

# **EXERCICES DE PHP**

## **MEMBRES DU GROUPE**

Fatoumata Dial

Mame Diarra Dieng

Mouhamadou Rassoul Name

Anna Ndoye

Ndeye Aïta Seck

# **Exercice 1 : Tri**

## **1- Tri par sélection**

Parcourir un tableau de  $n$  éléments et à trouver le numéro de l'élément le plus petit (exemple  $\text{tab}[\text{min}]$ ) tel que  $\text{tab}[i] \geq \text{tab}[\text{min}]$ . Une fois trouvé, on les échange  $\text{tab}[\text{min}]$  va prendre la place de  $\text{tab}[i]$  et  $\text{tab}[i]$  on va prendre la place de  $\text{tab}[\text{min}]$  avec l'aide d'une variable tampon et on fait la même chose pour tous les autres indices du tableau  $\text{tab}[i+1]$ ,  $\text{tab}[i+2]$ .

## **2- Tri par insertion**

On commence par considérer le deuxième élément du tableau en considérant que le premier est déjà trié. On compare le deuxième élément au premier élément s'il est inférieur le deuxième prend la place du premier et le premier prend la place du deuxième sinon on passe à l'élément trois du tableau. On compare le troisième élément aux deux premiers éléments s'il est inférieur au deuxième le deuxième prend sa place et vice versa sinon on passe au premier élément et on fait la même chose ensuite on passe au quatrième élément du tableau.

## **3- Tri à bulle ou par propagation**

On commence par comparer le premier élément au autres éléments du tableau. Si le premier rencontre un élément qui est inférieur à lui, il échange de place avec cet élément ainsi de suite s'il rencontre un autre qui est inférieur à lui jusqu'à finir. On passe au deuxième élément et on fait la même chose. On répète ce procédé avec tous les autres éléments du tableau jusqu'à la fin.

Autrement dit On cherche le plus petit élément du tableau, et on l'échange avec l'élément d'indice 0 on passe à l'indice 1. On recherche le second plus petit élément du tableau, et on l'échange avec l'élément d'indice 1. On continue ainsi jusqu'à ce que le tableau soit entièrement trié dans l'ordre croissant.

## **4- Tri par permutation**

L'algorithme commencera par comparer les deux premiers éléments. Il compare le premier et le deuxième élément si le premier  $>$  le deuxième, les éléments sont échangés, sinon rien ne change. On continue en comparant le premier et le troisième, si le premier  $>$  le troisième, les éléments sont échangés, sinon rien ne change. On procède ainsi jusqu'à la fin du tableau puis on passe au deuxième élément du tableau ainsi de suite jusqu'à la fin.

## **5- tri rapide ou Quicksort**

Le tri rapide fonctionne en choisissant un élément du tableau, appelé pivot, puis en permutant tous les éléments du tableau de telle sorte que tous ceux qui sont inférieurs au pivot soient à sa gauche et que tous ceux qui sont supérieurs au pivot soient à sa droite. Le pivot est ensuite placé à sa position définitive, à la fin de la sous-liste des éléments inférieurs.

On peut choisir le premier, le dernier élément, l'élément du milieu ou un élément quelconque du tableau comme pivot. On crée ensuite deux tableaux: un qui va stocker les éléments inférieurs au pivot et l'autre qui va stocker les éléments supérieurs au pivot. Ensuite pour chaque tableau trié on choisit à nouveau un pivot et on fera à nouveau la même chose

## **6-Tri par fusion**

L'algorithme fonctionne en divisant le tableau en deux sous-tableaux de taille égale, puis en triant récursivement ces deux sous-tableaux. Une fois que les deux sous-tableaux sont triés, ils sont fusionnés en un seul tableau trié.

## **7-Tri par tas**

Le tri par tas commence par initialiser le tas. Pour cela, on parcourt le tableau de gauche à droite et on insère chaque élément dans le tas.

Pour insérer un élément dans un tas, on le place à la fin du tas. Ensuite, on remonte le nœud jusqu'à ce qu'il respecte les conditions de tas.

Une fois que le tas est construit, on récupère le pivot, qui est la valeur maximale du tas. Le pivot est échangé avec le dernier élément du tableau.

Le tas est reconstruit sans le pivot. Pour cela, on parcourt le tas de droite à gauche et on descend chaque nœud jusqu'à ce qu'il respecte les conditions de tas.

L'algorithme se répète jusqu'à ce que le tableau soit entièrement trié. Pour cela, on réapplique l'algorithme aux sous-listes des éléments inférieurs et supérieurs au pivot.

## **8-Tri par introsort** (algorithme hybride qui combine les techniques du tri par fusion et du tri rapide)

L'algorithme commence par choisir un élément du tableau, appelé pivot. Le pivot peut être choisi de différentes manières, par exemple en choisissant le premier, le dernier ou un élément au hasard.

L'algorithme utilise ensuite une fonction de partitionnement pour placer tous les éléments inférieurs au pivot à sa gauche et tous les éléments supérieurs au pivot à sa droite.

L'algorithme applique ensuite le tri par introsort à la sous-liste des éléments inférieurs au pivot, puis à la sous-liste des éléments supérieurs au pivot.

Si la profondeur de récursivité dépasse une certaine limite, l'algorithme utilise un tri par tas pour trier la sous-liste.

Le choix du pivot est une étape importante du tri par introsort. Un bon choix du pivot peut améliorer considérablement l'efficacité de l'algorithme.

La fonction de partitionnement est responsable de placer tous les éléments inférieurs au pivot à sa gauche et tous les éléments supérieurs au pivot à sa droite.

## 9- **Tri par shell**

Le tri par Shell, ou Shell sort en anglais, est un algorithme de tri par comparaison. Il est basé sur l'idée de diviser le tableau en sous-tableaux de taille décroissante, puis de trier ces sous-tableaux individuellement.

L'algorithme de tri par Shell fonctionne comme suit :

### 1. **Choix de la distance**

L'algorithme commence par choisir une distance, qui est une valeur positive. La distance initiale est généralement choisie de manière à être supérieure à la moitié de la taille du tableau.

### 2. **Tri des sous-tableaux**

L'algorithme trie ensuite les sous-tableaux du tableau, en utilisant un algorithme de tri plus simple, tel que le tri par insertion. Les sous-tableaux sont définis comme suit :

- Le premier sous-tableau contient tous les éléments du tableau dont l'indice est un multiple de la distance.
- Le deuxième sous-tableau contient tous les éléments du tableau dont l'indice est un multiple de la distance/2, mais pas un multiple de la distance.
- Et ainsi de suite.

### 3. **Diminution de la distance**

L'algorithme répète ensuite les étapes 1 et 2, en diminuant la distance à chaque fois. La distance est généralement divisée par 2 à chaque itération.

### 4. **Fin de l'algorithme**

L'algorithme s'arrête lorsque la distance est égale à 1. À ce stade, le tableau est entièrement trié.

## **10-Tri par gnome**

Le tri par gnome fonctionne en choisissant deux éléments du tableau, appelés les gnomes, puis en les échangeant de place si le gnome de gauche est plus grand que le gnome de droite. Le processus se poursuit jusqu'à ce qu'un gnome soit plus grand que tous les éléments qui le suivent.

L'algorithme est récursif, ce qui signifie qu'il s'applique à nouveau aux sous-listes des éléments à gauche et à droite du gnome. Le processus se poursuit jusqu'à ce que le tableau soit entièrement trié.

# **Implémentation de 5 tris**

## **1- tri à bulle**

### **Code du tri**

```

<?
echo "Le tri à bulle";
echo "<br/>";
$tableau = [5, 1, 4, 2, 3, 9, 0, 8, 7, 6];
$n = count($tableau);

echo "Avant le tri, on a: ";
foreach ($tableau as $valeur)
{
    echo "$valeur    ";
}

for($i = 0; $i < $n - 1; $i++)
{
    for($j = 0; $j < $n - $i - 1; $j++)
    {
        if($tableau[$j] > $tableau[$j+1])
        {
            $tempon = $tableau[$j];
            $tableau[$j] = $tableau[$j+1];
            $tableau[$j+1] = $tempon;
        }
    }
}

echo "<br/>";
echo "Après tri, on a: ";
foreach ($tableau as $valeur)
{
    echo "$valeur    ";
}
?>

```

### Explication du code

Deux boucles for imbriquées sont utilisées pour parcourir le tableau.

La première boucle for parcourt les éléments du tableau.

La deuxième boucle for parcourt les éléments restants à comparer avec l'élément actuel.

Pour chaque paire d'éléments consécutifs, si l'élément actuel est supérieur à l'élément suivant, les deux éléments sont échangés de place.

Ce processus se répète jusqu'à ce que tous les éléments soient triés.

La première boucle ne parcourt que jusqu'à l'avant-dernier élément car le dernier élément sera déjà trié après les itérations précédentes. De même, la deuxième boucle ne parcourt que les éléments non triés restants.

## **Résultats**

---

Le tri à bulle

Avant le tri, on a: 5 1 4 2 3 9 0 8 7 6

Après tri, on a: 0 1 2 3 4 5 6 7 8 9

## **2- tri par fusion**

### **Code du tri**

```
<?php
echo "Le tri par fusion";
echo "<br/>";
$tableau1 = [0, 1, 4, 6, 9];
$tableau2 = [2, 3, 5, 7, 8];
$tableau3 = array();
$i1 = $i2 = $i = 0;

echo "Avant fusion, on a: ";
echo "<br/>";

echo "le premier tableau : ";
foreach ($tableau1 as $valeur)
{
    echo "$valeur    ";
}

echo "<br/>";

echo "le deuxième tableau : ";
foreach ($tableau2 as $valeur)
{
    echo "$valeur    ";
}
echo "<br/>";

while ($i1 < count($tableau1) && $i2 < count($tableau2))
{
    if ($tableau1[$i1] < $tableau2[$i2])
    {
        $tableau3[$i] = $tableau1[$i1];
        $i1++;
    }
    else
    {
        $tableau3[$i] = $tableau2[$i2];
```



```

7 fusion.php > ...
    while ($i1 < count($tableau1))
    {
        $tableau3[$i] = $tableau1[$i1];
        $i1++;
        $i++;
    }

    while ($i2 < count($tableau2))
    {
        $tableau3[$i] = $tableau2[$i2];
        $i2++;
        $i++;
    }

    echo "<br/>";
    echo "Après fusion, on a: ";
    foreach ($tableau3 as $valeur)
    {
        echo "$valeur    ";
    }

?>

```

### **Explication du code**

On définit deux tableaux avec des valeurs initiales qu'on souhaite fusionner par la suite.

\$tableau3 = array(); : Cela initialise un tableau vide \$tableau3.

\$i1 = \$i2 = \$i = 0; : Cela initialise trois variables \$i1, \$i2, et \$i à 0 qui vont servir à parcourir les tableau1, 2 et 3.

foreach (\$tableau1 as \$valeur) : Cela commence une boucle qui itère à travers chaque élément du tableau \$tableau1 en le stockant dans la variable \$valeur.

while (\$i1 < count(\$tableau1) && \$i2 < count(\$tableau2)) : Cela commence une boucle while qui continue tant que les indices \$i1 et \$i2 sont inférieurs à la longueur de \$tableau1 et \$tableau2 respectivement.

Les deux boucles while suivantes fusionnent les éléments restants de \$tableau1 et \$tableau2 dans \$tableau3.

### **Résultats**

Le tri par fusion

Avant fusion, on a:

le premier tableau : 0 1 4 6 9

le deuxième tableau : 2 3 5 7 8

Après fusion, on a: 0 1 2 3 4 5 6 7 8 9

### **3- tri par insertion**

```

1  <?php
2
3      echo "Le tri par insertion";
4      echo "<br/>";
5
6      $tableau = [5, 1, 4, 2, 3, 9, 0, 8, 7, 6];
7
8      echo "Avant tri, on a: ";
9      foreach ($tableau as $valeur)
10     {
11         echo "$valeur ";
12     }
13
14     $n = count($tableau);
15
16     for ($i = 1; $i < $n; $i++)
17     {
18         $valeur_actuelle = $tableau[$i];
19         $j = $i - 1;
20
21         while ($j >= 0 && $tableau[$j] > $valeur_actuelle)
22         {
23             $tableau[$j + 1] = $tableau[$j];
24             $j--;
25         }
26
27         $tableau[$j + 1] = $valeur_actuelle;
28     }
29
30     echo "<br/>";
31     echo "Après tri, on a: ";
32     foreach ($tableau as $valeur)
33     {
34         echo "$valeur ";
35     }
36     ?>

```

### Explication du code

Un tableau de nombres non triés est défini.

Le code affiche le contenu du tableau avant le tri.

Une boucle for parcourt les éléments du tableau à partir du deuxième élément jusqu'à la fin.

Pour chaque élément, sa valeur est stockée dans une variable temporaire.

Une boucle while est utilisée pour déplacer les éléments du tableau vers la droite tant que leur valeur est supérieure à celle de l'élément actuel, en commençant par l'élément précédent.

Lorsque la bonne position est trouvée (c'est-à-dire que la valeur de l'élément précédent est inférieure ou égale à la valeur de l'élément actuel), l'élément actuel est inséré à cet endroit en décalant les éléments du tableau vers la droite.

Ce processus se répète jusqu'à ce que tous les éléments du tableau soient triés.

Une fois le tri terminé, le code affiche le contenu du tableau après le tri.

### Résultats

Le tri par insertion

Avant tri, on a: 5 1 4 2 3 9 0 8 7 6

Après tri, on a: 0 1 2 3 4 5 6 7 8 9

## **4- tri par permutation**

### **Code du tri**

```

1  <?php
2      echo "Le tri par permutation";
3      echo "<br/>";
4      $tableau = [5, 1, 4, 3, 2];
5      $n = count($tableau);
6
7      echo "Avant le tri, on a: ";
8      foreach ($tableau as $valeur)
9      {
10         echo "$valeur ";
11     }
12
13     for ($i = 0; $i < $n - 1; $i++)
14     {
15         $minIndex = $i;
16         for ($j = $i + 1; $j < $n; $j++)
17         {
18             if ($tableau[$j] < $tableau[$minIndex])
19             {
20                 $minIndex = $j;
21             }
22         }
23
24         $temp = $tableau[$i];
25         $tableau[$i] = $tableau[$minIndex];
26         $tableau[$minIndex] = $temp;
27     }
28
29     echo "<br/>";
30     echo "Après le tri, on a: ";
31     foreach ($tableau as $valeur)
32     {
33         echo "$valeur ";
34     }
35     ?>

```

### Explication du code

Nous parcourons le tableau avec la variable \$i. À chaque itération, nous sélectionnons l'élément minimum restant dans la partie non triée du tableau. Pour cela, nous avons une boucle interne (\$j) qui parcourt les éléments restants après \$i. Nous mettons à jour l'index de l'élément minimum (\$minIndex) lorsque nous trouvons un élément plus petit. À la fin de la boucle interne, nous avons trouvé l'élément minimum pour la partie non triée du tableau. Nous échangeons ensuite cet élément avec l'élément à la position \$i, qui devient ainsi trié.

Ce processus se répète jusqu'à ce que tous les éléments soient triés.

## **Résultats**

---

Le tri par permutation

Avant le tri, on a: 5 1 4 3 2

Après le tri, on a: 1 2 3 4 5

## **5- tri par sélection**

```

1  <?
2  $tableau = [5, 1, 4, 2, 3, 9, 0, 8, 7, 6];
3  echo "Avant le tri par sélection :";
4  foreach ($tableau as $valeur)
5  {
6      echo "$valeur    ";
7  }
8
9  echo "<br>";
10 $n = count($tableau);
11
12 for ($i = 0; $i < $n-1; $i++)
13 {
14     $indexMin = $i;
15
16     for ($j = $i + 1; $j < $n; $j++)
17     {
18         if ($tableau[$j] < $tableau[$indexMin])
19         {
20             $indexMin = $j;
21         }
22     }
23
24     $tempon = $tableau[$i];
25     $tableau[$i] = $tableau[$indexMin];
26     $tableau[$indexMin] = $tempon;
27 }
28
29 echo "<br/>";
30 echo "Après le tri on a: ";
31 foreach ($tableau as $valeur)
32 {
33     echo "$valeur    ";
34 }
35 ?>

```

### Explication du code

Un tableau de nombres est défini avec des valeurs à trier.

La fonction count() est utilisée pour obtenir la taille du tableau et stocker cette valeur dans la variable \$n.

Une boucle for parcourt le tableau du début jusqu'à l'avant-dernier élément (\$i < \$n-1).

À chaque itération de cette boucle, une variable \$indexMin est initialisée à l'index actuel (\$i).

Une autre boucle for parcourt les éléments du tableau à partir de l'élément suivant ( $j = i + 1$ ) jusqu'au dernier élément ( $j < n$ ).

À chaque itération de cette boucle interne, si l'élément actuel ( $\text{tableau}[j]$ ) est plus petit que l'élément situé à l'index minimum actuel ( $\text{tableau}[\text{indexMin}]$ ), alors  $\text{indexMin}$  est mis à jour avec  $j$ .

À la fin de la boucle interne, l'élément minimum trouvé est échangé avec l'élément à l'index  $i$ , assurant que l'élément le plus petit est placé à la bonne position dans le tableau. Enfin on affiche le tableau trié

### **Résultats**

Avant le tri par sélection : 5 1 4 2 3 9 0 8 7 6

Après le tri on a : 0 1 2 3 4 5 6 7 8 9



# **Exercice 2 : Régénération de formulaire**

Nous avons créé trois fichiers

--le premier fichier qui s'intitule Fonction.php

```

1  <?php
2      function formulaire()
3      {
4      ?>
5          <center> <header>Formulaire</header> </center>
6
7          <section>
8
9              <form action="formulaire3.php" method="post">
10                 <div class="label">Age </div>
11                 <div class="input">
12                     <input type="number" name="Age" id="age" ><br>
13                 </div>
14                 <?php
15                     if(isset($_POST['Age'])&& empty($_POST['Age'])) {
16                     {
17                         echo"<span class='erreur'>L'Age n'est pas saisi</span>";
18                     }
19                 ?>
20                 <div class="label">Prenom</div>
21                 <div class="input">
22                     <input type="text" name="Prenom" ><br>
23                 </div>
24                 <?php
25                     if(isset($_POST['Prenom'])&& empty($_POST['Prenom'])) {
26                     {
27                         echo"<span class='erreur'>Le prénom n'est pas saisi</span>";
28                     }
29                 ?>
30                 <div class="label"></div>
31                 <div class="input">
32
33                     <center> <br>
34                     <input type="submit" value="Valider">
35                     </center>
36                 </div>
37
38             </div>
39
40         </form>
41
42     </section>
43
44 <?php }
45
46 function traitement($n, $v)
47 {
48     if(empty($n) || empty($v))
49     {
50         formulaire();
51     }
52     else if( $v < 0)
53     {
54         echo "<div>L'âge doit être supérieur à 0</div>";
55         formulaire();
56     }
57     else
58     {
59         $message="Bonjour  votre nom est ".$n." et votre age est ".$v." .";
60         echo $message;
61         //header('Location:message.php');
62         //exit();
63     }
64 }
65
66 ?>

```

--le deuxième fichier formulaire3.php

```
1  <?php
2  include("Fonctions.php");
3  ?>
4  <!DOCTYPE html>
5  <html lang="en">
6  <head>
7      <meta charset="UTF-8">
8      <meta name="viewport" content="width=device-width, initial-scale=1.0">
9      <title>Document</title>
10     <link rel="stylesheet" href="formulaire.css">
11 </head>
12 <body>
13
14 <?php
15 if(!isset($_POST['Prenom'])){ formulaire();}
16
17
18 else
19 {
20     traitement($_POST['Prenom'],$_POST['Age']);
21 }
22
23 ?>
24
25 </body>
26 </html>
```

--le troisième fichier formulaire.css

```
1  form
2  {
3      margin: 0 auto;
4      width: 600px;
5      height: 300px;
6      padding:12px;
7      border:5px;
8      border-color: □black;
9      border-radius:5px;
10     border-style:solid ;
11 }
12
13 input[type="submit"]
14 {
15     margin :0 auto;
16     margin-bottom: 2px;
17     width: 150px;
18     background-color:□gray;
19     padding: 5px;
20 }
21
22 header
23 {
24     color: ■aliceblue;
25     margin: 0 auto;
26     background-color: □black;
27     width: 250px;
28     margin-bottom: 12px;
29     font-size: 50px;
30 }
31
```

# Exercice 3 recherche de motif

Nous avons deux fichiers.

Le premier qui s'appelle **EnvoieFichier.html**

```
1 <html>
2 <head>
3
4   <title>Transférer</title>
5 </head>
6
7 <body BGCOLOR="#e3e3e3">
8   <form name="transférer" enctype="multipart/form-data" action="stocke.php" method="post">
9     <input name="monfichier" type="file">
10    <br>
11    Motif à surligner en bleu: <input type="text" name="motif">
12    <br>
13    <input type="submit" value="envoyer">
14  </form>
15 </body>
16 </html>
17
```

Le deuxième qui s'appelle **stocke.php**

```

1 <?php
2 $fichierNom = isset($_FILES["monfichier"]["name"]) ? $_FILES["monfichier"]["name"] : 'non renseigné';
3 $fichierTmp = isset($_FILES["monfichier"]["tmp_name"]) ? $_FILES["monfichier"]["tmp_name"] : 'non renseigné';
4 $fichierTaille = isset($_FILES["monfichier"]["size"]) ? $_FILES["monfichier"]["size"] . " octets" : 'non renseigné';
5
6 echo "Fichier original: $fichierNom<br>";
7 echo "Fichier temporaire: $fichierTmp<br>";
8 echo "Taille du fichier: $fichierTaille<br>";
9
10 if (!isset($_FILES["monfichier"]) || empty($_FILES["monfichier"]["name"]) || empty($_POST['motif']))
11 {
12     echo "<br><br>";
13     echo "Fichier non renseigné ou motif non renseigné";
14     die();
15 }
16
17 $fichier = $_FILES["monfichier"]["tmp_name"];
18
19 if (!$f = fopen("$fichier", "r"))
20 {
21     die("Problème d'ouverture du fichier!!!");
22 }
23
24 echo "<br><br>";
25 echo "Voici le contenu du texte avec motif surligné en bleu:";
26 echo "<br>";
27
28 $motif = $_POST['motif'];
29
30 while (!feof($f))
31 {
32     $ligne = fgets($f);
33     $ligne = str_replace($motif, "<span style='color: blue; background-color: blue;'>$motif</span>", $ligne);
34     echo $ligne . "<br>";
35 }
36
37 echo "<br>";
38 fclose($f);
39 ?>
40

```

# Exercice 4 : Session

```
1  <?php
2
3  define("MAX", 5);
4
5  session_start();
6
7  if (!isset($_SESSION["c"])) {
8      $_SESSION["c"] = 1;
9      echo "<h3> Bienvenue ! C'est votre " . $_SESSION['c'] . " ère connexion. <h3>";
10     $_SESSION["t"] = time();
11 } else {
12     if (time() - $_SESSION["t"] > MAX) {
13         session_destroy();
14         die("<h3>Trop tard ! Vous avez dépassé le temps de session.<h3>");
15     }
16     $_SESSION["c"]++;
17     $_SESSION["t"] = time();
18     echo "<h3> Bienvenue ! C'est votre " . $_SESSION['c'] . " ème connexion. <h3>";
19 }
20 }
21
22
23 ?>
24 <form action="Session.php" method="post">
25     <input type="submit" value="continuer"> avant 5 secondes
26 </form>
27
```

# Exercice 5 : base de données

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Liens</title>
5      <style>
6          th, td
7          {
8              text-align: center;
9          }
10     </style>
11 </head>
12 <body>
13     <center><table border="1" cellpadding="15" cellspacing="0">
14         <tr>
15             <th>Code</th>
16             <th>Nom du site</th>
17             <th>URL</th>
18             <th>Description</th>
19         </tr>
20         <?php
21             $host = "localhost";
22             $user = "root";
23             $password = "";
24             $database = "mysites";
25
26             $connection = mysqli_connect($host, $user, $password, $database);
27
28             if ($connection->connect_error)
29             {
30                 die("Erreur de connexion à la base de données : " . $connection->connect_error);
31             }
32
33             $query = "SELECT * FROM sites";
34             $result = mysqli_query($connection, $query);
35
36             if (!$result)
37             {
38                 die("Erreur lors de la requête : " . mysqli_error($connection));
39             }
40         </?php>
41     </table></center>
42 </body>
43 </html>
```

```
32
33         $query = "SELECT * FROM sites";
34         $result = mysqli_query($connection, $query);
35
36         if (!$result)
37         {
38             die("Erreur lors de la requête : " . mysqli_error($connection));
39         }
40         else
41         {
42             while ($row = mysqli_fetch_assoc($result))
43             {
44                 $Code = $row["Code"];
45                 $Nom = $row["Nom"];
46                 $URL = $row["URL"];
47                 $Description = $row["Description"];
48                 echo "<tr>\n";
49                 echo "<td>$Code</td>\n";
50                 echo "<td>$Nom</td>\n";
51                 echo "<td><a href=\"http://$URL\">$URL</a></td>\n";
52                 echo "<td>$Description</td>\n";
53                 echo "</tr>\n";
54             }
55         }
56
57         mysqli_close($connection);
58     <?php>
59 </table></center>
60 </body>
61 </html>
62
```



## base de données

```
MariaDB [mysites]> select * from sites;
```

Code	Nom	URL	Description
1	UCAD	<a href="http://www.ucad.sn">www.ucad.sn</a>	Ceci est le site de l'UCAD.
2	ESP	<a href="http://www.esp.sn">www.esp.sn</a>	Ceci est le site de l'ESP.
3	GOOGLE	<a href="http://www.google.com">www.google.com</a>	Ceci est le site de GOOGLE.
4	YOUTUBE	<a href="http://www.youtube.com">www.youtube.com</a>	Ceci est le site de YOUTUBE.

```
4 rows in set (0.000 sec)

MariaDB [mysites]>
```

## Résultats

Code	Nom du site	URL	Description
1	UCAD	<a href="http://www.ucad.sn">www.ucad.sn</a>	Ceci est le site de l'UCAD.
2	ESP	<a href="http://www.esp.sn">www.esp.sn</a>	Ceci est le site de l'ESP.
3	GOOGLE	<a href="http://www.google.com">www.google.com</a>	Ceci est le site de GOOGLE.
4	YOUTUBE	<a href="http://www.youtube.com">www.youtube.com</a>	Ceci est le site de YOUTUBE.