

Programmation Web Dynamique (INF 232)

Fiche de Travaux Dirigés / Travaux Pratiques / Travaux Personnels de l'Étudiant
Dr KENGNE T. Vianney

PARTIE I : Les langages HTML et CSS

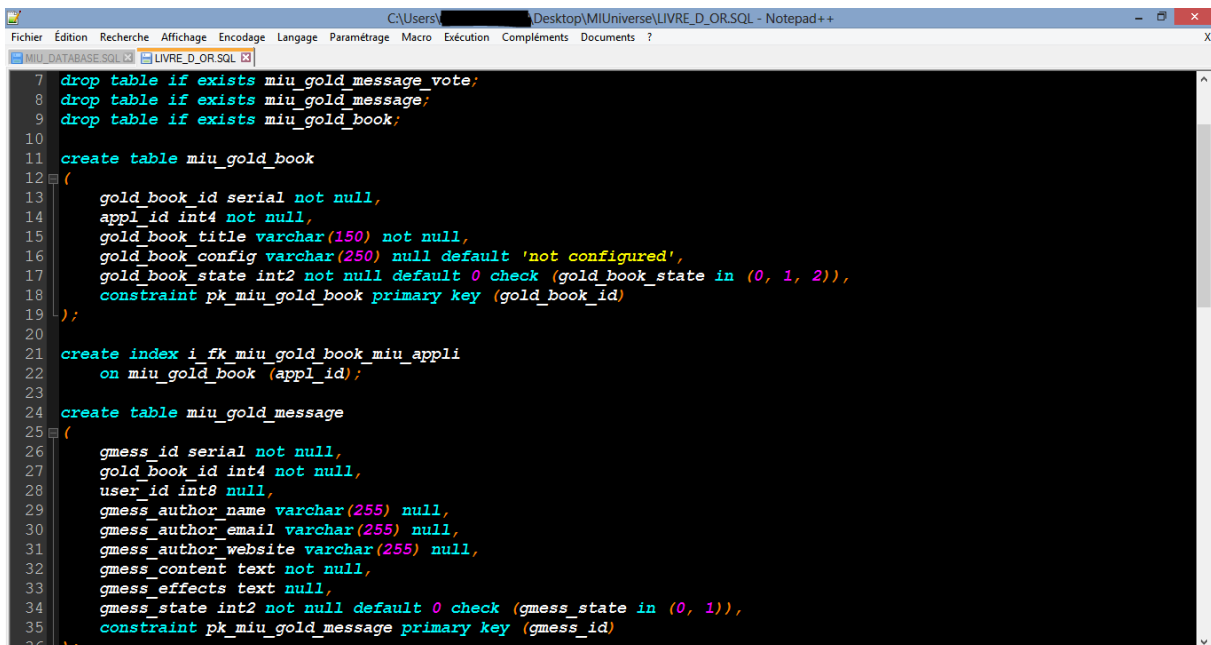
Travail 1 : Les outils de développement en HTML/CSS

Cet exercice a pour but principal de présenter aux apprenants les outils de développement et de débogage en HTML et en CSS.

1- Présenter les différentes classes de logiciel qui interviennent dans l'écriture et l'interprétation des codes HTML et CSS. Citer des exemples de logiciels pour chacune de ces classes.

2- Présenter l'éditeur *Notepad++*

- Rôle et fonctions principales (*coloration syntaxique, auto complétion, support de plusieurs langages...*)
- Coder un exemple basique (*un hello world !*) de page HTML/CSS



```

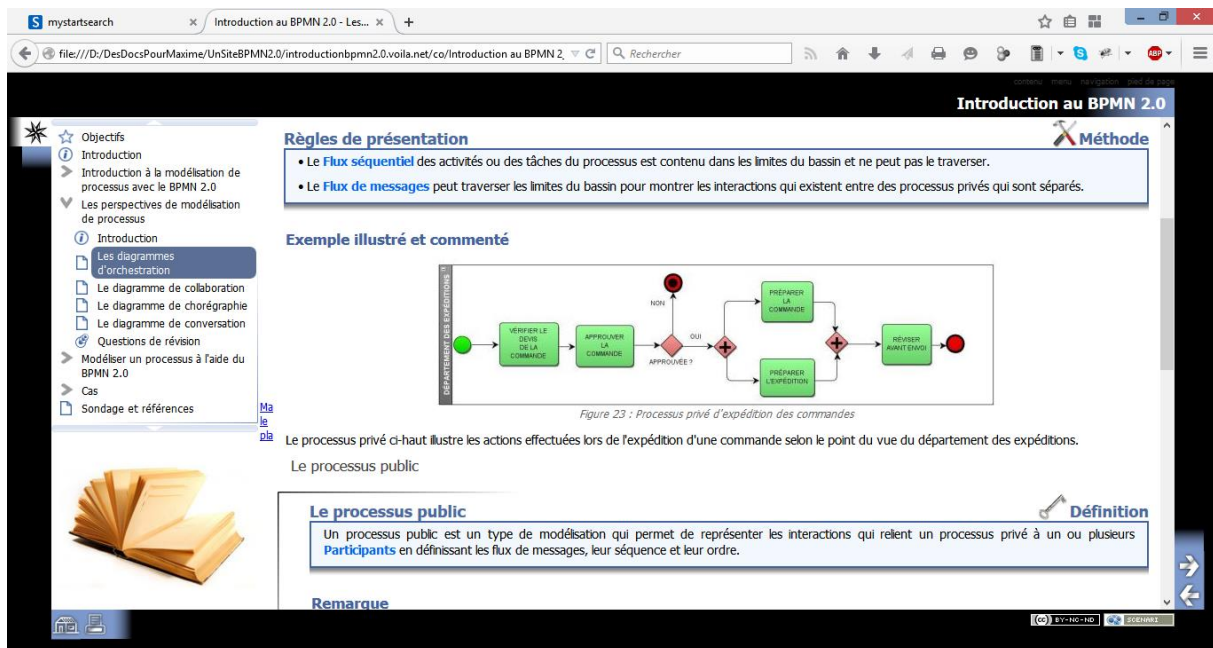
7 drop table if exists miu_gold_message_vote;
8 drop table if exists miu_gold_message;
9 drop table if exists miu_gold_book;
10
11 create table miu_gold_book
12 (
13     gold_book_id serial not null,
14     appl_id int4 not null,
15     gold_book_title varchar(150) not null,
16     gold_book_config varchar(250) null default 'not configured',
17     gold_book_state int2 not null default 0 check (gold_book_state in (0, 1, 2)),
18     constraint pk_miu_gold_book primary key (gold_book_id)
19 );
20
21 create index i_fk_miu_gold_book_miu_appli
22 on miu_gold_book (appl_id);
23
24 create table miu_gold_message
25 (
26     gmess_id serial not null,
27     gold_book_id int4 not null,
28     user_id int8 null,
29     gmess_author_name varchar(255) null,
30     gmess_author_email varchar(255) null,
31     gmess_author_website varchar(255) null,
32     gmess_content text not null,
33     gmess_effects text null,
34     gmess_state int2 not null default 0 check (gmess_state in (0, 1)),
35     constraint pk_miu_gold_message primary key (gmess_id)
36 );

```

Un code source sous *Notepad++* v6.7.5

3- Présenter le navigateur *Mozilla Firefox*

- Rôle et caractéristiques principales (*extensibilité, support, portabilité...*)
- Présenter les modules complémentaires (*addons*) en particulier l'outil de débogage *Firebug*.
- Interpréter l'exemple basique précédent grâce à *Firefox*.



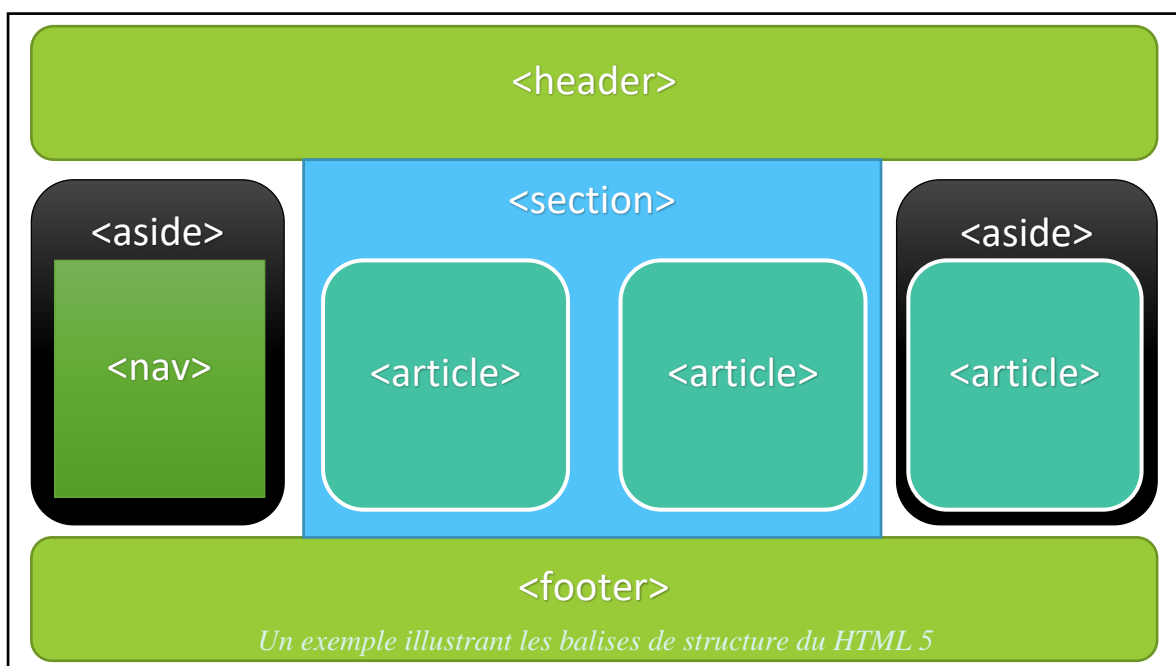
Une page HTML/CSS interprétée par Firefox 37.0.2

Travail 2 : La structure d'une page HTML/CSS

Cet exercice a pour buts principaux de :

- Présenter la structure d'un code HTML/CSS
- Présenter les différentes balises de structure d'une page HTML

- 1- Donner et commenter le code minimal d'une page HTML/CSS (page HTML contenant du CSS inclus de différentes façons (*interne* et *externe*))
- 2- Présenter les balises de structure `<div>`/`</div>` et ``/`` (insister sur les caractères de style *inline* et *block*).
- 3- Présenter et donner le rôle de chacune des balises structurantes du standard HTML 5 (`<header>`, `<nav>`, `<section>`, `<aside>`, `<article>`, `<footer>`...)



4- Identifier les balises de structure présentes dans le code de la page *HTML/CSS* de cette capture d'écran :



Page d'accueil d'un site WEB

5- Proposer deux codes sources *HTML/CSS* qui permettent d'obtenir la structure de la page ci-dessus :

- **Premier code** : Utiliser les balises `<div>` et `` pour structurer votre page.
- **Second code** : Utiliser les balises structurantes du standard HTML 5 pour la structuration de votre page. Respecter au maximum le rôle de chacune des balises.

Travail 3 : Manipulation des tableaux en HTML/CSS

Cet exercice a pour but principal de familiariser les apprenants aux procédés de création et de mise en forme des tableaux.

1- Présenter la notion de tableau en général

Gestion de la paie des étudiants en Master					
Identification		Informations salariales			Observations
Noms et Prénoms	Matricule	Heures de travail	Salaire conséquent	Salaire à percevoir	
Cellule (i,j) = Ligne (i) ∩ Colonne (j)					
Généré par The HTML Dynamic Tables Maker © 2015					

Entête du tableau

Corps du tableau

Pied du tableau

Un exemple de tableau

- 2- Présenter, en donnant le rôle de chacune d'entre elles, les balises qui permettent de créer des tableaux en *HTML/CSS* (`<table>`, `<caption>`, `<thead>`, `<tbody>`, `<tfoot>`, `<tr>`, `<th>`, `<td>`...).
- 3- Réaliser une page *HTML/CSS* dans laquelle on retrouve le tableau de la question 1 (*respecter la mise en forme et la structure du tableau*).
- 4- Modifier le code source précédent afin que l'entête du tableau soit fixe.

Travail 4 : Les formulaires en HTML/CSS

Cet exercice a pour but principal de familiariser les apprenants à la création et à la validation des formulaires grâce au *HTML/CSS*.

- 1- Présenter la notion de formulaire et déceler l'intérêt de ceux-ci.
- 2- Présenter, en donnant le rôle de chacune d'entre elles, les balises qui permettent de créer et d'enrichir des formulaires en *HTML/CSS* (`<form>`, `<input />`, `<label>`, `<textarea>`, `<button>`, `<select>`, `<optgroup>`, `<option />`, `<fieldset>`, `<legend>`...).
- 3- Présenter les différents effets causés par les valeurs de l'attribut *type* de la balise `<input />` :
 - Les zones de texte mono ligne : **text**
 - Les cases à cocher : **checkbox**
 - Les boutons radio : **radio**
 - Les boutons de soumission : **submit**, **image**
 - Les boutons de réinitialisation : **reset**
 - Les boutons sans effets : **button**
 - Les zones de mot de passe : **password**
 - Les zones de fichier : **file**
 - Les zones cachées : **hidden**
 - Les zones de saisies enrichies (*HTML 5*) : **email**, **url**, **tel**, **color**, **range**, **number**, **week**, **month**, **date**, **time**, **datetime**, **datetime-local**, **search**...
- 4- Présenter les autres attributs, leurs valeurs et leurs effets (**method**, **action**, **enctype**, **checked**, **selected**, **name**, **id**, **for**, **placeholder**, **required**, **autofocus**, **value**, **label**, **src**, **multiple**, **size**...).
- 5- Proposer un code source pour la récupération des informations sur un bachelier voulant s'inscrire en première année en Faculté des Sciences de l'université de Dschang.

Le bachelier doit fournir **ses informations d'identification** (son nom, sa date de naissance, son sexe, une mini biographie), **ses informations de contact** (email, numéro de téléphone, ville et pays de résidence), **sa formation** (plus grand diplôme obtenu, année d'obtention, note obtenue, une pièce justificative), **le cursus sollicité** (premier choix de filière, second choix de filière, troisième choix de filière, sa motivation) et **les informations de connexion** (pseudonyme, mot de passe). La mini biographie et la motivation sont optionnelles.

Le formulaire doit être muni d'un bouton de réinitialisation et d'un bouton de soumission avec image. Les données du formulaire seront contrôlées par le navigateur puis transmises suivant le standard approprié et seront traitées par la page *subscribe.php*.

Travail 5 : Le multimédia en HTML/CSS

Cet exercice a pour but principal de familiariser les apprenants à l'insertion, à la création et à la mise en forme du contenu multimédia dans une page *HTML/CSS*.

1- Présenter les différents types de médias (*images, graphiques 2D & 3D, vidéos, sons audio, programmes exécutables (flash, applet Java)...*).

2- Insérer et mettre en forme les images

- La balise ``
- La disposition des images par rapport au texte
- Créer et utiliser une carte sur votre image grâce à `<map>`

3- Insérer différents types de contenus multimédias grâce aux balises `<embed>` et `<object>`.

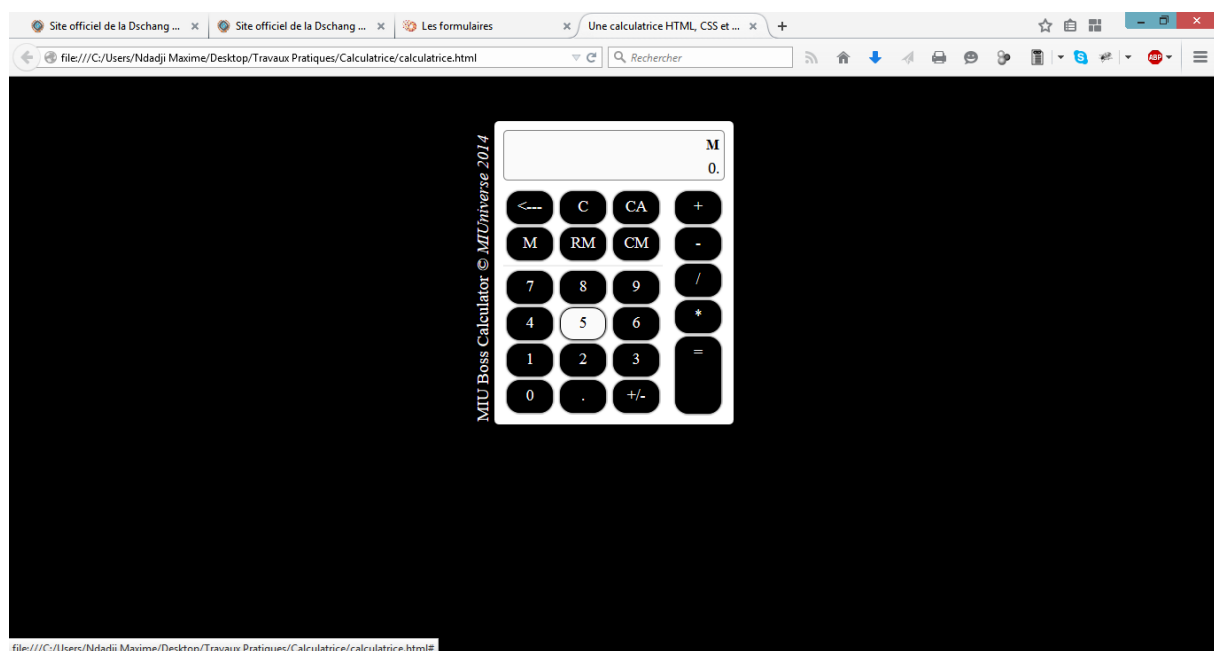
4- Insérer de l'audio et de la vidéo grâce aux balises `<audio>` et `<video>`

5- Coder une applet Java qui affiche le texte « *hello world !* » en rouge. Insérer votre applet dans une page HTML/CSS à un endroit que vous déciderez.

Travail 6 : Les tests en HTML/CSS

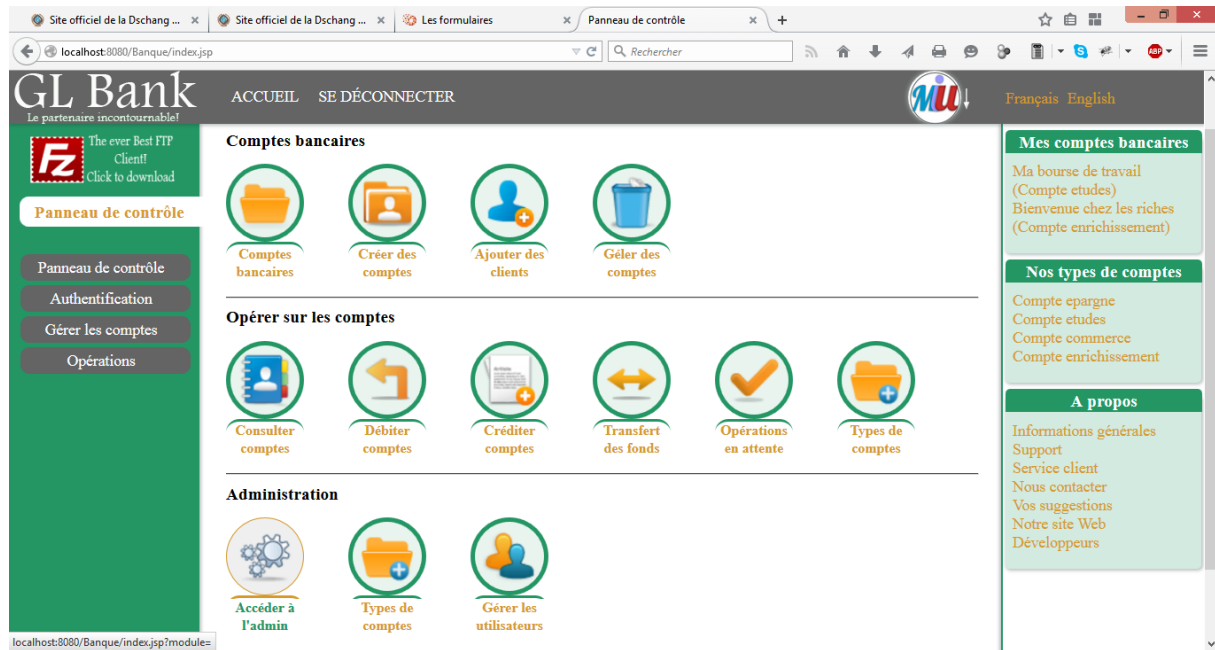
Cet exercice a pour but principal d'évaluer les apprenants afin de connaître le niveau de ces derniers en HTML/CSS.

1- Proposer un code source HTML/CSS permettant de réaliser la page de la capture d'écran ci-dessous :



L'IHM d'une calculatrice simple en HTML/CSS

2- Proposer un code source HTML/CSS permettant de réaliser la page de la capture d'écran ci-dessous :



Le panneau de contrôle du site d'un établissement de microfinance

Il est important de savoir que le **<header>**, le **<aside>** gauche et le **<aside>** droit sont fixes (ils restent à leurs positions même pendant un défilement).

PARTIE II : Introduction au JavaScript (JS)

Travail 7 : Les bases du JavaScript

Cet exercice a pour but principal de fixer les jalons du développement en JavaScript.

- 1- Présenter le langage JavaScript (*caractéristiques (langage de scripts, orienté objet par prototype, langage interprété, langage client-side), origine, utilité...*)
- 2- Présenter les relations qui existent entre JavaScript et HTML/CSS, JavaScript et Java.
 - JavaScript n'est pas Java mais JavaScript fait de la publicité à Java.
 - JavaScript permet de dynamiser les pages HTML/CSS en agissant sur la structure de ces dernières.
 - Le code JavaScript est inclus dans du code HTML grâce à la balise `<script>`. `<script>` peut être utilisé partout et le code JavaScript peut être *externe* ou *interne* au code HTML/CSS qui l'inclut.
- 3- Ecrire un script JavaScript qui affiche le texte « *hello world !* » dans une boîte de dialogue en utilisant les approches suivantes :
 - Ecrire l'instruction d'affichage dans *l'espace global*.
 - Ecrire une fonction qui réalise l'affichage et l'appeler par la suite.
- 4- Ecrire un script JS qui demande, grâce à des boîtes de dialogue, à l'utilisateur d'entrer son nom puis son prénom (*dans cet ordre*) et qui affiche une chaîne suivant le format *M/Mme/Mlle [prénom récupéré] [nom récupéré]*.

Par exemple, si l'utilisateur entre *Kenfack* et *Joël* alors le système affichera *M/Mme/Mlle Joël Kenfack*.

Travail 8 : Les nombres et les chaînes de caractères en JS

Dans cet exercice, les apprenants découvriront et apprendront à utiliser les objets JavaScript de manipulation des nombres et des chaînes de caractères.

- 1- Présenter l'objet **Number**
 - Création des instances de **Number**
 - Utilisation de ses propriétés statiques (**NAN**, **MAX_VALUE**, **MIN_VALUE**, **NEGATIVE_INFINITY**, **POSITIVE_INFINITY**...)
- 2- Présenter l'objet **Math**
 - Les différences avec l'objet **Number**
 - Utilisation de ses propriétés statiques (**PI**, **E**...)
 - Utilisation de ses méthodes statiques (**floor**, **ceil**, **round**, **pow**, **cos**, **sin**, **sqrt**, **random**, **min**, **max**...)
- 3- Présenter les fonctions globales (*de l'objet **window***) de gestion des données numériques (**parseInt**, **parseFloat**, **isFinite**, **isNaN**...).

4- Présenter l'objet *String*

- Création des instances de *String*
- Connaitre la longueur d'une chaîne grâce à la propriété *length*
- Présenter quelques méthodes de *String* (*toLowerCase*, *toUpperCase*, *concat*, *charAt*, *trim*, *indexOf*, *lastIndexOf*, *substring*, *substr*, *split*...).

5- Ecrire un script *JS* qui récupère, à travers une boîte de dialogue, une chaîne de la forme *noms=[un ou plusieurs noms];prenoms=[un ou plusieurs prénoms];age=[une valeur entière]*, transforme le/les noms en majuscule, transforme chacun des prénoms tels que leurs premières lettres uniquement soient en majuscule, convertit l'âge en binaire puis affiche le résultat suivant le format *[noms transformés] [prénoms transformés] a [âge transformé] ans en binaire*. Il n'est pas nécessaire d'effectuer les différents contrôles.

Voici quelques jeux d'exemples :

Chaîne d'entrée	Chaîne de sortie
noms=Tankeu Emaleu;prenoms=ludivine;age=20	TANKEU EMALEU Ludivine a 10100 ans en binaire
prenoms=GireSSe nana;age=10;noms=ewane	EWANE Giresse Nana a 1010 ans en binaire
noms=Limofouet;age=19	LIMOFOUET a 10011 ans en binaire

Travail 9 : Les Regex, la gestion du temps et les tableaux en JS

Les questions qui suivent vont contribuer à renforcer la capacité des apprenants à manipuler les expressions régulières (regex), les dates et les tableaux en JS.

1- Présenter les expressions régulières en JS

- Objectifs visés par les expressions régulières
- Construction d'une expression régulière pour la vérification d'un email valide
- Tester la correspondance d'un mot par rapport à une expression régulière grâce à la méthode *test* de l'objet *RegExp*.

2- Présenter le système de datation de JS

- La notion de Timestamp
- La manipulation du temps grâce à l'objet *Date*
 - Instanciation d'un objet de type *Date*
 - Présenter les méthodes des instances de *Date* (*getMonth*, *getDay*, *getDate*, *getFullYear*, *getHours*, *getMinutes*, *getSeconds*, *getTime*, *setTime*...).
- Présenter les fonctions temporelles *setTimeout*, *setInterval*, *clearTimeout* et *clearInterval*.

3- Présenter l'objet de création et de manipulation des tableaux en JS.

- Création des instances de *Array*
- Accéder à un élément du tableau
- Utiliser les méthodes de *Array* (*push*, *pop*, *shift*, *unshift*, *concat*, *indexOf*, *sort*, *reverse*, *splice*, *slice*, *isArray*)

Travail 10 : La manipulation du code HTML/CSS en JS

Le JS est exceptionnel parce qu'il permet de modifier la structure d'une page HTML/CSS. Ce travail a pour but de présenter les objets qui permettent d'accéder et de modifier le code HTML/CSS.

1- Accéder aux éléments du code HTML grâce au DOM

- Récupérer un élément en connaissant la valeur de son attribut *id* grâce à la méthode ***getElementById*** de l'objet ***document***.
- Récupérer tous les éléments d'une même famille (*définis avec le même tag*) grâce à ***document.getElementsByTagName***.
- Récupérer des éléments grâce aux méthodes ***querySelector*** et ***querySelectorAll*** de l'objet ***document***.

2- Agir sur un élément HTML

- Récupérer et modifier la valeur d'un attribut grâce aux méthodes ***getAttribute*** et ***setAttribute***
- Récupérer et modifier la valeur des attributs grâce aux propriétés (*href, value, className, id, src...*)
- Récupérer et modifier le code HTML d'un élément grâce à la propriété ***innerHTML***.
- Créer un nouvel élément HTML et l'ajouter à un élément grâce à ***createElement***, ***appendChild*** et ***insertBefore***.
- Supprimer un élément grâce à ***removeChild***

3- Récupérer et modifier le style d'un élément

- Grâce à la propriété ***style*** et aux styles (*width, height, color, backgroundColor...*).
- Récupérer le style grâce à la méthode ***getComputedStyle***.

4- Présenter les événements

- Intérêt et mode d'emploi des événements
- Présentation des événements (*click, blur, focus, load, submit, mouseout...*)

Travail 11 : Tests en JS

Cet exercice a pour but principal d'évaluer les capacités des apprenants en JS.

1- Ecrire un script *JS* et l'inclure dans la page *HTML* du **Travail 4 question 5** (formulaire d'inscription d'un bachelier...). Le script permettra de valider les informations présentes dans le formulaire avant sa soumission. Si les informations sont erronées alors les messages d'erreurs doivent être affichés dans la page. Si le formulaire est valide alors les informations sur l'étudiant seront ajoutées dans un tableau *HTML* de la page. Vous modifierez le code *HTML* en conséquence.

2- Ecrire un page *HTML/CSS/JavaScript* dans laquelle sera affichée une horloge numérique qui affichera l'heure courante suivant le format *HH:mm:ss*. L'heure devra être mise à jour toute les secondes.

3- Ecrire un script *JS* qui va permettre de faire fonctionner de façon standard la calculatrice du **Travail 6 question 1**.

PARTIE III : Le dynamisme du PHP

Travail 12 : Les outils pour le développement en PHP

Cet exercice a pour but principal de présenter le langage PHP et les outils qui sont nécessaires pour l'écriture des scripts PHP.

1- Présenter le langage *PHP* (caractéristiques (langage de scripts, langage server-side, approche procédurale, approche impérative et approche orientée objet, langage interprété), origine, utilité...)

2- Présenter *WampServer*

- Installation de *WampServer*
- Présentation et configuration du serveur *Apache*
- Présentation de *PHP* installé sur *Apache*
- Présentation du SGBD *MySql*

3- Réaliser un premier exemple (un script PHP qui affiche le texte « *hello world !* »)

- Présentation du dossier *www* de *WampServer*
- Créer un nouveau projet (sous dossier de *www*) de site Web dynamique.
- Créer les fichiers PHP (code HTML contenant du code PHP) grâce à *Notepad++*
- Démarrer le serveur et accéder à notre projet grâce au navigateur

Travail 13 : Les tableaux en PHP

Cet exercice a pour but principal d'enseigner la manipulation des tableaux en PHP.

1- Création et utilisation des variables de type tableau (*array*)

- Création et utilisation d'un tableau classique
- Création et utilisation d'un tableau associatif
- Parcourir et manipuler un tableau grâce à (*foreach*, *for*, *array_key_exists*, *in_array*, *array_search*).
- Transformer une chaîne de caractère en tableau grâce à *explode*

2- Proposer une structure de données basée sur les tableaux PHP pour représenter une liste d'étudiants. Un étudiant est caractérisé par son *nom*, son *matricule* et la liste d'*UV* auxquelles il est inscrit. Une *UV* est caractérisée par son *intitulé*, son *coefficient* et le *nombre d'heures de cours*.

Ecrire un script PHP qui crée une liste de 3 étudiants puis d'afficher les informations de cette dernière dans un tableau HTML.

Travail 14 : Les variables super globales en PHP

Cet exercice a pour but principal de présenter les variables globales proposées par PHP.

1- Récupérer les variables transmises à travers l'url ou par le biais d'un formulaire

- La forme générale d'une url qui contient des paramètres (<type de connexion>://<serveur>/<ressource>?paramètre1=valeur1&...¶mètrek=valeur k)
- Récupérer les valeurs transmises grâce aux variables super globales `$_GET`, `$_POST`, `$_REQUEST` et `$_FILES`

2- Ecrire un script PHP qui récupère les informations sur un étudiant grâce au formulaire du **Travail 11 question 1** les contrôle puis les affiche suivant une structure similaire à ceci (certaines informations ont été floutées) :

UNIVERSITE DE DSCHANG
THE UNIVERSITY OF DSCHANG
FACULTE DES SCIENCES
FACULTY OF SCIENCES

REPUBLIQUE DU CAMEROUN
République du Cameroun
Peace - Work - Faithfulness

FICHE DE PREINSCRIPTION

Année Académique : 2014/2015
N°préinscription : 111574

INFORMATION

Type Formation : Formation Initiale (Préuniversitaire)
Classe LMD/Niveau : Master 2
Mode d'admission : Etude de Dossier
Choix de formation : Licence en Services Distribués
Licence en Services Distribués

INFORMATIONS PERSONNELLES

Noms et prénoms : [Redacted]
Date et lieu de naissance : [Redacted]
Sexe : [Redacted]
Nationalité : [Redacted]
Département : [Redacted]
Situation familiale : [Redacted]
N°Téléphone : [Redacted]
Ville de résidence : [Redacted]
Email : [Redacted]

Langue principale : Français
Région : [Redacted]
Handicapé : [Redacted]
Capacité de port : [Redacted]
Situation emploi : [Redacted]

Parents / tuteurs

Noms père : [Redacted]
Profession père : [Redacted]
Ville de résidence : [Redacted]

Noms mère : [Redacted]
Profession mère : [Redacted]
Adresse : [Redacted]

Profil scolaire et Académique

Diplôme d'admission (DA) : [Redacted]
Niveau (DA) : [Redacted]
Année d'entrée : [Redacted]
Ville d'entrée : [Redacted]

Signature de l'étudiant (e) : [Redacted]
Le Chef Service de la Scolarité

3- Présentation des autres variables super globales

- Des variables présentes partout grâce à `$_SESSION`
- Définir des cookies avec `setcookie` et les récupérer grâce à `$_COOKIE`
- Connaître des informations sur le serveur grâce à `$_SERVER` et à `$_ENV`

4- Modifier le script de la question 2 de façon à conserver les informations de tous les bacheliers qui ont déjà été enregistrées.

Travail 15 : Les dates en PHP

Cet exercice a pour but principal d'enseigner la manipulation des dates en PHP.

1- Présentation de la fonction `date`

2- Manipuler le Timestamp grâce aux fonctions `time`, `mktime`

3- Ecrire un script PHP qui propose un formulaire à l'utilisateur pour récupérer sa date de naissance sous le format `jour-mois-année` et qui affiche ensuite le jour de la semaine qui a vu naître ce dernier. Par exemple si l'utilisateur entre `09-05-2015` alors on affichera *Samedi*.

Travail 16 : Les fichiers et l'accès aux bases de données en PHP

Dans ce qui va suivre, nous allons apprendre à sauvegarder les données des utilisateurs de manière permanente soit dans des fichiers soit dans des bases de données.

1- La création et la manipulation des fichiers

- Ouvrir un fichier grâce à la fonction *fopen* en spécifiant le mode d'ouverture (*r*, *r+*, *a*, *a+*)
- Fermer un fichier ouvert avec *fclose*
- Lire, se déplacer et écrire dans un fichier (*fgets*, *fseek*, *fputs*).

2- Ecrire un script PHP qui récupère les informations sur un contact (*noms*, *sexe*, *numéro de téléphone*, *email*) et les stocke dans un fichier nommé *annuaire.txt*. Vous devrez vous assurer qu'un contact soit enregistré une et une seule fois. Le script permettra aussi d'afficher la liste des contacts déjà présents dans l'annuaire. Pour faire simple, veuillez adopter la structure suivante pour le fichier *annuaire.txt* :

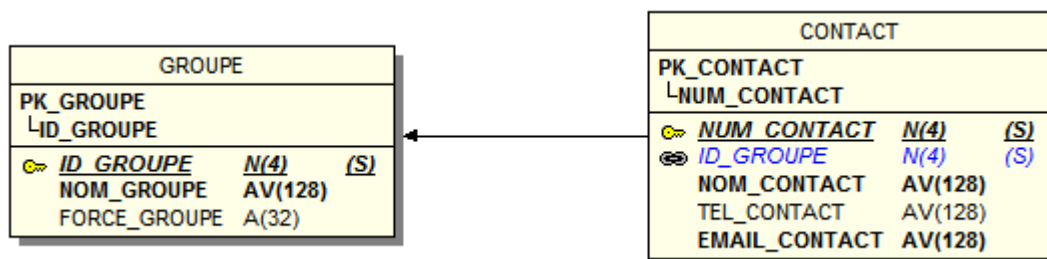
```
3 //Nombre de contacts actuellement répertoriés
Zeulekeu;Feminin;6942356;test@test.yahoo.fr
Mfambou james;Masculin;685235;j_mfam@tet.fr
Nkeng Paul;Masculin;;popol@gmail.com
```

3- Présenter *PhpMyAdmin*

- Rôle et interface graphique
- Création d'une nouvelle base de données
- Ajouter des tables à une base de données (*création des colonnes et ajout des contraintes*)
- Joindre des tables grâce à la contrainte de *clé étrangère*
- Exporter sa base de données

4- Réaliser toutes les opérations suivantes à partir de *PhpMyAdmin*

- Créer la base de données *annuaire* dont les tables sont représentées sur cette capture d'écran :



Les clés primaires s'incrémentent automatiquement, le nom de groupe, l'email et le téléphone sont unique.

- Ajouter des données à vos tables en utilisant 2 procédés (*le formulaire d'ajout proposé par PhpMyAdmin et le langage SQL*) différents.
- Afficher les données de chacune des tables grâce à des requêtes *SQL*

5- Manipuler une base de données à partir d'un script PHP

- Se connecter et se déconnecter à un serveur de base de données *MySQL* grâce à *mysql_connect* et *mysql_close*
- Se connecter à une base de données, y exécuter des requêtes *SQL*, récupérer les résultats et les interpréter (*mysql_select_db*, *mysql_query*, *mysql_fetch_array*, *mysql_fetch_assoc*...).

6- Proposer un script PHP qui affiche tous les contacts présents dans la base de données *annuaire* que vous avez créé précédemment.

7- Présenter l'approche *PDO*

- Activer les extensions PHP qui permettent d'utiliser *PDO*.
- Présenter les avantages de l'utilisation de l'interface *PDO* plutôt que les autres approches.
- Résoudre le problème de la **question 6** en utilisant l'approche *PDO*

Exercices d'approfondissement

Les travaux qui vont suivre vont permettre aux apprenants de créer des scripts PHP utiles. Aussi la structure des exercices est telle qu'on peut y déceler une bonne structure de projet Web dynamique.

Exercice 1 : Réalisation d'un système de pagination

Il arrive parfois, lorsque nous listons des données dans nos systèmes, que la liste soit très longue. Cela entraîne en général un effet touffu et par conséquent les données sont quasiment illisibles. Pour améliorer la lisibilité, on peut subdiviser nos données en plusieurs pages et permettre la navigation entre ces pages. Le système de pagination regroupe l'ensemble des données et des fonctions permettant de paginer nos listes et de naviguer entre les pages.

Pour cet exercice, nous considérons la base de données `bd_reunion_familiale` contenant l'unique table `invite` et hébergée sur un serveur `MySQL`. Le serveur de base de données est installé sur la machine d'adresse IP `192.168.25.230` sur notre réseau. On accède aux bases de données grâce au couple d'identifiants (`admin`, `massacre`). La table `invite` possède les colonnes `id`, `nom`, `prenom`, `nombre_uv`. La colonne `id` sert de clé primaire.

Notre objectif est d'afficher dans un tableau HTML, grâce à notre système de pagination, la liste des invités à la réunion familiale. Pour faire simple, nous allons supposer que chaque page devra afficher au maximum 10 invités. Le système devra fournir ensuite un panneau de navigation sur lequel on verra : un lien `page précédente` pour afficher la page de données précédente ; un lien `page suivante` pour afficher la page de données suivante ; le texte `Page x1/x2` où `x1` et `x2` sont deux entiers naturels représentant *le numéro de la page courante* et *le nombre total de pages* respectivement. On veillera à ce que les liens `page précédente` et `page suivante` soient désactivés à la première page et à la dernière page respectivement. Vous pouvez voir ce que nous voulons obtenir sur cette capture d'écran.

Sont invités à la réunion familiale,

Id	Noms et prénoms	Nombre d'UV
21	Toto Le Grand	13
22	Tata La Belle	10
23	Toto Le Grand	13
24	Tata La Belle	10
25	Toto Le Grand	13
26	Tata La Belle	10
27	Toto Le Grand	13
28	Tata La Belle	10
29	Toto Le Grand	13
30	Tata La Belle	10

[page précédente](#)

Page 3/4

[page suivante](#)

Notre système de pagination

Question 1 : Connexion à la base de données (fichier `connexion.php`)

1-) Proposez une fonction PHP nommée *connecter* qui utilisera la fonction `mysql_connect` (fonction de l'extension `php_mysql`) pour établir une connexion avec la base de données `bd_reunion_familiale`.

2-) Dites en quoi (du point de vue fonctionnel) cette façon de procéder diffère de celle qui utilise la classe PDO (classe de l'extension `php_pdo_mysql`).

Question 2 : Fonctions du fichier *pagination.php*

1-) Proposez une fonction PHP nommée `getTotalPages` qui retournera un nombre entier représentant le nombre total de pages à afficher. Vous pourrez vous servir de la fonction `ceil` qui permet d'effectuer un arrondi vers l'entier supérieur. On rappelle que la requête SQL `SELECT COUNT(*) AS nbre_total FROM invite` permet de stocker dans `nbre_total` le nombre de lignes total de la table `invite`.

2-) Proposez une fonction nommée `getData` qui prend en paramètre un numéro de page, calcule la plage de données à afficher, puis récupère si possible les lignes correspondantes dans la base de données. Pour indication, la plage de données pour la première page est 0, 10 et celle de la deuxième page est 10, 20. La fonction retourne un tableau contenant les données récupérées dans la base de données.

3-) Proposez une fonction nommée `layData` qui prend en paramètre le tableau précédent et retourne sous forme de chaîne de caractère le code du tableau HTML permettant d'afficher les données.

Question 3 : Création du fichier *invites.php*

Dans le fichier `invites.php`, nous voulons afficher les données sous forme de pages et construire notre panneau de navigation. Pour cela, nous allons supposer que la page `invites.php` sera indexée avec un paramètre transmis via l'URL. Ce paramètre sera nommé `page` et il désignera le numéro de la page à afficher. En l'absence de celui-ci ou s'il est inférieur ou égal à zéro, on considèrera que sa valeur est 1.

Proposez le code de la page `invites.php`. N'oubliez pas que vous avez tout ce dont vous avez besoin dans les fichiers `connexion.php` et `pagination.php`.

Exercice 2 : Réalisation d'un compteur de visiteurs connectés

On voudrait créer un système qui pourra informer nos internautes sur le nombre de visiteurs connectés sur le site à un instant donné. Pour cela, nous décidons de créer une table nommée `visiteur` dans notre base de données MySQL `bd_compteur`. La table `visiteur` permettra de stocker tous les visiteurs de notre site. La table `visiteur` possède deux colonnes : la colonne `addr_ip` qui permet de stocker l'adresse IP de la machine du visiteur ; la colonne `heure_requete` qui permettra de stocker l'heure de la dernière requête du visiteur.

Le fonctionnement de notre système sera donc le suivant : Lorsqu'un visiteur demande la page `news.php`, on récupère son adresse IP qui se trouve dans la variable `$_SERVER['REMOTE_ADDR']`. On réalise les actions suivantes :

- Si le visiteur est connecté (la table `visiteur` possède une entrée dont la colonne `addr_ip` vaut `$_SERVER['REMOTE_ADDR']`) alors on met à jour l'heure de la dernière requête du visiteur.
- Si le visiteur n'est pas connecté alors on le connecte en ajoutant une entrée dans la table `visiteur`.
- On supprime toutes les entrées de la table `visiteur` correspondant à plus de 10 minutes sans aucune requête (d'inactivité).
- On compte le nombre restant d'entrées dans la table `visiteur` et on l'affiche comme nombre de visiteurs actuellement connectés sur notre site.

Question 1 : Création de la bibliothèque (fichier `compteur.php`)

1-) Proposez une fonction PHP nommée `connecter` qui utilisera la fonction `mysql_connect` pour établir une connexion avec la base de données `bd_compteur`.

2-) Proposez une fonction nommée `addVisiteur` qui réalise les deux premières actions ci-dessus évoquées. Pour rappel la fonction PHP `time()` permet de récupérer l'heure (timestamp) actuelle. Pour la mise à jour de l'heure, vous pourrez utiliser la requête SQL `UPDATE visiteur SET heure_requete='time()' WHERE addr_ip = '$_SERVER['REMOTE_ADDR']'`.

3-) Ecrivez une fonction nommée `update` qui va permettre de réaliser la troisième de nos actions. Notez que `time() - (10*60)` permet de récupérer l'heure qu'il était 10 minutes plus tôt.

4-) Proposez une fonction nommée `compter` qui se sert des fonctions écrites précédemment et retourne le nombre de visiteurs actuellement connectés sur notre site.

Question 2 : Améliorations grâce au JavaScript

Pour afficher notre compteur de visite, nous avons utilisé le fichier `news.php` dont le code est le suivant :

```
<html>
  <body>
    <?php
      require_once('compteur.php');
      echo '<h2 id="compteur">Il y a ' .compter(). '
visiteurs</h2>' ;
    ?>
  </body>
</html>
```

Nous voulons à présent que notre compteur soit rafraîchi toutes les 20 secondes et ce sans que la page `news.php` soit rechargée. Nous avons donc écrit une fonction JavaScript nommée `getCompte` qui utilise la technologie AJAX et qui nous retourne le nombre de visiteurs actuellement connectés.

Proposez un script JavaScript qui utilisera la fonction `getCompte` toutes les 20 secondes et modifiera le DOM pour y afficher le nouveau nombre de visiteurs en ligne.

Exercice 3 : Manipulation d'une file d'attente

Toto est en train de réaliser le site dynamique de ses rêves. Malheureusement la durée de l'enregistrement des informations (`nom`, `email`, `âge`) sur un utilisateur par un administrateur est très grande. Pour résoudre le problème, toto décide d'utiliser une file d'attente pour stocker ces informations temporairement afin de les enregistrer de manière asynchrone. L'administrateur ne sera donc pas gêné par la lenteur de l'opération. Voici la conception de toto :

- La file d'attente sera représentée par une variable (`$_SESSION['file']`) stockée en session.
- La file sera un tableau associatif à trois valeurs : `tete` contiendra l'indice du premier élément de la file ; `position` contiendra l'indice suivant celui du dernier élément de la file ; `vecteur` sera un tableau contenant les éléments de la file.
- Un élément de la file sera un tableau associatif à trois valeurs : `nom` contiendra le nom du visiteur à enregistrer ; `email` contiendra l'adresse électronique du visiteur à enregistrer ; `age` contiendra l'âge du visiteur en question.

Toto fait appel à vous parce qu'il est très fatigué. Il vous explique que son vecteur n'est pas circulaire parce qu'il bénéficie du fait que les tableaux sont de tailles illimitées en PHP. Voici ce que toto vous demande ensuite de faire :

Question 1 : Formulaire et validation (fichier *infos.php*)

1-) Proposez un formulaire en HTML 5 pour la récupération des informations sur un utilisateur : L'`email` est requis.

2-) Proposez une fonction JavaScript pour la validation du formulaire précédent. On devra s'assurer que le nom est constitué d'au moins 4 lettres et que l'âge est bel et bien un entier naturel inférieur à 150.

Question 2 : Création de la bibliothèque (fichier *file.php*)

1-) Quelle est la particularité de la variable `$_SESSION['file']` ?

2-) Proposez les fonctions `initialiser` qui crée une file vide et la retourne, `taille` qui retourne le nombre d'éléments présents dans une file qu'elle prend en argument, `defiler` qui supprime et retourne l'élément en début d'une file prise en paramètre et `fileVide` qui permet de savoir si une file est vide.

3-) Proposez une fonction PHP nommée `enfiler` qui prend en paramètres un `nom`, un `email`, un `age` et une `file` puis construit l'élément à insérer et l'insère dans la file d'attente. La fonction retourne la file d'attente modifiée.

Question 3 : Enregistrer dans la base de données (fichier *enregistrer.php*)

On suppose la base de données MySQL nommée `file_db` et contenant la table `utilisateur`. La table `utilisateur` possède 4 colonnes : `id`, `nom`, `email`, `age`.

Proposer une fonction nommée `enregistrer` qui établit une connexion avec la base de données `file_db` et insère dans la table `utilisateur`, toutes les données d'une file d'attente prise en paramètre.

QUELQUES SUJETS TYPE EXAMEN

UNIVERSITE DE DSCHANG
FACULTE DES SCIENCES
Département de Maths-Info



REPUBLIQUE DU CAMEROUN
Paix - Travail - Patrie

Contrôle Continu : *Programmation web dynamique*

Classe : IN 2

Avril 2019

Durée : 01h15

Dr KENGNE T. Vianney

(Documents non autorisés. Répondre brièvement et ne répondre qu'aux questions posées.)

Question de mise en confiance : Définir le sigle XML.

- 1- Enumérer les principales balises et attributs servant à créer un tableau en HTML 5 et donner leurs rôles respectifs. **5pts**
- 2- Ecrire le sélecteur CSS ciblant les éléments suivants : « les liens de classe menulink, enfants directs de li, eux-mêmes enfants de listes à puces dont l'attribut id est défini » **2pts**
- 3- Présenter à l'aide d'un exemple les deux manières d'inclure du code JavaScript dans une page web. **2pts**
- 4- Recopier et compléter le tableau suivant en faisant correspondre les éléments de la colonne A à un ou plusieurs éléments de la colonne B (*réponse juste 1pt, mauvaise réponse -0.5pt, abstention 0pt*).

1	2	3	4	5	6	7	8	9	10	11

Colonne A		Colonne B	
1	Apache	A	Attribut utilisé dans la création des formulaires HTML 5
2	CSS	B	Environnement de développement des sites web dynamiques en PHP
3	XML	C	Ecoute sur le port 3307 par défaut
4	JavaScript	D	Intègre le concept de media queries dans sa version 3
5	MySQL	E	Format d'échange inspiré de la notation des objets en JS
6	CMS	F	Variable super globale
7	XAMPP	G	Format similaire au HTML mais moins général
8	HTML	H	Utilisé pour la mise en forme des pages web
9	required	I	Similaire au JSP et à l'ASP
10	HTTP	J	Hypertext Preprocessor
11	PHP	K	Joomla, Wordpress, Social Engine, etc.
		L	Contient MySQL, Apache, PHP etc.
		M	Ecoute sur le port 80 par défaut
		N	Format similaire au HTML mais plus général
		O	Système de Gestion des Base de Données
		P	Serveur Web très réputé
		Q	Protocole d'échange des données hypertextes
		R	Content Monitoring System
		S	Dérivé du SGML
		T	Langage interprété client-side

UNIVERSITE DE DSCHANG
FACULTE DES SCIENCES
Département de Maths-Info



REPUBLIQUE DU CAMEROUN
Paix - Travail - Patrie

Examen (Session Normale)

Programmation web dynamique (INF 232)

Classe : IN 2

Juin 2019

Durée : 03h

Dr KENGNE T. Vianney

(Documents non autorisés. Une bonne présentation de la copie est conseillée ! Bon courage !!!)

Question de mise en confiance : Donner la signification des sigles HTTP et PHP.

Exercice 1 (5.5pts)

- 5- Enumérer les principales balises et attributs servant à créer un tableau en HTML 5 et donner leurs rôles respectifs. 2.5pts
- 6- Ecrire le sélecteur CSS ciblant les éléments suivants : « les titres de niveau 3 de classe *menulink*, enfants directs de *li*, eux-mêmes enfants de listes à puces dont l'attribut *id* est défini et a comme valeur *list* » 1pt
- 7- Considérons une page web contenant l'élément suivant :
`<div id="monDiv myDiv" class="uneDiv divBleu">...</div>`
 - a- Cet élément est-il bien défini ? Justifier. 1pt
 - b- Proposer deux fonctions prédéfinies en JavaScript qui pourraient permettre de cibler cet élément et d'afficher le code HTML qu'il contient. 1pt

Exercice 2 (5pts)

- 1- Proposer un formulaire (maquette + code) HTML 5 pour l'enregistrement des utilisateurs sur votre site Web. Le formulaire devra permettre de récupérer le nom, l'email, l'âge et le sexe de l'utilisateur. 2pts
- 2- Proposer une fonction JavaScript *valider* pour la validation du formulaire précédent. Il faudra s'assurer que le nom ait au moins 4 caractères, que l'âge soit un entier compris entre 17 et 60 et que l'email soit valide du point de vue syntaxique. 2pts
- 3- Donner le code JavaScript qui permet de lier la fonction *valider* à l'évènement *submit* du formulaire. 1pt

Exercice 3 (4pts)

- 1- Un étudiant est caractérisé par son *nom*, son *matricule* et la *liste d'UV* auxquelles il est inscrit. Une *UV* est caractérisée par son *intitulé* et son *coefficient*. Ecrire un script PHP qui crée une liste de 2 étudiants puis affiche les informations de cette dernière dans un tableau HTML. (Utiliser les *tableaux associatifs* pour cet exercice). 2pts
- 2- Ecrire un script PHP qui propose un formulaire à l'utilisateur pour récupérer sa date de naissance et qui affiche ensuite le jour de la semaine qui a vu naître ce dernier. Par exemple si l'utilisateur entre la date 26-05-1992 alors on affichera *Tuesday*. 2pts

Problème (5.5pts)

On voudrait créer un système qui pourra informer nos internautes sur le nombre de visiteurs connectés sur notre site à un instant donné. Pour cela, nous décidons de créer une table nommée *visiteur* dans notre base de données MySQL *bd_compteur*. La table *visiteur* permettra de stocker tous les visiteurs de notre site. La table *visiteur* possède deux colonnes : la colonne *addr_ip* qui permet de stocker l'adresse IP de la machine du visiteur ; la colonne *heure_requete* qui permettra de stocker l'heure de la dernière requête du visiteur.

Le fonctionnement de notre système sera donc le suivant : lorsqu'un visiteur demande la page *news.php*, on récupère son adresse IP qui se trouve dans la variable `$_SERVER['REMOTE_ADDR']`. On réalise les actions suivantes :

- Si le visiteur est connu alors on met à jour l'heure de la dernière requête du visiteur ;
- Si le visiteur n'est pas connu alors on l'enregistre en ajoutant une entrée dans la table *visiteur* ;
- On supprime toutes les entrées de la table *visiteur* correspondant à plus de 10 minutes sans aucune requête (inactivité) ;
- On compte le nombre restant d'entrées dans la table *visiteur* et on l'affiche comme nombre de visiteurs actuellement connectés sur notre site ;

- 1- Proposez une fonction PHP dénommée *connecter* qui utilisera l'approche PDO pour établir une connexion avec la base de données *bd_compteur*. **1pt**
- 2- Proposez une fonction dénommée *addVisiteur* qui réalise les deux premières actions ci-dessus évoquées. Pour rappel la fonction PHP *time()* permet de récupérer l'heure (timestamp) actuelle. Pour la mise à jour de l'heure, vous pourrez utiliser la requête SQL *UPDATE visiteur SET heure_requete='.time().' WHERE addr_ip = \''. \$_SERVER['REMOTE_ADDR'].\'*. **1.5pts**
- 3- Ecrivez une fonction nommée *update* qui va permettre de réaliser la troisième de nos actions. Notez que *time()-(10*60)* permet de récupérer l'heure qu'il était 10 minutes plus tôt. **1pt**
- 4- Proposez une fonction nommée *compter* qui se sert des fonctions écrites précédemment et retourne le nombre de visiteurs actuellement connectés sur notre site. **1pt**

Pour afficher notre compteur de visite, nous avons utilisé le fichier *news.php* ci-dessous :

```
<html>
<body>
  <?php
    require_once('compteur.php');
    echo '<h2 id="compteur">Il y a '.compter().' visiteurs</h2>';
  ?>
</body>
</html>
```

Nous voulons à présent que notre compteur soit rafraîchi toutes les 20 secondes et ce sans que la page *news.php* soit rechargée. Nous avons donc écrit une fonction JavaScript nommée *getCompte* qui utilise la technologie AJAX et qui nous retourne le nombre de visiteurs actuellement connectés.

- 5- Proposez un script JavaScript qui utilisera la fonction *getCompte* toutes les 20 secondes et modifiera le DOM pour y afficher le nouveau nombre de visiteurs en ligne. **1pt**

UNIVERSITE DE DSCHANG
FACULTE DES SCIENCES
Département de Maths-Info



REPUBLIQUE DU CAMEROUN
Paix - Travail - Patrie

Examen (Rattrapage)

Programmation web dynamique (INF 232)

Classe : IN 2

Juillet 2019

Durée : 03h

Dr KENGNE T. Vianney

(Documents non autorisés. Une bonne présentation de la copie est conseillée ! Bon courage !!!)

Question de mise en confiance : Que faut-il faire avant de pouvoir manipuler les variables de session dans un script PHP ?

Exercice 1 : Questions à Choix Multiples (2.5pts)

NB : (Bonne réponse = 0.5pt, Mauvaise réponse = -0.25pt, abstention = 0pt)

1-) La fonction « **php_isset** » :

- a) Informe sur l'existence d'une variable donnée
- b) Est équivalente à la fonction **defined**
- c) Produit un effet contraire à celui produit par **unset**
- d) N'existe pas dans le langage PHP
- e) Rien de tout ceci

2-) La fonction JavaScript « **setInterval** » :

- a) Indique à l'interpréteur d'exécuter une fonction *f* donnée toutes les *t* millisecondes
- b) Permet de modifier le DOM
- c) Exécute une fonction de manière illimitée pendant un intervalle de temps
- d) Enclenche un compte à rebours à l'issue duquel, elle exécute une fonction
- e) Rien de tout ceci

3-) Que contient la variable « **\$_SERVER['REMOTE_ADDR']** » ?

- a) L'adresse IP de la machine serveur
- b) L'adresse IP du client
- c) Le nom du client
- d) Le nom de la machine serveur
- e) Rien de tout ceci

4-) Qu'affiche « **echo (time() - 5*60) ;** » ?

- a) Une date au format JJ/MM/AAAA
- b) Rien du tout
- c) Un nombre entier
- d) Une date et une heure
- e) Rien de tout ceci

5-) Que fait la fonction JavaScript « **querySelectorAll** » ?

- a) Elle insère un élément dans le DOM
- b) Elle définit l'orientation de l'élément zone
- c) Elle sélectionne tous les éléments du DOM correspondant à un sélecteur CSS

- d) Elle sélectionne les éléments ayant pour id « elem »
 e) Rien de tout ceci

Exercice 2 : Mise en œuvre de AJAX (3pts)

Soit l'extrait XML suivant (fichier *response.xml*) :

```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
  <packet id="1">
    <sender>John Doe</sender>
    <content>Bla bla bla</content>
  </packet>
  <packet id="2">
    <sender>Jane X</sender>
    <content>Blo blo blo</content>
  </packet>
</response>
```

- 4- Proposer un code JSON équivalent à ce code XML **1pt**
 5- Expliquer à l'aide de schémas les deux types de requêtes supportées par AJAX. **1pt**
 6- Proposer une fonction JavaScript qui demande de manière asynchrone, la ressource *response.xml* au serveur puis insère le contenu du premier paquet dans le DOM (comme contenu HTML de l'élément d'id *response*). **1pt**

Exercice 3 : Complétion de code (4pts)

```
<table>
  <tr>
    <td>Nom du cookie</td>
    <td>Valeur du cookie</td>
  </tr>
  <?php
    foreach(-1- as -2-){
    ?>
      <tr>
        <td><?php -3- $key; ?></td>
        -4-
      </tr>
    <?php
    }
  ?>
</table>
```

Code 1

```
<?php
if(-1-($ _FILES['fic']) && -2- == 0){
  $name = -3-
  $inf = pathinfo($name);
  $filename = -4-('.', -5-($name))[0];
  echo '<h1>Nom du fichier : '.$filename.'</h1>';
  echo '<h1>Extension : '-6- '</h1>';
}
?>
```

Code 2

```
<?php
$con = -1-(-2-, 'root', '', -3-);
$req = -4-
$res = -5-($con, $req);
-6-
?>
```

Code 3

NB : Compléter un code signifie reporter les paramètres *-i-* dudit code sur sa feuille de composition et ensuite donner pour chacun d'eux, un code pouvant les remplacer pour atteindre les objectifs fixés par la question.

- Compléter le Code 1 pour qu'il affiche dans un tableau HTML l'ensemble des cookies disponibles lors de son exécution. **1pt**
- Compléter le Code 2 pour qu'il affiche le nom et l'extension du fichier soumis via le formulaire lorsque celui-ci est présent. **1.5pts**
- Compléter le Code 3 pour qu'il se connecte à la base de données MySQL *bd_tp.php* situé sur une machine d'adresse IP 192.168.53.15, puis insère dans la table *etudiant(matricule, nom)* le tuple ("CM-UDS-18SCI1220", "Kenfack") et enfin se déconnecte du serveur MySQL. **1.5pts**

Exercice 4 : Ecriture d'un script PHP (3pts)

Ecrire un script PHP qui récupère les informations sur un contact (*noms, sexe, numéro de téléphone, email*) et les stocke dans un fichier nommé *annuaire.txt*. Pour faire simple, veuillez adopter la structure suivante pour le fichier *annuaire.txt* :

Zeulekeu;Feminin;6942356;test@test.yahoo.fr

Mfambou james;Masculin;685235;j_mfam@tet.fr

Problème : Création d'un système de pagination (7.5pts)

Il arrive parfois, lorsque nous listons des données dans nos systèmes, que la liste soit très longue. Cela entraîne en général un effet touffu et par conséquent les données sont quasiment illisibles. Pour améliorer la lisibilité, on peut subdiviser nos données en plusieurs pages et permettre la navigation entre ces pages. Le système de pagination regroupe l'ensemble des données et des fonctions permettant de paginer nos listes et de naviguer entre les pages.

Pour cet exercice, nous considérons la base de données *bd_reunion_familiale* contenant l'unique table *invite* et hébergée sur un serveur MySQL. Le serveur de base de données est installé sur la machine d'adresse IP 192.168.25.230 sur notre réseau. On accède aux bases de données grâce au couple d'identifiants (*admin, massacre*). La table *invite* possède les colonnes *id, nom, prenom, nombre_uv*. La colonne *id* sert de clé primaire.

Notre objectif est d'afficher dans un tableau HTML, grâce à notre système de pagination, la liste des invités à la réunion familiale. Pour faire simple, nous allons supposer que chaque page devra afficher au maximum 10 invités. Le système devra fournir ensuite un panneau de navigation sur lequel on verra : un lien *page précédente* pour afficher la page de données précédente ; un lien *page suivante* pour afficher la page de données suivante ; le texte *Page n1/n2* où *n1* et *n2* sont deux entiers naturels représentant le numéro de la page courante et le nombre total de pages respectivement. On veillera à ce que les liens *page précédente* et *page suivante* soient désactivés à la première page et à la dernière page respectivement. Vous pouvez voir ce que nous voulons obtenir sur cette capture d'écran.

Sont invités à la réunion familiale,

Id	Noms et prénoms	Nombre d'UV
21	Toto Le Grand	13
22	Tata La Belle	10
23	Toto Le Grand	13
24	Tata La Belle	10
25	Toto Le Grand	13
26	Tata La Belle	10
27	Toto Le Grand	13
28	Tata La Belle	10
29	Toto Le Grand	13
30	Tata La Belle	10

[page précédente](#) [Page 3/4](#) [page suivante](#)

Connexion à la base de données (fichier connexion.php)

1-) Proposez une fonction PHP dénommée *connecter* qui utilisera la fonction *mysql_connect* (fonction de l'extension *php_mysql*) pour établir une connexion avec la base de données *bd_reunion_familiale*. **1pt**

Fonctions de pagination (fichier *pagination.php*)

2-) Proposez une fonction PHP dénommée *getTotalPages* qui retournera un nombre entier représentant le nombre total de pages à afficher. Vous pourrez vous servir de la fonction *ceil* qui permet d'effectuer un arrondi vers l'entier supérieur. On rappelle que la requête « *SQL SELECT COUNT(*) AS nbre_total FROM invite* » permet de stocker dans *nbre_total* le nombre de lignes total de la table *invite*. **1.5pts**

3-) Proposez une fonction dénommée *getData* qui prend en paramètre un numéro de page, calcule la plage de données à afficher, puis récupère si possible les lignes correspondantes dans la base de données. Pour indication, la plage de données pour la première page est **0, 10** et celle de la deuxième page est **10, 10**. La fonction retourne un tableau contenant les données récupérées dans la base de données. **1.5pts**

4-) Proposez une fonction nommée *layData* qui prend en paramètre le tableau précédent et retourne sous forme de chaîne de caractère le code du tableau HTML permettant d'afficher les données. **1.5pts**

Création du fichier *invites.php*

Dans le fichier *invites.php*, nous voulons afficher les données sous forme de pages et construire notre panneau de navigation. Pour cela, nous allons supposer que la page *invites.php* sera indexée avec un paramètre transmis via l'URL. Ce paramètre sera nommé *page* et il désignera le numéro de la page à afficher. En l'absence de celui-ci ou s'il est inférieur ou égal à zéro, on considèrera que sa valeur est 1.

5-) Proposez le code de la page *invites.php*. N'oubliez pas que vous avez tout ce dont vous avez besoin dans les fichiers *connexion.php* et *pagination.php*. **2pts**

