

Module Programmation Internet.

TD 4

Patrick Poulingeas

Exercice 1.

Dans un fichier texte nommé « dates.txt », chaque ligne contient une date sous la forme :
numéro_du_jour/numéro_du_mois/numéro_de_l_année

Ecrire un script Perl qui affiche la date la plus récente contenue dans le fichier « dates.txt ».

Exercice 2.

Une entreprise a mis en place un Intranet avec un agenda commun à tous ses employés.

Cet agenda correspond à un fichier texte (« agenda.txt ») stocké sur le serveur Web de l'entreprise. Chaque ligne de ce fichier est constituée de 5 champs séparés par des « : ». Les champs sont :

- une semaine indiquée par un numéro entre 1 et 52
- un jour indiqué par un chiffre entre 1 et 6
- un nom d'employé
- une activité dans l'entreprise
- une tranche horaire sous la forme : « <heure début>h-<heure fin>h » où <heure début> et <heure fin> sont des entiers entre 7 et 20.

Voici un extrait du fichier « agenda.txt » :

40:5:Dupont:Redaction documentation projet A:8h-10h

40:5:Durand:Recette projet B:9h-11h

41:1:Dubois:Mise au point cahier des charges projet Z:8h-12h

40:5:Dupont:Deploiement projet A:14h-15h

Ecrire un formulaire permettant la saisie du nom d'un employé ainsi que d'un numéro de semaine (On ne fera pas de vérification avec JavaScript sur le numéro de semaine).

Quand on valide le formulaire, celui-ci est soumis à un script CGI en Perl.

Ecrire ce script qui va produire un document HTML contenant :

- Dans son titre : le nom de l'employé et le numéro de la semaine
- Dans son corps : un tableau contenant la totalité des activités de l'employé pour la semaine. Ce tableau contient pour chaque activité :
 - le nom de l'activité
 - le jour de l'activité (sous forme d'un nom de jour ; on remplace donc le chiffre « 1 » par « lundi », etc.)

- la plage horaire de l'activité
- la durée de l'activité en heures

Si l'employé n'existe pas (dans le fichier « agenda.txt ») ou que le numéro de semaine est incorrect, le script CGI l'indique avec un texte approprié. Si l'employé n'a pas d'activité pour la semaine considérée, le script CGI le signale aussi.

On utilisera la bibliothèque cgi-lib.pl pour récupérer les données du formulaire dans le script CGI.

On supposera que le script CGI et le fichier « agenda.txt » sont situés dans le même répertoire sur le serveur.

La personne chargée de maintenir l'agenda doit garantir qu'il n'y aura pas de « : » dans les expressions décrivant les activités de l'entreprise, qu'il n'y aura pas de tranche horaire erronée comme « 9h-8h », etc. Par contre, on ne supposera pas que les enregistrements du fichier « agenda.txt » sont triés par ordre croissant des semaines.

Exercice 3.

Ecrire un script Perl qui rajoute la date : « 12/4/2005 » au début du fichier « dates.txt » de l'exercice 1. Si le fichier n'existe pas, on le crée et le remplit avec cette date.

Exercice 4.

Ecrire une expression régulière permettant de vérifier qu'une chaîne :

- commence par « Mme » ou « Mlle » ou « M. »
- commence par un chiffre ou un signe (« + » ou « - »)
- se termine par un « \$ »
- contient au moins un caractère différent d'une lettre

Exercice 5.

Ecrire une instruction Perl utilisant une expression régulière et permettant d'extraire d'une chaîne \$s respectivement :

- ses deux premiers caractères
- ses deux premiers mots (un mot étant considéré comme une suite non vide de lettres minuscules ou majuscules)
- son dernier caractère non-numérique

Exercice 6.

Ecrire une instruction Perl utilisant une expression régulière et permettant de supprimer d'une chaîne \$s respectivement :

- les espaces en fin de chaîne
- la sous-chaîne « http:// » en début de chaîne