

Travaux Dirigés et Pratiques d'Architecture et Programmation du web

HLIN510

Michel Meynard

31 août 2016

1 Introduction, paramètres locaux

En cas de problème, veuillez lire les lignes suivantes avant de demander de l'aide. Ce TP est destiné à être exécuté dans un environnement comportant :

- un navigateur (client HTTP) ;
- un serveur HTTP configuré avec un module interprétant le PHP ;
- un serveur MySQL ;
- les droits suffisants pour écrire des pages PHP et manipuler des bases de données MySQL ;
- un outil d'administration phpMyAdmin.

Les paramètres actuels de ces outils dans le domaine `info-ufr.univ-montp2.fr` sont :

- `~/public_html` : le répertoire de vos documents web accessibles depuis n'importe quel serveur http. Vous y créez un sous-répertoire `ArchiToile` pour y stocker les pages et scripts du TP. Les scripts php peuvent être situés n'importe où sous `public_html`, doivent avoir le suffixe `.php` et être **lisibles** par tous (`chmod a+r toto.php`).
- `http://localhost/~pdupont/ArchiToile` : l'URL pour accéder à vos pages et scripts (localhost, 127.0.0.1, a1.info-ufr.univ-montp2.fr ou a1 étant le nom de votre machine : commande Unix `hostname`) ;
- `~/public_html/ArchiToile/index.html` : page d'accueil par défaut accessible par : `http://machine/~pdupont/ArchiToile`. Si cette page n'existe pas, alors la page `index.php` est recherchée ;
- DROITS :
 - vos répertoires doivent être traversables et lisibles : `chmod 755 public_html public_html/ArchiToile` ;
 - vos fichiers html doivent être lisibles : `chmod 644 public_html/index.html` ;
 - vos scripts php doivent être lisibles : `chmod 644 public_html/ArchiToile/hello.php` ;
 - vos cgi doivent être lisibles et exécutables : `chmod 755 monrep/moncgi`.
- MySQL : // Avant d'accéder au serveur, il vous faut vous inscrire en tant qu'utilisateur. Pour cela, il faut se rendre sur le site du SIF (`http://sif.info-ufr.univ-montp2.fr/`), vous connecter avec vos id et mots de passe Unix puis :
 - dans le bloc bleu "Mon Espace", cliquer sur "Mon Compte"
 - dans le bloc central, cliquer sur "Modifier"
 - faire défiler jusqu'au formulaire "Bases de Données", sélectionner OUI pour la liste MySQL
 - saisir un mot de passe spécifique à MySQL dont vous vous souviendrez ! Remarquez que votre identifiant Unix est votre id MySQL ainsi que le nom de votre BD personnelle alors que les mots de passe Unix et MySQL peuvent être différents ...
 - **Cliquez sur "Enregistrer"**Votre compte MySQL devrait être créé dans les 5 minutes ...
Pour tester votre compte, la machine exécutant le serveur `mysqld` est `venus` sur le port 3306 ;
- Pour se connecter à sa BD en mode console : `mysql -h venus -u pdupont pdupont -p`
Bien entendu, il faut saisir son mot de passe MySQL !
- phpMyAdmin : l'outil `phpMyAdmin` est également situé sur `venus`. Son URL est : `http://venus/phpMyAdmin`
- pour visualiser les erreurs PHP : créer un fichier `.htaccess` avec les directive suivantes :

```
php_flag display_startup_errors on
php_flag display_errors on
php_flag html_errors on
```

Vous pourrez trouver à l'adresse suivante `http://www.lirmm.fr/~meynard/ArchiToile/`, des exemples correspondant aux exercices suivants.

2 TP1 : le langage HTML5 et les feuilles de style

Exercice 1 (TD/TP 1/2 heure) On souhaite écrire le squelette d'un site de e-commerce en HTML5. Les impératifs suivants doivent être respectés :

- Le site doit avoir un en-tête contenant le nom de la société "INFOCOM" ainsi que sa raison d'être : vente de matériel informatique ;
- Le site doit avoir un pied de page contenant une publicité : "INFOCOM : le leader de la vente de matériel informatique" ainsi qu'un lien vers les conditions générales de ventes ;

- Un menu permettant de naviguer dans l'un des trois “rayons” du magasin : disques durs, claviers et souris, divers ;
 - 3 rayons présentant dans un tableau les produits, leur prix, et les caractéristiques techniques ;
- Bien entendu, vous respecterez la décomposition stricte entre contenu et présentation en ne recherchant aucune disposition ni embellissement particulier.

Exercice 2 (TP 1/2 H) Ecrire une feuille de style CSS externe `messtyles.css` afin de mettre en page le site comme dans la figure 1 avec un menu à gauche en petites capitales, une couleur de fond aquamarine, des titres en rouge, les en-têtes et pieds de page séparés par un trait et centrés. Si l’extension Firebug existe sur votre navigateur Firefox, vous pouvez l’utiliser en sélectionnant l’onglet CSS et en modifiant interactivement le rendu de la page. Une fois satisfait, copier les styles développés sous Firebug et copier les dans `messtyles.css`.



FIGURE 1 – Commerce stylé

Exercice 3 (TP 1/2 heure) Changer la feuille de style afin d’obtenir une disposition horizontale du menu.

2.1 Formulaires

On va maintenant tester les formulaires HTML5 et la validation côté client. Vous testerez les formulaires au moins sur 2 navigateurs afin de voir les différences de rendu et de validation.

Exercice 4 (TD/TP 1 heure) Ecrire un formulaire permettant à un utilisateur de remplir son panier tel que celui de la figure 2. On utilisera la méthode GET afin de visualiser les paramètres.

- La première zone de saisie sera destinée aux disques durs : un `number` de nom `qdd` permettra d’indiquer le nombre de disques durs à commander (précédé d’un label). Il sera suivi d’une liste déroulante permettant de choisir le modèle de disque dur.
- Pour les souris et claviers, on utilisera une série de cases à cocher qui auront toutes le nom `souris`. Ces produits ne peuvent donc être commandés qu’en un seul exemplaire !
- On utilisera une liste de boutons radio afin de sélectionner un des produits divers vendus par INFOCOM.
- Un champ OBLIGATOIRE permettra de saisir l’adresse email du client, et un autre champ permettra de saisir son nom qui ne devra contenir que des majuscules.
- Un bouton de soumission nous permettra de voir dans la barre d’adresse la façon dont les paramètres (`nom=valeur&...`) sont passés au serveur.

Exercice 5 (TD/TP 1/2 heure) Ecrire un formulaire qui permettra de jouer à la bataille navale dans un tableau de 5x5 cases comme dans la figure 3. Au début toutes les cases seront remplies de “?”, puis au fur et à mesure du jeu (que vous ne développerez pas), ils se transformeront en “BOUM” si un bateau a été touché ou en “PLOUF” !

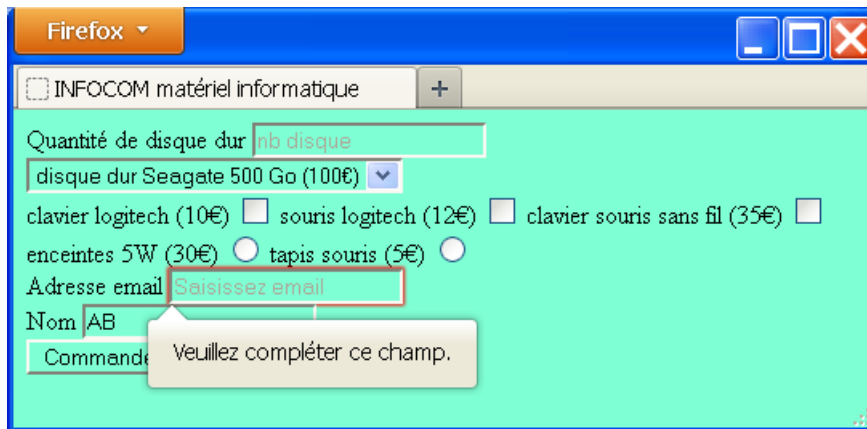
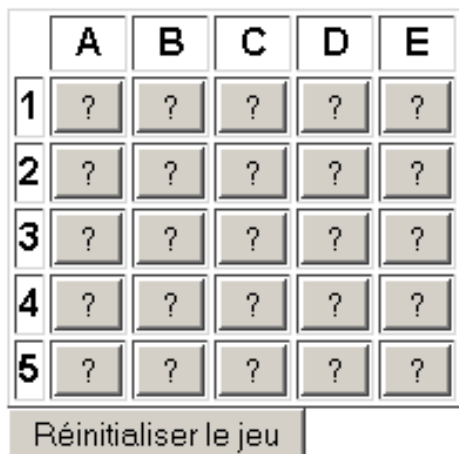


FIGURE 2 – Formulaire panier

Bataille navale



Bataille navale

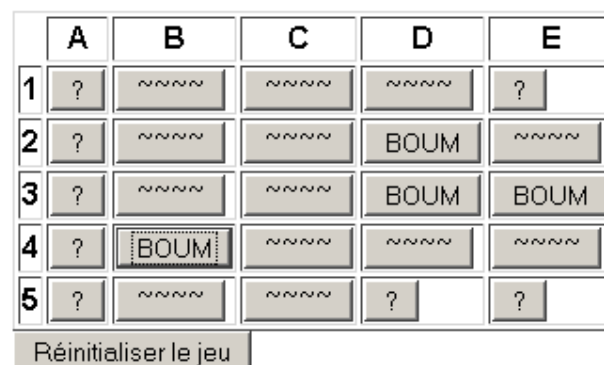


FIGURE 3 – Bataille navale

3 TP2 et 3 PHP

Exercice 6 (TP) Rappelons que `php` est un interpréteur qui peut être utilisée en ligne de commande comme `bash`, `python` :

- Ouvrez un terminal et lancez-y `php -a`;
- Taper `<?php` puis tapez des instructions `php` sans oublier les point-virgules.
- Pour finir, taper `?>` puis CTRL-C pour revenir au `bash`;

On peut également exécuter un fichier PHP : `php -f monfic.php`

Attention à vérifier la correspondance de version entre l'interprète `/bin/php` et le module PHP d'apache (voir `phpinfo()`).

Exercice 7 (Factorielle) On s'intéresse à la fonction factorielle définie par $0! = 1$ et $(n + 1)! = (n + 1) * n!$.

1. écrire la fonction factorielle en PHP ;
2. écrire en un seul fichier, une page permettant de saisir un nombre entier puis d'afficher sa factorielle ;
3. que se passe-t-il si le paramètre passé est incorrect ? Comment y remédier ?

Exercice 8 (Champs cachés et commande) On souhaite écrire une page permettant de calculer le montant d'une commande composées d'articles en quantités variables. Les articles et leur tarifs seront insérés dans un tableau PHP. Par exemple :

```
$larticle=array('marteau'=>10, 'tenaille'=>5, 'vis'=>5.2, 'clou'=>5.8,
'tournevis'=>7, 'ciseau'=>4, 'toile émeri'=>3);
```

Le formulaire devra donc comprendre un nombre de lignes de commandes variables qui augmentera au fur et à mesure que le client commandera de nouveaux articles. A chaque fois que le client validera une nouvelle ligne de commande (nom article, quantité), le formulaire devra être mis à jour pour refléter la bonne quantité et un total à payer correct.

1. Comment conserver l'information sur les articles et leur quantité commandée sans utiliser de session, ni de cookie, ni de fichier, ni de BD ?
2. écrire en un seul fichier, une page permettant de saisir une commande.
3. comment ajouter la suppression de lignes de commandes ?

Exercice 9 (Session et Mastermind) En utilisant la notion de session PHP, vous créez un site permettant de jouer au Mastermind. Le jeu crée un code aléatoire caché à 4 chiffres différents. Le joueur tente à chaque coup une combinaison de 4 chiffres différents et le jeu lui répond en indiquant le nombre de chiffres bien placés et le nombre de chiffres mal placés. Vous conserverez la liste des coups joués ainsi que leur résultat dans la session PHP. Vous modéliserez le jeu par une classe PHP5 qui mémorise le code aléatoire ainsi que la liste des essais successifs et de leurs résultats.

1. Décrire l'interface de la classe Mastermind.
2. Comment stocker une instance de Mastermind dans une SESSION PHP ?
3. Programmer la classe Mastermind dans le fichier Mastermind.php.
4. Ecrire le script du jeu.

4 TP4 MySQL et PDO

Exercice 10 (MySQL et Liste d'étudiants) On utilise une base de données MySQL d'étudiants possédant 3 tables ayant la structure suivante :

```
-- Structure de la table 'etudiant'
CREATE TABLE 'etudiant' (
  'nom' varchar(20) NOT NULL default '',
  'prenom' varchar(20) NOT NULL default '',
  'statut' char(2) NOT NULL default 'FI',
  'groupe' tinyint(1) NOT NULL default '0',
  'email' varchar(20) NOT NULL default '',
  'opt' enum('B','S','C','L','W') default NULL,
  'numStageA' tinyint(2) unsigned default '0',
  PRIMARY KEY ('nom','prenom'),
  UNIQUE KEY 'email' ('email')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 PACK_KEYS=1;
```

```

-- Structure de la table 'options'
CREATE TABLE 'options' (
  'code' char(1) NOT NULL default '',
  'nom' varchar(30) NOT NULL default '',
  'resp' varchar(30) NOT NULL default '',
  'email' varchar(30) NOT NULL default '',
  PRIMARY KEY ('code')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 PACK_KEYS=1;

-- Structure de la table 'stageA'
CREATE TABLE 'stageA' (
  'numStageA' tinyint(2) unsigned NOT NULL default '0',
  'sujet' varchar(255) NOT NULL default '',
  'entreprise' varchar(50) NOT NULL default '',
  'lieu' varchar(150) NOT NULL default '',
  'respEnt' varchar(150) NOT NULL default '',
  'respPeda' varchar(150) NOT NULL default '',
  PRIMARY KEY ('numStageA'),
  KEY 'numStageA' ('numStageA')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 PACK_KEYS=1;

```

Chaque étudiant est inscrit dans une option et dans un stage. Le contenu de la BD est visible à l'adresse : <http://www.lirmm.fr/~meynard/ArchiToile/crebas.sql.txt>.

1. Créer les tables dans votre BD MySQL grâce à l'onglet SQL de PhpMyAdmin.
2. Afficher la liste des noms d'étudiants grâce à un script PHP en utilisant PDO.
3. Ecrire un script PHP `stageA.php` permettant de regrouper les étudiants participant à un même stage d'analyse. Chaque stage sera présenté (numéro, sujet, responsable, tuteur, et enfin les étudiants y participant).
4. On souhaite visualiser les étudiants selon différents critères et écrire un formulaire PHP `trombino.php` comportant :
 - une liste des options avec choix multiples ;
 - une liste de l'ordre d'affichage avec choix unique parmi nom et prénom, statut, groupe, option ;
 - un bouton de validation
 Les choix d'affichage devront être mémorisés.

Exercice 11 (Cookies et Sessions) Afin de bien comprendre les mécanismes de session et de cookie, nous allons commencer une partie de Mastermind sur le site du Lirmm : <http://www.lirmm.fr/~meynard/ArchiToile/master.php>. Après avoir joué un coup "1234" et avoir soumis le formulaire, réaliser les manipulations suivantes :

1. afficher les cookies du lirmm. Repérer PHPSESSID et récupérer sa valeur.
2. supprimer ce cookie ; jouer une nouvelle proposition "5678" : que se passe-t-il ?
3. afficher les cookies du lirmm. Repérer PHPSESSID et comparer sa valeur avec la précédente. Que s'est-il passé ?
4. Refuser les cookies et tenter de jouer une partie. Que se passe-t-il ?
5. Passer le premier PHPSESSID comme paramètre de l'URI : <http://www.lirmm.fr/~meynard/ArchiToile/master.php?PHPSESSID=0c92dbddd4d1ada27ea223b0bc651ff2>. Que se passe-t-il ?
6. Si l'on fait maintenant une nouvelle proposition et que l'on clique sur OK, le premier jeu continue-t-il ?
7. Accepter à nouveau les cookies. Jouer plusieurs nouveaux coups, lancer une nouvelle partie et observer les cookies du lirmm : qu'en déduisez-vous ?
8. Saisissez l'URL <http://www.lirmm.fr/~meynard/ArchiToile/master.php> dans le navigateur. Que se passe-t-il le premier coup et le second coup joués ?
9. Fermer le navigateur et en relancer un autre. Saisissez l'URL : <http://www.lirmm.fr/~meynard/ArchiToile/master.php?PHPSESSID=0c92dbddd4d1ada27ea223b0bc651ff2> avec un PHPSESSID précédent. Retrouve-t-on l'ancienne partie ?

5 TP5 Cookies, authentification

Exercice 12 (Cookies et Mastermind) Réécrire le jeu de Mastermind en ajoutant la fonctionnalité suivante : un joueur peut sauvegarder une partie en cours dans un cookie en lui donnant un nom de son choix. Lorsqu'il joue, il

peut sauver la partie en cours, restaurer une ancienne partie, jouer une nouvelle partie, ou jouer un coup. Les cookies devront avoir une durée de vie d'un jour. Faites des essais en affichant les cookies, en les supprimant, ...

Exercice 13 (Authentification avec MySQL) A l'aide de PhpMyAdmin, créer une table : `utilisateur(login varchar(30), password varchar(50), nom ...)`. Ajouter quelques utilisateurs à cette table en ayant soin de crypter avec la fonction `md5()` le mot de passe de chaque utilisateur. Créer un script PHP qui nécessite l'authentification d'un utilisateur au long des différentes pages du site. L'authentification sera réalisée par un formulaire qui une fois posté et vérifié, sauvegardera le login dans la variable de SESSION.

Sur la page <http://www.lirmm.fr/~meynard/ArchiToile/authentif.php>, on peut se connecter sous deux identités :

- login etu de password etu;
- et prof, prof;

6 TP6 framework

Exercice 14 (Propel ou Doctrine) Les questions suivantes sont à réaliser dans **votre environnement (SIF ou perso.)** ! Les fichiers présentés correspondent à une **autre configuration** (Windows, Wamp, MySQL, en localhost). Ils vous faut donc effectuer les modifications dans chaque fichier de configuration.

1. suivant les informations du polycopié de cours, installer à la ligne de commande Composer puis propel;
2. Ecrire à la main le fichier `schema.xml` suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<database name="cmd">
<table name="client">
  <column name="id" type="INTEGER" required="true" primaryKey="true" />
  <column name="nom" type="varchar" size="40" required="true" description="Nom du client"/>
  <column name="prenom" type="varchar" size="40" description="Prénom du client"/>
</table>

<table name="cmd">
  <column name="id" type="INTEGER" required="true" primaryKey="true" />
  <column name="date" type="timestamp" required="true" description="Date de la commande"/>
  <column name="client_id" type="INTEGER" required="true" description="id du client"/>
  <foreign-key foreignTable="client" onDelete="restrict" onUpdate="cascade">
    <reference local="client_id" foreign="id"/>
  </foreign-key>
</table>
</database>
```

3. Ecrire à la main le fichier `build.properties` suivant :

```
propel.project = cmd
propel.database = mysql
propel.database.url = mysql://root@localhost/cmd
```

4. Lancez la génération du code sql : `propel-gen MonPropel sql`. Puis examinez le fichier `MonPropel/build/sql/schema.sql`
5. Lancez la génération du modèle objet : `propel-gen MonPropel om`. Puis examinez les fichiers `MonPropel\build\classes\cmd\om`
6. Lancez la création des tables dans la bd grâce à la commande suivante : `propel-gen.bat MonPropel insert-sql`. Vérifiez grâce à PhpMyAdmin que les deux tables ont bien été créées.
7. Ecrire à la main le fichier `MonPropel/runtime.xml` suivant :

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<config>
  <log>
    <ident>cmd</ident>
    <level>7</level>
  </log>
  <propel>
    <datasources default="cmd">
      <datasource id="cmd">
```

```

    <!-- the Propel adapter will usually be the same as phptype of connection DSN -->
    <adapter>mysql</adapter>
    <connection>
      <phptype>mysql</phptype>
      <database>cmd</database>
      <hostspec>localhost</hostspec>
      <username>root</username>
    <!--      <password></password> -->
    </connection>
  </datasource>
</datasources>
</propel>
</config>

```

. Ce dernier permet d'indiquer la configuration à l'exécution.

8. Lancez la création de la configuration à l'exécution en php grâce à la commande suivante : `propel-gen.bat MonPropel convert-conf`. Vérifiez le fichier `MonPropel/build/conf/cmd-conf.php`. Remarquons que les cibles `sql`, `om` et `convert-props` peuvent être demandées plus simplement avec la commande simplifiée : `propel-gen.bat MonPropel`.
9. Pour utiliser le modèle objet, il suffit maintenant d'écrire le fichier suivant :

```

<?php
set_include_path("build/classes" . PATH_SEPARATOR . get_include_path());
include_once 'propel/Propel.php';
Propel::init("build/conf/cmd-conf.php");

include_once 'cmd/Client.php';
$dup=new Client();
$dup->setNom("Dupont");
$dup->save();
echo "client créé";
?>

```

Lancer ce fichier depuis un navigateur et vérifiez que le client a bien été créé dans la BD !

10. Utiliser PhpMyAdmin afin de réaliser les transformations suivantes sur la BD :
 - transformer le moteur de stockage en `innoDB` afin de permettre les transactions ;
 - créer la table `article`(id int autoinc, designation varchar(50), prix decimal(12,2)) ;
 - créer la table `ligne`(cmd_id int, article_id int , qte int) ;
 - ajouter les contraintes d'intégrité référentielle (gestion des relations) suivantes : `cmd_id` ref `cmd(id)`, `article_id` ref `article(id)`, `client_id` ref `client(id)`
 - peupler les tables avec des clients, des commandes, des articles, des lignes de commandes de quantité d'articles.
11. après avoir sauvé `schema.xml` dans un autre fichier, recréez-le en utilisant la commande `propel` : `propel-gen.bat MonPropel reverse`.
12. régénérez tout le modèle objet et vérifiez les nouvelles classes. S'il y a des erreurs, déboguez-les !