TP09 : PHP – HTML : LA RÉCUPÉRATION DES PARAMÈTRES

I - PARAMÈTRES PASSÉS DANS L'URL

Le code suivant permet d'afficher, un damier de 10 lignes sur 10 colonnes : damier.php (voit TP)

```
<table border=1 width=600>
<?php
$ligne=10;
$colonne=10;
 for ($i=0;$i<$ligne; $i++){
    echo '<tr>';
    for ($j=0;$j<$colonne; $j++){
     if(($j+$i)%2==0)
          echo '<td> </td>';
    else
          echo '<td bgcolor=black> </td>';
   }
   echo '</tr>';
 }

?>
</table>
```

Comment faire si l'on souhaite afficher, un damier de 20 lignes sur 10 colonnes ou de 5 lignes sur 8 colonnes etc.....
On peut toujours rentrer dans le code PHP et changer la valeur des variables. Mais si vous n'avez pas accès à ce code, la seule solution si vous n'utilisez pas de formulaire, c'est de transmettre ces paramètres à l'aide de l'URL.
On va ajouter à la fin de l'URL « **?ligne=20&colonne=10** », l'URL deviendra :

http://localhost/~stephane/damier.php?ligne=20&colonne=10

Le point d'interrogation "?" dit au navigateur que les paramètres suivants sont des variables**.** Il faudra modifier votre code PHP pour récupérer ces paramètres.

http://localhost/~stephane/damier.php?ligne=20&colonne=10

Ainsi, en modifiant les valeurs de ligne et colonne dans l'URL, vous modifiez la présentation du damier. Vous pouvez également passez en paramètres, la couleur du damier
http://localhost/~stephane/damier.php?ligne=20&colonne=10&couleur1=red&couleur2=green

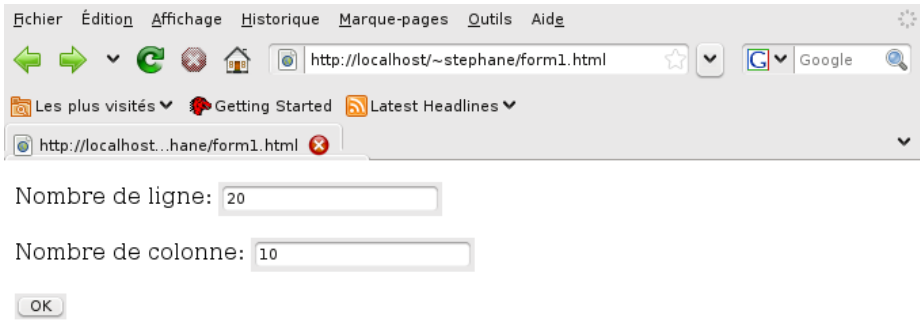
Que va afficher : print_r($REQUEST);
De quel type de variable s'agit-il ? Comment afficher son contenu?

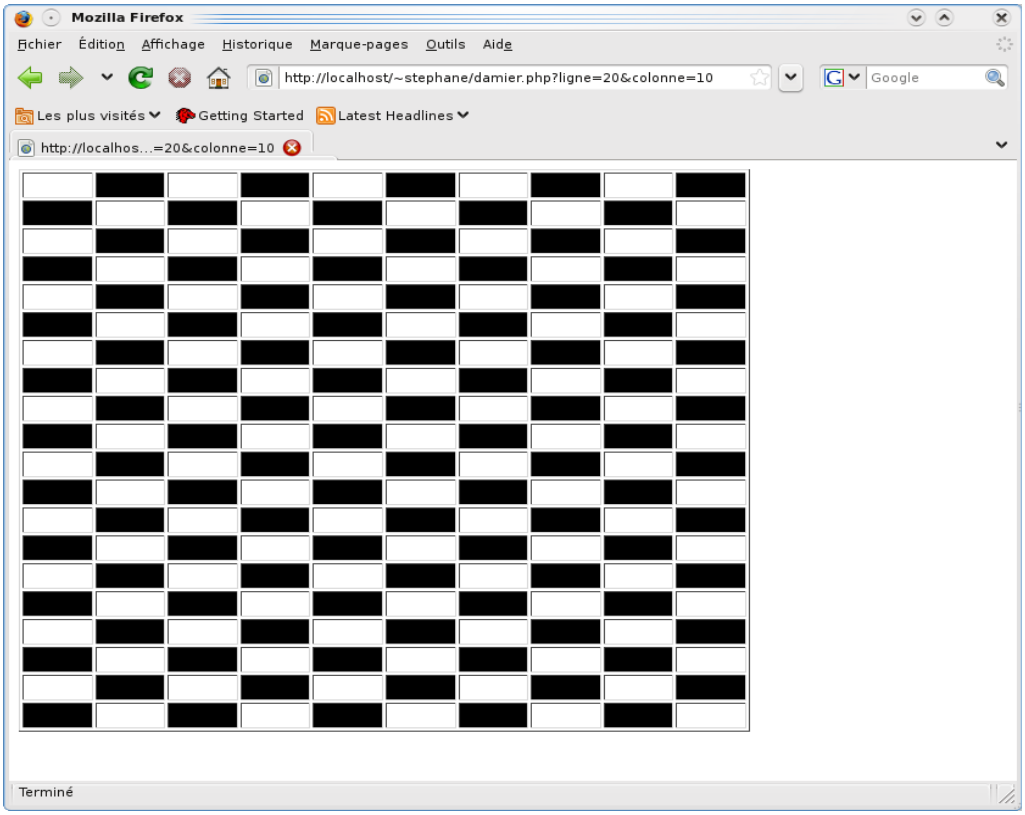
II - PARAMÈTRES PASSÉS À L'AIDE D'UN FORMULAIRE

Généralement pour passer des paramètres dans l'URL, on utilise des formulaires :
exemple : form1.html

```
<form action='damier.php' method='get'>
      <p>Nombre de ligne: <input type="text" name="ligne" /></p>
      <p>Nombre de colonne: <input type="text" name="colonne" /></p>
      <input type="submit" value="OK" />
</form>
```

Nous avons deux zones de saisies , et un bouton. Lorsque l'utilisateur va « cliquer » sur le bouton OK, le navigateur va construire l'URL suivante :
.....damier.php?ligne=nombre_saisie&colonne=nombre_saisie





Le champ action='damier.php', indique le nom du fichier à afficher (si ce fichier se trouve dans un autre répertoire par exemple le répertoire dessin vous devez indiquer action='dessin/damier.php'), les champs « name » indiquent le nom des paramètres à transmettre qui vont prendre comme valeur les valeurs saisies par l'utilisateur.

MÉTHODE GET ET POST

Dans un formulaire vous devez indiquer, la méthode utilisée pour envoyer vos données, il existe 2 méthodes :
La méthode GET (celle qui est utilisée par défaut si rien n'est renseigné) fait circuler les informations du formulaire en clair dans la barre d'adresse en suivant le format ci-après :

```
Exemple d'url créée à partir de la méthode GET d'un formulaire
http://www.unsite.com/chemin/scriptphp.php?var1=valeur1
```

Pour récupérer les variables vous pouvez utiliser **$_REQUEST['var1']** ou **$_GET['var1']**

La méthode POST, quant à elle, transmet les informations du formulaire de manière masquée **mais non cryptée**. Le fait de ne pas afficher les données dans l'URL ne signifie en rien qu'elles sont sécurisées. Elle est préférée lorsqu'il y'a un nombre important de données à transmettre ou bien lorsqu'il faut envoyer des données sensibles comme des mots de passe.
Pour récupérer les variables vous pouvez utiliser **$_REQUEST['var1']** ou **$_POST['var1']**

III - PRÉCAUTIONS DE SÉCURITÉ

Lorsque vous traitez des données issues d'un formulaire, vous devez vous assurer que ces données ont bien été envoyées et quelles ont des valeurs exploitables par votre script PHP. A l'aide des 3 fonctions suivantes, essayer de sécuriser votre script damier.php. Vous devez vérifier dans votre script que les variables ligne et colonne sont bien définies, qu'elles ne sont pas vides et de types numériques

isset()*Détermine si une variable est affectée*

bool **isset** (mixed var , *mixed var* , ...)
isset renvoie TRUE si la variable var est définie, FALSE sinon.
Exemple avec isset

```
//si la variable var est définie, alors le texte est affiché
if (isset($var)) {
  echo 'Cette variable existe, donc je peux l\'afficher.';
}
```

empty() — *Détermine si une variable contient une valeur non nulle*

Retourne **FALSE** si *var* a une valeur non-vide et différente de zéro.
Exemple #1 Une comparaison simple empty() / isset().

```
<?php
$var=0;
//Evalué à vrai car $var est vide ou égale à 0
if (empty($var)){
 echo '$var vaut soit 0, vide, ou pas définie du tout';
}
//Evalué à vrai car $var est défini
if(isset($var)){
  echo '$var est définie même si elle est vide';
}
?>
```

is_numeric — *Détermine si une variable est un type numérique*

bool **is_numeric** (mixed `$var`)

Détermine si la variable donnée est de type numérique

```
$var=15;
$val='test';
if (is_numeric($var)){
 echo '$var est une variable numérique';
}
if (!is_numeric($val)){
 echo '$val n'est pas une variable numérique';
}
```

IV - APPLICATION :

1 - Ajouter 2 listes déroulantes dans votre formulaire pour choisir les couleurs du damier.

2 – Créer un formulaire qui demandera le login et le mot de passe de l'utilisateur, les données seront envoyées à un script authentification.php, dans lequel vous allez initialiser , un tableau associatif, du type clé: login, et valeur : mot de passe
$aut=array('sguyon' => 'aerft' , 'lkila' => 'azerty' ,.....)
Ce script devra indiquer si l'accès au site est permis ou non.