

## MIF 13 – Programmation Web – Examen

Durée : 1 heure 30 – Documents autorisés – Ordinateurs, calculatrices et téléphones portables interdits

### Questions de cours (barème : 14 points)

1. À quoi correspond un header « Content-Type » dans une requête HTTP et quand ne doit-il pas être utilisé ?  
Il permet de spécifier l'encodage des paramètres de la requête, à condition qu'ils soient envoyés dans le corps de cette requête.  
Il ne doit pas être utilisé dans une requête qui n'a pas de corps, comme GET, HEAD ou OPTIONS.
2. En quoi peut-on dire que le protocole AJP, utilisé dans Tomcat, fonctionne de façon similaire à CGI ?  
Il fait l'interface entre un serveur Web et un autre processus : AJP encapsule les requêtes dans des objets Java, les transmet au processus (Catalina), et désencapsule les réponses.
3. En JSP, quelle différence y a-t-il entre déclarer une variable dans un scriptlet et la déclarer dans une déclaration JSP ? À quoi cela correspond-il techniquement ?  
Dans un scriptlet, c'est une variable locale : elle sera dans le corps de la méthode de service de la servlet générée.  
Dans une déclaration, c'est une variable globale : ce sera une variable d'instance de la servlet.
4. Citez deux différences entre les mécanismes de gestion des Javabeans intégrés à un container de servlets et celui des beans Spring.  
Javabeans : instanciation par constructeur sans paramètre ; Spring : appel d'un constructeur avec les paramètres définis dans la configuration  
Javabeans : utilisation d'accesses simples (get, set) via jsp:getProperty / setProperty ; Spring : accès à toutes les méthodes  
Javabeans : pas de gestion des dépendances ; Spring : résolution du référentiel de dépendances entre les beans  
Gestion des scopes différente
5. Pourquoi dit-on que l'interface d'un service REST « est » le protocole HTTP et quelles sont les limites de cette expression ? Vous pouvez donner un exemple pour illustrer votre réponse.  
La sémantique des méthodes HTTP contraint les opérations sur les ressources exposées (GET -> récupérer, PUT : créer...).  
Cependant, une partie de cette sémantique est également portée par le format des représentations échangées et cette partie n'est pas documentée (que se passe-t-il si l'on fait un PUT avec uniquement un numéro de téléphone vers l'URI d'une personne : crée-t-on une personne sans nom, ou reçoit-on une erreur ? Que renvoie un GET sur une liste de ressources : les ressources ou des pointeurs (URI) sur ces ressources ?).
6. À quoi sert le plus souvent l'API DOM (HTML) dans une fonction de callback en JavaScript après une requête AJAX ?  
À ajouter les résultats de la requête dans la page.
7. Donnez deux raisons pour lesquelles le code suivant ne fonctionnera pas :  
`jQuery("#monElementDiv").innerHTML = jQuery.get("http://example.com");`  
1) la fonction jQuery renvoie un tableau d'éléments DOM wrapped dans des objets jQuery, alors qu'innerHTML s'applique directement à un élément DOM.  
2) une requête AJAX nécessite une fonction de callback ; on ne peut pas directement en affecter le résultat.

### Étude de cas (barème : 8 points)

Suite à la COP21, le Père Noël a décidé de rajouter à son site Web des informations sur le bilan carbone des jouets qu'il livre chaque année aux enfants. Son site actuel est fondé sur une application Web en Java et lui permet de gérer :

- Les produits (cadeaux de tous types)
- Les prospects (enfants ayant envoyé une lettre de demande de cadeau)
- Les demandes (cadeaux demandés, ordre des souhaits)
- Le suivi du stock de cadeaux (choisir les cadeaux à livrer en fonction des demandes et de la disponibilité)

Cette application a été conçue avec Spring Web MVC et comporte donc déjà des composants comme des contrôleurs (produit, prospect, demande...), des beans (POJOs contenant les classes de modèle correspondantes) et les JSP appropriées pour composer les vues HTML. Le Père Noël vous a choisi(e) pour rendre cette application plus dynamique et plus éco-responsable. Pour cela, vous devez la modifier pour permettre (en plus de ce qui existe déjà) de :

- rendre certaines parties de l'application RESTful et exposer les différentes propriétés :
  - o des cadeaux : images, lieu de fabrication, coût d'acheminement...
  - o des prospects : prénoms, adresse de livraison...

- requêter les images des cadeaux et les afficher dynamiquement (en AJAX) dans les pages,
- requêter, calculer (à l'aide d'une fonction JavaScript `calculCarbone(produit, prospect)` fournie ; cette fonction renvoie une valeur numérique indiquant un nombre de grammes de CO2) et afficher le bilan carbone des cadeaux, en fonction des données disponibles à propos des cadeaux (lieu de fabrication...) et des prospects (lieu de livraison).

### Conception (barème : 5 points)

8. Listez les différents éléments que vous allez rajouter à cette application, en indiquant leur nature : classes Java, pages statiques, scripts, servlets, JSP, beans (précisez le scope)...

CECI EST UN EXEMPLE DE REPONSE – TOUT AUTRE STRUCTURE SENSEE EST TOUT AUSSI VALIDE.

Côté serveur :

Modification des contrôleurs pour prendre en compte les requêtes sur différentes méthodes http et sur les sous-ressources

Amélioration du modèle (beans) : produit (scope = prototype pour Spring, application en servlet API « classique »), prospect (scope = session)

Ajout de vues (XML ou JSON) : produit, prospect (-> négociation de contenus)

Amélioration de la vue (JSP pour HTML) : produit pour appeler les scripts côté client

Côté client :

Script qui requête les images et les ajoute à la vue produit

Script qui requête les données carbone, lance le calcul du bilan carbone et l'ajoute à la vue produit

Bibliothèque JS directe : métier (jQuery)

9. Décrivez, à l'aide de diagrammes UML appropriés, leurs communications lorsqu'un utilisateur clique sur un lien lui permettant de visualiser un cadeau. Les éléments du SI externes à votre partie de l'application seront modélisés comme des boîtes noires.

Faire un diagramme de séquence ou de communication. Dans tous les cas, un diagramme dynamique, représentant :

Côté client : L'utilisateur clique sur un lien → requête synchrone

Côté serveur : contrôleur Spring → contrôleur délégué → interrogation du modèle (cadeau) → renvoie les infos (nom, description...) du cadeau → retour au contrôleur délégué → composition de la vue côté serveur (JSP)

Côté client : Affichage de la vue → onload : script de requêtage asynchrone image → requête(s) asynchrone(s) → callback : affichage de l'image

Côté client : Affichage de la vue → onload : script de requêtage asynchrone carbone : infos carbone du cadeau + infos localisation du prospect → callback : appel de la fonction de calcul une fois toutes les réponses arrivées → affichage du bilan carbone

### 10. Programmation (barème : 3 points)

11. Écrivez l'ensemble des scripts JavaScript qui gèrent la partie calcul du bilan carbone, depuis la réception de la page principale jusqu'à l'affichage des données ajoutées.

Cf. ci-dessus pour le dernier script :

body onload : 1 fonction qui lance 2 scripts de requêtage asynchrones : infos carbone du cadeau + infos carbone du prospect (chaînage des appels ou lancement en parallèle) → callback : appel de la fonction de calcul une fois toutes les réponses arrivées (si appel en parallèle, vérifier que toutes les données des 2 réponses sont disponibles) → affichage du bilan carbone

# Partiel de Programmation Web

Licence 3<sup>ème</sup> année

Mars 2016

Le but de ce partiel est de créer le site web suivant :

**Etudiant**

- Nom :
- Prénom :
- Numéro d'étudiant :

**Sujet de l'examen**

**Le HTML est un langage de programmation**  
vrai  faux

**Le PHP est un langage de programmation**  
vrai  faux

**Je vais avoir 20/20 à cet examen**  
vrai  faux

**Vérier**

## Aucun document n'est autorisé

**Exercice 1. Squelette HTML** *Écrire une page web contenant les informations suivantes :*

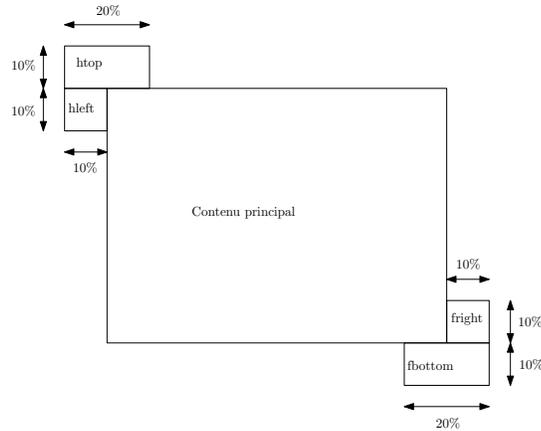
— les métadonnées :

1. Le titre de la page est "Sujet du Partiel".
2. La page est encodée en utf-8.
3. Préciser le nom de l'auteur de la page (vous êtes cet auteur).
4. Inclure les fichiers css suivants : position.css et qcm.css. Ces deux fichiers se trouvent dans le répertoire CSS.

— le contenu :

1. deux en-têtes vides ayant chacun un identifiant (htop et hleft).
2. Le contenu **principal** contient un article, contenant lui-même deux sections.
  - une section dont l'en-tête est "Etudiant". Le contenu de la section est une liste contenant votre nom, votre prénom et votre numéro d'étudiant.
  - une section dont la classe est qcm, dont l'entête est "Sujet de l'examen". Le contenu sera défini en PHP ultérieurement.
3. deux pieds de page vides ayant chacun un identifiant (ftop et fleft).

**Exercice 2. CSS : positionnement** *Dans cet exercice, on souhaite écrire une feuille de style permettant de positionner le contenu principal, les en-têtes et les pieds de pages. Vous devrez réunir **au maximum** les propriétés communes à plusieurs balises dans une seule et même règle.*



- attention, les en-têtes et les pieds de pages qui apparaissent dans une règle sont uniquement ceux qui sont des descendants direct de la balise body.
- la position de toutes ces balises est calculée par rapport aux bords du navigateur et aucune d'entre elles ne bouge si l'utilisateur fait défiler la page.
- Les en-têtes et les pieds de pages ont une couleur de fond vert foncé.
- Le contenu principal a une bordure noire de 2 pixels de large.
- Le contenu principal prend 80% de la page en hauteur et en largeur, bordure incluse.

**Exercice 3. PHP : la classe QCM** Écrire le fichier qcm.php contenant une classe nomée QCM, qui possède au moins deux variables, questions et réponses, et les méthodes suivantes :

1. ajouter\_question qui prend en entrée une question et sa réponse et l'ajoute au qcm.
2. un constructeur qui ne prend rien en entrée et ajoute les trois questions (voir page 1 du sujet) et les réponses adéquates à l'aide de la méthode ajouter\_question.
3. afficher\_QCM qui crée un formulaire HTML contenant toutes les questions contenues dans la classe QCM et tel que :
  - les réponses du formulaire ne doivent pas être visibles dans l'url,
  - le script PHP appelé par le formulaire est verif.php (à définir plus tard),
  - chaque question est encapsulée dans une balise générique de type bloc, dont la classe est qcm\_question,
  - le bouton de validation a pour identifiant qcm\_submit.
4. vérifier\_QCM ne prend rien en entrée et utilise la variable superglobale adéquate afin de compter le nombre de bonnes réponses remplies dans le formulaire. La fonction affiche un paragraphe en HTML contenant le nombre de bonne réponses.

**Exercice 4. CSS : le qcm** Dans cet exercice, on souhaite écrire une feuille de style relative au contenu principal contenu principal,

- un article possède une marge extérieure de 50 pixels en haut et 30 pixels sur la gauche,
- les en-têtes des sections sont en italique et soulignés,
- le texte de l'en-tête d'un QCM est centré,
- les questions du qcm sont séparés par 15 pixels,
- la marge intérieure de **toutes les balises** incluse dans une question du qcm est de 15 pixels.
- la première ligne d'une question du qcm est en gras.
- lorsque le curseur de la souris passe sur une question, sa couleur de fond devient vert clair.
- le bouton de validation du qcm est rouge, le texte est de couleur blanche et de taille large. Par défaut, le bouton a une bordure rouge d'une largeur de 2 pixels.
- lorsque l'utilisateur clique sur le bouton, le texte et la bordure deviennent noir .

**Exercice 5. PHP : les scripts**

1. Écrire le script PHP qui se trouve dans la deuxième section de l'article. Ce script crée un qcm et l'affiche.
2. Écrire le script verif.php qui appelle la méthode vérifier\_QCM.

## CORRECTION EXAMEN

- Q.C.M individuel sur feuille (aucun document autorisé), durée : 20 minutes
- Les réponses sont à remplir sur ce document qui doit être rendu
- Une seule réponse correcte par question, et donc une seule réponse autorisée! (sinon 0 pour la question)
- Une réponse correcte = 1 point (total sur 20)

**Nom :**

**Prénom :**

**Groupe :**

Questions	Réponses
1. Quel élément HTML permet d'intégrer du JavaScript dans une page ?	<input type="checkbox"/> <js> <input type="checkbox"/> <javascript> <input checked="" type="checkbox"/> <script> <input type="checkbox"/> <scripting>
2. Quelle est la syntaxe correcte pour référencer un script externe appelé monScript.js dans une page ?	<input checked="" type="checkbox"/> src="monScript.js" <input type="checkbox"/> link="monScript" <input type="checkbox"/> name="monScript" <input type="checkbox"/> href="monScript.js"
3. Où insère-t-on du JavaScript (code ou lien) dans une page ?	<input type="checkbox"/> <head> <input type="checkbox"/> <body> <input checked="" type="checkbox"/> <head> et <body> <input type="checkbox"/> <meta>
4. Comment ajouter un commentaire en JavaScript ?	<input type="checkbox"/> <!-- Ceci est un commentaire --> <input type="checkbox"/> " Ceci est un commentaire " <input type="checkbox"/> # Ceci est un commentaire # <input checked="" type="checkbox"/> /* Ceci est un commentaire */
5. En JavaScript, comment déclarer une variable maVariable pour stocker une valeur de type Number ?	<input type="checkbox"/> var Number maVariable ; <input checked="" type="checkbox"/> var maVariable = new Number() ; <input type="checkbox"/> Number maVariable ; <input checked="" type="checkbox"/> var maVariable ;
6. Que renvoie l'exécution du code message="bonjour" ; alert(typeof message) ; ?	<input type="checkbox"/> boolean <input type="checkbox"/> "typeof message" <input type="checkbox"/> undefined <input checked="" type="checkbox"/> string
7. Qu'affiche le code alert("10"+"5") ; ?	<input type="checkbox"/> "10"+"5" <input type="checkbox"/> 15 <input checked="" type="checkbox"/> 105 <input type="checkbox"/> "15"
8. Qu'affiche le code isNaN("20") ; ?	<input type="checkbox"/> 20 <input type="checkbox"/> "20" <input type="checkbox"/> true <input checked="" type="checkbox"/> false
9. Que renvoie l'exécution du code alert(4 >= 4 && (2 > 3    (6%3 < 1))) ; ?	<input type="checkbox"/> erreur <input type="checkbox"/> undefined <input checked="" type="checkbox"/> true <input type="checkbox"/> false
	<i>suite sur la page suivante...</i>

Questions	Réponses
10. Comment écrit-on en JavaScript la condition pour représenter l'alternative <b>Si x est égal à 4</b> ?	<input type="checkbox"/> if (x = 4) then <input type="checkbox"/> if x equal 4 <input checked="" type="checkbox"/> if (x == 4) <input type="checkbox"/> if x = 4 then
11. Comment écrit-on en JavaScript la condition pour représenter l'alternative <b>Si x n'est pas égal à 2</b> ?	<input type="checkbox"/> if (x <> 2) <input checked="" type="checkbox"/> if (x != 2) <input type="checkbox"/> if x <> 2 <input type="checkbox"/> if (x != 2) then
12. Quelle est la syntaxe correcte en JavaScript pour l'instruction de boucle <b>for</b> ?	<input type="checkbox"/> for i = 1 to 4 <input type="checkbox"/> for (i = 0; i <= 4) <input type="checkbox"/> for (i <= 4; i++) <input checked="" type="checkbox"/> for (i = 0; i <= 4; i++)
13. Quelle est la syntaxe correcte en JavaScript pour l'instruction de boucle <b>while</b> ?	<input type="checkbox"/> while i = 1 to 7 <input checked="" type="checkbox"/> while (i <= 7) <input type="checkbox"/> while (i <= 7; i++) <input type="checkbox"/> while (i = 0; i <= 4; i++)
14. Comment créer une fonction en JavaScript qui se nomme <b>maFonction</b> ?	<input type="checkbox"/> function = maFonction() {}; <input type="checkbox"/> function : maFonction() {}; <input checked="" type="checkbox"/> function maFonction() {}; <input checked="" type="checkbox"/> maFonction = new function() {}; 
15. En JavaScript, comment exécuter une fonction déjà créée qui s'appelle <b>maFonction</b> ?	<input type="checkbox"/> call function maFonction(); <input type="checkbox"/> call maFonction(); <input checked="" type="checkbox"/> maFonction(); <input type="checkbox"/> maFonction.run();
16. Si je veux changer l'url de la page courante, le composant <b>BOM (Browser Object Model)</b> inclus dans JavaScript me propose l'objet :	<input type="checkbox"/> history <input checked="" type="checkbox"/> location <input type="checkbox"/> navigator <input type="checkbox"/> screen
17. En JavaScript, quelle propriété me permet de modifier l'élément HTML <code>&lt;p&gt;bonjour&lt;/p&gt;</code> ?	<input checked="" type="checkbox"/> document.getElementsByTagName("p").innerHTML <input type="checkbox"/> document.getElementsByName("p").innerHTML <input type="checkbox"/> document.getElementsByTagName("p").content <input type="checkbox"/> document.getElementById("p").content
18. En JavaScript, comment modifier la couleur de fond de l'élément HTML initialisé par <code>elem = document.getElementById("myElement")</code> ?	<input checked="" type="checkbox"/> elem.style.backgroundColor="blue"; <input type="checkbox"/> elem.background-color="blue"; <input type="checkbox"/> elem.style.background-color="blue"; <input type="checkbox"/> elem.backgroundColor="blue";
19. Cliquer avec la souris sur le bouton de l'élément HTML <code>&lt;input type="button" onclick="alert("A")" onmouseover="alert("B")"/&gt;</code> affiche à l'écran :	<input type="checkbox"/> Rien du tout <input checked="" type="checkbox"/> Uniquement B <input type="checkbox"/> B, puis ensuite A <input type="checkbox"/> A, puis ensuite B
20. Parmi ces évènements, quel est l'intrus?	<input type="checkbox"/> load <input type="checkbox"/> change <input type="checkbox"/> focus <input checked="" type="checkbox"/> click

## MIF 13 – Programmation Web – Examen

Durée : 1 heure 30 – Documents autorisés – Ordinateurs, calculatrices et téléphones portables interdits

### Questions de cours (barème : 10 points)

1. Côté serveur, à quoi correspond la partie « Vue » du modèle MVC lors du traitement d'une requête AJAX ? Quel type d'objet correspond le mieux à son implémentation ?
2. Quelle est la différence fondamentale de fonctionnement entre une redirection HTTP et l'utilisation de la méthode `forward()` d'un `RequestDispatcher` ?
3. À quoi cela sert-il d'écrire du JSON dans la partie droite de l'affectation d'une variable en JS ?
4. Si vous deviez utiliser en même temps GWT et jQuery dans une application, à quel problème vous heurteriez-vous ? Citez un exemple de fonctionnalité qui fait doublon entre ces deux outils.
5. En quoi peut-on dire qu'une fonction de callback semble se dérouler dans un thread séparé de celui qui l'a appelée ?

### Étude de cas (barème : 11 points)

Comme chaque année, l'UCBL propose à ses usagers un service de cartes de vœux électroniques. Cette année, c'est vous qui êtes chargé(e) de concevoir cette application Web. Cette application fonctionne de la façon suivante : un utilisateur se connecte à l'application par le CAS. Une fois authentifié, il arrive sur la page d'accueil. De là, il peut visualiser l'aspect général de la carte de vœux et a accès à un lien lui permettant d'envoyer cette carte par messagerie électronique. S'il clique sur ce lien, il accède à un formulaire qui possède les champs suivants :

- Adresse de l'expéditeur (déjà remplie en fonction de l'adresse mail de l'utilisateur dans l'annuaire LDAP et non modifiable)
- Adresse du destinataire
- Titre de la carte
- Texte de la carte
- Choix de la langue de la carte (« français » ou « anglais »)
- Case à cocher pour recevoir ou non un message de confirmation
- Bouton envoyer

S'il clique sur le bouton envoyer, la carte est envoyée à l'adresse indiquée et une page de confirmation / erreur d'envoi s'affiche. S'il a demandé une confirmation, il reçoit un message électronique lui indiquant le résultat de cette opération.

Pour réaliser cette application, vous disposez des composants suivants :

- Une classe `Carte` qui ne possède qu'une méthode statique nommée `getHtml(String titre, String texte, String langue)`. Cette méthode génère le code HTML de la carte de vœux à envoyer.
- Une classe `Message` possédant les champs :
  - o nom de l'expéditeur
  - o adresse du destinataire
  - o titre du message (peut être le titre de la carte, mais pas nécessairement)
  - o texte (ou code HTML) du messageLa classe `Message` contient également une méthode `envoyer()` qui s'occupe de l'envoi par messagerie électronique.
- Une classe `User` qui représente l'utilisateur authentifié. Cette classe permet d'accéder à l'ensemble des champs de l'annuaire LDAP pour l'utilisateur. Pour l'instancier, il suffit d'appeler son constructeur, avec en paramètre le login de l'utilisateur. Les champs sont ensuite accessibles à l'aide d'accesseurs (« getters »).

**Cette application ne s'appuiera pas sur un framework et mettra en œuvre un pattern Web MVC push-based.**

### Conception (barème : 4 points)

6. Listez les différents éléments nécessaires au fonctionnement de votre application, en indiquant leur nature : pages statiques, scripts, servlets, JSP, JavaBeans (précisez le scope)...
7. Décrivez, à l'aide de diagrammes UML appropriés, leurs communications lorsqu'un utilisateur authentifié clique sur le bouton de validation du formulaire, à partir de la réception des requêtes du client. Les éléments du SI externes à votre application seront modélisés comme des boîtes noires.

### Structure de l'application (barème : 8 points)

8. Écrivez le contrôleur de cette application.

### Programmation (barème : 3 points)

Vous allez maintenant rajouter une partie dynamique à cette application : lorsque l'utilisateur tape des caractères dans le champ destinataire, l'application lui suggère des personnes parmi ses contacts. Pour cela, vous disposez d'un composant côté serveur qui interroge la liste de contacts de l'utilisateur et répond aux requêtes GET sur l'URL « `getContacts?start=xxx` », en renvoyant les contacts qui commencent par la valeur du paramètre « `start` ». Vous n'avez pas à vous préoccuper de son implémentation.

La réponse est un document XML dont la DTD est :

```
< !ELEMENT liste (contact)*>
< !ELEMENT contact (#PCDATA)>
< !ATTLIST contact adresse CDATA #REQUIRED>
```

9. Écrivez la fonction JavaScript de callback qui :

- affiche l'ensemble des contacts suggérés dans un div dont l'ID est « `suggestions` » et rend ce div visible
- fait en sorte que chacun de ces contacts soient cliquables et que lorsque l'utilisateur clique sur l'un d'eux, le div « `suggestions` » devienne invisible, et l'adresse de messagerie du contact soit ajoutée au champ d'ID « `destinataire` ».

# Examen de programmation web

## Formations par alternance

ENSIIE - 2014/2015

### Partie 1 - Architectures web

**Question 1** - Qu'est-ce que le world wide web ?

- un système de réseaux d'ordinateurs interconnectés
- un programme permettant d'afficher des fichiers multimédia
- un système hypertexte fonctionnant sur internet
- un système identifiant un document ou un fragment de document

**Question 2** - Qu'est-ce qu'un site dynamique ?

- un site dont le contenu évolue avec le temps
- un site dont le contenu est fixé dans les fichiers du site
- un site dont les pages contiennent des animations
- un site dont le contenu peut être modifié sans modification du code source des pages

**Question 3** - Quels sont les tiers intervenant dans l'architecture *trois tiers* d'une application web ?

- client-application-ressource
- client-application-serveur
- navigateur-client-serveur
- navigateur-application-serveur

### Partie 2 - HTML

**Question 4** - Le langage HTML a pour objectif principal de spécifier :

- la structuration d'un document web
- la mise en forme d'un document web
- l'accès aux données d'un document web
- les langages utilisés dans un document web

**Question 5** - Quelle balise HTML contient les informations destinées au navigateur et aux moteurs de recherche ?

- body
- head
- header
- content

**Question 6** - Parmi les balises HTML suivantes, laquelle est une balise vide (autofermante) ?

- p
- img
- table
- tr

**Question 7** - Quelle est la balise permettant d'indiquer un titre de niveau le plus haut (titre principal de la page) dans une page HTML ?

- header
- h1
- h6
- p

**Question 8** - À quoi sert la ligne `<meta charset="utf-8">` ?

- à indiquer le type du document au navigateur
- à préciser l'encodage des mots clés de la page web
- à transmettre un ensemble de caractères au serveur web
- à indiquer au navigateur le jeu de caractères utilisé dans la page web

**Question 9** - Quelle ligne correspond à une ligne de tableau en HTML ?

- `<th><td>Jeanne</td><td>Dupond</td></th>`
- `<table><td>Jeanne</td><td>Dupond</td></table>`
- `<tr><td>Jeanne</td><td>Dupond</td></tr>`
- `<tr><tc>Jeanne</tc><tc>Dupond</tc></tr>`

**Question 10** - En HTML5, quelle balise vaut-il mieux utiliser pour déclarer un menu de navigation ?

- `<nav>`
- `<section>`
- `<div class="menu">`
- `<span>`

## Partie 3 - CSS

**Question 11** - Quelle utilisation HTML du sélecteur CSS `alert` défini à la ligne suivante est correcte ?  
`.alert{color : red;}`

- `<p style="alert">Attention</p>`
- `<p.alert>Attention</p>`
- `<alert>Attention</alert>`
- `<p class="alert">Attention</p>`

**Question 12** - Comment changer la taille des légendes de tableau pour qu'elles s'affichent en 20px ?

- `caption{char-size : 20px}`
- `caption{size : 20px}`
- `caption{text-size : 20px}`
- `caption{font-size : 20px}`

## Partie 4 - PHP

**Question 13** - Dans une balise de formulaire, comment indiquer la page à laquelle les informations entrées par l'utilisateur seront envoyées ?

- `<form link="cible.php">`
- `<form target="cible.php">`
- `<form action="cible.php">`
- `<form href="cible.php">`

**Question 14** - Qu'affichent les lignes PHP suivantes :

```
$boisson=vin; echo "il a goûté plusieurs ${boisson}s";
```

- il a goûté plusieurs `${boisson}s`
- il a goûté plusieurs boissons
- il a goûté plusieurs vin

- il a goûté plusieurs vins

**Question 15** - Si l'on a déclaré un tableau associatif PHP comme sur la ligne suivante, comment afficher le réalisateur de `vertigo` ?

```
$realisateurs=array('vertigo' => 'Hitchcock', 'alien' => 'Scott', 'kagemusha' => 'Kurosawa');
```

- `echo $realisateurs[$vertigo];`
- `echo $realisateurs['vertigo'];`
- `echo realisateurs['vertigo'];`
- `echo $realisateurs.vertigo;`

**Question 16** - Quelle fonction PHP permet d'établir une connexion avec une base de données PostgreSQL ?

- `pg_connect`
- `pg_connexion`
- `pg-connect`
- `pg-query`

**Question 17** - Quel est l'intérêt d'utiliser l'extension PDO de PHP ?

- les accès aux données sont indépendants du SGBD utilisé
- l'accès aux données est indépendant du langage utilisé par le serveur web
- les erreurs d'accès à la base de données s'affichent
- la connexion à la base de données est automatique

**Question 18** - Dans une application web respectant l'architecture MVC, dans quelle partie placer une fonction récupérant des informations dans la base de données ?

- modèle
- vue
- contrôleur
- aucune des trois

**Question 19** - De quel côté les codes Javascript et PHP sont-ils respectivement exécutés ?

- côté client pour Javascript et serveur pour PHP
- côté serveur pour Javascript et client pour PHP
- côté serveur pour les deux
- côté client pour les deux

**Question 20** - Dans un site web, une même page peut-elle être incluse dans plusieurs autres pages ?

- oui avec la fonction `include` de PHP
- oui c'est automatique si les fichiers sont dans le même dossier
- non
- oui si les noms des fichiers commencent par les mêmes caractères

**Question 21** -

## Partie 5 - UML

**Question 22** - Quel diagramme permet de décrire le comportement attendu d'un système du point de vue des utilisateurs ?

- un diagramme de classes
- un diagramme de séquence système
- un diagramme d'état
- un diagramme de cas d'utilisation

**Question 23** - Quel diagramme UML permet de décrire le déroulement temporel d'un cas d'utilisation ?

- un diagramme de classes
- un diagramme de séquence système

- un diagramme d'état
- un diagramme de cas d'utilisation

**Question 24** - La relation entre deux cas d'utilisation peut être tout sauf :

- une inclusion
- une réalisation
- une généralisation
- une extension

**Question 25** - Les relations entre acteurs et cas d'utilisation sont des :

- extensions
- réalisations
- généralisations
- associations



**Sujet UE NFA016 : Développement web (1)**

Année universitaire 2016 – 2017  
Examen première session : 06/02/2017  
Responsable : Olivier PONS

Durée : 3 heures

Tout document *papier* autorisé. Tout support électronique est interdit : pas d'ordinateur, de tablette, de liseuse...

**Les téléphones mobiles et autres équipements communicants doivent être éteints et rangés dans les sacs pendant toute la durée de l'épreuve.**

*Le barême est donné à titre indicatif; il est susceptible de modifications.*

Sujet de **13** pages, celle-ci comprise.

## Exercice 1 3 points

Écrivez le code HTML permettant de réaliser la page représentée sur la figure ci-après. **Toute balise ouvrante doit être explicitement fermée.**

On précise que

- la photo du cnam, est dans le fichier `cnam.png`, situé dans le dossier `images` (lui-même situé dans le même dossier que votre page html)
- l'en-tête contient le code `css` suivant (vous n'avez pas besoin de le recopier) :

```
table, th, td {border: solid 2pt black;}
```

C'est la seule règle `css`, le reste correspond au rendu par défaut de la structure décrite par le code HTML

## Le CNAM Paris



L'entrée du CNAM :

Le Cnam Paris est situé **rue Saint Martin**.

Le Cnam délivre des formations dans de nombreux domaines :

- Sciences dures:
  - Mathématiques ;
  - Physique ;
  - Biologie ;
- Sciences Humaines : ...

### Un tableau

Formations

	Tronc commun	Spécialisation	
		Web	Mobile
Parcours	Java	HTML	Android
		PHP	

## correction

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Exercice 1</title>
    <style>
      table, td, th {border: solid 1px black;border-collapse: collapse; padding: 2px;}
    </style>
  </head>
  <body>
    <h1> Le CNAM Paris</h1>
    <p>L'entrée du CNAM : </p>
    <p>Le Cnam Paris est situé <strong>rue Saint Martin.</strong></p>
    <p>Le Cnam délivre des formations dans de nombreux domaines : </p>
    <ul>
      <li> Sciences dures:
        <ul>
          <li> Mathématiques ;</li>
          <li> Physique ; </li>
          <li> Biologie ;</li>
        </ul>
      </li>
      <li> Sciences Humaines : ... </li>
    </ul>
    <h2> Un tableau </h2>
    <table>
      <caption>Formations</caption>
      <tr>
        <th rowspan=' 4' >Parcours</th>
        <th>Tronc commun</th>
        <th colspan=' 2' >Spécialisation</th>
      </tr>
      <tr>
        <td></td>
        <th>Web</th>
        <th>Mobile</th>
      </tr>
      <tr>
        <td rowspan=' 2' >Java</td>
        <td>HTML</td>
        <td rowspan=' 2' >Android</td>
      </tr>
      <tr>
        <td></td>
        <td></td>
        <td></td>
      </tr>
    </table>
  </body>
</html>
```

```

        <td>PHP</td>
    </tr>
</table>
</body>
</html>

```

## Exercice 2 2 points

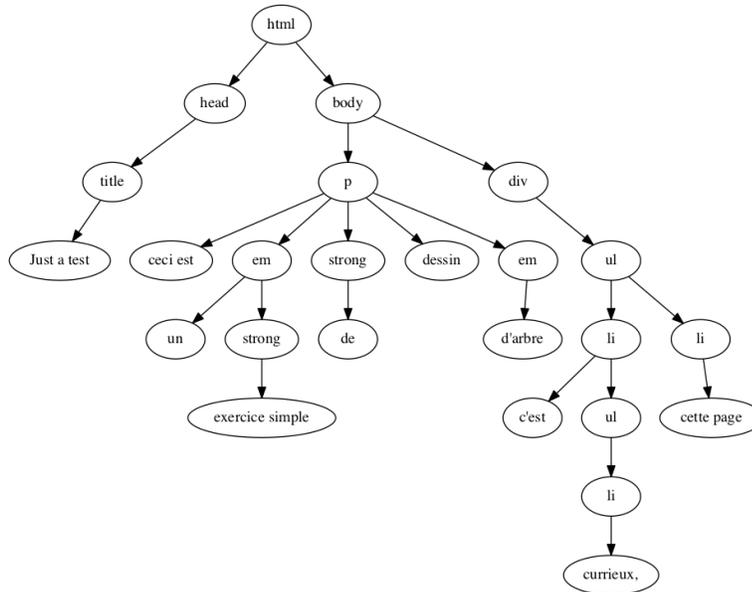
Dessinez, sans oublier les nœuds texte l'arbre correspondant au code source HTML suivant :

```

<!doctype html>
<html><head><title>Just a test</title>
</head><body><p>Ceci est <em>un <strong>exercice</strong>
simple </em><strong>de</strong> dessin <em>
d' arbre!</em></p>
<div><ul><li> c'est...<ul><li> curieux,</li></ul>
</li><li>cette page.</li></ul></div>
</body>
</html>

```

### correction



## correction

### Exercice 3 4 points

Dans les questions qui suivent, vous ne pouvez pas modifier le HTML.

#### Question 3.1 0,5 points

Soit le fragment de HTML :

```
<p class="question">Quelle est la valeur de PI ?</p>
<p class="reponse">3.14 ?</p>
```

proposez les règles CSS pour que le paragraphe qui comprend la réponse à la question n'apparaisse pas.

#### Question 3.2 1 point

Si j'ai le texte suivant :

```
<p id="enigme" class="question">Qu'ai-je dans ma poche ?</p>
<p>Sa main ?</p>
```

Et la CSS suivante :

```
p {font-weight: bold;}
p {color: blue;}
p.question {color: red;}
#enigme {color: black;}
```

1. Comment sera le premier paragraphe ?
2. Comment sera le second paragraphe ?

#### Question 3.3 1,5 points

Étant donné le fragment de code HTML suivant

```
<h1> Exercices </h1>
<div class="exercice">
  <h1>Applications</h1>
  <p> dans cet exercice, on écrira des CSS</p>
  <div>Une remarque...</div>
  <div>
    <div> Ce texte est en vert</div>
  </div>
</div>
<p>Une remarque inoffensive</p>
```

proposez les règles CSS pour que :

- “Applications” soit en italiques ;
- tout le texte de l’exercice soit en rouge ;
- sauf le texte “ce texte est en vert” qui sera en vert.

### Question 3.4 1 points

Soit les css :

```
div {border: solid 1px black; margin: 1mm;}  
.s {display: inline-block;}
```

et le html :

```
<div>  
  <div>  
    <h1>titre</h1>  
    <p> texte...</p>  
  </div>  
  <p> un texte...</p>  
  <div class="s">  
    <h1>titre</h1>  
    <p> texte...</p>  
  </div>  
  <div class="s">  
    <h1>titre</h1>  
    <p> texte...</p>  
  </div>  
</div>
```

Dessinez la page html telle qu’elle apparaîtra dans un navigateur. On rappelle que la largeur d’un élément inline-block est égale à la largeur de son contenu.

### correction

#### question 1

```
.reponse {display:none}
```

ou

```
.reponse {visibility:hidden}
```

#### question 2

1. noir, gras
2. bleu gras

### question 3

```
.exercice {color: red;}
.exercice h1 {font-style: italic;}
.exercice div div {color: green;}
```

### question 4

<b>titre</b> texte...	
un texte...	
<b>titre</b> texte...	<b>titre</b> texte...

## Exercice 4 Javascript (6 points)

On donne le code HTML suivant :

```
<html>
  <head>
    <title>Répartition des votes</title>
    <style>
      td:nth-child(3) {border:solid;width:90%}
      td:nth-child(3) div {width:0}
    </style>
    <script>
function EstInf100(n1,n2,n3){
  //A COMPLETER (Q1)
}
function redimensionne(id,n){
  //A COMPLETER (Q2)
}
    </script>
  </head>
  <body>
    <table>
      <tr>
        <th> Candidat 1</th>
        <td><input type="text" id="c1" size="3"/></td>
        <td><div id="vc1" style="background-color:blue;"></div></td>
      </tr>
      <tr>
        <th> Candidat 2</th>
```

```

        <td><input type="text" id="c2" size="3"/></td>
        <td><div id="vc2" style="background-color:yellow;"></div></td>
    </tr>
    <tr>
        <th> Candidat 3</th>
        <td><input type="text" id="c3" size="3" /></td>
        <td><div id="vc3" style="background-color:green;"></div></td>
    </tr>
</table>
<button onclick="visualiser()">visualiser</button>
</body>
</html>

```

### Question 4.1 0.5 points

Complétez la fonction javascript `EstInf100` qui prend trois arguments numériques et renvoie `true` si leur somme est inférieure ou égale à 100 et `false` sinon.

Attention, on ne demande pas d'affichage ; la fonction renvoie juste une valeur `true` ou `false`.

### Question 4.2 1 points

Complétez la fonction `redimensionne` qui prend en argument un `id` et un nombre `n` et assigne `n%` comme largeur (`width`) de l'élément désigné par l'`id`.

La figure ci-dessous montre le résultat de l'appel : `redimensionne("vc3", 100)` ;

Avant l'appel de fonction

Après l'appel, on peut visualiser que l'élément a désormais une largeur de 100% grâce au fond vert qui occupe tout la largeur du parent.

### Question 4.3 2 points

Écrivez une fonction `verification1` qui prend en argument l'`id` d'une zone de saisie (`input`) et vérifie que la valeur entrée est

- un nombre
- positif

— inférieure ou égale à 100;

Si ce n'est pas le cas, la valeur est remplacée par 0 dans la zone de saisie correspondante et on ajoute un fond rouge à la zone. Si c'est le cas on garde ou on remet un fond blanc sur la zone de saisie.

Dans tous les cas la valeur (ou 0 si il y a eu remplacement) est renvoyée

Avant première ligne vide et dernière ligne non numérique

Après première et dernière lignes à 0 sur fond rouge

### Question 4.4 2.5 points

Ecrire une fonction `visualiser` qui appelle la fonction `verification1` sur chacune des trois zones de saisie et vérifie que la somme des trois nombres renvoyés est inférieure à 100.

Si ce n'est pas le cas elle affichera (dans un alert) un message d'erreur pour le signaler et terminera, sinon elle redimensionnera chacun des `div` (d'id respectif `vc1,vc2,vc3` en fonction des valeurs saisies

### correction

```
function EstInf100(n1,n2,n3){  
    return n1+n2+n3 <=100;  
}
```

```
function redimensionne(id,n){  
    document.getElementById(id).style.width=n+"%";  
}
```

```
function verification1(ci){  
    var c=document.getElementById(ci);
```

```

    var cv=parseFloat(c1.value);
    if(isNaN(cv) || cv <0 || cv >100) {
        cv=0;
        c.value=cv;
        c.style.backgroundColor="red";
    }else {
        cc.style.backgroundColor="white";
    }
    return cv;
}

function visualiser(){
    var res=[];
    for(var i=1;i<=3;i++){
        res[i]=verification1("c"+i);
    }
    if( !EstInf100(res[1],res[2],res[3])){
        alert("la_somme_des_pourcentage_est_>_a_100_%%:" +
            res[1]+ res[2] +res[3]);
    }else{
        for(var i=1;i<=3;i++){
            redimensionne("vc"+i,res[i]);
        }
    }
}

```

## Exercice 5 Javascript (6 points)

On donne le code HTML suivant :

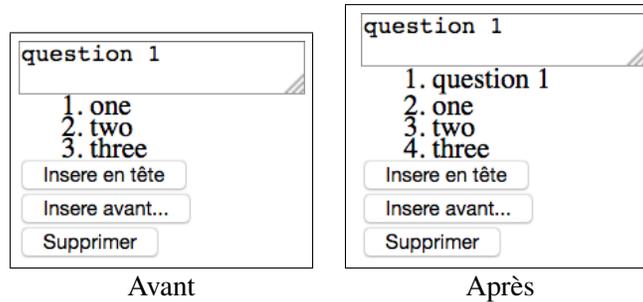
```

<body>
  <textarea id="datas"></textarea>
  <ol id="liste"><li>one</li><li>two</li><li>three</li></ol>
  <button onclick="insererEnTete()"> en tête</button><br/>
  <button onclick="insererAvant()"> avant ...</button><br/>
  <button onclick="supprimer()">Supprimer</button>
</body>

```

### Question 5.1 1 points

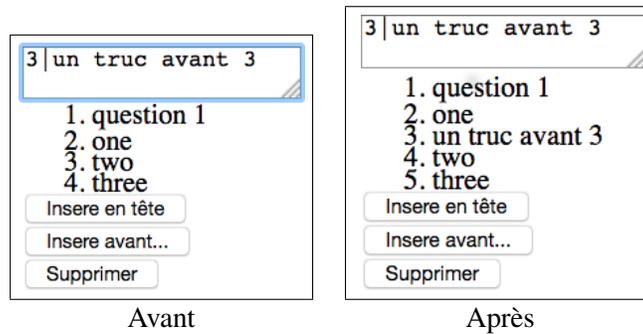
Écrivez la fonction `insererEnTete` qui récupère le texte saisi dans le `textarea` et l'insère comme premier item de la liste.



### Question 5.2 2.5 points

Écrivez la fonction `insereAvant` qui récupère le texte saisi dans le `textarea` sous la forme `indice|texte` (c'est-à-dire, un indice numerique, suivi du caractère '|', suivi de texte).

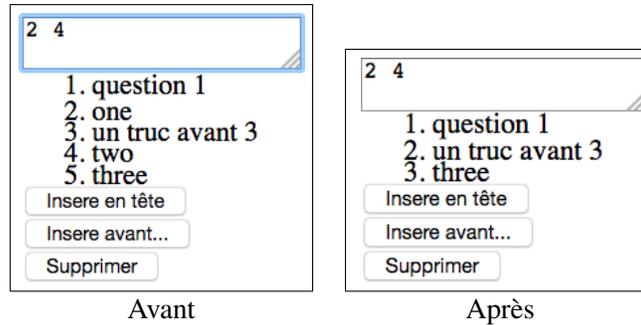
Et insère le texte dans la liste avant le l'item dont on a donné l'indice.



### Question 5.3 2,5 points

Écrivez la fonction `supprimer` qui recupere le texte saisi dans le `textarea` sous la forme `indice1 indice2 ... indicen` (c'est-à-dire, des indices numerique séparés par des espaces).

Et supprime de la liste les items correspondants à ces indices



Pour toutes ces questions on suppose que les données saisies sont toujours correctes (c'est-à-dire, au format attendu, et que les indices correspondent toujours à des items existants).

## correction

```

function insererEnTete() {
    var liste=document.getElementById("liste");
    var data=document.getElementById("datas").value;
    liste.innerHTML("<li>"+data+"</li>"+liste.innerHTML;
}

function insererAvant() {
    var liste=document.getElementById("liste");
    var data=document.getElementById("datas").value;
    var numeros=data.split("|");
    var li=document.createElement("li");
    liste.insertBefore(li,liste.childNodes[parseInt(numeros[0]-1)]);
    li.appendChild(document.createTextNode(numeros[1]));
}

function supprimer() {
    var liste=document.getElementById("liste");
    var data=document.getElementById("datas").value;
    var numeros=data.split("_");
    var l=numeros.map((x)=>(liste.childNodes[parseInt(x-1)]));;
    for(var i=0;i<numeros.length;i++){
        liste.removeChild(l[i]);
        alert(numeros[i]-1);
    }
}

```

## EXAMEN de PROGRAMMATION en PHP

Durée : 1h45

Tous documents autorisés

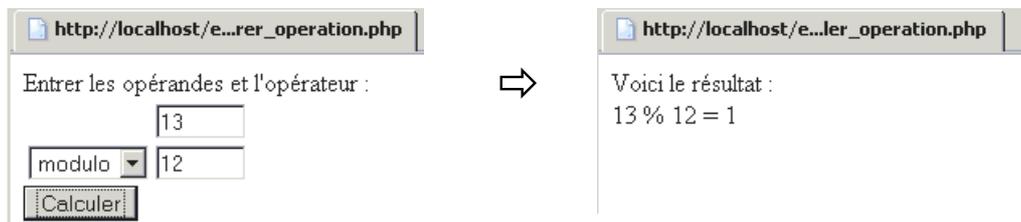
Seul le code sera ramassé et noté. Les exercices, indépendants, peuvent être traités dans un ordre quelconque.

Vous êtes vivement encouragés à détailler votre démarche en commentant abondamment le code.

### Exercice 1 : Page web dynamique par formulaire

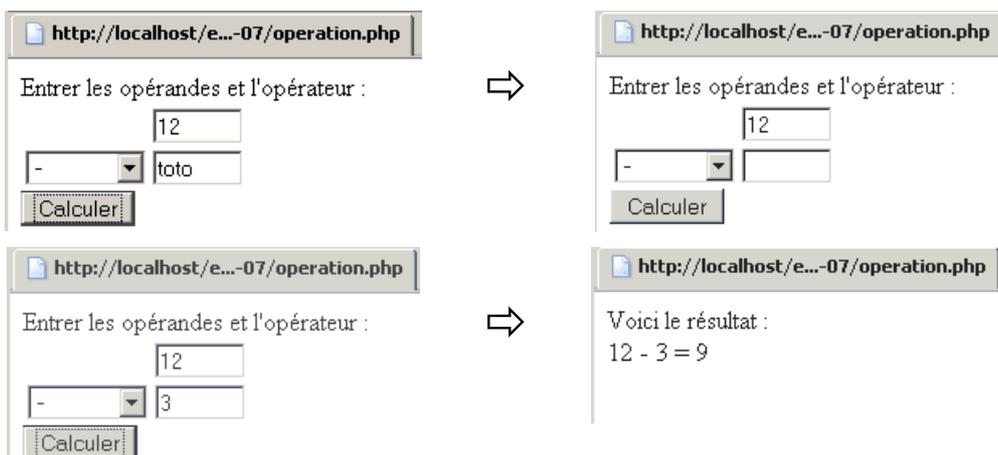
1. Écrire une petite application en ligne permettant de saisir des nombres, puis de calculer le résultat de différentes opérations sur ceux-ci. Pour cela, on écrira successivement :

- un formulaire *entrer\_operation.html*, qui permet de saisir les nombres et l'opérateur (addition, soustraction, multiplication, division ou modulo) ;
- un script *calculer\_operation.php*, qui permet de calculer et d'afficher le résultat de l'opération définie dans le formulaire précédent :



*Remarque* : Dans cette première version, on ne fera pas de test de validité des saisies de l'utilisateur (les nombres non saisis seront considérés comme égaux à 0).

2. Le programme doit être robuste aux erreurs de saisie, en proposant à nouveau à l'utilisateur la saisie des opérandes si les valeurs entrées ne sont pas numériques. Réécrire les 2 programmes précédents en un seul (*operation.php*) effectuant la même chose que précédemment, mais qui teste en plus si les saisies sont correctes. Si c'est le cas, on affiche le résultat ; sinon, on redemande d'effectuer la saisie :



## Exercice 2 : Page web dynamique fondée sur une base de données

Sous MySQL, on dispose de la base de données suivante (qui a servi de base de travail en TP). On souhaite créer une page PHP pour pouvoir changer la personne qui assure l'enseignement d'une matière :

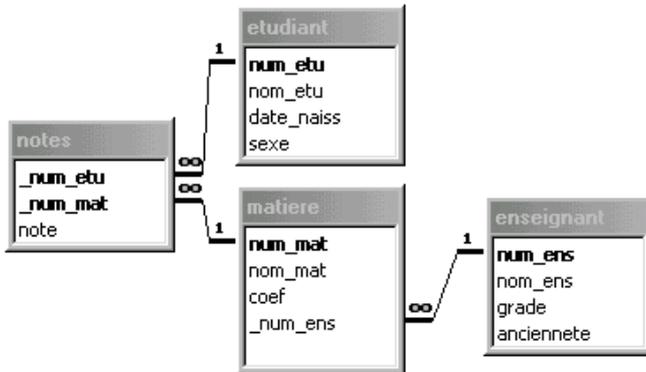


Schéma relationnel de la base

http://localhost/...6-07/mat\_ens.php

Matières et enseignants

Numéro	Nom	Coef	Enseignant
1	Programmation	3	Dupond, Auguste
2	Bases de données	2	Boyer, Julie
3	G.P.A.O.	2	Bertrand, Pierre
4	Logique	1	Simon, Etienne
5	Statistiques	1	Simon, Etienne

Modifier

Bertrand, Pierre  
Dupond, Auguste  
Aubois, Jean  
Boyer, Julie  
Vincent, Paul  
Simon, Etienne

mat\_ens.php

Pour cela, on demande :

- d'écrire une fonction PHP `select(...)` (paramètres à compléter) qui retourne une chaîne contenant le code HTML utile pour créer une liste déroulante (telle que celles de la figure ci-dessus). Cette fonction prend en paramètre le nom de l'objet liste, un tableau associatif d'éléments (sur le modèle `clé=>valeur`) et une chaîne représentant la valeur sélectionnée par défaut.

Exemple :

```
select('maliste', array(1=>'Un',2=>'Deux',3=>'Trois'), 2)
retourne la chaîne suivante :
```

```
"<select name='maliste'>
  <option value=1>Un</option>
  <option value=2 selected>Deux</option>
  <option value=3>Trois</option>
</select>"
```

- d'écrire le script `mat_ens.php` incluant la fonction précédente et le programme principal permettant de changer l'enseignant d'une matière grâce à une liste déroulante (tel que sur l'exemple ci-dessus).
- d'écrire le script de traitement (`update_mat_ens.php`) effectuant les changements d'enseignants correspondants dans la base.