

Durée 2 heures.

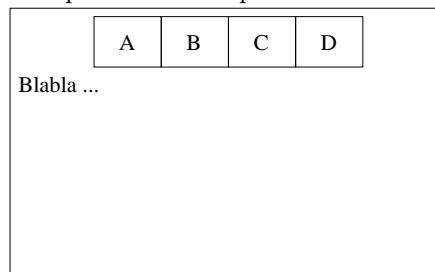
Technologie du Web  
juin 2006

**Exercice 1 :** Voici la partie “body” d’une page XHTML.

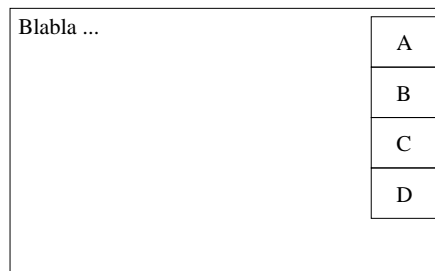
```
<body>
  <div id="menu">
    <p class="choix"> A </p>
    <p class="choix"> B </p>
    <p class="choix"> C </p>
    <p class="choix"> D </p>
  </div>
  <div id="contenu">
    Blabla.....
  </div>
</body>
```

Vous devez concevoir 2 pages de style pouvant lui être associées. Notez que les paragraphes de la classe `choix` symbolisés ici par A,B,C,D sont destinés à contenir des textes très courts 1 ou quelques mots. Le bloc “contenu” (ici Blabla ..) peut, lui, contenir du code XHTML quelconque et un texte long.

Question 1 : Voici une première présentation attendu pour le texte. Le menu est situé en haut. Il est centré sur la page. Le contenu prend toute la place située sous le menu.



Question 2 : Le menu est situé à droite et occupe 20% de la largeur. Le contenu se place sur les 80% à gauche.



**Exercice 2 :** (Répondre sur le formulaire) Une base de données est consacrée à la gestion d’une bibliothèque publique. Cette base comprend les tables suivantes

- La table **lecteurs** contient les informations sur tous les lecteurs enregistrés à la bibliothèque (qu’ils aient des livres en cours d’emprunt ou pas). Chaque lecteur est porteur d’une carte qui possède un numéro d’identification. Les attributs de la table **lecteurs** sont
  - `carte` (entier) : numéro de la carte d’emprunteur encore appelé “numéro de lecteur”
  - `nom` (varchar(30))
  - `age` (entier)
- La table **livres** contient les informations sur tous les livres de la bibliothèque. Chaque livre porte un numéro d’identification. Cette table comporte les attributs
  - `id` (entier) : numéro d’identification du livre (ou “numéro de livre”).

- titre (varchar(30))
- catégorie (varchar(30))

La catégorie est une chaîne dont la valeur appartient un ensemble assez restreint de possibilités. Par exemple “roman”, “encyclopedia”, “bande dessinée” etc). La table des livres peut contenir quelques dizaines de milliers de lignes.

- La table **emprunts** répertorie tous les emprunts en cours. Notez bien que seuls les ouvrages en cours d’emprunt y figurent. Dès qu’un livre est restitué, il disparaît de la table **emprunts**. Attributs de la table :
  - qui (entier) : numéro du lecteur
  - quoi (entier) : numéro du livre emprunté
  - quand (date) : date d’emprunt.

Question 1 : (Répondre sur le formulaire)

Donnez les requêtes SQL permettant d’obtenir les résultats mentionnés ci-dessous. Vous veillerez à faire les requêtes les plus simples possibles (en évitant les jointures inutiles, par exemple). Il est inutile d’utiliser des sous-requêtes.

1. La liste des lecteurs âgés entre 7 et 12 ans (inclus), classée par âges croissants. Informations à produire : le nom et l’âge.
2. Le nombre de lecteurs de moins de 7 ans.
3. Le nombre de lecteurs de chaque âge, liste classée par âges décroissants. Informations à produire : âge et nombre.
4. Le nombre de lecteurs pour chaque tranche d’âges, liste classée par tranches d’âges décroissants. Les tranches d’âges couvrent chacune 10 ans : 0 à 9 ans (tranche 0), 10 à 19 (tranche 1), 20 à 29 (tranche 2) etc... Informations à produire : numéro de la tranche d’âge et le nombre.
5. Nombre de lecteurs ayant un livre en cours d’emprunt.
6. Nombre de lecteurs ayant au moins un livre emprunté depuis plus de 21 jours. Notez que la fonction MySQL permettant d’obtenir le nombre de jours entre la date courante est une date  $d$  est `to_days(curdate())-to_days(d)`. Par simplicité, vous la noterez en abrégé `depuis(d)`
7. Nombre de livres en cours d’emprunt pour chaque catégorie. Informations à produire : la catégorie et le nombre.
8. La liste des livres empruntés par le lecteur numéro  $x$ . Cette liste fera apparaître sur chaque ligne le nom de l’emprunteur et le titre de l’ouvrage.

Question 2 : (Répondre sur la copie)

On se propose de créer une nouvelle table nommée **categories** qui contient 2 champs : un identifiant de catégorie (un entier) et le texte décrivant la catégorie. Elle pourrait contenir par exemple des lignes comme

1	"roman"
2	"bande dessinée"

On modifierait aussi la structure de la table **livres** pour que la catégorie soit un entier (l’identifiant défini dans la nouvelle table) au lieu de la chaîne.

Que pensez-vous de cette proposition (argumentez) ?

Question 3 : **La procédure d’emprunt de livre.** (Répondre sur la copie)

Il s’agit de concevoir une interface de saisie permettant au bibliothécaire d’enregistrer un nouvel emprunt. Un même lecteur peut détenir au maximum 3 livres. La procédure d’emprunt doit vérifier que le nombre de livres que le lecteur veut emprunter additionné au nombre de livres qu’il détient déjà n’excède pas 3.

Une page XHTML statique nommée **emprunts.html** contient un formulaire pour la saisie du numéro de lecteur et de 3 numéros de livre. L’utilisateur n’est pas tenu de fournir 3 numéros de livre mais peut en remplir seulement 1 ou 2. Ce formulaire constitue une “demande d’emprunts”. Sa validation fait appel à un script PHP **valid\_emprunts.php** qui va

- Si les emprunts sont valides, mettre à jour la base de données et renvoyer au navigateur une page XHTML indiquant “emprunts enregistrés”.

- Si les emprunts ne sont pas valides, renvoyer au navigateur une page expliquant la cause de l'erreur qui peut être :
  - Le numéro de lecteur est incorrect (aucun emprunt n'a bien sûr pu être enregistré).
  - Les emprunts demandés feraient dépasser le nombre maximal d'emprunts pour ce lecteur, compte tenu d'ouvrages en cours d'emprunt. Aucun nouvel emprunt n'est enregistré et la page renvoyée indique le nombre maximal d'emprunts possibles pour ce lecteur.
  - Certains numéros de livres sont incorrects ou correspondent à des livres déjà en cours d'emprunt. La page renvoyée indique quels sont les demandes incorrectes (les demandes d'emprunt correctes sont néanmoins traitées et enregistrées dans la base).

**3.a** Donner le code XHTML de la page `emprunts.html`.

**3.b** Donner le code PHP de la page `valid_emprunts.php`. *Attention : afin de faciliter la lecture du code, toutes les commandes SQL générées par ce script devront être mémorisées au préalable dans des variables PHP et la définition de ces variables PHP consacrées aux requêtes SQL devront être regroupées (sous peine d'être ignorées lors de la correction).* Vous ne vous préoccupez pas des instructions de connexion à la base de données.