

Travaux Dirigés et Pratiques d'Architecture et Programmation du web

HLIN510

Michel Meynard

30 novembre 2020

1 Introduction, paramètres locaux

Ce TP est destiné à être exécuté dans un environnement comportant :

- un navigateur (client HTTP) type Firefox ou Chrome(ium)
- un serveur HTTP (Apache, Nginx) configuré avec un module interprétant le PHP ; ce serveur peut être local (localhost) ou distant ;
- un serveur MySQL avec un compte d'utilisateur (login, password) et une BD existante sur laquelle l'utilisateur a tous les droits courants ;
- les droits suffisants pour écrire des pages PHP et les transférer (filezilla) ou les déposer sur la machine du serveur HTTP à l'endroit convenable (htdocs, public_html, ..) ;
- un outil d'administration phpMyAdmin pour gérer la BD.

La solution la plus simple est d'installer un xAMP (LAMP pour Linux, WAMP pour Windows, ...) qui contient un serveur Apache, un serveur MySQL, un module PHP pour Apache et une application PHPMyAdmin. Une interface graphique permet de lancer les serveurs et d'ouvrir une page Web à la racine des documents Webs (<http://localhost:8888/MAMP/>).

Une autre solution pour un serveur HTTP/PHP servant le répertoire courant consiste à lancer ce qui suit depuis la ligne de commande puis d'aller visualiser l'URL <http://localhost:8090/mapage.html> :

```
$ php -S localhost:8090
```

2 TP1 : le langage HTML5, les feuilles de style, Bootstrap

Exercice 1 (TD 1/2 heure)

On souhaite concevoir un site de e-commerce en HTML5 et Bootstrap. Les impératifs suivants doivent être respectés :

- Le site doit être **responsive** : utilisable sur mobile aussi bien que sur ordinateur
- Il est destiné à la vente de matériels informatiques répartis dans des rayons (tablette, PC Portable, ...)
- Le site doit être agrémenté de photos des produits, et dans l'en-tête un carrousel de photos dynamisera la présentation
- Un menu permet de naviguer dans l'un des "rayons" du magasin
- Un produit est présenté avec un nom, une photo, un prix, et les caractéristiques techniques le concernant

Le site **Hard Discounter** est représenté dans la figure 1 à la page 2.

Les questions suivantes concernent les éléments HTML présents sur le site **Hard Discounter** de la figure 1 :

1. Quel est l'élément HTML5 qui doit représenter le bandeau haut (titre et carrousel) ?
2. Quel est l'élément HTML5 qui doit représenter la colonne de gauche permettant de naviguer sur le début d'un rayon ?
3. Quel sont les éléments HTML5 qui doivent représenter les titres tels que Portables ou Hard Discounter ?
4. Quel est l'élément HTML5 qui doit représenter le bouton bleu **Commander** ?
5. Quel est l'élément HTML5 qui doit représenter le champ de saisie indiquant la **quantité** d'articles à commander et quel sont ses deux attributs **indispensables** ? Décrivez l'utilité des 2 attributs.
6. Quel est l'élément HTML5 qui doit représenter le conteneur de ce champ de saisie indiquant la **quantité** ? Décrivez 2 attributs fondamentaux de ce conteneur.
7. Quel est l'élément HTML5 présent dans la colonne de gauche en deux exemplaires permettant de naviguer sur les rayons Portables et Tablettes ? Décrivez l'attribut fondamental de ce lien.
8. Quel est l'élément HTML5 présent dans la colonne centrale en deux exemplaires Portables et Tablettes qui est la cible d'un lien ? Décrivez l'attribut fondamental de cette cible.

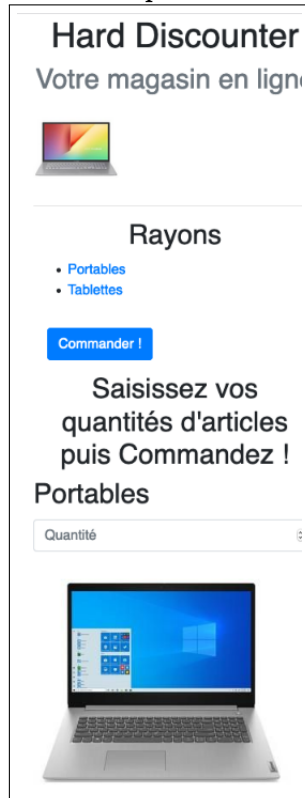
Exercice 2 (TD 1/2H responsive et Bootstrap)

Les questions suivantes concernent les éléments HTML présents sur **grand écran** du site **Hard Discounter** de la figure 1 en supposant que la mise en page a été effectuée avec Bootstrap (framework CSS) :

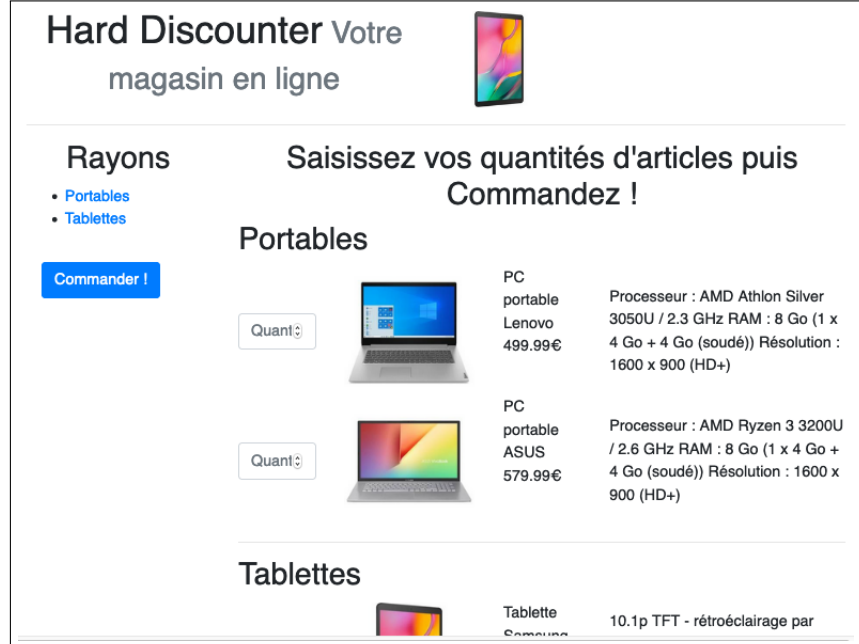
1. Quel est le nombre de colonnes de la grille Bootstrap ?

FIGURE 1 – Site de commerce responsive

Sur un téléphone mobile



Sur écran d'ordinateur



2. Quel est le nom de la classe de plus haut niveau permettant de définir un conteneur dont la largeur prend 100% de l'écran de manière continue ?
3. Quel est le nom d'élément générique HTML utilisé habituellement par toutes les classes de disposition (layout) de Bootstrap (conteneur, ligne, colonne) ?
4. Quelles sont les tailles d'écran utilisées pour spécifier la mise en page ?
5. Si l'on définit une unique mise en page pour les écrans sm (small), à quelles tailles s'appliquent cette mise en page ?
6. Si l'on définit une unique mise en page pour les écrans sm, comment sera la mise en page sur un écran xs ?
7. En supposant une unique mise en page pour les écrans sm, dessiner sur la photo **grand écran** les principales classes de style Bootstrap en indiquant la largeur des colonnes utilisées ...
8. On vous demande d'ajouter un rayon **Mobile** et un téléphone mobile ainsi qu'un pied de page contenant 3 colonnes centrées de même taille avec : un lien pour revenir en début de page, un titre rappelant la marque Hard Discounter ainsi qu'un lien **Nous contacter** permettant d'ouvrir son application de mail avec comme destinataire `contact@harddiscounter.com`. Ecrire le code HTML5 correspondant.

Exercice 3 (TP 1 heure HTML et PHP)

Récupérez sur Moodle le fichier compressé contenant :

- `commande.html` le fichier contenant le source des exercices 1 et 2
- `data.php` le fichier PHP définissant une chaîne JSON définissant les données et deux fonctions permettant d'analyser le JSON et de fournir le catalogue complet ou un seul article désigné par son identifiant
- le répertoire `img` qui contient les photos des articles avec un nom constitué de leur identifiant suivi de l'extension `jpg`

L'un des objectifs de cet exercice consiste à produire une facture en PHP lorsqu'on soumet le formulaire de commande une fois rempli par l'utilisateur. La vue de la facture est dans la figure 3

1. Intégrez les nouveaux éléments (mobile, footer, ...) de l'exercice 2 dans le fichier `commande2.html` et testez la visualisation sur 2 navigateurs (Chrom(ium) et Firefox).
2. Ouvrez les outils de développement (F12 sur Firefox et Chrome) et passez en revue la structure grâce à l'inspecteur HTML (flèche placée à l'extrême gauche de la barre d'onglets)
3. Visualisez votre code pour différentes tailles d'écrans grâce à l'icone représentant un mobile (à gauche dans Firefox, à droite sur Chrome)

FIGURE 2 – Facture

Facture

Quantité	Article	Prix Unitaire HT	T.V.A.	T.T.C.
2	PC portable Lenovo	399.99 €	100.00 €	999.98 €
3	PC portable ASUS	463.99 €	116.00 €	1,739.97 €
1	Tablette Samsung	172.15 €	43.04 €	215.19 €
	TOTAL		591.03 €	2,955.14 €

4. Ecrivez un squelette HTML `facture.html` qui respecte le rendu de la figure 3
5. Recopiez votre `facture.html` en un fichier `facture.php` qui inclura `data.php` et qui devra effectuer les calculs de la facture à l'aide des paramètres du tableau superglobal `$_GET[]` rempli par la soumission du formulaire de `commande2.html`. **Programmation !**
6. Pour finir, transformer `commande2.html` en `commande.php` qui devra afficher le catalogue de `data.php` comme dans la figure 1
7. Une ultime étape consisterait à écrire un fichier `index.php` qui incluerait le fichier `commande.php` et qui lorsqu'il serait *actionné* par la soumission incluerait le fichier `facture.php` ...

Exercice 4 (TD/TP 1/2 heure)

Ecrire un formulaire qui permettra de jouer à la bataille navale dans un tableau HTML de 5x5 cases comme dans la figure 3. Au début toutes les cases seront remplies de "?", puis au fur et à mesure du jeu en PHP (**que vous ne développerez pas**), ils se transformeront en "X" (ou BOUM) si un bateau a été touché ou en "~" s'il n'y a rien à cette position !

Bataille navale

	A	B	C	D	E
1	<input data-bbox="272 1220 326 1255" type="text" value="?"/>	<input data-bbox="370 1220 423 1255" type="text" value="?"/>	<input data-bbox="467 1220 521 1255" type="text" value="?"/>	<input data-bbox="565 1220 618 1255" type="text" value="?"/>	<input data-bbox="662 1220 716 1255" type="text" value="?"/>
2	<input data-bbox="272 1276 326 1312" type="text" value="?"/>	<input data-bbox="370 1276 423 1312" type="text" value="?"/>	<input data-bbox="467 1276 521 1312" type="text" value="?"/>	<input data-bbox="565 1276 618 1312" type="text" value="?"/>	<input data-bbox="662 1276 716 1312" type="text" value="?"/>
3	<input data-bbox="272 1333 326 1369" type="text" value="?"/>	<input data-bbox="370 1333 423 1369" type="text" value="?"/>	<input data-bbox="467 1333 521 1369" type="text" value="?"/>	<input data-bbox="565 1333 618 1369" type="text" value="?"/>	<input data-bbox="662 1333 716 1369" type="text" value="?"/>
4	<input data-bbox="272 1390 326 1425" type="text" value="?"/>	<input data-bbox="370 1390 423 1425" type="text" value="?"/>	<input data-bbox="467 1390 521 1425" type="text" value="?"/>	<input data-bbox="565 1390 618 1425" type="text" value="?"/>	<input data-bbox="662 1390 716 1425" type="text" value="?"/>
5	<input data-bbox="272 1446 326 1482" type="text" value="?"/>	<input data-bbox="370 1446 423 1482" type="text" value="?"/>	<input data-bbox="467 1446 521 1482" type="text" value="?"/>	<input data-bbox="565 1446 618 1482" type="text" value="?"/>	<input data-bbox="662 1446 716 1482" type="text" value="?"/>

Bataille navale

Bravo, vous avez coulé un navire !

	A	B	C	D	E
1	<input data-bbox="820 1220 873 1255" type="text" value="?"/>	~	~	<input data-bbox="917 1220 971 1255" type="text" value="?"/>	<input data-bbox="1015 1220 1068 1255" type="text" value="?"/>
2	<input data-bbox="820 1276 873 1312" type="text" value="?"/>	<input data-bbox="917 1276 971 1312" type="text" value="?"/>	~	X	~
3	<input data-bbox="820 1333 873 1369" type="text" value="?"/>	~	~	<input data-bbox="917 1333 971 1369" type="text" value="?"/>	<input data-bbox="1015 1333 1068 1369" type="text" value="?"/>
4	<input data-bbox="820 1390 873 1425" type="text" value="?"/>	<input data-bbox="917 1390 971 1425" type="text" value="?"/>	<input data-bbox="1015 1390 1068 1425" type="text" value="?"/>	~	<input data-bbox="1112 1390 1166 1425" type="text" value="?"/>
5	<input data-bbox="820 1446 873 1482" type="text" value="?"/>	<input data-bbox="917 1446 971 1482" type="text" value="?"/>	<input data-bbox="1015 1446 1068 1482" type="text" value="?"/>	<input data-bbox="1112 1446 1166 1482" type="text" value="?"/>	<input data-bbox="1209 1446 1263 1482" type="text" value="?"/>

FIGURE 3 – Bataille navale

3 TP2 et 3 PHP

Exercice 5 (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 6 (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 7 (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 8 (Session et Mastermind)

En utilisant la notion de session PHP, vous créerez 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 9 (MySQL et Liste d'étudiants)

On utilisera une base de données MySQL d'étudiants possédant 3 tables ayant la structure suivante :

```
etudiant (nom,prenom,statut,groupe,email,opt,numStageA)
PRIMARY KEY (nom,prenom)
FOREIGN KEY (opt) REFERENCES options(code)
FOREIGN KEY (stageA) REFERENCES stageA(numStageA)
```

```
options (code,nom,resp,email)
PRIMARY KEY (code)
```

```
stageA (numStageA,sujet,entreprise,lieu,respEnt,respPeda)
PRIMARY KEY (numStageA)
```

Chaque étudiant est inscrit dans une option et dans un stage. Le script SQL de création de la BD est téléchargeable sur Moodle. L'accès aux données en PHP se fera à travers PDO.

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 validationLes choix d'affichage devront être mémorisés.

Exercice 10 (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 Lirrm : <http://www.lirrm.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 lirrm (sur firefox : préférences, vie privée, supprimer des cookies spécifiques). 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 lirrm. 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.lirrm.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 lirrm : qu’en déduisez-vous ?
8. Saisissez l’URL <http://www.lirrm.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.lirrm.fr/~meynard/ArchiToile/master.php?PHPSESSID=0c92dbddd4d1ada27ea223b0bc651ff2>
avec un PHPSESSID précédent. Retrouve-t-on l’ancienne partie ?

5 TP5 Cookies, authentification

Exercice 11 (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 12 (Authentification avec MySQL)

A l’aide de PhpMyAdmin, créer ou utiliser la 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://localhost/~meynard/.../authentif.php>, on peut se connecter sous deux identités :

- login `un` de password `un` ;
- et `deux`, `deux` ;

On créera également un formulaire de déconnexion (désauthentification) !

6 TP6 framework

Exercice 13 (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 doctrine ;
2. Ecrire à la main les fichiers php des entités **annotées** correspondant à la BD des étudiants, stage et projet vues au TP précédent.
3. Ecrivez un premier script PHP permettant d’afficher la liste des étudiants.
4. Réécrivez en doctrine le formulaire de visualisation des étudiants selon une sélection et un ordre d’affichage...