

More or Less – PHP et Cookies

Prérequis : Avoir des bases en HTML / CSS / PHP et sur les Cookies

Avoir un serveur Apache installé sur votre appareil

Dans ce tuto, nous allons vous apprendre à créer votre propre jeu du « More or Less » en PHP et en utilisant les cookies.

Le principe du jeu est d'attribuer aléatoirement une valeur comprise entre 0 et 100 (dans ce cas présent) à une variable, et l'utilisateur doit tenter de trouver ce nombre en le moins de coups possibles à l'aide des indications retournées par l'ordinateur.

Nous utiliserons le système de cookies, qui permettront de sauvegarder les valeurs correspondant au nombre de coups joués, ainsi que la valeur à deviner.

1ère étape : Créer le corps du jeu en HTML

Pour commencer, il est impératif de créer le corps du jeu. Ici, nous aurons seulement une balise formulaire, pour que l'utilisateur valide sa réponse, et que le serveur y réponde en fonction. Bien que ce soit du code HTML, du code PHP y sera intégré par la suite. Il aura donc une extension .php.

Pour notre part, ce sera index.php, et se présentera sous cette forme :

```
index.php x
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <title>More or Less</title>
6 </head>
7 <body>
8   <form action="index.php" method="post">
9     <label>Saisissez une valeur (entre 0 et 100) : </label>
10    <input type="number" name="valeur">
11    <input type="submit" name="submit">
12  </form>
13  <p>Nombre de coups joués : </p>
14 </body>
15 </html>
16
```

Nous utiliserons la méthode POST, qui permettra de récupérer les valeurs insérées dans l'input et les réutiliser.

2ème étape : Création des cookies

En second temps, nous allons créer les cookies, pour stocker la valeur à deviner, ainsi que le nombre de coups joués.

ATTENTION : Un cookie se déclare TOUJOURS avant tout code HTML.

A screenshot of a code editor window titled 'index.php'. The code is as follows:

```
1 <!-- Initialisation des cookies -->
2 <?php
3 setcookie('valeur', rand(0,100));
4 setcookie('nombreCoups', 0);
5
6 ?>
7
```

Hein ? Mais, il ne faut pas 3 paramètres à un cookie ? Il me semblait qu'il lui fallait déclarer une date d'expiration, non ?

Mais dans ce cas présent, lorsque nous ne précisons pas de date d'expiration, cela signifie qu'à la fermeture du navigateur, ce dernier sera automatiquement effacé. Ainsi, lorsque l'utilisateur reviendra, une nouvelle partie sera lancée.

3ème étape : Initialisation des conditions / comparaisons

Maintenant que nos cookies sont créés, il va falloir interagir avec le formulaire.

Avec la gestion d'erreurs et affichage des messages, notre code ressemble maintenant à ça :

```
index.php
1  <!-- Initialisation des cookies -->
2  <?php
3  setcookie('valeur', rand(0,100));
4  setcookie('nombreCoups', 0);
5
6  ?>
7
8  <!DOCTYPE html>
9  <html lang="fr">
10
11  <head>
12  <meta charset="UTF-8">
13  <title>More or Less</title>
14  </head>
15
16  <?php
17
18  // Initialisation des message d'erreur et de succès
19  $msgErr = $msgSucc = "";
20  $msg = "";
21  // Lorsque l'utilisateur clique sur le bouton submit
22  if (isset($_POST['submit'])) {
23      if(!empty($_POST['valeur'])){
24          //Si l'utilisateur a saisi un chiffre, nous faisons apparaître un message contenant ce chiffre
25          $msg .= "La valeur " . $_POST['valeur'] . " a été saisie";
26      } else {
27          //Si l'utilisateur clique sur le bouton sans avoir saisi de valeur, un message d'erreur apparait
28          $msgErr .= "Aucune valeur n'a été saisie";
29      }
30  }
31
32  ?>
33
```

Maintenant que les messages d'erreurs sont gérés, on va s'atteler aux comparaisons, en ajoutant des conditions sur les nombres hors du champs (0 - 100). Pour ce faire, le "empty" de la ligne 22 a été modifié par un "isset", étant donné que le "empty" n'accepte pas la valeur 0. Pour appeler un cookie, il faut utiliser la variable \$_COOKIE['sonNom']. Ici par exemple, nous appellerons les cookies grâce aux variables \$_COOKIE['valeur'] et \$_COOKIE['nombreCoups'].

```

15
16 <?php
17
18 // Initialisation des message d'erreur et de succès
19 $msgErr = $msgSucc = "";
20 // Lorsque l'utilisateur clique sur le bouton submit
21 if (isset($_POST['submit'])) {
22     if(isset($_POST['valeur'])){
23         // Si la valeur saisie est inférieure, l'ordinateur affiche "C'est plus !"
24         if ($ _POST['valeur'] < 101 && $ _POST['valeur'] >= 0 ) {
25             if ($ _POST['valeur'] < $_COOKIE['valeur']) {
26                 $msgErr .= "C'est plus !";
27             // Si la valeur saisie est supérieure, l'ordinateur affiche "C'est moins !"
28             } else if ($ _POST['valeur'] > $_COOKIE['valeur']) {
29                 $msgErr .= "C'est moins !";
30             } else {
31                 // Si la valeur saisie est égale, l'ordinateur affiche "C'est gagné !"
32                 $msgSucc .= "C'est gagné !";
33             }
34         } else {
35             $msgErr .= "Tapez une valeur comprise entre 0 et 100";
36         }
37     } else {
38         //Si l'utilisateur clique sur le bouton sans avoir saisi de valeur, un message d'erreur apparait
39         $msgErr .= "Aucune valeur n'a été saisie";
40     }
41 }
42
43 ?>
44

```

```

40
41 <body>
42 <!-- Création du formulaire, et affichage du nomnbre de coups joués -->
43 <form action="index.php" method="post">
44     <label>Saisissez une valeur (entre 0 et 100) : </label>
45     <input type="number" name="valeur">
46     <input type="submit" name="submit">
47 </form>
48 <p>Nombre de coups joués : <?php echo $_COOKIE['nombreCoups'] ?></p>
49 <span><?php echo "<font color='red'>$msgErr</font>" ;?></span>
50 <span><?php echo "<font color='green'>$msgSucc</font>" ;?></span>
51 </body>
52 </html>
53

```

Tout fonctionne ! Mais .. Lorsque je rentre 2 fois la même valeur, il m'affiche une fois "C'est plus", et la fois d'après, "C'est moins". Comment ça se fait? Pareil, le compteur de coups joué reste à 0?

Et pour cause, lorsque l'utilisateur clique sur "Valider", la page est rafraîchie, et il en est de même pour le cookie, qui se voit attribuer une nouvelle valeur aléatoire.

Pour le cookie de "coups joués", il va falloir créer une incrémentation pour que ce dernier augmente de 1 à chaque coup joué.

Commençons par le cookie des “coups joués”.

Pour pouvoir l’incrémenter, il faut donc réaliser une modification du cookie en lui même.

```
index.php x
1 <!-- Initialisation des cookies -->
2 <?php
3 setcookie('valeur', rand(0,100));
4 setcookie('nombreCoups', ($_COOKIE['nombreCoups']+1));
5
6 ?>
7
```

Ici, le cookie aura donc une incrémentation de 1 après chaque rafraîchissement de page, correspondant à une soumission d’une valeur.

Ok, jusque là, ça marche, le nombre de coups joués augmente bien de 1 à chaque fois. Mais qu’en est t-il de la valeur qui change à chaque coups?

On y vient ! Une méthode efficace revient à dire : “Si le cookie n’existe pas, tu le créé. Si il existe déjà, tu ne fais rien”. De cette manière, le cookie ne sera définie qu’une fois, et la valeur ne sera pas modifiée !

Voilà comment procéder :

```
index.php x
1 <!-- Initialisation des cookies -->
2 <?php
3 { if (!isset($_COOKIE['valeur'])) {
4   setcookie('valeur', rand(0,100));
5 } }
6 setcookie('nombreCoups', ($_COOKIE['nombreCoups']+1));
7
8 ?>
9
```

Cool ! Ca fonctionne enfin ! Par contre, on ne peut plus rejouer.. C’est plutôt amusant comme jeu, alors comment faire pour recommencer une partie ?

Il est possible de recommencer une partie en ajoutant par exemple un petit bouton, qui apparaît en même temps que le message de victoire par exemple.

Cependant, il faudra aussi procéder au reset des Cookies. Pour ça, il faudra les détruire, puis les redéfinir, par exemple lorsque l’utilisateur clique sur un bouton “nouveau jeu” :


```

56
57 <body>
58 <!-- Création du formulaire, et affichage du nombre de coups joués -->
59 <form action="index.php" method="post">
60 <label>Saisissez une valeur (entre 0 et 100) : </label>
61 <input type="number" name="valeur">
62 <input type="submit" name="submit">
63 </form>
64 <p>Nombre de coups joués : <?php echo $_COOKIE['nombreCoups'] ?> </p>
65 <span><?php echo "<font color='red'>$msgErr</font>" ;?></span>
66 <span><?php echo "<font color='green'>$msgSucc</font>" ;?></span>
67 <?php if ($msgSucc !== ""): ?>
68 <form action="index.php" method="post">
69 <input type="submit" name="newgame" value="Recommencer">
70 </form>
71 <?php endif ?>
72
73 </body>
74 </html>
75

```

Et au niveau des cookies, cela donnerait:

```

index.php x
1 <!-- Initialisation des cookies -->
2 <?php
3
4 // Si l'utilisateur clique sur "Recommencer", les cookies sont détruits
5 if (isset($_POST['newgame'])){
6     setcookie('valeur', '', time()-100000);
7     setcookie('nombreCoups', '', time()-100000);
8     Header('Location: index.php');
9 }
10
11 // Les cookies étant détruits, il faut les réinitialiser
12 if (!isset($_COOKIE['valeur'])){
13     setcookie('valeur', rand(0,100));
14     setcookie('nombreCoups', 0);
15     Header('Location: index.php');
16 } else {
17     setcookie('nombreCoups', ($_COOKIE['nombreCoups']+1));
18 }
19
20 ?>
21

```

Et voilà, vous avez un jeu du “C’est plus, c’est moins” totalement fonctionnel !

A vous de trouver un moyen de l’améliorer, par exemple en faisant afficher une liste des nombres saisis

Enjoy !