

Réseaux 1

TP 9 – Traitement des formulaires HTML

Objectif : introduire au traitement des formulaires html

Note : vous placerez vos fichiers html et cgi sur nyx, dans un répertoire ~/web/rx1/TD9

1. Récupérer les données transmises

Un petit formulaire d'essai

- 1.1. Construisez un petit formulaire d'inscription et qui demandera les informations suivantes :
 - nom (20 caractères au maximum)
 - age (entre 10 et 30)
 - sexe (boutons radio : fille, garçon)
 - adresse e-mail
 - sport (cases à cocher : foot, judo, danse ..)
 - catégorie (liste déroulante : benjamin, minime, cadet ..)
- 1.2. Ecrire le programme CGI qui récupère les données transmises par le formulaire et les retourne dans une page html. Pour réaliser l'extraction et le décodage de la chaîne de données transmises, vous utilisez le programme `query.c` donné en Annexe.
- 1.3. Ajouter en Javascript une fonction qui vérifie les conditions suivantes :
 - tous les champs sont renseignés
 - l'age est un entier compris entre 10 et 30
 - l'adresse mail contient un @ et un « . »Si ces conditions ne sont pas vérifiées, on refuse d'exécuter l'action associée au formulaire.

2. Suivi de session à l'aide de champs cachés

Jeu du nombre secret

- 2.1. Ecrire un programme CGI en C qui propose de jouer au nombre secret.

Le programme a un nombre secret que l'utilisateur doit trouver. Ce dernier sait au départ dans quel intervalle se trouve ce nombre et il peut proposer à chaque tour de jeu un nombre. Si c'est le nombre secret, on lui indique qu'il a gagné, sinon on lui indique si le nombre proposé est trop petit ou trop grand par rapport au nombre secret. Programmer ce jeu, on suppose que c'est toujours le même nombre qui est à trouver.
Suggestion : utilisez un champ caché pour compter le nombre d'essais successifs.

QCM sur les capitales

- 2.2. Ecrire un programme CGI en C permettant de construire un petit QCM sur les capitales.

Les questions seront stockées dans un fichier texte Unix avec le format suivant : une ligne par question comportant le nom d'un pays suivi des capitales à choisir, suivi du numéro de la bonne réponse. Les champs sont séparés par un caractère « : »
par exemple :
Finlande:Oslo:Stockholm:Helsinki:Copenhague:3
Hongrie:Budapest:Varsovie:Prague:Bucarest:1
On supposera que pour chacune des questions, une seule réponse est possible (boutons radios)

Au départ le programme envoie l'ensemble des questions et les propositions de réponses. L'utilisateur pourra envoyer le questionnaire et connaître ses résultats en cliquant sur un bouton après avoir répondu à toutes les questions. Le programme retourne les résultats, avec un rappel de chacune des questions, et la réponse que l'utilisateur a sélectionné (de couleur verte si elle est correcte et rouge sinon). En fin des résultats le nombre d'erreurs total.

- 2.3. Ecrire une deuxième version, permettant d'envoyer les questions les unes après les autres sur des pages séparées. Vous utiliserez des champs cachés pour conserver les réponses.

3. Annexe – le programme query.c

Note : ce programme est fourni par Apache, il se trouve sur *nyx* dans `/home/www/site.DEPTINFO/cgi-src` (`query.c` utilise les fonctions de base contenues dans le programme `util.c`)

```
#include <stdio.h>
#ifndef NO_STDLIB_H
#include <stdlib.h>
#else
char *getenv();
#endif
#include <string.h>

typedef struct {
    char name[128];
    char val[128];
} entry;

void getword(char *word, char *line, char stop);
char x2c(char *what);
void unescape_url(char *url);
void plustospace(char *str);

main(int argc, char *argv[]) {
    entry entries[100];
    register int x,m=0;
    char *cl;

    printf("Content-type: text/html%c%c",10,10);

    if(strcmp(getenv("REQUEST_METHOD"),"GET")) {
        printf("This script should be referenced with a METHOD of GET.\n");
        exit(1);
    }

    cl = getenv("QUERY_STRING");
    if(cl == NULL) {
        printf("No query information to decode.\n");
        exit(1);
    }
    for(x=0;cl[0] != '\0';x++) {
        m=x;
        getword(entries[x].val,cl,&');
        plustospace(entries[x].val);
        unescape_url(entries[x].val);
        getword(entries[x].name,entries[x].val, '=');
    }

    printf("<H1>Query Results</H1>");
    printf("You submitted the following name/value pairs:<p>%c",10);
    printf("<ul>%c",10);

    for(x=0; x <= m; x++)
        printf("<li> <code>%s = %s</code>%c",entries[x].name,
            entries[x].val,10);
    printf("</ul>%c",10);
}
```