

TP n°3 - sessions et cookies

Site: [Université de Reims Champagne-Ardenne](#)

Cours: INFO0303 - Technologies Web 2

Livre: TP n°3 - sessions et cookies

Imprimé par: CYRIL RABAT

Date: vendredi 15 septembre 2023, 10:53

Table des matières

1. Introduction aux sessions

2. Jouons avec les cookies

3. Rejeu de formulaires

1. Introduction aux sessions

Pour débiter avec les sessions en PHP, nous allons développer un jeu de labyrinthe. Pour commencer, nous supposons que le labyrinthe est représenté par un tableau, défini comme une constante :

```
define("LABY", [  
    [ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ],  
    [ 1, 0, 0, 0, 0, 1, 0, 0, 0, 1 ],  
    [ 1, 0, 1, 1, 0, 1, 0, 1, 1, 1 ],  
    [ 1, 0, 0, 1, 0, 1, 0, 1, 2, 1 ],  
    [ 1, 1, 0, 1, 0, 1, 0, 0, 0, 1 ],  
    [ 1, 0, 0, 1, 1, 1, 0, 1, 1, 1 ],  
    [ 1, 0, 1, 1, 0, 0, 0, 0, 0, 1 ],  
    [ 1, 0, 1, 1, 0, 1, 0, 1, 0, 1 ],  
    [ 1, 0, 0, 0, 0, 1, 0, 1, 0, 1 ],  
    [ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]  
]);
```

La valeur 1 correspond à un mur, la valeur 2 est l'arrivée et la valeur 0 correspond à un vide. Le joueur commence à la position (1, 1) et celle-ci est stockée à l'aide de deux variables de session `posX` et `posY`.

1. Écrivez une fonction `init` permettant d'initialiser le jeu (uniquement la position, pour le moment).
2. Écrivez une fonction `displayForm` permettant d'afficher un formulaire contenant les boutons "Reset", "Gauche", "Haut", "Bas" et "Droite". Elle utilise les variables de session pour savoir s'il y a un mur ou non en haut, à droite, en bas ou à gauche. S'il y a un mur, le bouton est désactivé (attribut `disabled`). Si le joueur est sur l'arrivée, on affiche un message "Bravo !!!!".
3. Écrivez le script PHP général. Il appelle la fonction `init` si les variables de session n'existent pas. Si un bouton a été cliqué, la position est mise à jour (vérifiez bien que le joueur ne va pas dans un mur).

Nous souhaitons maintenant ajouter la possibilité au joueur d'avoir une aide (3 fois au cours de la partie). Lorsqu'il presse le bouton "Help", son trajet est affiché (pas les murs, juste les cases parcourues). L'affichage peut être réalisé très simplement avec un tableau HTML (on peut spécifier une largeur et une hauteur pour chaque élément "td" et on peut spécifier une couleur de fond pour les cases parcourues).

4. Comment stocker tous les mouvements du joueur ?
5. Faites une fonction `displayHelp` permettant d'afficher les précédents mouvements.
6. Adaptez le script pour ajouter cette fonctionnalité (n'oubliez pas de modifier la fonction `init`).

2. Jouons avec les cookies

Pour tester l'utilisation des cookies, nous allons écrire plusieurs scripts.

1. Écrivez un script `index.php` qui vérifie si l'utilisateur a déjà visité le site ou non en utilisant un cookie. Quand l'utilisateur arrive sur le site pour la première fois, le script affiche : « Je ne vous connaissais pas, mais maintenant si ! ». Sinon, le script affiche « Je vous connais ». Utilisez une durée de validité de quelques secondes pour faciliter vos tests.
2. Créez deux autres scripts : un qui permet de supprimer le cookie (`delete.php`) et un autre qui permet d'indiquer si le cookie existe ou non (`state.php`). Ajoutez sur chaque script, un lien vers les deux autres scripts pour simplifier les tests.
3. Lorsque vous créez le cookie, vérifiez à l'aide de `print_r` que le cookie n'existe pas dans le tableau `$_COOKIE` après l'appel à `setcookie`. De même, lors de la suppression, vérifiez que le cookie existe toujours dans le tableau `$_COOKIE`.
4. Que se passe-t-il lorsque le cookie est créé et que vous accédez au script `state.php` via un onglet de navigation privée ? Peut-on créer des cookies dans un onglet de navigation privé ?
5. À l'aide des outils de navigation de votre navigateur, vérifiez les données envoyées.
6. Refaites les tests précédents en utilisant un cookie de session.

3. Rejeu de formulaires

L'une des problématiques des applications Web est d'éviter le rejeu des formulaires. Cela intervient lorsqu'un utilisateur valide à nouveau un même formulaire. Nous allons simuler le contenu d'une base de données par un fichier texte où chaque ligne contient un enregistrement (une chaîne de caractères).

1. Écrivez un script `index.php` qui affiche la liste des chaînes de caractères présentes dans le fichier `db.txt`, ainsi qu'un lien vers le script `add.php`. Vous devez gérer le cas où le fichier `db.txt` n'existe pas.
2. Écrivez un script `add.php` qui contient un formulaire avec un simple textarea pour permettre à l'utilisateur d'écrire une chaîne de caractères. Lorsque le formulaire est validé, le script `add.php` est appelé à nouveau. En cas d'erreur, le formulaire est affiché de nouveau (avec un message d'erreur), sinon l'enregistrement est ajouté à la fin du fichier puis un message est affiché pour prévenir l'utilisateur. N'oubliez pas un lien qui permet de retourner sur le script `index.php`.

Pour ajouter une ligne à la fin d'un fichier, récupérez le contenu sous forme de tableau avec la fonction `file` puis ajoutez une nouvelle case au tableau avec les données saisies. Utilisez ensuite la fonction `file_put_contents` en spécifiant le tableau comme nouveau contenu.

3. Vérifiez le bon fonctionnement de votre application en ajoutant plusieurs enregistrements.

Maintenant que votre application est prête, vous pouvez tester le rejeu du formulaire :

4. Ajouter un enregistrement. Lorsque vous avez le message qui indique que l'enregistrement a été ajouté, rechargez la page plusieurs fois. Retournez sur le script principal et vérifiez le résultat obtenu.

Pour empêcher le rejeu, nous générons un identifiant aléatoire que nous plaçons dans le formulaire (champ caché) et en session. Lors du traitement du formulaire, on vérifie que les deux valeurs sont identiques (et on supprime l'entrée en session).

Pour générer un identifiant unique, vous pouvez utiliser la fonction PHP `uniqid`.

5. Modifiez vos scripts. N'hésitez pas à écrire des fonctions pour la génération et la récupération de l'identifiant.