

Examen du module Programmation Internet

Jeudi 19 mai 2005

Durée : 1h30mn

Tous les documents écrits sont autorisés.

Le barème est donné à titre indicatif.

- **Exercice 1 (5 points).**

Ecrire un code HTML permettant d'afficher un texte, un champ de texte et deux boutons comme sur la figure ci-dessous :

Compteur :

39

Ajouter du code JavaScript permettant de :

1. Démarrer ou redémarrer un compteur (dont la valeur s'affiche dans le champ de texte) quand on clique sur le bouton « Lancer ».
2. Arrêter le compteur quand on clique sur le bouton « Stopper ».

Quand on lance le compteur, il est régulièrement incrémenté de 1 et ce processus ne peut s'interrompre qu'en appuyant sur le bouton « Stopper » (ou en fermant le navigateur).

Au chargement de la page, le compteur est à 0 et est arrêté.

Quand on redémarre le compteur après l'avoir arrêté, il repart de sa valeur courante (Il n'est donc pas réinitialisé à 0).

On ne tiendra pas compte des problèmes de débordement numérique pouvant affecter la valeur du compteur.

- **Exercice 2 (3 points).**

Ecrire une fonction JavaScript qui calcule le produit des éléments d'un tableau qu'elle reçoit en paramètre. Si le tableau est vide, la fonction retourne la valeur spéciale « undefined » (Pour cela, on utilise la construction syntaxique suivante : « return; »). La fonction n'a pas à contrôler le fait que les éléments du tableau soient des nombres ou pas.

- **Exercice 3 (12 points).**

1) Une société vendant différents produits met en place sur son serveur Web une page permettant à un internaute de passer une commande.

Ecrire le code HTML et JavaScript permettant d'obtenir le formulaire de saisie représenté et décrit ci-dessous :

Nom :

Prénom :

Adresse e-mail :

Produit sélectionné :

Prix du produit : euros

Le produit sélectionné est un élément de la liste suivante :

Aucun, Livre, CD, DVD

Le champ correspondant au prix doit être mis à jour automatiquement par un code JavaScript quand l'internaute choisit un élément de la liste.

La société a choisi les tarifs suivants :

Aucun → 0
Livre → 10
CD → 15
DVD → 22

Indication : Quand l'élément sélectionné dans la liste change, un événement « Change » est engendré par le navigateur. On dispose de l'attribut « onChange » pour associer un gestionnaire à cet événement.

La soumission du formulaire doit amener l'exécution d'un script CGI référencé par l'URI suivante : /cgi-bin/stocker.pl

Quand on clique sur le bouton de soumission du formulaire (le bouton « Commander »), du code JavaScript doit vérifier que le nom, le prénom et l'adresse e-mail ont été saisis par l'internaute. Si ce n'est pas le cas, une boîte de message signalant **précisément** l'information manquante doit apparaître avant que l'internaute ne recommence sa saisie.

2) On vous demande à présent d'écrire le code Perl du script CGI « stocker.pl ».

Remarque : Vous pouvez faire appel, si besoin, à la bibliothèque « cgi-lib.pl » pour écrire les scripts CGI.

- Dans un premier temps, le script CGI effectue les vérifications suivantes :
 - Le nom n'est constitué que des caractères suivants : lettres, espace, apostrophe, tiret.
Exemples de noms valides : D'Arsonval, Saint-Just, De Vinci, a-b'
Exemples de noms invalides : ZX81, 10\$, Molière
 - Le prénom n'est constitué que des caractères suivants : lettres et tiret.
Exemples de prénoms corrects : Louis-Ferdinand, Stephanie, -x
Exemples de prénoms incorrects : Pierre-François, Stéphanie, D2R2
 - Le produit choisi n'est pas « Aucun ».

Si l'une des trois vérifications échoue, le script retourne une page HTML avec un texte signalant une erreur dans les données envoyées.

- Dans un second temps (si les vérifications ont réussi), le script sauvegarde dans un fichier « commandes.txt » - que l'on suppose dans le même répertoire que le script – la commande de l'internaute sous la forme :

numéro_commande/nom/prénom/adresse_e_mail/produit

« numéro_commande » est un entier positif incrémenté de 1 à chaque nouvelle commande.

Si le fichier « commandes.txt » n'existe pas, le numéro de la première commande sera 1.

Exemple d'un fichier « commandes.txt » possible :

521/Dupont/Eric/dupond.eric@machin.org/Livre
 522/Durand/Herve/durandh12@machin.fr/DVD
 523/Dupont/Emilie/dupond.eric@machin.org/DVD

On remarquera que le fichier ne commence pas nécessairement par un numéro de commande égal à 1 (Quand une commande a été traitée par la société, elle l'efface de ce fichier pour la stocker dans une base de données).

Après la sauvegarde dans le fichier « commandes.txt », le script CGI renvoie finalement une page HTML indiquant que la commande de l'internaute a été enregistrée.

- 3) Une page HTML est mise en place par la société pour permettre à un client de consulter l'ensemble des commandes qu'il a passées et qui sont en traitement (Ces commandes sont stockées dans le fichier « commandes.txt »).

Un client est identifié par le triplet :

(nom, prénom, adresse e-mail)

Ecrire le code HTML permettant de produire le formulaire suivant :

Nom :

Prénom :

Adresse e-mail :

Quand on clique sur le bouton de soumission « Rechercher », les données sont transmises au script CGI dont l'URI est : /cgi-bin/rechercher.pl

Ecrire ensuite le code Perl du script CGI « rechercher.pl » :

- Si le client est référencé dans le fichier « commandes.txt », le script retourne un document HTML avec un tableau contenant toutes ses commandes. Exemple de tableau produit :

Numéro commande	Produit	Prix
4	CD	15 euros
6	CD	15 euros

- Si l'on ne trouve pas le client dans le fichier « commandes.txt », le script renvoie un document HTML avec comme corps un titre de niveau 1 de couleur rouge dont le texte est : « Vous n'avez actuellement aucune commande ».

- Si le fichier « commandes.txt » n'existe pas ou est vide, le script renvoie un document HTML avec comme corps un titre de niveau 1 de couleur jaune dont le texte est : « Aucune commande actuellement en traitement ».

4) En interne, la société a développé un script CGI en Perl permettant de rechercher dans le fichier « commandes.txt » le client le plus important (c'est-à-dire celui dont le montant total des commandes est le plus élevé). Il est bien entendu possible qu'il n'y ait pas qu'un seul client dont le montant total des commandes est le plus élevé.

Ecrire ce script Perl qui renvoie une page HTML avec un tableau contenant :

- l'identifiant (nom + prénom + e-mail) du client le plus important
- le nombre de commandes du client
- la somme totale due par le client

S'il y a plusieurs clients « les plus importants », on les affichera tous dans le tableau.

Exemple de tableau produit par le script CGI :

Nom	Prénom	Adresse e-mail	Nombre de commandes	Total en euros
Durand	Albert	durand@truc.org	2	30
Dupont	Thierry	dth@truc.fr	3	30

Si le fichier « commandes.txt » n'existe pas ou est vide, le script retourne une page HTML spécifiant qu'il n'y a actuellement aucune commande.

5) On reprend les vérifications du script du 2.

On souhaite à présent ajouter un contrôle sur l'adresse e-mail.

On considérera qu'une adresse e-mail sera valide si elle est sous la forme :

première partie@deuxième partie

La première partie est formée d'une suite de termes séparés par des points. Un terme est une suite non vide de lettres ou de chiffres. Il y a au moins un terme dans la première partie.

La deuxième partie est formée d'une suite de termes séparés par des points. Il y a au moins deux termes dans la deuxième partie.

Exemples d'adresses e-mail considérées comme correctes :

dupont27@truc.fr
 jean.dupont@truc.fr3
 33.durand22@mail.conf.3ia
 12@2.5

Exemples d'adresses e-mail considérées comme incorrectes :

@truc.fr
 dupont@truc
 dupont@truc.fr@truc.fr
 dupont.@truc.fr
 .dupont@truc.fr
 bonjour dupont@truc.fr

Ecrire le code Perl à ajouter pour tester l'adresse e-mail. Si la vérification échoue, le script retourne au navigateur une page HTML contenant un paragraphe avec le texte suivant écrit en gras : « Adresse e-mail incorrecte ».