

PHP et Base de données

Pour cette fiche, nous utiliserons le logiciel Xampp.

1 Initiation à Xampp

- Ouvrir le panneau de contrôle de **Xampp**.
- Démarrer les module **Apache** et **SQL**.
- Ouvrir le dossier **explorer**.
- Dans le dossier **htdocs** , créer un dossier **NSI**.
- Dans ce dossier,nous allons créer notre page web, enregistrer sous le nom **ma_page.php** :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ma page</title>
    <meta charset = "utf-8">
    <link rel ="stylesheet" href="ma_feuille_style.css" />
  </head>
  <body>

    <?php
      echo '<p>Bonjour le monde </p>';
    ?>

  </body>
</html>
```

- Pour ouvrir la page sur un navigateur, l'adresse est : **localhost/NSI/ma_page.php** (ou **1127.0.0.1/NSI/ma_page.php**).

2 Base du PHP

2.1 Les variables

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ma page</title>
    <meta charset = "utf-8">
    <link rel ="stylesheet" href="ma_feuille_style.css" />
  </head>
  <body>

    <?php
```

```
$nb_1 = 5;
$nb_2 = 13;
$somme = $nb_1 + $nb_2 ;
$phrase = "La somme est : $somme";
$phrase_1 = 'La somme est : $somme';
$phrase_2 = 'La somme est : '.$somme;
echo "<p> $phrase </p>" ;
echo "<p> $phrase_1 </p>" ;
echo "<p> $phrase_2 </p>" ;
?>

</body>
</html>
```

2.2 Les tableaux

```
<?php
$mon_tableau = array( 8,9,4,7);
$mon_tableau[2] = 15 ;
$taille = sizeof($mon_tableau ) ;
echo "<p> $taille </p>" ;
for ( $i = 0 ; $i<$taille ; $i++)
{
    echo "<p> $mon_tableau[$i] </p>" ;
}
foreach($mon_tableau as $element )
{
    echo "<p> $element </p>" ;
}
$indice = $taille -1;
while ( $indice >= 0 )
{
    echo "<p> $mon_tableau[$indice] </p>" ;
    $indice-- ;
}
?>
```

2.3 Les fonctions

```
<?php
function jeu($nb_1,$nb_2)
{
    $tirage = rand($nb_1,$nb_2);
    echo("<p> $tirage </p>");
}
jeu(8,12);
jeu(25,32);
?>
```

Exercice 1 :

1. Écrire la fonction `table_alea` qui renvoie une matrice dont la taille est donnée en argument, et dont les éléments sont valeurs aléatoires.
2. Écrire la fonction `affiche` qui prend en argument une matrice et l'affiche dans la page html.
3. Modifier la fonction `affiche` pour changer aléatoirement la couleur de fond de la case.

2.4 Formulaires

Le but du formulaire est d'interagir avec l'utilisateur.

GET et **POST** sont des méthodes d'accès définies dans le protocole HTTP et reprises dans la spécification HTML.

Le choix de la méthode dépend de la façon dont les données sont reçues, de la taille et la nature des données.

2.4.1 Méthode Post

Dans une nouvelle page, nommée `test_post.php` :

```
<body>
  <form method="post" action="test_post.php">
    Nom : <input type="text" name="nom" />
    Prenom : <input type="text" name="prenom" />
    <input type="submit" name="bouton" value="Valider" />
  </form>

  <?php
  if(isset($_POST['bouton'])){
    $nom = $_POST["nom"] ;
    $prenom = $_POST["prenom"] ;
    echo "<p> Bien venu $nom $prenom </p>" ;
  }

  ?>
</body>
```

2.4.2 Méthode Get

Dans une nouvelle page, nommée `test_get.php` :

```
<body>
  <form method="get" action="test_get.php">
    Nom : <input type="text" name="nom" />
    Prenom : <input type="text" name="prenom" />
    <input type="submit" name="bouton" value="Valider" />
  </form>

  <?php
  if(isset($_GET['bouton'])){
    $nom = $_GET["nom"] ;
    $prenom = $_GET["prenom"] ;
    echo "<p> Bien venu $nom $prenom </p>" ;
  }

```

```
}  
?>  
</body>
```

Exercice 2 : Pour vérifier la bonne connexion sur un site, nous décidons de vérifier le mot de passe et un caractère précis parmi les suivants :

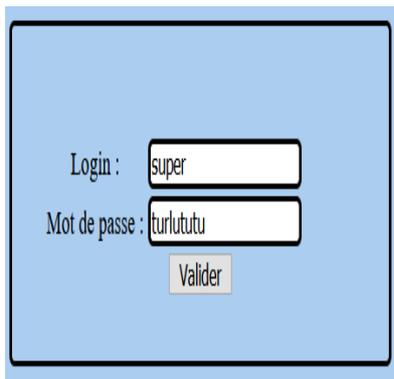
- La couleur préférée,
- L'animal préférée,
- Le département de naissance

Pour ce faire, nous allons construire trois pages :

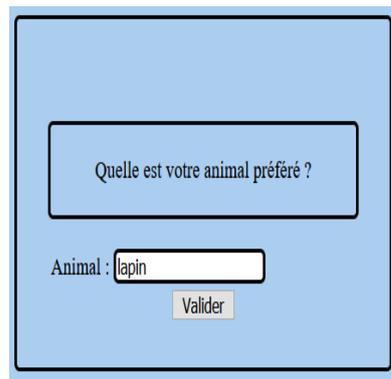
- La page d'accueil qui demande le login et le mot de passe,
- La page de vérification, qui vérifie les information précédente et propose un caractère au hasard à compléter,
- Enfin la page finale, si tous les critères sont convenables.

Les informations seront stockées dans un tableau associatif :

```
$identite = array('login' => 'super' , 'mdp' => 'turlututu' ,  
'couleur' => 'vert' , 'animal' => 'lapin' , 'departement' => '95' ) ;
```



Page d'accueil



Page de vérification



Page bienvenue

3 Création d'une base de donnée

Ouvrir l'interface phpMyAdmin : <http://127.0.0.1/phpmyadmin/>.

- Télécharger le code sql :
http://sebjmaths.free.fr/NSI_term/exercices/jeux_videos.sql
- Ouvrir l'onglet **SQL**, puis copier les requêtes du fichier.

Exercice 3 : Donner les tables suivantes :

1. Donner la liste des jeux vidéos.
2. Donner le nombre de jeux enregistrés.
3. Donner le nombre de jeux dont le genre est aventure.
4. Donner le nombre de jeux par console.
5. Donner la liste des jeux notés par 'A'.
6. Donner la moyenne du jeu 'Uncharted 4'.
7. Donner la moyenne du jeu de `code_produit = '711719454113'` , en précisant son support.
8. Donner dans l'ordre décroissant de leur note moyenne les jeux.

4 PHP et SQL

Dans le dossier `NSI`, nous allons créer un nouveau dossier `jeux_videos`, puis une nouvelle page : `index.php` :

```
<body>
<h1> Liste des jeux video </h1>
<?php
$serveur = "localhost" ;
$user = "root" ;
$mdp = '' ;
$bd = 'jeux_video' ;
$idcon = mysqli_connect($serveur , $user , $mdp , $bd );
$requete = "SELECT * FROM jeux" ;
$quete = mysqli_query($idcon , $requete) ;
echo "<table border>
<caption > Tableau des jeux </caption>
<tr>
<th> Titre </th>
<th> Genre </th>
<th> Developpeur </th>
</tr> ";
while ( $ligne = mysqli_fetch_array($quete))
{ echo "<tr>
    <td> {$ligne['titre']}
    </td>
    <td> {$ligne['genre']}
    </td>
    <td> {$ligne['developpeur']}
    </td>
</tr>";
}
echo "</table>" ;
mysqli_close($idcon);
?>
</body>
```

Exercice 4 :

- Ajouter un tableau des consoles.
- Ajouter le tableau des moyennes par jeux.

Exercice 5 : Concevoir une page d'enregistrement d'un produit (`nouvelle_entree.php`) qui demande code, le titre, le genre et le développeur pour un jeu vidéo.

Exercice 6 : Concevoir une page d'inscription (`nouvelle_inscription.php`) qui demande le nom, le prénom et un mot de passe. Si la personne est déjà inscrite, une page avec un message d'erreur sera générée.