (sans compter que `weblint` les traite comme des erreurs). Nous allons donc donner une solution "plus propre". Pour cela, dans notre nouvelle feuille de style `produitb.css`, nous mettons

```
.rougeG { font-weight : bold ; color: #990000 ; }
.bleuG  { font-weight : bold ; color: blue ;    }
```

La différence avec le style précédent réside seulement dans les points en début de nom de style. Nous utilisons alors ces styles en disant que ce sont des "attributs universels" via le mot *class*, soit le texte

```
<font class="rougeG">AAA</font>
...
<font class="bleuG">moins de 10 euros</font>
```

Du coup, `weblint` est content et cette solution marche avec pratiquement tous les navigateurs. On trouvera ce nouveau fichier à l'adresse

http://www.info.univ-angers.fr/pub/gh/n_p1b.htm

2. Javascript

La fonction javascript à écrire ressemble à

```
<script language="javascript">
  function ecritSite(cmt,adr) {
    document.write("<DT><a href=\""+adr+"\">"+adr+"</a></DT>") ;
    document.write("<DD>"+cmt+"</DD>") ;
  } // fin de fonction ecritSite
</script>
```

et on on peut l'utiliser ainsi (remarquer les anti-slashes dans les chaines de caractères transmises) :

```
<DL>
  <script language="javascript">
    ecritSite("sans doute \"le meilleur\" moteur de recherches",
              "http://www.google.fr")
    ecritSite("Département Informatique de l'Université d'Angers",
              "http://www.info.univ-angers.fr/")
  </script>
</DL>
```

ce que vous pouvez vérifier grâce à l'URL

<http://www.info.univ-angers.fr/pub/gh/desSites.htm>

3. Php et MySql

Pour créer l'annuaire de personnes des stages, il faut simplement définir les champs Nom, Statut et Tel. Nous proposons le programme suivant

```
USE test ;

DROP TABLE IF EXISTS annuStages ;

CREATE TABLE annuStages (
    Nom          varchar(60) ,
    Statut       text ,
    Telephone    text
) ; # fin de Create
```

Nous avons repris pour Nom le type trouvé dans `describe stagesMass` pour le champ de même nom. Ce type est compatible avec la description de responsable puisque les personnes sont soit des stagiaires ("nom"), soit des maitres de stages ("responsable").

Le statut sera une chaîne de caractère, à savoir "stagiaire" ou "maitre de stage".

Enfin pour le téléphone, on utilise le même type que pour `telstg` et `tel` de la base des stages.

Pour le remplissage, nous utilisons deux instructions `REPLACE...SELECT`, la première pour les stagiaires et la seconde pour les maitres de stage. Dans la mesure où `REPLACE` ajoute des données puisque nous n'avons pas défini d'index, il est prudent de supprimer les anciens enregistrements, soit le programme :

```
USE test ;

DELETE FROM annuStages ;

REPLACE INTO annuStages (Nom,Statut,Telephone)
SELECT distinct(nom),"stagiaire",telstg FROM stagesMass
WHERE LENGTH(nom)>0 AND LENGTH(telstg)>0
; # fin de replace pour les stagiaires
```

```

REPLACE INTO annuStages (Nom,Statut,Telephone)
  SELECT distinct(trim(responsable)),"maitre de stage",
    tel FROM stagesMass
  WHERE LENGTH(trim(responsable))>0 AND LENGTH(tel)>0
; # fin de replace pour les maitres de stage

```

Si maintenant on veut utiliser PHP pour afficher les mêmes informations, il suffit de passer en revue un **SELECT** de la base pour retenir les enregistrements si les champs **NOM** et **TELSTG** sont simultanément remplis ou si les champs **RESPONSABLE** et **TEL** sont simultanément remplis, soit le programme :

```
<?
```

```

mysql_connect("sirius","anonymous","anonymous") ;
mysql_select_db("test") ;

echo " <h2>Liste des stagiaires</h2> " ;

# requête stagiaires

$debreq = " SELECT nom,telstg FROM stagesMaitrise " ;
$finreq = " WHERE LENGTH(nom)>0 AND LENGTH(telstg)>0 " ;
$req     = mysql_query("$debreq $finreq ;") ;

# affichage de la liste des stagiaires

echo "<table border=1>" ;

while ($ligr=mysql_fetch_array($req)) {

    $nom    = $ligr["nom"] ;
    $tel    = $ligr["telstg"] ;
    echo "<tr><td> $nom </td><td> $tel </td></tr><p>\n";

} ; # fin de tant que

echo "</table>" ;

```

```

echo " <h2>Liste des maitres de stage</h2> " ;

# requête maitres de stage

$debreq = " SELECT responsable,tel FROM stagesMaitrise" ;
$finreq = " WHERE LENGTH(responsable)>0 AND LENGTH(tel)>0 " ;
$req     = mysql_query("$debreq $finreq ;") ;
echo " $debreq $finreq" ;

# affichage de la liste des maitres de stages
...

```

Vous trouverez l’affichage correspondant à la page :

http://www.info.univ-angers.fr/pub/gh/aff_Ann.php

En ce qui concerne le choix d’un programme mySql ou PHP, il est certainement préférable dans tous les cas de figures de passer par PHP, ne serait-ce que pour la gestion de l’affichage.

4. Applet Java

La page *Web* demandée ne comporte qu’un simple appel de l’applet :

```

<html><head><title>examj.htm (java)</title></head>
<body background="beige.jpg">
<h1>Utilisez le panneau suivant pour préparer vos</h1>
<h1>références de sites Web :</h1>
<blockquote>
<applet code="prepref.class"
        width="800"
        height="200">
</applet>
<blockquote>
</body>
</html>

```

Pour le programme *Java*, on peut essayer progressivement de gérer une ligne avec un "label", puis avec un "textfield" avant de passer à une deuxième ligne puis une troisième, en faisant un copier-coller des instructions précédentes. On obtient alors un programme comme :

```
// (gH)  --  prepref.java

import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class prepref extends java.applet.Applet implements ActionListener {

    Label lab1 =
        new Label("Indiquer ici l'adresse du site ",0) ;
        /* la valeur 0 est mise LEFT */

    Label lab2 =
        new Label("Indiquer ici le commentaire ",0 ) ;

    Label lab3 =
        new Label(" cliquer ensuite sur le bouton nommé Ok",0) ;

    Label lab4 = new Label(" ",0) ;
    Label lab5 = new Label(" ",0) ;
    Label lab6 = new Label(" ",0) ;

    TextField  tf1 = new TextField(70) ;
    TextField  tf2 = new TextField(70) ;

    Button      bout = new Button(" Ok ") ;

    Panel lig1 = new Panel() ;
    Panel lig2 = new Panel() ;
    Panel lig3 = new Panel() ;
    Panel lig4 = new Panel() ;
    Panel lig5 = new Panel() ;
    Panel lig6 = new Panel() ;
```

```

public void init() {

    setLayout( new GridLayout(6,1)) ;

    add(lig1) ; lig1.add(lab1) ; lig1.add(tf1) ;
    add(lig2) ; lig2.add(lab2) ; lig2.add(tf2) ;
    add(lig3) ; lig3.add(lab3) ;
    add(lig4) ; lig4.add(lab4) ;
    add(lig5) ; lig5.add(lab5) ;
    add(lig6) ; lig6.add(lab6) ;

    tf1.setText(" http://www.google.fr ") ;
    tf2.setText(" le meilleur moteur de recherches ") ;

    lig3.add(bout) ;
    bout.addActionListener(this) ;

} ; // fin de init

public void actionPerformed(ActionEvent e) {
    lab4.setText(" VOICI CE QUE VOUS POUVEZ RECOPIER ") ;
    String href = tf1.getText() ;
    lab5.setText(" <dt><a href=\""+href+"\">"+href+"</a></dt>" ) ;
    lab6.setText(" <dd>"+tf2.getText()+"</dd>" ) ;
    repaint() ;
} ; // fin de actionPerformed

} ; // fin de classe prepref

```

Toutefois, avec un peu de recul, il serait plus propre de définir des tableaux de "label" et de "textfield" ; on pourrait alors tout initialiser par des boucles. Nous laissons le soin au lecteur d'écrire cette nouvelle version. On pourra tester l'applet à l'adresse suivante :

http://www.info.univ-angers.fr/pub/gh/exam_j.htm

5. Xml

On peut écrire de nombreux fichiers *XML* pour le stage selon que l'on veut utiliser des éléments imbriqués ou des couples (attribut,valeur). Un fichier "purs éléments" ressemble à

```
<stage>
  < sujet>      </sujet>
  < an>        </an>
  < categorie>  </categorie>
  < stagiaire>
    < nom>      </nom>
    < prenom>   </prenom>
    < numtel>   </numtel>
    < adr>      </adr>
    < mail>     </mail>
  </stagiaire>
  < entreprise>
    < nom>      </nom>
    < mds>     </mds>    <!-- maitre de stage -->
    < adr>
      < rueetc> </rueetc>
      < codepost> </codepost>
      < fincp>   </fincp> <!-- fin code postal -->
    </adr>
    < telent> </telent> <!-- telephone entreprise -->
    < telmds> </telmds> <!-- telephone mds -->
    < telfax> </telfax> <!-- numero de fax -->
    < mailmds> </mailmds>
  </entreprise>
</stage>
```

Nous avons décidé ici de donner une version longue du stage, contrairement à l'énoncé qui simplifiait compte tenu de la durée de l'examen.

On notera qu'on a pensé à mettre un élément `fincp` pour inclure le CEDEX français comme dans 49045 ANGERS CEDEX 2.

Un fichier *XML* qui serait "tous couples (attribut,valeur)" ressemblerait plutôt à

```
<stage sujet="" an="" categorie="">
  <stagiaire nom="" prenom="" numtel=""
    adr="" mail="" />
  <entreprise nom="" mds=""
    adr_rueetc=""
    adr_codepost=""
    adr_fincp=""
    telent="" telmds="" telfax="" mailmds="" />
</stage>
```

Et bien sur des panachés entre ces deux fichiers extrêmes sont possibles. La bonne solution (si tant est qu'il n'y en ait qu'une seule) doit se trouver en regardant ce qu'on veut en faire, comme le suggère l'énoncé.

Il est déconseillé d'essayer d'écrire les grammaires correspondantes car des outils logiciels comme XMLSPY ou la page *Web*

http://www.hitsw.com/xml_utilites/

le feront automatiquement et mieux que nous, et qui plus est, sans erreur dans les fichiers-grammaires.

Pour finir, on notera que l'adresse

<http://www.info.univ-angers.fr/pub/gh/xmti2005.zip>

fait référence à l'archive des fichiers et programmes présentés dans la solution de l'examen.