Image anti-bot et fichier wav

Auteur : Arnaud
Créé le : 22/10/2006 à 13h02
Modifié le : 02/12/2006 à 21h36
Noter et commenter ce tutorial
Imprimer ce tutorial

Vous vous apprêtez à lire un tutorial rédigé par un membre de ce site. Malgré tout le soin que ce membre a pu apporter au tutorial, nous ne pouvons pas garantir que les informations contenues sur cette page sont exactes à 100%. Merci de garder cela en tête lorsque vous lirez cette page ;o)

Votre site est envahi par des inscriptions de bot (des robots qui s'inscrivent plusieurs fois à des sites dans le but de publier des messages publicitaires, sur un livre d'or, un forum etc).
Il existe de nombreuses solutions pour empêcher les bots d'agir, comme proposer un champ avec une opération mathématique à résoudre (simple à réaliser), mais dans ce tutoriel nous allons créer une image aléatoire composée de chiffres, avec un lien vers un fichier son qui contiendra la formule vocale du code afin que les personnes mal-voyantes puissent s'inscrire sur votre site.
Nous allons tout d'abord simplement ( ) générer l'image aléatoire. Dans ce TP elle sera composé uniquement de chiffres mais rien ne vous empêche d'inclure des lettres.
Nous procéderons donc ainsi,

- Créer une image aléatoire et vérifier que celle-ci correspond au code
- Nous verrons comment générer le fichier wave (fichier son) qui dictera aux visiteurs les chiffres à inscrire dans le champ du formulaire.

Allez au boulot

Sommaire du chapitre :

L'image aléatoire

Notre script sera composé de 3 fichiers.

- Le fichier formulaire.php contiendra l'image et le formulaire d'inscription par exemple (ici, il vérifiera si le code est le bon, mais rien ne vous empêche de faire une autre page)
- Le fichier image.php qui sera l'image png générée par php
- Le fichier sound.php qui générera un fichier wave

On va commencer par le fichier image.php.

L'image

Ce que l'on veut

- Un nombre à 5 chiffres, ce nombre sera utilisé comme une chaîne de caractères.
- Une image de ces chiffres.
- Des éléments "perturbateurs", tels qu'une barre.

Pour obtenir une image de ce type :

39959
Code aléatoire

La chaîne aléatoire

Une chaîne simple composée de 5 chiffres.

Code : PHP

```php
function chaineAleatoire($nombre = 5)
{
	$chaine = '';
	for($i = 0; $i < $nombre; $i++)
	{
		$chaine .= mt_rand(0,9);
	}
	return $chaine;
}
```

image.php

Il existe d'autres méthodes bien entendu.

Je vous conseille quand même de lire tout le cours PHP de M@teo21 en particulier le cours sur la librairie GD si ce n'est déjà fait afin de comprendre le code suivant :

L'image

Code : PHP

```php
function image($chaine)
{
	$largeur = (strlen($chaine) * 10); // Environ la largeur d'un caractere
	$hauteur = 20; // La bonne hauteur
	$image = imagecreate($largeur,$hauteur);

	$blanc = imagecolorallocate($image, 255, 255, 255); // On colore tout en blanc
	$noir = imagecolorallocate($image, 0, 0, 0);

	$milieuHauteur = ($hauteur / 2) - 8; // Pour centrer le texte en hauteur
	imagestring($image, 6, (strlen($chaine) / 2 ), $milieuHauteur, $chaine, $noir);
// On ecrit au milieu ( augmenter le nombre de caractères pour voir ;) )
	ImageRectangle ($image, 1, 1, $largeur - 1, $hauteur - 1, $noir); // Le rectangle
autour pour l'esthétique
		$hauteur1 = mt_rand(2,$hauteur); // Barre aléatoire
		$hauteur2 = mt_rand(2,$hauteur); // Fin de la barre aléatoire

	ImageLine ($image, 2,$hauteur1, $largeur - 2, $hauteur2, $noir); // Barre
aléatoire
	ImageLine ($image, 2, $milieuHauteur + 8, $largeur - 2, $milieuHauteur + 8,
$noir); // Barre standard
	imagepng($image); // On dessine
}
```

image.php (suite)

Et de deux

Enregistrez le fichier image.php avec ces deux fonctions.

Le Formulaire

Code html basique

Ce code est le code d'un formulaire simple, nous incorporerons l'image de validation après.

Code : HTML

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" ><head>
   <title>Image Anti-Bot</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<link rel="stylesheet" media="screen" type="text/css" title="" href="" />
</head>

<body>
<div>
	<form action="" method="post">
		<label>Pseudo : <input type="text" name="pseudo" /></label><br />
		<img src="image.php?nombre=5" alt="" /><br /> <!-- on appelle l'image php
-->
		<label>Cle : ( respectez la case ) <input type="text" name="cle"
/></label><br />
		<input type="submit" value="envoyer" />
	</form>
</div>
</body>
</html>
```

formulaire.php

Vérifions que le code de l'image correspond à ce qui est entré

Nous aurons besoin des sessions car le code sera stocké dans une variable de session là encore référez vous au cours de php de ce site

Code : PHP

```php
session_start();
```

image.php (tout en haut)
formulaire.php (tout en haut)

En haut de formulaire.php et de image.php

De plus, image.php est normalement une image nous avons donc besoin de le spécifier au navigateur par le biais d'un header :

Code : PHP

```php
header ("Content-type: image/png");
```

image.php (tout en haut)

Code : PHP

```php
$chaine = chaineAleatoire($_GET['nombre']); // Comme on l'appelle avec ?nombre=5 (
caracteres )
$_SESSION['chaine'] = $chaine; // Tout l'interet des sessions ;)
image($chaine); // On genere l'image avec la chaine obtenu.
```

image.php (tout en bas)

Maintenant, il va falloir modifier le fichier qui "réceptionne" le formulaire, dans notre cas ce sera formulaire.php.

1. On vérifie si on a cliqué sur le bouton .
2. On vérifie si le code est le même.

Code : PHP

```php
<?php
session_start();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
<head>
   <title></title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<link rel="stylesheet" media="screen" type="text/css" title="" href="" />
</head>

<body>
<?php
if(isset($_POST['cle']))
{
        if($_POST['cle'] == $_SESSION['chaine'])
        {
                echo 'Votre nom est '. $_POST['pseudo'];
        }
        else
        {
                echo 'Cle non identifié';
                echo '<br /><a href="formulaire.php">Ressayer</a>';
        }
}
else
{
?>
<div>
        <form action="" method="post">
                <label>Pseudo : <input type="text" name="pseudo" /></label><br />
                <img src="aleatoire.php?nombre=5" alt="" /><br />
                <label>Cle : ( respectez la case ) <input type="text" name="cle"
/></label><br />
                <input type="submit" value="envoyer" />
        </form>
</div>
<?php
}
?>
</body>
</html>
```

formulaire.php (final)

A ce stade vous avez déjà un bon formulaire anti-bot, fonctionnel, mais il manque quelque chose, en effet, le web se veut accessible, et vous devez y contribuer.
Imaginons donc la situation suivante, votre espace-membre donne accès à un contenu riche et intéressant.
Une personne ayant des problèmes de vue se rend sur votre site à l'aide d'un navigateur auditif ou braille etc. Vous commencez à comprendre ?
Cette personne ne peut s'inscrire. Comme sur les sites professionnels, je vais vous apprendre à créer un fichier wave qui "lira" l'image à vos visiteurs !

Le fichier wave

Je ne suis pas sûr que vous comprendrez toutes les notions utiliséesfi ici, néanmoins vous pouvez vous renseigner sur le format wave, il existe quelques sites qui vous en apprendrons un peu plus sur la façon dont est écrit un fichier wave, voici quelques unes de ces adresses :

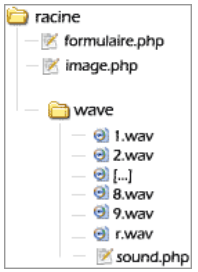
- http://www.freesoundeditor.com/incagen.html?docwave.htm-main
- http://crteknologies.free.fr/programmation/ressources/wav.html
- http://www.sonicspot.com/guide/wavefiles.html
- http://www.neurotraces.com/scilab/scilab2/node24.html

- http://www.borg.com/~jglatt/tech/wave.htm

Ensuite regardez du côté de pack(), unpack().

Je vais essayer d'être le plus clair possible tout de même.
Vous devez savoir qu'un fichier contient généralement un en-tête, ici c'est sur celui-ci que nous allons le plus travailler l'en-tête d'un fichier est écrite en binaire, heureusement les fonctions pack() et unpack savent faire la transcription entre nombres entiers, chaînes de caractères etc. et langage binaire. La première transcrit une variable d'une forme que l'on peut traiter avec php (int, string, bool etc) à une forme binaire (suite de 0 et de 1). La fonction unpack fait l'inverse.
Chaque fichier a son propre en-tête et son propre type d'encodage des données. Ici nous nous occuperons d'un fichier wave, mais le principe est le même pour les zip, les images etc.

Voici l'arborescence du script fini,



Arborescence

Nous avons déjà écrit les deux fichiers : formulaire.php et image.php
Le fichier sound.php est une classe (lien vers un tuto d'un zéro)"wave" composé des méthodes suivantes :

- __construct : le constructeur, qui entraîne la génération du fichier wave.
- analyse : Analyse l'en-tête d'un fichier wave et retourne soit toutes les caractéristiques soit une seule (taille par exemple).
- get_datas : renvoi les données brutes d'un fichier wave sans l'en-tête.
- package : mélange les données brutes de plusieurs wave et ajoute l'en-tête approprié. Renvoi le fichier final.

Liens vers les fichiers wav : Télécharger.

Analyse d'un fichier wave

Il faut savoir qu'un fichier wave est composé d'un en-tête (de 44 octets en l'occurrence ici) et du contenu à proprement parler du fichier.
Cet en-tête permet d'avoir des informations concernant la vitesse d'échantillonnage, la durée, le nombre de bits etc.

Je vous épargne tout cela seulement voila une fonction que j'ai écrite qui gère l'en-tête :
Code : PHP

```
                public function analyse($wav,$arg = FALSE)
                {
                        if(file_exists('../'.$wav))
                        {

                                $entete_unpack =
'a4file_type/Lfile_size/a4file_id/A4nom_zone1/Ltaille_zone1/SFormatTag/SChannels/LSamplesP

                                $fp = fopen('../'.$wav,'r');
                                $file = fread($fp, 44 );

                                $entete = unpack($entete_unpack,$file);
                                fclose($fp);


                                if( ($entete['file_type'] !==
chr(0x52).chr(0x49).chr(0x46).chr(0x46) ) OR ( $entete['file_id'] !==
chr(0x57).chr(0x41).chr(0x56).chr(0x45) ) )
                                {
                                        /* La fonction trigger_error genère une erreur
comme celles qui apparaissent lorsque vous oublier un argument d'une fonction, un point
virgule etc... ( consultez la doc pour plus d'informations ) */
                                        trigger_error('Le fichier '.$wav.' n ''est pas un
fichier wav',E_USER_ERROR); // je double l'apostrophe pour mieux colorer le code sur le
site, mais dans vos code echaper juste une seule apostrophe : \'
                                }

                                if(!$arg)
                                        return $entete;
                                else
                                        return $entete[$arg];
                        }
                        else
                        {

                                trigger_error('fonction analyse, le fichier '.$wav.' est
neccessaire mais manquant', E_USER_ERROR);
                                return FALSE;
                        }

                }
```

sound.php

Si vous ne comprenez pas tout ce qui est écrit, ce n'est pas très grave sachez seulement que la fonction prend le nom d'un fichier wav, l'ouvre, lit l'en-tête, et renvoi au choix :

- Tout l'en-tête
- Une seule caractéristique

Le constructeur de la classe

Le paramètre à lui transmettre est le nombre à dicter, le constructeur lancera la procédure de création du wave en appelant les bonnes fonctions :

Citation : Paramètre du constructeur

(int) nombre

Ce qui en php donne cela :

Code : PHP

```
public function __construct($nombre)
                {
                        if(!empty($nombre))
                        {
                                $this->nombre = $nombre.'r'.$nombre; // Nombre, "je
repete", nombre
                                $strlen = strlen($this->nombre);

                                $binary = ''; // Contiendra les donnees, juste les
donnees.

                                /* Parcours chaque fichier wav qui contient un nombre
utilisé dans la chaine */
                                for($i = 0; $i < $strlen; $i++)
                                {
                                        $binary .=
$this->get_datas('../'.$this->nombre{$i}.'.wav');
                                }

                                $this->package($binary);
                        }
                }
```

sound.php (constructeur)

Ca ne devrait pas vous poser de problème normalement  .

Récupérer les données du fichier wav

Notre fonction get_datas(), doit récupérer les données brutes, c'est à dire le son proprement dit d'un fichier wav, c'est juste l'utilisation de fopen, fread, fclose. Seulement la fonction analyse() va nous donner la taille en octets des données brutes.

Voici la fonction get_datas, j'explique en commentaires.

Code : PHP

```
private function get_datas($wav)
                {
                        if(file_exists('../'.$wav))
                        {
                                $fp = fopen('../'.$wav,'r');
                                fseek($fp, 44 ); // On place le curseur à 44 octets,
juste après les données

                                $datas = fread($fp, $this->analyse($wav, 'taille_data'));
// On lit juste la taille des données d'après analyse().

                                fclose($fp);

                                return $datas; // On renvoi au constructeur qui ajoutera
les données à la fin de la chaine des données.
                        }
                        else
                        {
                                trigger_error('fonction analyse, le fichier '.$wav.' est
neccessaire mais manquant', E_USER_ERROR);
                                return FALSE;
                        }
                }
```

Ce code n'est lui non plus pas trop compliqué, seules les fonctions analyse() et package() (que nous allons voir) sont un peu plus difficile car elles utilise des fonctions de traitement binaire, et donc il faut connaître la structure d'un

fichier wav.

Package, on crée le fichier wav final

Notre fonction package doit assemblé toutes les données des fichiers requis, ajouter un en-tête et envoyer le tout au navigateur.
Allez je vous laisse .

Vous n'y arrivez pas ? C'est un peu normal, c'est la fonction la plus compliquée de notre classe.

La voici :

Code : PHP


```php
private function package($datas)
		{
			$entete_pack = 'a4La4A4LSSLLSSA4L'; // Entete d'un fichier wav transcrit pour la fonction pack / unpack, cela veut dire que l'on transcrit en binaire 4 chaine, un entier long non signé, 4 chaines etc...

			$entete = $this->analyse('../'.$this->nombre{0}.'.wav'); // On prend l'entete du premier fichier utilisé comme base pour les autres, si les autres n'ont pas les même caractéristiques, on arrete tout (voir plus bas)

			for($i = 1;$i<strlen($this->nombre); $i ++)
			{
				$entete2 = $this->analyse('../'.$this->nombre{$i}.'.wav');
				/* Verification des en-têtes. */
				if($entete !== $entete2)
				{
					if($entete['Channels'] !== $entete2['Channels'])
					{
						trigger_error('fonction package, un ou plusieurs fichiers wad n''ont pas le même nombre de canaux',E_USER_ERROR); // Noté, j'ai encore doublé l'apostrophe
					}

					if($entete['BitsPerSample'] !== $entete2['BitsPerSample'])
					{
						trigger_error('fonction package, un ou plusieurs fichiers wad, n''on pas le même nombres d''échantillons par seconde', E_USER_ERROR);
					}

					if($entete['SamplesPerSec'] !== $entete2['SamplesPerSec'])
					{
						trigger_error('fonction package, un ou plusieurs fichiers wad, n''ont pas la même fréquence d''échantillonage', E_USER_WARNING);
					}
				}
			}

			/* On calcule la taille des donnees que l'on à creer selon la formule de soundeditor (voir liens tout en haut)*/
			$entete['taille_data'] = strlen($datas);
			$entete['taille_data'] /= $entete['BlockAlign'];
			$entete['taille_data'] *= $entete['BlockAlign'];

			$entete['file_size'] = 44 + strlen($datas); // Taille totale du fichier

			/* on "pack" l entete et on y ajoute les donnees */
			/* Transcription en binaire */
			$binary = pack($entete_pack, $entete['file_type'],$entete['file_size'],$entete['file_id'],$entete['nom_zone1'], $entete['taille_zone1'],$entete['FormatTag'],$entete['Channels'],$entete['SamplesPerSec'],$entete['BlockAlign'],$entete['BitsPerSample'],$entete['nom_data'],$entete['taille_data']);
			$binary .= $datas; // Ajoute les données à la fin et c'est bon !!

			/* On transmet ou enregistre */
				header('Content-type: audio/x-wav'); // Indique que le navigateur recoit un fichier wav
				header('Content-Disposition: attachment; filename="code.wav"'); // Indique que l'on transmet le fichier directement
				echo $binary; // Sortie santard.
		}
```

Code final de la classe sound.php

Code : PHP


```php
<?php
	class wave
	{
		public $nombre;
		public function __construct($nombre = 0)
		{
			if(!empty($nombre))
			{
				$this->nombre = $nombre.'r'.$nombre;
				$strlen = strlen($this->nombre);

				$binary = ''; // Contiendra les donnees, juste les
donnees.

				for($i = 0; $i < $strlen; $i++)
				{
					$binary .=
$this->get_datas('../'.$this->nombre{$i}.'.wav');
				}

				$this->package($binary);
			}
		}

		public function analyse($wav,$arg = FALSE)
		{
			if(file_exists('../'.$wav))
			{

				$entete_unpack =
'a4file_type/Lfile_size/a4file_id/A4nom_zone1/Ltaille_zone1/SFormatTag/SChannels/LSamplesP

				$fp = fopen('../'.$wav,'r');
				$file = fread($fp, 44 );

				$entete = unpack($entete_unpack,$file);
				fclose($fp);


				if( ($entete['file_type'] !==
chr(0x52).chr(0x49).chr(0x46).chr(0x46) ) OR ( $entete['file_id'] !==
chr(0x57).chr(0x41).chr(0x56).chr(0x45) ) )
				{
					trigger_error('Le fichier '.$wav.' n\'est pas un
fichier wav',E_USER_ERROR);
				}

				if(!$arg)
					return $entete;
				else
					return $entete[$arg];
			}
			else
			{

				trigger_error('fonction analyse, le fichier '.$wav.' est
neccessaire mais manquant', E_USER_ERROR);
				return FALSE;
			}

		}


		private function get_datas($wav)
		{
			if(file_exists('../'.$wav))
			{
				$fp = fopen('../'.$wav,'r');
				fseek($fp, 44 );

				$datas = fread($fp, $this->analyse($wav, 'taille_data'));

				fclose($fp);

				return $datas;
			}
```

Ouf, c'est fini, oui votre classe fonctionne très bien, cependant il faut l'inclure avec le formulaire.

Script fini

Nous verrons ici comment faire marcher votre formulaire en entier.

Modifications

Le formulaire

Voici les modifications à apporter au fichier formulaire pour qu'il fonctionne avec les autres fichiers.
Image pour lancer le fichier wave : 

Code : PHP

```php
<?php
session_start();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" ><head>
   <title></title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<link rel="stylesheet" media="screen" type="text/css" title="" href="" />
</head>

<body>
<?php
if(isset($_POST['cle']))
{
        if($_POST['cle'] == $_SESSION['chaine'])
        {
                echo 'Votre nom est '. $_POST['pseudo'];
                session_destroy();
        }
        else
        {
                echo 'Cle non identifié';
                echo '<br /><a href="formulaire.php">Ressayer</a>';
        }
}
else
{
?>
<div>
        <form action="" method="post">
                <label>Pseudo : <input type="text" name="pseudo" /></label><br />
                <img src="aleatoire.php?nombre=5&type=numerique" alt="" /><a
href="wave/sound.php"><img src="hp.png" alt="Je ne peux pas voir le message" title="Je ne
peux pas voir le message" /></a><br />
                <label>Cle : ( respectez la case ) <input type="text" name="cle"
/></label><br />
                <input type="submit" value="envoyer" />
        </form>
</div>
<?php
}
?>
</body>
</html>
```

formulaire.php

On a juste rajouté un lien vers sound.php

Le fichier sound.php

Il faut rajouter

Code : PHP

```php
session_start();
```

En haut et

Code : PHP

```php
$wav = new wave($_SESSION['chaine']);
```

En bas

Ce qui donne donc :

Code : PHP

```php
<?php
	session_start();

	class wave
	{
		public $nombre;
		public function __construct($nombre = 0)
		{
			if(!empty($nombre))
			{
				$this->nombre = $nombre.'r'.$nombre;
				$strlen = strlen($this->nombre);

				$binary = ''; // Contiendra les donnees, juste les
donnees.

				for($i = 0; $i < $strlen; $i++)
				{
					$binary .=
$this->get_datas('../'.$this->nombre{$i}.'.wav');
				}

				$this->package($binary);
			}
		}

		public function analyse($wav,$arg = FALSE)
		{
			if(file_exists('../'.$wav))
			{

				$entete_unpack =
'a4file_type/Lfile_size/a4file_id/A4nom_zone1/Ltaille_zone1/SFormatTag/SChannels/LSamplesP

				$fp = fopen('../'.$wav,'r');
				$file = fread($fp, 44 );

				$entete = unpack($entete_unpack,$file);
				fclose($fp);


				if( ($entete['file_type'] !==
chr(0x52).chr(0x49).chr(0x46).chr(0x46) ) OR ( $entete['file_id'] !==
chr(0x57).chr(0x41).chr(0x56).chr(0x45) ) )
				{
					trigger_error('Le fichier '.$wav.' n\'est pas un
fichier wav',E_USER_ERROR);
				}

				if(!$arg)
					return $entete;
				else
					return $entete[$arg];
			}
			else
			{

				trigger_error('fonction analyse, le fichier '.$wav.' est
neccessaire mais manquant', E_USER_ERROR);
				return FALSE;
			}

		}


		private function get_datas($wav)
		{
			if(file_exists('../'.$wav))
			{
				$fp = fopen('../'.$wav,'r');
				fseek($fp, 44 );

				$datas = fread($fp, $this->analyse($wav, 'taille_data'));

				fclose($fp);
```


sound.php

Vous avez maintenant un bon formulaire anti-bot et accessible aux personnes handicapées.

Idées d'améliorations

- Vous pouvez incorporer des lettres aux chiffres du code.
- Changer la voix, mettre la votre ou une voix féminine, lien vers des outils de synthèse vocale, ou ici pour le synthétiseur qui a servi dans ce tuto
- Ralentir le rythme en ajoutant des blancs entre les chiffres (sons vides).

J'espère que cela vous a servi, et que c'était pour vous une courte introduction dans la manipulation de fichiers en binaire.

Auteur : Arnaud
Noter et commenter ce tutorial
Imprimer ce tutorial