

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 les templates, AJAX, 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 demandé 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, les TPs de découverte d'application du cours sont destinés à vous familiariser avec les bases des langages et à vous apprendre les bonnes pratiques liées à leur utilisation.

Chaque TP donne un objectif minimal à atteindre, ainsi que quelques objectifs secondaires pour les plus rapides.

Vous devrez également mettre en pratique ce que vous avez appris en réalisant un site web, dont vous choisirez vous-mêmes le sujet.

L'enchaînement des TPs se fera directement en lien avec le cours.

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 directs, 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.

Mise en pratique

Vous réaliserez un site web de votre choix. L'objectif est de démontrer que vous maîtrisez les concepts vu en cours et pendant les TPs de découverte.

Vous aurez plus de liberté quant à la mise en œuvre des solutions, n'hésitez donc 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 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 donc 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

Le planning de la séquence est donné sur la figure **1**

TLI 1 : bases des langages et technologies de l'Internet																
		S1	S2	S3	Séance 4	S5	Séance 6	Séance 7	S8	Séance 9	S10	Séance 11	Séance 12			
		C1	C2	C3	TP1	C4	TP2	TP3	C5	TP4	C6	TP5	TP6			
		Intro HTML WAI	CSS JS 1	JS 2 DOM	HTML Accessib. Bonne Prat.	PHP Base	CSS JS	JS DOM	XML	Formulaires PHP1	BD	HttpRequest Intro AJAX	Stockage données et connaissances (SQL et XML)			
Cours		2	2	2		2			2		2					
TP					4		4	4		4		4	4			
4ETI	A	Mercredi 8:00-12:15	BM	BM	BM	BM	MS	DC	BM	MS	DC	FP	MS	BM		
	B	Vendredi 13:30-17:45					25 nov.			2 déc.	9 déc.		MS	DC	16 déc.	6 janv.
	C	Mardi 13:30-17:45	5/11	12/11	16/11	26/11	DC	MS	DC	BM	MS	DC	DC	BM	MS	DC
	D	Lundi 8:00-12:15	10-12h	10-12h	10-12h	8h-10h	27 nov.		4 déc.	11 déc.	18 déc.	8 janv.	15 janv.	TB	CP	TB
							TB	CP	TB	CP	TB	CP	TB	CP	TB	CP
							24 nov.		1 déc.	8 déc.	15 déc.	8h-10h	5 janv.	12 janv.		
							MS	BM	BM	MS	BM	MS	MS	BM	BM	MS
							23 nov.		30 nov.	7 déc.	14 déc.		4 janv.	11 janv.		

SEANCE TP	SEANCE COURS
-----------	--------------

BM	Bruno Mascret
FP	Françoise Perrin
DC	Dino Cosmas
MS	Mohammed Sallami
CP	Charles Perrin
TB	Timothe Bordiga

FIGURE 1 – Planning des séances – TLI Tronc commun

Évaluation

Il y a aura 3 évaluations pour le module complet :

- une évaluation de TP pour la première partie du module, sous forme de QCMs (4 ou 5 en début de cours ou de TP) ;
- une évaluation de TP pour la deuxième partie du module, sous forme de QCMs (2 ou 3 en début de cours ou de TP) ;
- 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	i
TPs mise en pratique	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, contenu en HTML (TP 1)	3
Exercice 1 : Prise en main de HTML	3
Exercice 2 : A la recherche des balises perdues...	4
Exercice 3 : Audit de site	5
Exercice 4 : A vous de jouer !!!	5
2 Web statique : Mise en forme de document avec CSS (TP 2)	6
Exercice 1 : Prise en main de css	6
Exercice 2 : Audit de site	8
Exercice 3 : A vous de jouer !!!	8
3 Web pseudo–statique : Formulaires HTML5, javascript et ARIA (TP 3)	9
Exercice 1 : Prise en main de javascript	9
Exercice 2 : Accessibilité avancée avec ARIA	10
Exercice 3 : C'est à vous!	10
4 Vers le (vrai) dynamisme avec JS et ajax (TP 4)	11
Exercice 1 : Prise en main d'HttpRequest et d'ajax	11
Exercice 2 : Passage de paramètres à une requête	12
Exercice 3 : C'est à vous!	13
5 Web dynamique avec PHP et SQL (TP 5)	14
Exercice 1 : Mise en place du serveur	14
Exercice 2 : Prise en main de SQL	16
Exercice 3 : Travail à réaliser	17
Exercice 4 : C'est à vous!	23
6 Web dynamique avec PHP, XML, SQL et les formulaires (TP 6)	24
Exercice 1 : Découverte de XML	24
Exercice 2 : C'est à vous!	26

II Travaux pratiques du projet « fil rouge »	27
7 Cahier des charges des TPs « fil rouge »	29
7.1 Motivations pédagogiques sur le choix du sujet d'étude	29
7.2 Présentation générale du sujet d'étude	29
7.3 Contraintes techniques	30
7.3.1 Accessibilité et versions HTML	30
7.3.2 Environnement d'exécution	30
7.3.3 Base de connaissances	31
7.3.4 Graphisme	31
7.4 Contraintes de réalisation	31
7.4.1 Page d'accueil	31
7.4.2 Autres pages	31
7.5 Fonctionnalités à implémenter (première partie ou tronc commun)	32
7.5.1 Consultation des pathologies, critères de filtrage	32
7.5.2 Recherche de pathologies par mot-clé	32
7.5.3 Compte utilisateur	32
8 Première ébauche statique du site	33
8.1 Méthodologie	33
8.2 Travail à réaliser	33
8.3 Ne pas oublier...	34
9 Projet : Dynamisation de site avec PHP	35
9.1 Travail à réaliser	35
9.2 Formulaires HTML et PHP Objet	36
9.2.1 Objectifs	36
9.3 Bases de données et PHP	36
9.3.1 Création de la base	36
9.3.2 La librairie de référence pour les BD en PHP : PDO	36
10 XML et XSD	38
10.1 Travail à réaliser	38
11 Webservices et ressources REST	39
11.1 Objectifs	39
11.2 Prise en main : réalisation d'un WS simple	39
11.2.1 Objectifs	39
11.2.2 Première technique : URL rewriting côté serveur web	40
11.2.2.1 Configuration du serveur Apache (mod_rewrite)	40
11.2.3 Deuxième technique : URL rewriting via PHP	40
A Bibliographie	41
B Acronymes	43

Première partie

Travaux pratiques d'application du cour et de découverte

1. Web statique : Accessibilité, contenu en HTML (TP 1)

Objectifs

1. savoir évaluer la qualité de documents **HTML** ;
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** ;
5. comprendre les limites de la validation automatisée.



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**

Durée recommandée : 1H30 maximum

Récupérez le fichier [tp1/tp1-valider.html](#). Ce fichier **HTML** simple comporte de nombreuses erreurs (c'est peu dire...).

(1.1) Corrigez les erreurs de non–respect des normes Vous pouvez pour cela utiliser un validateur automatique de votre choix qui vous indiquera les erreurs de codage et les erreurs d'accessibilité.

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.

(1.3) Le menu du document a été (mal en plus) réalisé à l'aide d'une table. Qu'en pensez-vous ?



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 : A la recherche des balises perdues...

Durée recommandée : 30 minutes maximum

De nombreuses balises HTML ne sont pas toujours connues ou utilisées à bon escient...

Cherchez à améliorer le document HTML fourni avec les balises suivantes :

- nav
- footer et header
- abbr
- em et strong
- acronym
- l'attribut *alt* des images
- l'attribut tabindex
- l'attribut accesskey

Exercice 3 : Audit de site

Durée recommandée : 30 minutes maximum

Que pensez-vous des sites web des ministères français <http://www.social-sante.gouv.fr/> et belge <http://www.handicap.fgov.be/fr> responsables entre autres de l'accessibilité et des personnes en situation de handicap ?

Que proposeriez-vous si on vous demander de réaliser un audit de ces sites ?

Exercice 4 : A vous de jouer !!!

Durée recommandée : 1h30 maximum

Concevez et programmez la structure et le contenu d'une page web.

Si vous êtes à l'aise, écrivez plusieurs pages.

Essayez d'utiliser le plus de balises différentes !

2. Web statique : Mise en forme de document avec CSS (TP 2)

Objectifs

1. savoir évaluer la qualité de documents **HTML** et **CSS** ;
2. réaliser des audits simples d'accessibilité en conformité avec les **WCAG** de la **WAI** ;
3. savoir corriger un document sur la structure/le contenu (**Web statique : Accessibilité, contenu en HTML (TP 1)**) et la mise en forme ; en améliorer l'accessibilité ;
4. comprendre le fonctionnement de **CSS** ;
5. savoir adapter la mise en forme d'un document en fonction du contexte (responsive design) ;
6. comprendre les limites de la validation automatisée.

Exercice 1 : Prise en main de css

2.1.1 Feuille de style principale

Durée recommandée : 1H maximum

Récupérez le fichier [tp2/tp2-valider.css](#).

Placez-les dans le même répertoire que le fichier **HTML** que vous avez corrigé lors du TP **Web statique : Accessibilité, contenu en HTML (TP 1)**.

 **Ajoutez la ligne suivante dans l'entête de fichier html**

```
<link rel="stylesheet" href="tp2-valider.css">
```

(1.1) Que permet de faire cette ligne de code ? Que se passe-t-il lorsque vous actualisez la page **HTML ?**

(1.2) 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** ([?], <https://jigsaw.w3.org/css-validator>).

Soyez exigeant lors de la validation !!!

Pour cela, régler votre validateur **CSS** pour qu'au minimum :

- il utilise CSS 3;
- il teste tous les média;
- il vous affiche les avertissements;
- il vous prévienne lorsque vous utilisez des extensions propriétaires de CSS (efficaces uniquement avec certains navigateurs);

(1.3) Que pensez-vous de l'organisation de cette page web ? En appliquant la méthodologie du cours, réorganisez les ressources afin de mettre en évidence une séparation entre la partie *vue* et la partie *contenu*.

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

2.1.2 Alternate Style Sheets

Durée recommandée : 1H maximum

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.).

(1.5) Commencez par lire cet article pour vous familiariser avec la problématique : <http://www.alsacreations.com/article/lire/1615-cest-quoi-le-responsive-web-design.html>

(1.6) 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>

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



Encore le site alsacreation...

...et ce ne sera pas la dernière fois que nous y ferons référence !

Pour vous aider, voici un excellent article ([Alsacreation, 2014, Raphael Goetter]) sur le sujet plein de bonnes astuces : <http://www.alsacreation.com/astuce/lire/1177-une-feuille-de-styles-de-base-pour-le-web-mobile.html>.

Exercice 2 : Audit de site

Durée recommandée : 30 minutes maximum

Étudiez de nouveau les sites web des ministères français <http://www.social-sante.gouv.fr/> et belge <http://www.handicap.fgov.be/fr>¹ en vous focalisant sur la mise en forme et les capacités d'adaptation contextuelle.

Que proposeriez-vous si on vous demander de réaliser un audit de ces sites ?

Exercice 3 : A vous de jouer !!!

Durée recommandée : 1h30 maximum

Concevez et programmez la mise en forme de votre page/site web commencé(e) lors du précédent TP.

Décidez quelle technique accessible vous allez utiliser (responsive ou adaptive).

Essayez d'utiliser le plus de propriétés différentes !

Commencez par vous occuper des blocs principaux, puis du contenu de ces blocs (du plus grand au plus petit conteneur).

1. toujours responsables entre autres de l'accessibilité et des personnes en situation de handicap...

3. Web pseudo–statique : Formulaires HTML5, javascript et ARIA (TP 3)

Objectifs

1. savoir utiliser des script et des fonctions javascript simples ;
2. utiliser ARIA pour rendre accessible javascript ;
3. réaliser des formulaires html5 accessibles ;

Exercice 1 : Prise en main de javascript

Durée recommandée : 1h maximum

Récupérez l'archive du TP et regardez les documents fournis.

La page proposée contient des instructions javascript.

(1.1) Comprenez ce que fait le script. Est-il correctement déclaré ?

(1.2) Le script pourrait être écrit directement dans la page : en quoi cela poserait-il problème

(1.3) Améliorez l'initialisation du formulaire

En effet, au chargement de la page, la somme calculée est égale à 99€.

Quel événement (voir le cours) permettrait d'appeler la fonction *getTotal* lorsque le formulaire est entièrement chargé ? Sur quelle(s) balise(s) cet événement est-il utilisable ?

Réalisez les modifications nécessaires.

(1.4) Vérifier que la date de départ est bien postérieure à la date d'arrivée.

(1.5) Modifiez le formulaire en proposant des repas (midi et/ou soir), et en adaptant le tarif.

Pour les plus rapides...

S'il vous reste du temps uniquement :

(1.6) Ajoutez un colorchooser qui permettra de changer la couleur de fond du formulaire.

Exercice 2 : Accessibilité avancée avec ARIA

Durée recommandée : 1h maximum

Nous avons évoqué en cours l'accessibilité de javascript avec ARIA.

(2.1) Lisez l'article suivant (si vous avez été en cours, vous pouvez commencer directement à la section ARIA). C'est un classique de l'accessibilité.

<http://www.lesintegristes.net/2008/12/09/introduction-a-wai-aria-traduction/>.

(2.2) Utilisez ARIA pour indiquez les zones de modification , et plus particulièrement la zone d'affichage du montant.

(2.3) Placez les attributs role sur votre page personnelle

Exercice 3 : C'est à vous !

Durée recommandée : 2h maximum

Utilisez ARIA pour améliorer l'accessibilité de votre page personnelle.

Incluez un formulaire HTML5 dont vous vérifierez la cohérence à l'aide de javascript.

Incluez des scripts de votre choix (vous pouvez en chercher sur le Net).

4. Vers le (vrai) dynamisme avec JS et ajax (TP 4)

Objectifs

1. comprendre le principe d'ajax et des requêtes distantes ;
2. savoir utiliser javascript pour dynamiser une page web ;
3. utiliser ARIA pour rendre accessible ajax ;
4. commencer à prendre plus d'autonomie dans la recherche de solutions ;
5. commencer à devenir critique sur ce que vous trouvez sur internet.

Principe du TP

Nous allons utiliser javascript afin d'aller récupérer une valeur disponible sur une ressource (**Uniform Resource Locator (URL)**) distante (localisée sur une machine différente).

Nous verrons lors du prochain TP comment programmer une telle ressource, mais ce n'est pas l'objectif de cette séance.

Le principe est représenté sur la figure 4.1 :

Notez tout de suite qu'à aucun moment, il n'y a besoin d'actualiser la page HTML dans le navigateur ! C'est tout l'intérêt de cette forme de dynamisme.

Les actions sont exécutées en arrière plan, invisibles à l'utilisateur.

Exercice 1 : Prise en main d'HttpRequest et d'ajax

durée maximum conseillée : 1h

Nous allons ajouter une fonctionnalité au formulaire du TP **Web dynamique avec PHP, XML, SQL et les formulaires (TP 6)** : il s'agit d'aller interroger un site web afin qu'il nous indique le prix des chambres.

(1.1) Affichez la ressource <http://bruno.mascret.fr/tli/ressources/prix.php> Cette ressource indique le prix actuel de la chambre. C'est cette valeur qu'il vous faut récupérer pour mettre à jour l'affichage et le calcul de prix.

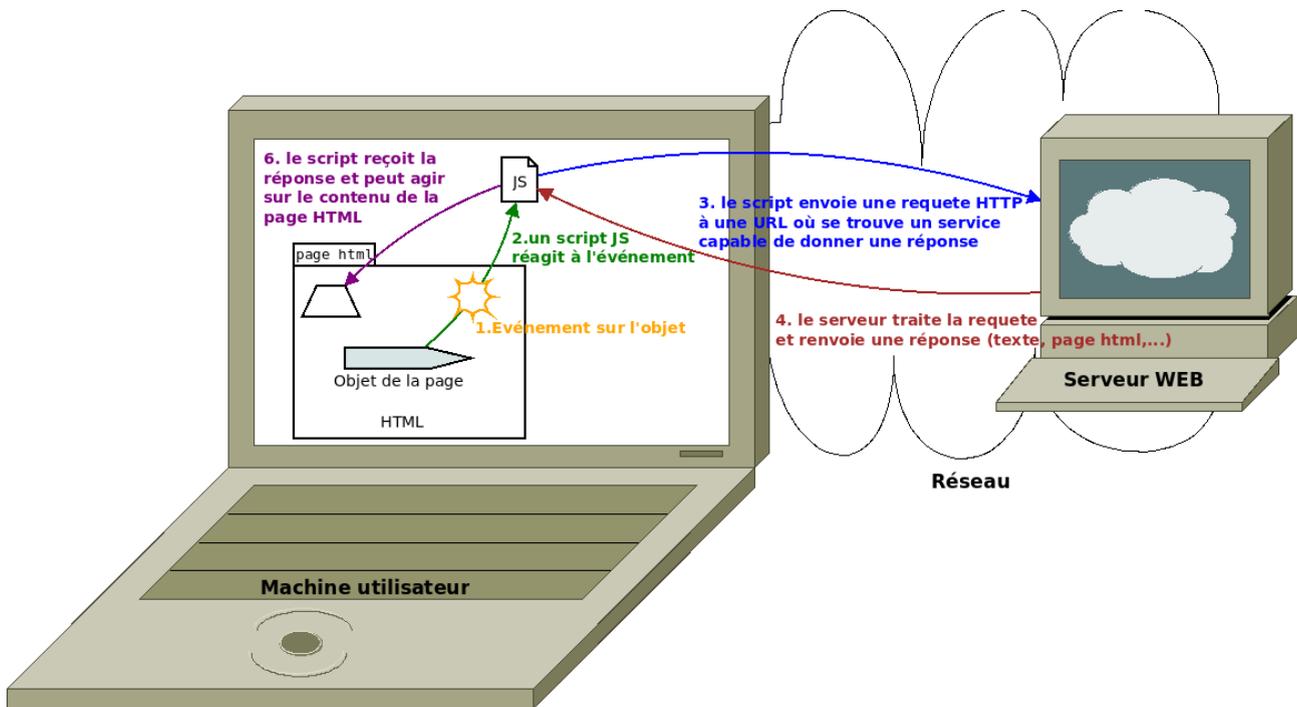


FIGURE 4.1 – Principe du fonctionnement d'AJAX

(1.2) Récupérez l'archive du TP et consultez le fichier [tp4/scripts/exemple.js](#) Que fait ce script ?

(1.3) Modifiez le fichier [tp3/formulaire.html](#) afin de faire fonctionner le script fourni .

Il vous faudra :

- inclure le script ;
- lier l'appel d'une fonction à un événement ;
- modifier la zone d'affichage du prix du formulaire, à vous de trouver avec quoi et comment ¹
- modifier le calcul du prix du script du TP précédent (getTotal).

Exercice 2 : Passage de paramètres à une requête

durée maximale conseillée : 1h

Nous avons vu en cours qu'il existait deux manières de passer des variables à une URL : la méthode GET et la méthode POST.

(2.1) Ajoutez une variable GET de nom *nb* représentant un nombre de nuits à l'url <http://bruno.mascret.fr/tli/ressources/prix.php> .

Si vous ne trouvez pas comment faire, reprenez le premier cours, où on parle d'URL.

1. il est évident que vous ferez cela de manière accessible!!!!

Vous constatez normalement que la réponse n'est pas la même en fonction du nombre de nuits (valeur de *nb*).

(2.2) Modifiez la fonction *send* de *httprequest* et le script *exemple.js* afin de prendre en compte le nombre de nuits pour le calcul du prix.

(2.3) Adaptez votre code afin de permettre une actualisation dynamique du prix. Pensez aux événements qui doivent être pris en charge...

4.2.1 Pour les plus rapides

(2.4) La ressource <http://bruno.mascret.fr/tli/ressources/prixp.php> traite les requêtes où *nb* est transmis par la méthode POST . Proposez un script équivalent à celui que vous avez réalisé avec la méthode GET permettant d'arriver au même résultat.

Quel intérêt représente cette deuxième manière de procéder ?

Exercice 3 : C'est à vous !

Vous trouverez facilement des sites proposant des ressources (météo, heure, etc.). Cherchez des services web utilisant http-get, post ou REST

En voici quelques-un :

- météo : <http://www.webservices.net/globalweather.aspx?op=GetWeather> ; utilise deux variables POST : *CityName* et *CountryName* ;
- convertisseur de température : https://php.radford.edu/~jcdavis/D2L/classes/it425/lectmat/webserv/tempws_api.php ; méthode get
- google chart : le principe est cette fois d'inclure le script js de google qui s'occupera de la communication : https://developers.google.com/chart/interactive/docs/quick_start
- phases de la lune : <http://www.currentplanetarypositions.com/cgi-bin/mpsyndicate.cgi>

(3.1) Ajoutez à vos pages HTML des fonctionnalités basées sur XMLHttpRequest et ajax.

5. Web dynamique avec PHP et SQL (TP 5)

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

Objectifs

1. savoir mettre en place un serveur web et un serveur de base de données sous Linux ;
2. utiliser un script php ;
3. apprendre les bases du langage **SQL** et savoir réaliser des requêtes simples ;
4. utiliser **SQL** avec **PHP**.

Exercice 1 : Mise en place du serveur

Durée recommandée : 45 min maximum

Vous allez installer un serveur web (apache), un module d'interprétation de **PHP**, un serveur de base de données (mySQL) et un outil de gestion de bases de données (phpmyadmin).

Cette configuration est souvent nommée serveur **Linux Apache MySQL PHP (LAMP)**.

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

Vous pouvez également travailler sur vos propres machines si vous préférez, **mais vous prenez un risque notamment si vous êtes sous windows ou mac.**



Raccourci pour ouvrir un terminal sous Linux

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

5.1.1 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
```

Login et password de mysql

Lors de l'installation de `mysql`, il vous sera demandé de donner un nom d'utilisateur et un mot de pass pour l'administrateur.

Notez ces informations car nous en aurons besoin prochainement.

5.1.2 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. regardez¹ 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>.

5.1.3 Configuration des messages d'erreurs

Copiez le fichier `tp5/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 débogage.

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 utiliser le comparateur `meld` sous linux (`sudo apt-get install meld`).

Écrivez 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
```

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

Exercice 2 : Prise en main de SQL

Durée recommandée : 2h45 maximum

ATTENTION : vous n'aurez probablement pas le temps de réaliser toutes les requêtes. Faites attention au temps et gardez bien 30min pour la dernière partie.

L'objectif de ce TP est de vous permettre d'acquérir rapidement les bases du langage **SQL** dont nous aurons besoin dans le cadre de ce module afin de manipuler et d'interroger des bases de données avec **PHP**.

Les bases de données font l'objet d'un module complet qui vous sera pour certains dispensé plus tard dans votre formation.

Nous ne verrons ici que le minimum nécessaire qu'un ingénieur doit connaître.

Ce TP est une adaptation du TP de Vincent COUTURIER lui-même révisé par Françoise Perrin et Martine Breda en 2011 et 2012.

5.2.1 Prise en main de l'environnement

Vous venez d'installer un environnement serveur qui inclus un serveur mysql, un client mysql et un logiciel client de manipulation (phpmyadmin).

PHPmyadmin est un logiciel écrit en **PHP** dont le but est de fournir une interface web pour manipuler des bases de données.

Il est normalement accessible, une fois installé, à l'adresse <http://localhost/phpmyadmin>.



Connectez-vous à l'interface phpmyadmin

Le nom d'utilisateur et le mot de passe sont ceux du serveur mysql.

5.2.2 Installation d'une nouvelle base de donnée

L'interface générale de PHPmyadmin est représentée sur la figure 5.1.

(2.1) Créez une nouvelle base de donnée

1. aller sur l'onglet « Bases de données » ;
2. indiquez un nom de base de données (par exemple « cinema » **sans accent ni espace**) ;
3. choisissez « utf8mb4_unicode_ci » ;
4. validez ; vous devriez voir apparaître un message de confirmation et le nom de votre base de donnée dans la barre latérale de PHPmyadmin.

(2.2) Importez les tables et leurs contenus Le script [tp5/cinema.sql](#) contient la déclaration des tables, de leur structure, des relations entre tables et des contenus. Il s'agit d'une série de commandes **SQL**, séparées par des ;.

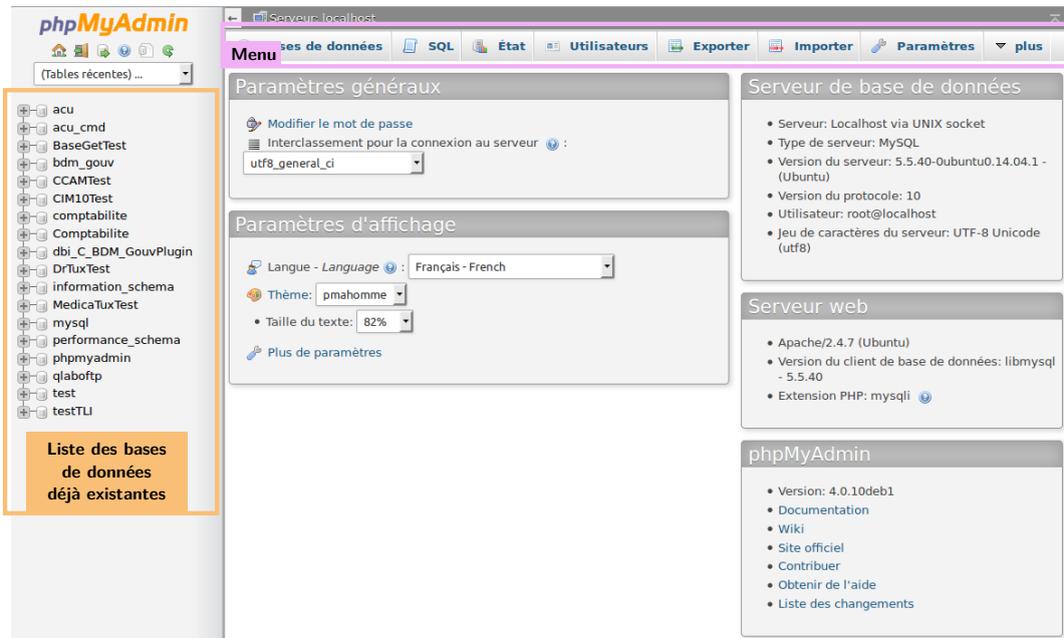


FIGURE 5.1 – Accueil de PHPmyadmin

 Ouvrez [tp5/cinema.sql](#) dans un éditeur texte et regardez son contenu

Essayez de comprendre ce que le script réalise, dans les grandes lignes.

Pour utiliser ce script sur la base de données que vous avez créée (« cinema ») :

1. sélectionnez la base de données créée dans la barre latérale ;
2. allez dans l'onglet « importer » ;
3. en utilisant le bouton parcourir, chargez le fichier [tp5/cinema.sql](#) ;
4. validez (bouton exécuter tout en bas) ;
5. un message vous indique que 90 requêtes ont été exécutées, et la base de données « cinema » doit maintenant contenir 8 tables.

5.2.3 Vue relationnelle

L'onglet « concepteur » de PHPmyadmin vous donne un aperçu graphique de l'organisation des tables.

La figure 5.2 vous propose un aperçu graphique possible.

Exercice 3 : Travail à réaliser

Guide de référence : cours et <http://dev.mysql.com/doc/refman/5.0/fr/tutorial.html>.

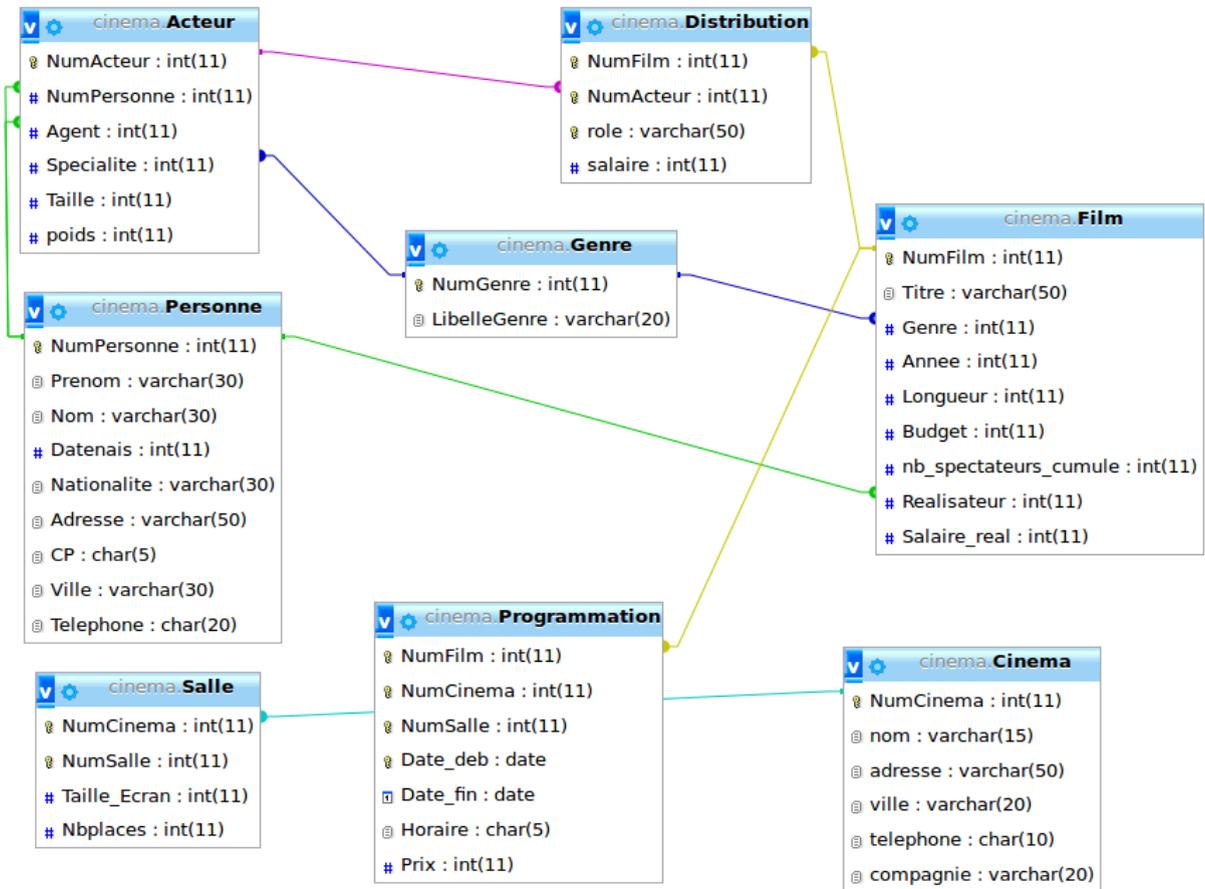


FIGURE 5.2 – Vue relationnelle des tables de la base de donnée « cinema »

💡 Saisir une requête SQL dans PHPmyadmin
 L'onglet « SQL » de PHPmyadmin vous permet de taper, exécuter et déboguer n'importe quelle requête compatible avec MySQL.

Programmer les requêtes suivantes :

(3.1) Afficher toutes les personnes habitant sur une avenue

	NumPersonne	Prenom	Nom	Datenais	Nationalite	Adresse	CP	Ville	Telephone
<input type="checkbox"/> Modifier Copier Effacer	1	Steven	Spielberg	1947	Americain	5é Avenue	NULL	New York	555111
<input type="checkbox"/> Modifier Copier Effacer	10	Woody	Allen	1935	Americain	10e Avenue	NULL	New York	4435574
<input type="checkbox"/> Modifier Copier Effacer	11	Mike	Leblanc	1968	Americain	10e Avenue	NULL	New York	5435789
<input type="checkbox"/> Modifier Copier Effacer	14	Sam	Neil	1947	Americain	15e Avenue	NULL	New York	555555
<input type="checkbox"/> Modifier Copier Effacer	15	Laura	Dern	1967	Americain	13e Avenue	NULL	New York	555444
<input type="checkbox"/> Modifier Copier Effacer	16	Jeff	Goldblum	1952	Americain	Avenue M. Foch	NULL	Miami	555333
<input type="checkbox"/> Modifier Copier Effacer	17	Walter	Matthau	1920	Americain	Avenue M. Bloch	NULL	San Francisco	555123
<input type="checkbox"/> Modifier Copier Effacer	18	Chris	Campion	1966	Americain	Avenue G. Bush	NULL	Washington	555124

(3.2) Affecter comme numéro de téléphone la valeur « N'a pas d'amis » à toutes les personnes qui n'ont pas renseigné leur numéro de téléphone (commencer éventuellement

par afficher la liste ci-dessous) 5 lignes affectées

(3.3) Afficher les noms et prénoms des réalisateurs habitant une ville commençant par la lettre 'N'

nom	prenom
Spielberg	Steven
Allen	Woody
Chabat	Alain

(3.4) Afficher tous les films (TITRE, ANNEE, REALISATEUR) qui n'ont pas été réalisés par Spielberg

titre	annee	Nom réalisateur
La jeune fille et la mort	1995	Polanski
Pirates	1986	Polanski
Les Misérables	1925	Frescourt
Les Misérables	1933	Bernard
Itinéraire d'un enfant gâté	1988	Lelouche
Robert et Robert	1978	Lelouche
Persona	1966	Bergman
Manhattan	1979	Allen
Maris et femmes	1992	Allen
Gazon Maudit	1994	Balasko
Obélix et Asterix : Mission Cléopâtre	2003	Chabat

(3.5) Afficher pour chaque réalisateur (nom, prénom) et chaque film (titre) son salaire à la minute de film

nom	prenom	titre	Salaire du réalisateur à la minute
Polanski	Roman	La jeune fille et la mort	909.0909
Polanski	Roman	Pirates	1449.2754
Spielberg	Steven	La couleur pourpre	6578.9474
Spielberg	Steven	La liste de Schindler	4291.8455
Spielberg	Steven	Jurassic Parc	8000.0000
Frescourt	Henri	Les Misérables	3.8986
Bernard	Raymond	Les Misérables	10.7143
Lelouche	Claude	Itinéraire d'un enfant gâté	559.4406
Lelouche	Claude	Robert et Robert	319.1489
Bergman	Ingmar	Persona	102.5641
Allen	Woody	Manhattan	576.9231
Allen	Woody	Maris et femmes	892.8571
Balasko	Josiane	Gazon Maudit	551.7241
Chabat	Alain	Obélix et Asterix : Mission Cléopâtre	1811.5942

(3.6) Afficher pour chaque film, les nom et prénom des acteurs et leur salaire (afficher le titre du film par ordre alphabétique et le salaire par ordre décroissant)

titre	nom	prenom	salaire
Gazon Maudit	Chabat	Alain	110000
Gazon Maudit	Balasko	Josiane	100000
Jurassic Parc	Goldblum	Jeff	10000000
Jurassic Parc	Neil	Sam	5000000
Jurassic Parc	Dem	Laura	4000000
Les Miserables	Frescourt	Henri	1000
Manhattan	Allen	Woody	12000
Maris et femmes	Allen	Woody	80000
Obelix et Asterix : Mission Cléopatre	Chabat	Alain	150000
Pirates	Matthau	Walter	150000
Pirates	Campion	Chris	120000

(3.7) Quels sont les acteurs dramatiques (nom, prénom) qui ont joué dans un film de Spielberg

Prenom	Nom
Sam	Neil
Laura	Dem

(3.8) Quels sont les cinémas indépendants Bordelais, avec le film correspondant, qui passent un film de Woody Allen de partir 22 heures dans une salle d'au moins 100 places et d'écran de taille supérieure à $30m^2$

nom	titre
Utopia	Manhattan

(3.9) Afficher tous les genres de film et les titres des films associés à chaque genre

LibelleGenre	Titre
Drame	La couleur pourpre
Drame	Les Miserables
Drame	Les Miserables
Drame	Itineraire d'un enfant gate
Comedie	Robert et Robert
Comedie	Manhattan
Comedie	Maris et femmes
Comedie	Gazon Maudit
Comedie	Obelix et Asterix : Mission Cléopatre
Aventure	Pirates
Science-fiction	Jurassic Parc
Historique	La liste de Schindler
Thriller	La jeune fille et la mort
Thriller	Persona
Fantastique	NULL

(3.10) Afficher les cinémas dont les salles n'ont pas été saisies dans la base

nom
Decavision

(3.11) Ajouter la salle Nø1 au cinéma « Decavision » 1 ligne insérée**(3.12) Afficher les programmations dans toutes les salles de tous les cinémas (même si la salle n'a pas de programmation)**

Nom	NumSalle	Titre	Date_deb	Date_fin	Horaire	Prix
Utopia	1	NULL	NULL	NULL	NULL	NULL
Utopia	2	Manhattan	2005-02-12	2005-02-28	22:00	8
Utopia	3	La jeune fille et la mort	2005-01-28	2005-04-14	15:00	8
UGC	1	Jurassic Parc	2005-01-28	2005-04-07	15:35	10
UGC	2	Itineraire d'un enfant gate	2005-01-13	2005-02-22	20:30	10
UGC	3	NULL	NULL	NULL	NULL	NULL
Pathe	1	NULL	NULL	NULL	NULL	NULL
Pathe	2	NULL	NULL	NULL	NULL	NULL
Vigo	1	Persona	2005-01-01	2005-12-31	18:00	9
Decavision	1	NULL	NULL	NULL	NULL	NULL

(3.13) Quel est le total des salaires des acteurs du film « Jurassic Parc ».

Montant des salaires
19000000

(3.14) Donner le nombre de films par genre

LibelleGenre	Nombre
Aventure	1
Comedie	5
Drame	4
Historique	1
Science-fiction	1
Thriller	2

(3.15) Supprimer de la table Genre ceux qui ne sont relatifs à aucun film

1 ligne supprimée

(3.16) Trouver le genre des films des années 90 dont le budget moyen (du genre) dépasse 10.000.000 \$

LibelleGenre	Budget moyen
Science-fiction	30000000.0000

(3.17) Afficher les réalisateurs qui n'habitent pas dans les villes où il y a un cinéma

nom	prenom
Allen	Woody
Balasko	Josiane
Bergman	Ingmar
Chabat	Alain
Frescourt	Henri
Lelouche	Claude
Polanski	Roman
Spielberg	Steven

(3.18) Trouver le titre et l'année du film le plus long

Titre	Annee	Longueur
Les Miserables	1925	513

(3.19) Trouver le titre des films qui passent dans aucun cinéma de la compagnie UGC. Répondre à cette requête de 2 façons : avec IN (et/ou NOT IN) et avec EXISTS (ou NOT EXISTS)

Titre
La jeune fille et la mort
Pirates
La couleur pourpre
La liste de Schindler
Les Miserables
Robert et Robert
Persona
Manhattan
Maris et femmes
Gazon Maudit
Obelix et Asterix : Mission Cléopâtre

(3.20) Afficher les acteurs qui ont gagné plus (en tout) que la somme des salaires d'un réalisateur quelconque

Nom	Prenom	Salaire total
Allen	Woody	92000
Balasko	Josiane	100000
Campion	Chris	120000
Chabat	Alain	260000
Dem	Laura	4000000
Goldblum	Jeff	10000000
Matthau	Walter	150000
Neil	Sam	5000000

(3.21) Afficher les acteurs qui ont gagné plus (en tout) que tous les réalisateurs (pour tous les films réalisés)

Nom	Prenom	Salaire total
Dem	Laura	4000000
Goldblum	Jeff	10000000
Neil	Sam	5000000

(3.22) Donner le nom et le prénom des réalisateurs qui ont un salaire plus élevé sur un film que leurs acteurs (acteurs du même film).

Prenom	Nom
Roman	Polanski
Henri	Frescourt
Woody	Allen
Alain	Chabat

Exercice 4 : C'est à vous !

Durée recommandée : 30 min maximum

Fabriquez une base de données SIMPLE (2 ou 3 tables maximum) pour votre application.

Lors du prochain TP, nous utiliserons **PHP** pour réaliser des requêtes **SQL** sur cette base.

6. Web dynamique avec PHP, XML, SQL et les formulaires (TP 6)

L'objectif de ce TP est de découvrir **XML**, de l'utiliser avec **PHP** et d'utiliser les langages vus à ce jour pour produire un premier programme dynamique de gestion de formulaire.

Objectifs

1. comprendre **XML** et savoir spécifier un modèle simple de document ;
2. utiliser un script php pour manipuler **XML** ;
3. concevoir et mettre en œuvre un petit programme utilisant les technologies vues à ce jour.

Exercice 1 : Découverte de XML

durée conseillée : 2h

6.1.1 **XML** par l'exemple : un format normalisé, MathML, et un format « maison »

Regardez le fichier [tp6/xml/maths.xhtml](#) dans un navigateur web (firefox de préférence) et ouvrez-le également dans un éditeur de texte (kate de préférence).

(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) Même question pour le fichier [tp6/xml/cinema.xml](#). Quelles différences d'affichage constatez-vous ?

(1.3) Pourquoi votre navigateur gère-t-il différemment les deux fichiers ?

(1.4) Intégrez quelques formules mathématiques sur vos pages. Quelles sont les principales difficultés que pose le format MathML ?

6.1.2 Définition de modèle de document XML

Nous avons vu en cours qu'un document XML doit être associé à un modèle permettant de garantir le bon respect des règles d'utilisation d'un format XML.

Nous avons également vu qu'il existe deux techniques possibles : l'utilisation de Document Type Definition (DTD) ou l'utilisation de schémas XML (format XML Schema Definition (XSD)).

(1.5) A quoi sert le fichier tp6/xml/xhtml-math11-f.dtd ? Regardez les fichiers [tp6/xml/document.xml](#), [tp6/xml/windob.xsd](#) et [tp6/xml/windob.dtd](#).

(1.6) A quoi servent ces fichiers ?

(1.7) Modifiez tp6/xml/document.xml afin de lui associer une DTD Vous validerez¹ ensuite le document afin de vérifier sa syntaxe xml (validation du langage xml) et sa conformité au modèle (respect des contraintes du modèle). Ne corrigez pas les erreurs.

(1.8) Modifiez tp6/xml/document.xml afin de lui associer un schéma XSD Vous validez² ensuite le document afin de vérifier sa syntaxe xml (validation du langage xml) et sa conformité au modèle (respect des contraintes du modèle). Ne corrigez pas les erreurs.



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

6.1.3 XPATH : un langage de parcours d'arbre XML

Le langage xpath permet de parcourir un document XML et de sélectionner les données répondant à une contrainte.

On peut par analogie dire que SQL est à une base de données ce qu'XPath est à un document XML.

Voici quelques éléments simples d'utilisation de XPath :

Syntaxe générale : *noeuds[restrictions]/enfants[restriction]*

1. voir les outils dans la boîte d'aide plus bas
2. voir les outils dans la boîte d'aide plus bas

Exemples :

- `noeud/*` : tous les fils de noeud
- `noeud/@*` : tous les attributs de noeud
- `noeud/*|noeud/@*` : tous les fils et les attributs de noeud
- `noeud/a` : tous les fils de noeud de type a
- `noeud/b[1]` : le premier fils de type b de noeud
- `noeud/b[position()>1 and @style="gras"]` : le 5ème des fils de type b de noeud qui n'est pas le premier fils de type b de noeud et qui a un attribut style valant "gras"
- `..` : le noeud père
- `//a` : tous les noeuds de type a du document ; SYNTAXE A EVITER car très coûteuse en performance

(1.9) Rendez-vous à l'adresse <http://www.whitebeam.org/library/guide/TechNotes/xpathtestbed.rhtm> et chargez le document <tp6/xml/cinema.xml> Ce site propose d'exécuter des requêtes XPATH sur un document XML.

(1.10) Réalisez des requêtes XPATH sur le document, par exemple :

- quels sont les cinémas indépendants ?
- quels sont les cinémas qui proposent au moins deux films ?
- quels sont les cinémas qui proposent au moins une séance après 20h ?
- etc....

Exercice 2 : C'est à vous !

Vous avez utilisé de nombreuses technologies durant ce module.

Pour terminer (en beauté), nous vous proposons de réaliser un petit programme permettant de consulter des informations dans une base de données et/ou un fichier XML, et de les utiliser via un formulaire html.

C'est à vous de définir :

- quelles technologies utiliser, et à quel moment ;
- le scénario (cas d'utilisation) de votre programme ;
- les vérifications à faire, notamment lors de l'utilisation d'un formulaire HTML.

Si vous avez correctement avancé votre projet personnel, cette dernière partie ne devrait pas vous prendre plus de 2 heures...

Afin de vous faciliter la vie, nous vous fournissons une classe php dans le répertoire <tp6/lib>.

Elle vous permet facilement d'exécuter des requêtes SQL en php sur une base de données.

Pour xpath et xml, consultez la documentation et l'exemple donné à <http://php.net/manual/fr/domxpath.query.php>

Deuxième partie

Travaux pratiques du projet « fil rouge »

7. Cahier des charges des TPs « fil rouge »

7.1 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 !

7.2 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 7.1) ;
2. de pouvoir n'afficher que certaines des pathologies en fonction de différents critères (type de pathologie, choix des méridiens¹, etc., voir la table 7.2) ;
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 poncturent, 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 7.1 – 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 7.2 – Critères pour les filtres. Les filtres peuvent se combiner : sélectionner les pathologies de méridien, interne, pour poumon et foie

7.3 Contraintes techniques

7.3.1 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.

7.3.2 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 **LAMP**.

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

7.3.3 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](#)].

7.3.4 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.

7.4 Contraintes de réalisation

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

7.4.1 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 \(RSS\)](#)).

7.4.2 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.

7.5 Fonctionnalités à implémenter (première partie ou tronc commun)

7.5.1 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.

7.5.2 Recherche de pathologies par mot-clé

Vous implémenterez une fonctionnalité de recherche de pathologie par mot-clé.

Les mot-clés sont associés aux symptômes dans la base de connaissance.

Cette fonctionnalité ne sera accessible qu'aux utilisateurs authentifiés.

7.5.3 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-clé (cf. [7.5.2](#)).

8. Première ébauche statique du site

Durée conseillée : 4h

8.1 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**.

8.2 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 prenez soin de découvrir les fonctionnalités nécessaires mais non exprimées directement par le client¹

(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

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.

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 !

8.3 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...

9. Projet : Dynamisation de site avec PHP

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

Durée conseillée : 4h

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 !)

Vous utiliserez 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>

9.1 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.).

Consultez l'exemple donné dans l'archive du TP ([projet/](#)).

(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

Regardez comment fonctionne phpdocumentor [[documentor, 2014](#)], et produisez la documentation de vos scripts PHP.

Installation de phpdocumentor

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

9.2 Formulaires HTML et PHP Objet

Durée indicative : 2h

9.2.1 Objectifs

L'objectif est d'utiliser des formulaires pour :

- envoyer des paramètres pour construire des pages dynamiques sur les pathologies (filtrage, recherche...);
- gérer la connexion de l'utilisateur.

Ces formulaires seront accessibles (HTML, CSS et ARIA).

Ils pourront utiliser AJAX pour rendre l'utilisation du site plus conviviale et ergonomique.

9.3 Bases de données et PHP

L'objectif du TP est d'utiliser la base de donnée mysql fournie par le client (voir modèle figure 9.1).

Durée indicative : 2h

9.3.1 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 [projet/bd/acuBD.sql](#)

9.3.2 La librairie de référence pour les BD en PHP : PDO

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>

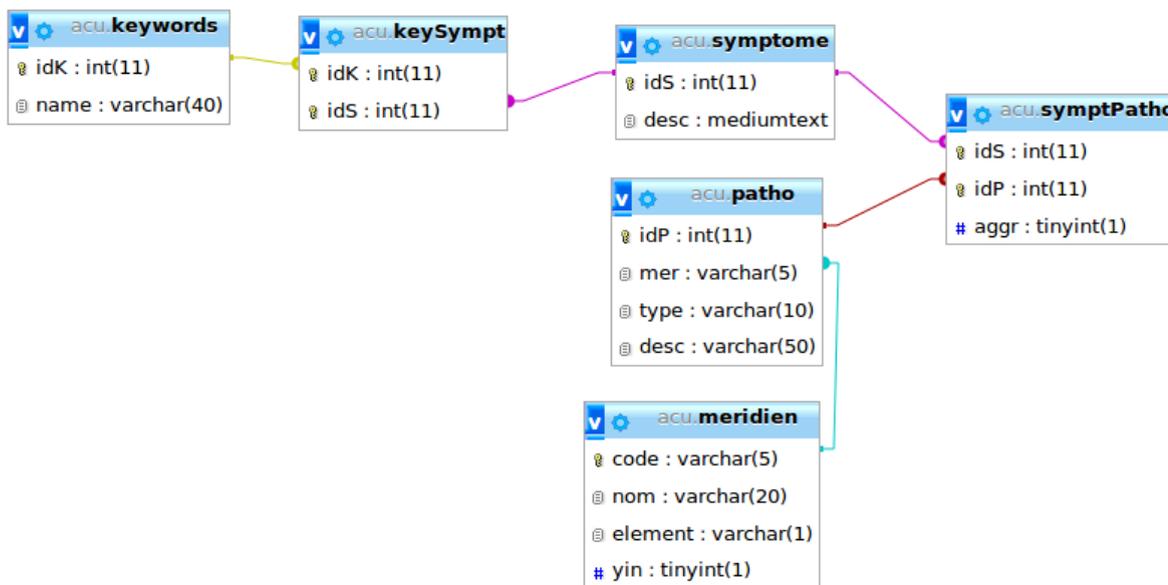


FIGURE 9.1 – Schéma de la base de donnée des acupuncteurs

10. XML et XSD

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

10.1 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.

11. Webservices et ressources REST

11.1 Objectifs

Réalisez un web service autre, de votre choix, utilisant des technologies vues dans ce module. Ce WEB service devra toutefois utiliser le modèle XML de pathologie réalisé précédemment.

11.2 Prise en main : réalisation d'un WS simple

11.2.1 Objectifs

L'objectif de ce petit TP d'application est de réaliser un web service simple basé sur l'architecture **REST**.

Il s'agit de produire un web service pour une petite calculatrice exécutant les opérations basiques suivantes :

- addition de deux entiers ;
- soustraction de deux entiers ;
- multiplication de deux entiers ;
- division de deux entiers.

Le web service aura la forme suivante :

[adresse_site/calculatrice/\[opération\]/nombre1/nombre2](#)

Exemple :

<http://www.cpe.fr/calculatrice/addition/5/8> affichera 13.



Ressources

- *rewriting avec apache* : <http://www.urlrewriting.fr/tutoriel-reecriture.htm>
- *site sur les expressions régulières* : <http://www.regular-expressions.info/>
- *tutoriel sur les expressions régulières* : <http://regexone.com/>
- *rewriting en PHP* : <http://www.urlrewriting.fr/tutoriel-urlrewriting-sans-moteur-rewrite.htm>
- *utilisation de Simple Object Access Protocol (SOAP) avec PHP* : la librairie NuSoap <http://sourceforge.net/projects/nusoap/> et un tutoriel : <http://vivien-brissat.developpez.com/tutoriels/php/soap/>

11.2.2 Première technique : URL rewriting côté serveur web

11.2.2.1 Configuration du serveur Apache (mod_rewrite)

1. activer le module rewrite : `a2enmod rewrite`
2. vérifier sa présence dans la liste des modules activés (répertoire `mod_enabled` de `apache2`)
3. autoriser la redirection dans la configuration du site actif :

```
<Directory /your/path>  
    AllowOverride All  
</Directory>}
```

4. redémarrer apache.

Méthode détaillée à <https://help.ubuntu.com/community/EnablingUseOfApacheHtaccessFiles>

11.2.3 Deuxième technique : URL rewriting via PHP

Exploitation d'un « hack » du fichier de gestion des erreurs 404.

Voir <http://www.urlrewriting.fr/tutoriel-urlrewriting-sans-moteur-rewrite.htm>

A. Bibliographie

- [Alsacreation, 2014] Alsacreation (2014). Alsacreation, communauté d'apprentissage des standards du web. <http://www.alsacreation.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.
- [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. 43

CSS Cascading Style Sheets. i, 4, 6, 7, 33, 34

DTD Document Type Definition. 25

HTML HyperText Markup Language. i, iv, v, 3, 4, 6, 7, 30, 33, 34

LAMP Linux Apache MySQL PHP. 14, 30, 35

MVC Modèle Vue Contrôleur. 35

PHP PHP: Hypertext Preprocessor, anciennement Personal Home Page. i, 14–16, 23, 24, 35

QCM Questionnaire à Choix Multiples. iii

REST REpresentational State Transfer. i, 39

RSS Really Simple Syndication. 31

SOAP Simple Object Access Protocol. 39

SQL Structured Query Language. i, 14, 16, 18, 23, 25, 26

TLI Technologies et Langages de l'Internet. i, iii

URL Uniform Resource Locator. 11, 12

W3C World Wide Web Consortium. 3, 6

WAI Web Accessibility Initiative. 3, 6

WCAG Web Content Accessibility Guidelines. 3, 4, 6

XML eXtensible Markup Language. i, 24–26, 38, 43

XSD XML Shema Definition. 25