Travaux Pratiques
Technologies et langages de l'Internet
5 ETI spécialité « Systèmes Informatiques Distribués »

Objectifs pédagogiques des TPs

Le module Technologies et langages de l'Internet (TLI) a pour principal objectif de vous faire découvrir ou redécouvrir les principaux langages de programmation utilisés de nos jours sur Internet : HyperText Markup Language (HTML) pour le contenu des pages, Cascading Style Sheets (CSS) pour la présentation, PHP: Hypertext Preprocessor, anciennement Personal Home Page (PHP) et javascript pour la réalisation de pages dynamiques, Structured Query Language (SQL) pour l'utilisation de bases de données, eXtensible Markup Language (XML) pour la représentation de données et la réalisation de formats personnalisés.

Ceux qui d'entre vous qui suivront la seconde partie du module (soit en spécialité, soit en tronc commun) découvriront également eXtended Stylesheet Language (XSL) pour la manipulation et la transformation des documents XML, les Application Programming Interface (API) de programmation XML Simple API for XML (SAX) et Document Object Model (DOM), l'architecture REpresentational State Transfer (REST) et les web services.

Il serait illusoire d'imaginer qu'au terme de ce module, vous serez expert dans toutes ces technologies : le temps imparti à leur étude est insuffisant, et ce n'est d'ailleurs en aucun cas l'objectif de ce module. *En revanche*, il vous sera demander de comprendre et de mettre en œuvre ces technologies de manière exhaustive, en appliquant les bonnes pratiques qui vous seront enseignées.

Il est donc primordial pour vous de maîtriser non pas les langages étudiés mais d'en comprendre l'intérêt, de savoir comment et quand les utiliser, et de connaître les bonnes pratiques de conception et de réalisation de solutions web.

Pour vous faciliter la tâche, différents types de TPs vous seront proposés :

1. des TPs de découverte d'application du cours (partie I du document) : destinés à vous familiariser avec les bases des langages et à vous apprendre les bonnes pratiques liées à leur utilisation ;
2. des TPs de mise en pratique concrète (partie II du document) : ces TPs « fil rouge » seront directement liés à un problème concret pour lequel il vous sera demander de proposer des solutions en utilisant les compétences et connaissances acquises en cours et durant la réalisation des TPs de découverte et d'application du cours ;
3. des TPs projet (partie III du document) où il vous sera demandé d'imaginer vous–mêmes une problématique, une conception et une réalisation.

L'enchaînement des TPs se fera directement en lien avec le cours. La séquence pédagogique (voir Séquence pédagogique) *ne suit pas l'ordre de lecture de ce document*, nous avons préféré regrouper les TPs par nature plutôt que par thématique pour des raisons de cohérence de contenu.

Un temps indicatif sera proposé pour chaque TPs : le contenu de ce module est très dense, il est important de vous astreindre à respecter les temps conseillés. Si cela est trop difficile, il vous faudra probablement compenser par du travail personnel en dehors des heures de face–à–face avec vos enseignants.

TPs de découverte et d'application du cours

Les TPs de découverte et d'application du cours comportent de petits exercices ou de petites activités qui vous permettront d'acquérir les bonnes pratiques de conception et de réalisation. Très guidés et directifs, ils ne requièrent aucun prérequis (mis à part ce qui a été introduit lors des cours).

Au terme du module, il vous sera demandé de maîtriser complètement leur exécution.

TPs projet fil rouge

Les TPs « fil rouge » vous guideront tout au long du module pour aboutir à une réalisation complète de site internet incluant toutes les technologies et les pratiques du cours. Ils prendront la forme d'un cahier des charges auquel vous devrez apporter des solutions.

Cette année, la thématique de ces TPs est la réalisation d'un site permettant de consulter les principales pathologies en médecine traditionnelle chinoise.

Vous aurez plus de liberté quant à la mise en œuvre des solutions, n'hésitez onc pas à vous démarquer en vous montrant originaux et innovants !

Il vous sera surtout demandé de respecter les bonnes pratiques de codage et de conception. L'aspect graphique est secondaire, il vaut mieux réaliser un site bien programmé qu'un site esthétiquement intéressant. Ne passez donc pas trop de temps sur l'aspect graphique, sauf si vous êtes à l'aise.

Afin de garantir un suivi pédagogique efficace, **il est obligatoire de travailler en binôme sur ces projets**. En fonction des effectifs, un seul trinôme par groupe sera accepté.

TPs projet de synthèse

Ces derniers TPs prendront la forme de projets personnalisés : il vous sera demandé d'imaginer de nouvelles fonctionnalités pour le site et de les mettre en œuvre.

Vous aurez donc une grande liberté tant au niveau de la conception que de la réalisation.

Vous travaillerez par groupe de 4 à 5 étudiants.

Séquence pédagogique

Première partie

— séance 1, *mercredi 1/10 am* : cours 1 (2h) + Web statique : Accessibilité, HTML et CSS (TP 1) (2h)
— séance 2, *mercredi 1/10 pm* : Fil rouge : Première ébauche statique du site (TP 2) (4h)
— séance 3, *jeudi 2/10 am* : cours 2 (1h) + Web dynamique avec PHP (TP 3) (1h) + Fil rouge : Dynamisation de site avec PHP (TP 4) (2h)
— séance 4 : *vendredi 3/10 am* : cours 3 (2h) + PHP avancé : gestion de formulaires, variables web globales, objet (TP 5) (30 min) + Fil rouge : Formulaires HTML et PHP Objet (TP 6) (1h30)

— séance 5 : *vendredi 3/10 pm* : SQL, PHP, PDO, et injections SQL (TP 7) (1h) + Fil rouge : Bases de données et PHP (TP 8) (3h)
— séance 6, *mercredi 8/10 pm* : cours 4 (2h) + XML par l'exemple (TP 9) (2h)
— séance 7, *jeudi 9/10 am* : Fil rouge : XML (TP 10) (2h) + Évaluation (2h)

Deuxième partie

— séance 8, *jeudi 9/10 pm* : cours 5 (1) + XSL 1 par l'exemple (TP 11) (3h)
— séance 9, *mercredi 15/10 am* : cours 6 (1) + XSL 2 par l'exemple (TP 12) (3h)
— séance 10, *jeudi 16/10 am* : Fil rouge : Transformation de contenu XML par XSL (TP 13) (4h)
— séance 11, *jeudi 16/10 pm* : cours 7 (1) + Les API de manipulation XML et XSL (TP 14) (1h) + Fil rouge : Un agrégateur de Flux rss paramétrable (TP 15) (2h)
— séance 12 : *vendredi 17/10 am* : cours 8 (2h) + Web service et architecture REST (TP 16) (2h)
— séance 13 : *vendredi 17/10 pm* : Fil rouge : Webservices : agrégation rss paramétrée et ressources REST (TP 17) (4h)
— séance 14, *jeudi 23/10 am* : Fil rouge : Préparation de l'évaluation (TP 18) (2h) + Évaluation (2h)

Projet

Le module se terminera par la réalisation d'un petit projet (Projet de fin de module).
— séance 15 : projet (4h)
— séance 16 : projet (4h)
— séance 17 : projet (2h) + Concours/Évaluation (2)

Évaluation

Il y a aura quatre évaluations pour le module :
— une évaluation de TP pour la première partie du module ;
— une évaluation de TP pour la deuxième partie ;
— une note de projet ;
— une évaluation de synthèse (devoir) individuelle.

Évaluation des TPs

Les TPs d'application du cours ne feront pas l'objet d'une évaluation en soi.

Les autres TPs et projet seront évalués conjointement par les enseignants et les étudiants qui les ont réalisés.

Devoir de synthèse

Le devoir de synthèse sera réalisé en ligne. Il sera composé pour un tiers de Questionnaire à Choix Multiples (QCM) ou de questions à réponse courte et pour le reste de questions ouvertes.

Table des matières

Objectifs pédagogiques des TPs i
TPs de découverte et d'application du cours ii
TPs projet fil rouge ii
TPs projet de synthèse ii
Séquence pédagogique ii
Évaluation iii

I Travaux pratiques d'application du cour et de découverte 1

1 Web statique : Accessibilité, HTML et CSS (TP 1) 3
Exercice 1 : Prise en main de HTML 3
Exercice 2 : Prise en main de css 4
Exercice 3 : Prise en main de javascript 5

2 Web dynamique avec PHP (TP 3) 6
Exercice 1 : Mise en place du serveur 6
Exercice 2 : Prise en main de php 7

3 PHP avancé : gestion de formulaires, variables web globales, objet (TP 5) 8

4 SQL, PHP, PDO, et injections SQL (TP 7) 9

5 XML par l'exemple (TP 9) 10
Exercice 1 : Intégrer un format XML sur le web 10

6 XSL 1 par l'exemple (TP 11) 12
Exercice 1 : Prise en main des fichiers 12
Exercice 2 : Questions dont vous devez comprendre les réponses 12

7 XSL 2 par l'exemple (TP 12) 14
Exercice 1 : Prise en main de SAXON 14
Exercice 2 : Travail à réaliser 14

8 Les API de manipulation XML et XSL (TP 14) 16

9 Web service et architecture REST (TP 16) 17

II Travaux pratiques du projet « fil rouge » 19

10 Cahier des charges des TPs « fil rouge » 21
10.1 Motivations pédagogiques sur le choix du sujet d'étude 21
10.2 Présentation générale du sujet d'étude 21

10.3 Contraintes techniques . . . 22
10.3.1 Accessibilité et versions HTML . . . 22
10.3.2 Environnement d'exécution . . . 22
10.3.3 Base de connaissances . . . 23
10.3.4 Graphisme . . . 23
10.4 Contraintes de réalisation . . . 23
10.4.1 Page d'accueil . . . 23
10.4.2 Autres pages . . . 23
10.5 Fonctionnalités à implémenter (première partie) . . . 23
10.5.1 Consultation des pathologies, critères de filtrage . . . 23
10.5.2 Recherche de pathologies par mot–clef . . . 24
10.5.3 Compte utilisateur . . . 24
10.6 Fonctionnalités à implémenter (deuxième partie) . . . 24
10.6.1 Fiches pathologie . . . 24
10.6.2 Agrégation de flux rss . . . 24
10.6.3 Web service . . . 24
10.6.4 Publication des ressources en REST . . . 24

11 Première ébauche statique du site (TP 2) 25
11.1 Méthodologie . . . 25
11.2 Travail à réaliser . . . 25
11.3 Ne pas oublier... . . . 26

12 Dynamisation de site avec PHP (TP 4) 27
12.1 Travail à réaliser . . . 27

13 Formulaires HTML et PHP Objet (TP 6) 28
13.1 Objectifs . . . 28
13.2 Travail à réaliser . . . 28

14 Bases de données et PHP (TP 8) 30
14.1 Création de la base . . . 30
14.2 Travail à réaliser . . . 30

15 XML (TP 10) 31
15.1 Travail à réaliser . . . 31

16 Transformation de contenu XML par XSL (TP 13) 32

17 Un agrégateur de Flux rss paramétrable (TP 15) 33

18 Webservices : agrégation rss paramétrée et ressources REST (TP 17) 34

19 Préparation de l'évaluation (TP 18) 35

III Projet de fin de module 37

A Bibliographie 39

B Acronymes 41

Première partie

Travaux pratiques d'application du cour et de découverte

1. Web statique : Accessibilité, HTML et CSS (TP 1)

Objectifs

1. savoir évaluer la qualité de documents HTML et CSS ;
2. réaliser des audits simples d'accessibilité en conformité avec les Web Content Accessibility Guidelines (WCAG) de la Web Accessibility Initiative (WAI) ;
3. savoir corriger un document et en améliorer l'accessibilité ;
4. comprendre le fonctionnement de HTML, CSS et javascript ;
5. comprendre les limites de la validation automatisée.

Durée recommandée : 2H maximum

Votre e–bibliothèque

La principale difficulté sur le web, notamment en ce qui concerne les « nouvelles technologies » et la conception de pages web, ne sera pas de trouver de l'information, mais de disposer de sites et d'articles de référence (en plus original qu'openclassroom, même si cette ressource est souvent intéressante !).

Commencez–donc à constituer une bibliothèque de pointeurs que vous jugez intéressant, n'hésitez ni à partager vos découvertes entre vous, ni à demander l'avis de votre enseignant sur telle ou telle ressource !

Exercice 1 : Prise en main de HTML

Récupérez le fichier `tp1/tp1-valider.html`. Ce fichier HTML simple comporte de nombreuses erreurs.

(1.1) Corrigez les erreurs de non–respect des normes Vous pouvez pour cela utiliser un validateur automatique de votre choix.

Nous vous recommandons néanmoins TotalValidator, mais vous pouvez également utiliser le validateur en ligne du World Wide Web Consortium (W3C).

Attention, ne modifiez pas les identifiants et les classes des balises, sauf si nécessaire !

(1.2) Corrigez les erreurs d'accessibilité du document Aidez–vous des messages d'erreurs des validateurs : ils vous indiquent quelles sont les probables erreurs et surtout vous fournissent des liens vers la documentation technique permettant de corriger ce type d'erreurs.

Outils pour le développement web

Firefox comporte de nombreux outils de développement par défaut (voir http://developer.mozilla.org/fr/docs/Outils).
*Pour afficher les consoles : **ctrl+alt+c** (inspecteur) ou **ctrl+F12** (firebug).*
D'autres outils peuvent également être utiles, parmi eux :
***TotalValidator**[a] est un logiciel permettant de valider une page Web tant au niveau du respect des normes que des directives d'accessibilité.*
Il existe également une extension pour Firefox[b] facilitant l'utilisation du logiciel.
Pour l'installer :

1. *téléchargez le logiciel pour votre plateforme et installez–le ;*
2. *téléchargez et installez l'extension pour firefox ;*
3. *paramétrez l'extension dans firefox :*
   (a) *dans le menu* outils, *choisissez « Total Validator Options » ;*
   (b) *renseignez les champs « path » ; l'extension a besoin du répertoire d'installation de TotalValidator et de l'exécutable java ;*
   (c) *indiquez votre niveau d'exigence pour la validation : HTML (auto detect) et accessibility (WCAG 2.0 AAA).*

***WebDeveloper**[c] est une extension très pratique de Firefox, notamment pour la réalisation des feuilles de style CSS.*

a. http://www.totalvalidator.com/
b. http://www.mozilla.org/fr/firefox/
c. http://chrispederick.com/work/web-developer/

Exercice 2 : Prise en main de css

1.2.1 Feuille de style principale

Récupérez maintenant le fichier `tp1/tp1-valider.css` ainsi que le répertoire `tp1/img`.

Placez–les dans le même répertoire que le fichier HTML que vous venez de corriger.

Ajoutez la ligne suivante dans l'entête de fichier html

<link rel="stylesheet" href="tp1-valider.css">

(2.1) Corrigez les erreurs de non–respect des normes. Vous pouvez pour cela utiliser un validateur automatique de votre choix. L'extension *WebDeveloper* en propose plusieurs dont celui du W3C.

(2.2) Que pensez–vous de l'organisation de cette page web ?. En appliquant la méthodologie du cours, ré–organisez les ressources afin de mettre en évidence une séparation entre la partie

vue et la partie *contenu*.

(2.3) Observez le rendu visuel du document. Que constatez-vous ? Corrigez les erreurs d'accessibilité qui n'ont pas été détectées par les validateurs.

1.2.2 Alternate Style Sheets

Nous avons vu en cours qu'il était possible d'adapter le rendu visuel d'une page web à différents dispositifs de sortie (imprimante, mobile, etc.).

(2.4) Ajoutez une feuille de style supplémentaire pour l'impression. Vous veillerez à ce que le document imprimé contienne bien l'ensemble des informations de la page HTML, notamment les adresses des liens.

Le site alsa creation [Alsacreations, 2014]

Le site alsa creation *(http://www.alsacreations.com/ est une des meilleures ressources francophones en matière de design web accessible.*
Vous y retrouverez la plupart des bonnes pratiques du cours et de nombreux tutoriels.
Voici par exemple celui pour les feuilles de style d'impression ([Alsacreations, 2014, Dew]) : http://www.alsacreations.com/tuto/lire/586-feuille-style-css-print-impression.html

(2.5) Ajoutez également une feuille de style pour un rendu sur portable ou en basse résolution.

Encore le site alsa creation...

...et ce ne sera pas la dernière fois que nous y ferons référence !
Pour vous aider, voici un excellent article ([Alsacreations, 2014, Raphael Goetter]) sur le sujet plein de bonnes astuces : http://www.alsacreations.com/astuce/lire/1177-une-feuille-de-styles-de-base-pour-le-web-mobile.html.

Exercice 3 : Prise en main de javascript

La page proposée contient des instructions javascript.

(3.1) Comprenez ce que fait le script. Est-il bien placé ? Corrigez–le et essayez également de le modifier et de l'appliquer à d'autres éléments de la page.

(3.2) Le script est écrit directement dans la page : remplacez-le par un appel à un fichier externe comme vu en cours

2. Web dynamique avec PHP (TP 3)

L'objectif de ce TP est de vous permettre de mettre en place un serveur web et de prendre en main PHP.

Durée : 1h maximum

Exercice 1 : Mise en place du serveur

Afin de disposer de tous les droits d'administration, vous utiliserez des machines virtuelles proposées par **virtualbox** sur les machines TPs.

Vous pouvez également travailler sur vos propres machines si vous préférez.

Raccourci pour ouvrir un terminal

ctrl* + *alt* +*t *(fonctionne sous la plupart des distributions linux)*

2.1.1 Installation des paquets

Vous pouvez consulter la procédure **ligne de commande** dans la documentation ubuntu : http://doc.ubuntu-fr.org/lamp.

Installation des paquets

sudo apt-get install *apache2 php5 mysql-server libapache2-mod-php5 php5-mysql phpmyadmin*

2.1.2 Tests

1. vérifiez dans un navigateur web que le serveur web fonctionne à l'adresse http://localhost ou à l'adresse 127.0.0.1 ;
2. regardez[1] puis copiez le fichier tp3/phptest.php sur le répertoire du serveur (/var/www/ normalement) ;
3. vérifiez que PHP fonctionne en consultant l'adresse http://localhost/phptest.php ;
4. regardez si phpmyadmin fonctionne correctement : http://localhost/phpmyadmin.

1. prenez l'habitude de regarder la documentation PHP sur http://www.php.net lorsque vous utilisez une nouvelle fonctionnalité

2.1.3 Configuration des messages d'erreurs

Copiez le fichier `tp3/erreursphp.php` sur le serveur et essayez d'y accéder.

Si votre serveur est en mode production, vous devriez avoir une page blanche, ce qui rend difficile le debuggage.

Suivant le type de machine sur laquelle tourne le serveur (développement, production), le fichier `/etc/php5/apache2/php.ini` n'est pas configuré de la même manière.

Réglage des messages

En général, des configurations par défaut sont proposées dans le répertoire `/usr/share/php5/`.
Comparez les fichiers `/usr/share/php5/php.ini-development` et `/usr/share/php5/php.ini-production`.
*Vous pouvez utilisez le comparateur **meld** sous linux (**sudo apt-get install** meld).*

Écrasez le fichier `/etc/php5/apache2/php.ini` en utilisant la version de développement proposée.

Redémarrez le serveur et consultez de nouveau l'adresse `http://localhost/erreursphp.php`. Qu'en concluez–vous ? En quoi cela est–il inquiétant ?

Pour redémarrer proprement le serveur apache

***sudo service** apache2 restart*

Exercice 2 : Prise en main de php

L'objectif de l'activité est de se familiariser avec PHP.

(2.1) Récupérer le répertoire `tp3/exo` et copiez le contenu sur le serveur web.

(2.2) Que fait le script exo.php ? Quels problèmes d'organisation et de sécurité pose-t-il ?

(2.3) Proposez les améliorations nécessaires (droits, structuration) pour rendre ce script plus sécurisé et réutilisable

(2.4) En utilisant phpdocumentor[documentor, 2014], produisez la documentation de ce script

Installation de phpdocumentor

***sudo apt-get install** php5-dev php-pear*
***sudo pear channel-discover** pear.phpdoc.org*
***sudo pear install** phpdoc/phpDocumentor*

3. PHP avancé : gestion de formulaires, variables web globales, objet (TP 5)

Pour simplifier, ce TP est fusionné en 5ETI avec le tp fil rouge Formulaires HTML et PHP Objet (TP 6).

4. SQL, PHP, PDO, et injections SQL (TP 7)

Pour simplifier, ce TP est fusionné en 5ETI avec le tp fil rouge Bases de données et PHP (TP 8).

5. XML par l'exemple (TP 9)

Durée : 30'

Exercice 1 : Intégrer un format XML sur le web

5.1.1 Un format normalisé : MathMl

Regardez le fichier `tp9/maths.xhtml`.

(1.1) Quelles différences observez-vous avec un document XHTML classique ? Essayez d'afficher ce document dans un navigateur différent de firefox. Conclusion ?

(1.2) Intégrez quelques formules mathématiques sur vos pages. Quelles sont les principales difficultés que pose ce format ?

5.1.2 Un format spécifié mais non standard : MusicXML

Lorsque vous utilisez MusicXML, vous devriez avoir un message indiquant qu'il n'existe aucune information de rendu permettant d'afficher le contenu. C'est ce qui se produit avec le fichier `tp9/musique.xml`.

Deux solutions sont possibles :

1. utiliser un plugin spécialisé (voir celui proposé par la société myriad) http ://www.myriad-online.com/en/products/mmplugin.htm

2. implémenter vous-même un rendu (en XSL par exemple) ; nous verrons cette solution dans la 2ème partie du cours

Ressources

Quelques tutoriels bien réalisés :

- *http://www.gchagnon.fr/cours/xml/index.html*, *tutoriel XML* ;
- *Schéma XML : http://www.gchagnon.fr/cours/xml/schema.html* ;
- *http://www.w3schools.com/xml/default.asp (tutoriel)* ;
- *http://www.w3schools.com/dtd/ pour les dtd* ;
- *http://www.w3schools.com/Schema/ pour les schémas xml*[a] ;

Validateurs :

- *document xml : http://www.w3schools.com/xml/xml_validator.asp*
- *schéma : http://www.utilities-online.info/xsdvalidation/*

a. pensez bien à ne PAS réaliser de schemas hiérarchiques

6. XSL 1 par l'exemple (TP 11)

durée : 3 à 4h

L'objectif de ce TP est de prendre en main XSL.

Exercice 1 : Prise en main des fichiers

Vous disposez d'un document au format xml représentant des données d'un livre pouvant contenir des expressions mathématiques (`tp11/document.xml`).

L'objectif est de réaliser la transformation de ce document en document HTML. Le résultat attendu est fourni : `tp11/document.html`.

Il vous est demandé avant toute chose de **comprendre** la logique de XSL, aussi, **une solution vous est proposée** : `tp11/document.xsl`.

(1.1) Affichez le document xml dans un navigateur, puis décommettez la deuxième ligne du fichier et actualisez. Explications ?

Exercice 2 : Questions dont vous devez comprendre les réponses

Voici une liste de questions qui vous permettront de comprendre comment le document a été construit.

Essayez d'abord de répondre par vous–mêmes, sollicitez les intervenants et/ou regardez la solution en cas de blocage.

(2.1) Combien de phrases, de mots et d'expressions mathématiques y-a-t-il dans le document ? Utiliser count(xpath)

(2.2) Afficher une table des matières du document Se servir des expressions xpath, des templates match et éventuellement des fonctions following et following-sibling

(2.3) Combien y-a-t-il d'expressions mathématiques "complexes" (qui utilisent au moins une fraction ou une racine carrée) ? Utiliser une variable, et un for-each

(2.4) Y a-t-il un titre de niveau 1 non suivi de titre de niveau 2 ?

(2.5) Y a-t-il deux titres avec le même nom ? Si c'est le cas, renommer le deuxième titre en ajoutant (#2) .

(2.6) Mettre les titres en majuscule

(2.7) Faites une fonction (template name) qui affiche un message si les lettres grecques Delta ou alpha sont présentes dans une expression . Elle prendra en paramètre un noeud de type "math"

(2.8) Faites un rendu html de tout ça ! Pour les maths, utiliser xsl :copy-of

Et pour la suite ?

Ne vous limitez pas à ce qui est demandé : posez–vous des problèmes, essayez de trouver des solutions, demandez conseil !
Bref, soyez CURIEUX !

Documentation

Voici une sélection de ressources particulièrement utiles :

- *LE site de référence de Dave Pawson en XSL : http://www.dpawson.co.uk/ ; voir plus particulièrement les XSL FAQ ;*
- *Un site pas mal pour débuter : http://www.zvon.org (ou directement http://www.zvon.org/xxl/XSLTutorial/Books/Book1/index.html)*
- *les fonctions très utiles de Priscilla Walmsley (XSL 2) : http://www.functx.com*
- *http://www.mulberrytech.com/xsl/xsl-list/ : LA liste sur xsl*

7. XSL 2 par l'exemple (TP 12)

durée : 3-4h

Objectif : L'objectif de ce TP est de découvrir XSL 2 et d'utiliser l'API java de SAXON pour interpréter des feuilles XSL en version 2.

Exercice 1 : Prise en main de SAXON

SAXON, le parser xsl2 de référence

Saxon est un analyseur/parseur xsl permettant d'exécuter une transformation xsl (http ://www.saxonica.com). Il existe maintenant en version libre (home édition) et non-libre avec des extensions (Professional et enterprise editions).
Il peut être utilisé en ligne de commande, ou intégré directement dans des développements en java ou dotnet (les deux api existent).
Nous allons utiliser l'api java de saxon, en ligne de commande pour ce tp.

Utilisation de SAXON

Récupérer les fichiers tp12/saxon.jar et tp12/saxon-dom.jar.
Exécuter votre feuille de style en ligne de commande :

```
java -jar saxon.jar -o sortie.html -xsl:document.xsl document.xml
```

Exercice 2 : Travail à réaliser

(2.1) Transformez votre feuille xsl en xsl 2 Que se passe-t-il lorsque vous essayez d'obtenir un rendu dans un navigateur web ? Pourquoi ?

Changer la version d'une feuille XSL

Il suffit de modifier la déclaration :

```
<xsl:stylesheet version="1.0"
    xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
```

en

```
<xsl:stylesheet version="2.0"
    xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
```

(2.2) Utilisez saxon en ligne de commande pour interpréter `tp12/document.xml` . Conclusion ?

(2.3) Ajoutez un type à vos variables Exemple :

```
<xsl:variable name='var' select='local-name(.)'/>
<xsl:variable name='var' as='xs:string' select='local-name(.)'/>
```

Typage

Pensez à ajouter un namespace pour le nommage des variables dans la première balise :

```
<xsl:stylesheet version="2.0"
    xmlns:xsl='http://www.w3.org/1999/XSL/Transform'
    xmlns:xs='http://www.w3.org/2001/XMLSchema'>
```

(2.4) Réalisez une fonction estComplexe qui prend en paramètre un noeud math et qui renvoie un booléen (vrai si l'expression contient des racines ou des fractions)

Namespaces et fonctions

Une bonne pratique consiste à ajouter un namespaces personnalisé pour vos fonctions, afin de les différencier des fonctions standards.

(2.5) Utilisez votre fonction par exemple :

```
<xsl:template match select="m:math">
    <xsl:choose>
        <xsl:when test="mesFonctions:estComplexe(.)">
            <xsl:text>Trop complexe</xsl:text>
        </xsl:when>
        <xsl:otherwise>
            <xsl:copy-of select="."/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>
```

(2.6) En vous inspirant des fonctions écrites par Priscilla Walmsley (`http://www.functx.com`), réalisez des fonctions utilisant les expressions régulières, les manipulations de chaînes et les manipulations d'arbres .

Utilisez les syntaxes xslt et non xquery (vous avez les exemples de codes).

(2.7) Intégrez et testez vos fonctions

(2.8) Pour aller plus loin... Il est dit de xsl2 (bientôt de xsl3) que sa maîtrise est **un art**. Si vous souhaitez progresser dans la maîtrise de ce langage, lire et comprendre les FAQ de Dave Pawson est un bon début !

8. Les API de manipulation XML et XSL (TP 14)

9. Web service et architecture REST (TP 16)

Deuxième partie

Travaux pratiques du projet « fil rouge »

10. Cahier des charges des TPs « fil rouge »

10.1 Motivations pédagogiques sur le choix du sujet d'étude

Le sujet d'étude que nous vous proposons pourra vous sembler étrange : il s'agira en effet de travailler pour une association d'acupuncteurs en médecine traditionnelle chinoise.

Dans votre vie professionnelle, il vous sera très souvent demandé en tant qu'informaticien de proposer des solutions pour des sujets d'étude dont vous ignorez complètement les périmètres. Aussi, il nous a semblé pertinent de vous confronter à un problème dont le sujet d'étude sera pour la très grande majorité (voire la totalité) d'entre vous complètement nouveau.

Sachez toutefois que les données que vous utiliserez sont des données réelles.

Certains aspects de la base de connaissance ont toutefois été simplifiés, afin de ne pas complexifier outre mesure la prise en main du contexte d'étude.

Un autre intérêt est que ce problème a réellement été posé : même si le cahier des charges a été (très) allégé, il correspond à une vraie demande de la part de vrais acupuncteurs.

Enfin, que l'on y croit ou non, de nombreux symptômes ainsi que leurs associations sont plutôt amusants voire parfois déroutants. Cela vous encouragera à vérifier l'exhaustivité de vos requêtes... et sera peut–être un moyen de vous faire sourire en TP et/ou de détendre l'atmosphère !

10.2 Présentation générale du sujet d'étude

L'association des acupuncteurs soucieux de l'accessibilité (AAA...) vous a sollicité pour la conception et la réalisation d'une plateforme en ligne dont les principales fonctionnalités seraient ·

1. de disposer d'un service en ligne leur permettant de consulter la liste des symptômes des principales pathologies en acupuncture (voir table 10.1) ;
2. de pouvoir n'afficher que certaines des pathologies en fonction de différents critères (type de pathologie, choix des méridiens[1], etc., voir la table 10.2) ;
3. de rechercher les pathologies comportant certains symptômes.

1. un méridien en acupuncture est un chemin composé de différentes branches lié à un organe, à un viscère ou à un « merveilleux vaisseau ». Sa partie superficielle contient les fameux « points » d'acupuncture que les praticiens ponctureront, chauffent ou massent. Les différentes branches des chemins n'empruntent pas forcément des trajets anatomiques ou physiologiques, ce qui est l'un des mystères que la médecine occidentale n'a pas encore pu expliquer scientifiquement.

Catégorie de pathologie	Caractéristiques possibles	exemple
Pathologies de méridien	— interne — externe	— méridien du poumon interne — méridien du rein externe
Pathologies d'organe/viscère (tsang/fu)	— plein — vide — chaud — froid	— poumon vide — poumon froid — rate vide et froid — foie chaud
Pathologies des tendino-musculaires (jing jin)	néant	jing jin du rein
Pathologie des branches (voies luo)	— vide — plein	— voie luo du poumon vide — voie luo du rein pleine
Pathologies des merveilleux vaisseaux	néant	pathologie du Ren Mai

TABLE 10.1 – Types de pathologies en acupuncture. Il existe d'autres types de pathologies qui ne seront pas utilisées dans l'application, et qui ne figurent pas dans la base de connaissances fournie.

Critère	valeurs possibles	exemple
Méridien	Nom du méridien (20 méridiens en tout)	Poumon, Ren Mai, rein, Yanq Qiao Mai
Type de pathologie	méridien, organe/viscère, luo, merveilleux vaisseaux, jing jin	sélectionner les pathologies voie luo
Caractéristiques	plein, chaud, vide, froid, interne, externe	sélectionner les pathologies de vide

TABLE 10.2 – Critères pour les filtres. Les filtres peuvent se combiner : sélectionner les pathologies de méridien, interne, pour poumon et foie

10.3 Contraintes techniques

10.3.1 Accessibilité et versions HTML

L'ensemble du site respectera les normes d'accessibilité de niveau AAA.

Il sera écrit en XHTML1.1 ou HTML 5.

De plus, toute page du site devra être accessible de la page d'accueil en 3 interactions au maximum (clic, raccourci, etc.).

La navigation devra être possible sans souris ou sans clavier.

10.3.2 Environnement d'exécution

Le site devra être affichable et utilisable à partir d'une version de Firefox de moins de deux ans.

La partie dynamique (serveur) devra fonctionner sur une architecture Linux Apache MySQL PHP (LAMP).

Toutefois, vous restez libres des outils et environnements de développement que vous souhaitez utiliser pour arriver à ce résultat.

10.3.3 Base de connaissances

La base de connaissances vous est fournie sous la forme d'un script SQL de génération de base de données.

Vous ne pouvez en aucun cas modifier les tables de cette base, mais vous êtes libres d'ajouter de nouvelles tables si vous en avez besoin.

La base de connaissance a été réalisée à partir du Vademecum d'acupuncture traditionnelle de Jean Motte [Motte, 2014].

10.3.4 Graphisme

La partie graphique doit avant tout éviter les erreurs d'accessibilité et d'ergonomie.

Il ne vous est pas demandé de faire un site esthétiquement joli, mais simple et fonctionnel.

S'il vous reste du temps en fin de module, vous pourrez améliorer cet aspect à ce moment–là.

10.4 Contraintes de réalisation

Toutes les pages doivent être consultables en basse résolution et disposer d'une mise en page spéciale pour l'impression.

10.4.1 Page d'accueil

La page d'accueil doit disposer d'un menu et d'un formulaire d'identification pour les utilisateurs.

Elle contiendra également une zone permettant d'afficher des informations conjoncturelles (ce sera dans la deuxième partie du TP cette zone qui sera alimentée par les flux Really Simple Syndication (RSS).

10.4.2 Autres pages

Vous êtes totalement libre de l'organisation des autres pages (nombre, architecture, etc...).

Vous réaliserez cependant une page d'information présentant les développements que vous avez réalisés, vos sources, les auteurs, ainsi que les ressources bibliographiques et la webographie que vous avez utilisé.

10.5 Fonctionnalités à implémenter (première partie)

10.5.1 Consultation des pathologies, critères de filtrage

Vous réaliserez une page permettant d'afficher la liste des pathologies.

Ces pathologies pourront faire l'objet de filtrage comme indiqué en introduction.

Vous vous efforcerez d'optimiser au maximum votre solution en limitant autant que possible les requêtes sur le serveur MySQL.

Vous êtes totalement libres dans le choix de l'interface graphique.

10.5.2 Recherche de pathologies par mot–clef

Vous implémenterez une fonctionnalité de recherche de pathologie par mot–clef.

Les mot–clefs sont associés aux symptôme dans la base de connaissance.

Cette fonctionnalité ne sera accessible qu'aux utilisateurs authentifiés.

10.5.3 Compte utilisateur

Vous proposerez un système de gestion des utilisateurs (inscription, login, session, etc.).

Un utilisateur connecté aura la possibilité d'accéder à la fonctionnalité de recherche de pathologies par mot–clef (cf. 10.5.2).

10.6 Fonctionnalités à implémenter (deuxième partie)

10.6.1 Fiches pathologie

A l'aide de XSL, vous réaliserez à partir du format XML que vous avez défini des fiches d'information sur les pathologies.

Chaque pathologie devra avoir sa propre fiche.

10.6.2 Agrégation de flux rss

La page d'accueil permettra à un utilisateur authentifié d'afficher les flux RSS auxquels il aura souhaité s'abonner.

10.6.3 Web service

Vous implémenterez la recherche par mot–clef sous la forme d'un web service.

10.6.4 Publication des ressources en REST

En utilisant l'architecture REST, vous publierez les fiches des pathologies sous formes de ressources web.

11. Première ébauche statique du site (TP 2)

Durée conseillée : 4h

11.1 Méthodologie

La méthodologie a été présentée en cours. Vous pouvez si vous le souhaitez l'adapter à vos besoins, en imaginer une qui vous parle plus, etc.

Quoiqu'il en soit, les exigences resteront les mêmes !

Les informations techniques dont vous avez besoin sont sur internet...

N'hésitez cependant pas à faire valider ce que vous avez trouvé et **maintenez votre webographie**.

11.2 Travail à réaliser

(2.1) Constituez–vous en binôme Il est interdit de travailler seul sur le projet fil rouge, sauf contre indication du responsable du module et raison valable (absences aux premiers TP principalement).

En cas de nombre impairs d'étudiants dans le groupe, seul un groupe sera autorisé à travailler à trois.

(2.2) Lisez le cahier des charges ! Vous pouvez vous arrêter à la liste des premières fonctionnalités **inclue**.

(2.3) En appliquant la méthode vue en cours, ou une méthode équivalente, réalisez les maquettes « papier–crayon » des pages du site. Vous penserez notamment à prévoir les réactions aux événements, et prendrez soin de découvrir les fonctionnalités nécessaires mais non exprimées directement par le client [1]

1. Ce point, très fréquent, doit normalement faire l'objet d'allers–retours du cahier des charge entre le demandeur et le réalisateur, puis de validations par le client. Dans le cadre de notre TP, vous prendrez les initiatives qui vous semblent pertinentes pour que les fonctionnalités souhaitées par le client soient exhaustivement traitées.

(2.4) Réalisez les maquettes statiques (en HTML et CSS) de vos pages. Vous réaliserez notamment les formulaires, qui seront pour le moment passifs (il n'y aura pas d'action côté serveur lorsqu'ils seront validés).

Vous pouvez cependant déjà prévoir et réaliser des contrôles javascript.

(2.5) Associez les feuilles css demandées

Attention aux pertes de temps !

Nous ne vous demandons pas de maîtriser en quelques heures HTML et CSS !
Le temps passe souvent vite quand on cherche à régler de petits détails. C'est formateur, mais il ne faut pas oublier que vous êtes en temps limité.
Profitez donc au maximum de la présence des enseignants qui pourront vous dépanner ! Cherchez d'abord une solution, et si vous ne trouvez rien au bout de 15 minutes, sollicitez–les !

11.3 Ne pas oublier...

— de citer vos sources
— de vérifier que vous avez le droit d'utiliser telle ou telle ressources
— de bien respecter TOUS les termes des licences (citation explicite, etc.)
— de maintenir votre webographie...

12. Dynamisation de site avec PHP (TP 4)

L'objectif de ce TP est de rendre votre site dynamique en utilisant PHP.

Durée conseillée : 2h

Les indications seront données dans le cas d'une architecture LAMP uniquement.

Faire du vrai Modèle Vue Contrôleur (MVC) avec les templates (ou du meilleur en tous cas !)

Si vous vous sentez à l'aise, vous devriez essayer d'utiliser un système de templating pour votre site.

Voici un bon tutoriel du site openclassrooms [Openclassrooms, 1999, Torejy] pour le moteur de template smarty [New Digital Group, 1999] : `http://fr.openclassrooms.com/informatique/cours/un-moteur-de-template-smarty`

12.1 Travail à réaliser

(1.1) Mettez en place une architecture dynamique pour votre site Vous veillerez à anticiper l'organisation physique des scripts et des ressources sur votre serveur (arborescence, droits, etc.).

(1.2) Réalisez les différents scripts php dont vous avez besoin Vous veillerez à vous approcher le plus possible d'une architecture MVC.

(1.3) Maintenez l'accessibilité de votre site...

(1.4) Générez (et maintenez par la suite) la documentation de vos scripts PHP

13. Formulaires HTML et PHP Objet (TP 6)

Version simplifiée : récupérerez l'archive du TP qui contient tout le matériel nécessaire.

13.1 Objectifs

L'objectif est d'exploiter les données du formulaire fourni pour :

- permettre la construction d'un objet Reservation
- effectuer les traitements à l'aide de méthodes de cette classe (calcul du prix, vérification du code promotionnel)
- renvoyer une réponse à l'utilisateur (une page HTML dynamique indiquant si la réservation a bien eu lieu, et sinon pourquoi

13.2 Travail à réaliser

(2.1) regardez les exemples fournis

(2.2) Créez la classe Reservation (patron de base fourni).

(2.3) Écrivez une fonction, en dehors de la classe Reservation, dont le rôle sera de collecter et de vérifier les données du formulaire . Cette fonction reverra un tableau de données contenant les données à utiliser pour construire une instance de *Reservation*, ou *null* en cas de données erronées.

(2.4) Écrivez un fonction, toujours en dehors de la classe Reservation, qui permet de donner un rendu d'une instance de Reservation .

(2.5) Assemblez le tout ! . Vous pouvez avoir un algorithme qui ressemble à ça :

Version basique de l'algorithme

```php
<?php
    include('lib/control/verifications.php');
    include('lib/model/classes/clsReservation.php');
    include('lib/view/render/renderResa.php');

    $data = verif($_POST);
    if($data==null){
        renderResaNOK();
    }
    else{
        $resa = new Reservation($data);
        $resa->calculePrix();//verifiera aussi le code
           promotionnel
        renderResa($resa);
    }
?>
```

(2.6) BONUS : réalisez en javascript une requête HTTP permettant de calculer directement le prix de la réservation sans avoir à valider le formulaire.

14. Bases de données et PHP (TP 8)

L'objectif du TP est d'utiliser la base de donnée mysql fournie par le client.

14.1 Création de la base

- créez une base de donnée avec php my admin (interclassement : utf8mb4_unicode_ci)
- créez un utilisateur avec les droits d'écriture et un utilisateur avec les droits de lecture seulement
- importez la base de donnée avec le script `tp8/bdAcu.sql`

14.2 Travail à réaliser

Vous arrivez à la fin de la première partie du cahier des charges.

Pensez à réaliser :

- une classe pour représenter une pathologie
- des requêtes préparées
- des connexions sécurisées
- vous protéger contre les principales attaques « faciles », notamment l'injection SQL...

Ressources

sur les bases de données et PDO : http://studio.jacksay.com/tutoriaux/php/connection-mysql-avec-pdo

protection contre injection : http://php.net/manual/fr/pdo.quote.php

15. XML (TP 10)

L'objectif de cette partie est de réaliser une modélisation en XML d'un format de représentation d'une pathologie.

15.1 Travail à réaliser

(1.1) Réaliser un modèle représentatif d'une pathologie

(1.2) Réaliser la dtd de ce modèle

(1.3) Réaliser le schema de ce modèle Vous devez être en mesure de justifier vos différents choix.

16. Transformation de contenu XML par XSL (TP 13)

17. Un agrégateur de Flux rss paramétrable (TP 15)

18. Webservices : agrégation rss paramétrée et ressources REST (TP 17)

19. Préparation de l'évaluation (TP 18)

Troisième partie

Projet de fin de module

A. Bibliographie

[Alsacreations, 2014] Alsacreations (2014). Alsacreations, communauté d'apprentissage des standards du web. http://www.http://www.alsacreations.com/. consulté en septembre 2014.

[documentor, 2014] documentor, P. (2014). Php documentor, auto-documentation tool for php. http://www.phpdoc.org/. consulté en septembre 2014.

[Motte, 2014] Motte, J. (2014). *Vade-mecum d'acupuncture traditionnelle*. Guy Trédaniel éditeur, 2ème édition edition.

[New Digital Group, 1999] New Digital Group, I. (1999). Smarty, template engine. http://www.smarty.net/. consulté en septembre 2014.

[Openclassrooms, 1999] Openclassrooms (1999). Open class rooms, cours et tutoriels en ligne. http://fr.openclassrooms.com/. consulté en septembre 2014.

B. Acronymes

API Application Programming Interface. 1

CSS Cascading Style Sheets. 1, 5, 7, 8

DOM Document Object Model. 1

HTML HyperText Markup Language. 1, 5, 7, 8

PHP PHP: Hypertext Preprocessor, anciennement Personal Home Page. 1

QCM Questionnaire à Choix Multiples. 4

REST REpresentational State Transfer. 1

SAX Simple API for XML. 1

SQL Structured Query Language. 1

TLI Technologies et langages de l'Internet. 1

WAI Web Accessibility Initiative. 7

WCAG Web Content Accessibility Guidelines. 7, 8

XML eXtensible Markup Language. 1

XSL eXtended Stylesheet Language. 1