



Ecole Nationale des Sciences Appliquées Al-Hoceima

1ere année cycle ingénieur

La transformation digitale et intelligence artificielle

Le rapport du Tp 3



Réalisé par :

- Maach Zineb

Annee universitaire:

2023-2024

-l'exécution d'un fichier PHP:

- Afin d'exécuter un code PHP , on doit suivre les etapes suivantes:

- **Activer Apache dans XAMPP :**

Assurez-vous que le module Apache est activé en XAMPP, en cliquant sur le bouton "Start à côté de "Apache".

- **Créer un fichier PHP dans le dossier htdocs :**

-Accédez au dossier où XAMPP est installé. Par défaut, il es installé dans"C:\xampp" sous Windows.

-Dans le dossier XAMPP, ouvrez le dossier "htdocs". C'est l'endroit où vous devez placer vos fichiers PHP pour qu'ils soient accessibles via votre serveur web local. Renommez le fichier avec l'extension ".php", par exemple "monfichier.php".

- **Accéder au fichier via le navigateur :**

-Ouvrez votre navigateur web

-Tapez "localhost" dans la barre d'adresse du navigateur et appuyez sur Entrée. Cela vous dirigera vers la page d'accueil de XAMPP.

-Pour accéder à votre fichier PHP, ajoutez le nom de votre fichier PHP à l'URL. Par exemple, si vous avez nommé votre fichier "monfichier.php", tapez "localhost/monfichier.php" dans la barre d'adresse du navigateur et appuyez sur Entrée.

-Vous devriez voir le résultat de l'exécution de votre code PHP dans le navigateur.

-En suivant ces étapes, vous pourrez exécuter votre code PHP avec XAMPP sur votre serveur web local.

Exercices 01 :

-**Le but:** Ecriture d'un programme PHP qui affiche tous les nombres d'Armstrong inférieurs à 1000 sous forme d'une liste à puce.

-Expliquons le principe de cet exercice:

DÉBUT

Étape 1 → Prendre la variable \$i

Étape 2 → Séparer tous les chiffres de \$i.

Étape 3 → Trouver la valeur au cube de chaque chiffre.

Étape 4 → Ajouter toutes les valeurs au cube ensemble.

Étape 5 → Enregistrer la sortie dans la variable \$s.

Étape 6 → Si \$s est égale à \$i, afficher Nombre d'Armstrong.

Étape 8 → Si \$s n'est pas égale à \$i, l'afficher comme étant un Nombre d'Armstrong.

FIN

-En appliquant cet algorithme, et on obtient le script suivant:

```
Ex1.php
1  <?php
2  echo "<ul>";
3  for($i=0; $i<=1000; $i++){
4      $s=0;
5      $temp=$i;
6      while($temp >0){
7          $r= $temp % 10;
8          $s += pow($r,3);
9          $temp /= 10;
10     }
11     if($s == $i)
12         echo "<li> $i </li>";
13     }
14     echo "</ul>";
15     ?>
```

-Après l'avoir ouvrir, on obtient:

← ↻ ⓘ localhost/Ex1.php

- 0
- 1
- 153
- 370
- 371
- 407

Exercice 02 :

-**Le but:** Ecrire un programme PHP qui effectue une suite de tirages de nombres aléatoires jusqu'à obtenir une suite composée de deux nombres pair suivis d'un nombre impair.

- Expliquons le principe de cet exercice:

DÉBUT

Étape 1 → Définir une fonction pour vérifier si un nombre est pair .

Étape 2 → Générer un nombre aléatoire entre 100 et 1000.

Étape 3 → Extraire les chiffres du nombre généré.

Étape 4 → Si le premier chiffre est pair, le deuxième chiffre est pair et le troisième chiffre est impair : Afficher le nombre généré.

Étape 5 → Sinon, revenir à l'étape 2.

FIN

-En appliquant cet algorithme, et on obtient le script suivant:

```

Ex2.php
1  <?php
2  function is_pair(int $n){
3      if($n % 2==0){
4          return TRUE;
5      }
6      else
7          return FALSE;
8  }
9  menu:
10 $x= rand(100,1000);
11 $ch= (string)$x;
12 $ch0=(int)$ch[0];
13 $ch1=(int)$ch[1];
14 $ch2=(int)$ch[2];
15 if( is_pair($ch0) and is_pair($ch1) and !is_pair($ch2) ){
16     echo $x;
17 }
18 else
19     goto menu;
20 ?>

```

-Le resultat: A chaque fois , il y'a une generation d'un nombre qui verifie les conditions .

← ↻ ⓘ localhost/Ex2.php

467

← ↻ ⓘ localhost/Ex2.php

263

← ↻ ⓘ localhost/Ex2.php

481

Exercice 04 :

-Le but: Ecriture d'un programme PHP qui permet de générer aléatoirement un nombre de trois chiffres N. Ensuite il effectue des tirages aléatoires jusqu'à réobtenir le même nombre N puis il affiche le nombre d'essais réalisés.

-Expliquons le principe de cet exercice:

DÉBUT

Étape 1 → Générer un nombre aléatoire N entre 100 et 1000.

Étape 2 → Afficher le nombre généré.

Étape 3 → Initialiser un compteur de tirages à 0.

Étape 4 → Tant que le nombre aléatoire M n'est pas égal à N, on génère un nouveau nombre aléatoire M entre 100 et 1000 et en incrémentant le compteur de tirages, jusqu'à on obtient un nouveau N.

Étape 5 → Afficher le nombre de tirages effectués pour obtenir un nouveau N.

FIN

-En appliquant cet algorithme, on obtient:

a)- Avec l'instruction **'while'**:

```
Ex4.php
1  <?php
2  $N= rand(100,1000);
3  echo " le nombre c'est :", $N,"<br>";
4  $M=0;
5  $temp=1;
6  while($M != $N){
7      $M= rand(100,1000);
8      if($N == $M)
9          break;
10     else
11         $temp++;
12 }
13 echo "On a effectué ", $temp,"de tirages pour obtenir une autre fois le nbr:", $M;
14
15 ?>
```

-Le resultat: A chaque fois Un nombre est genere, il s'effectue plusieurs de tirages afin qu'on obtient a nouveau le meme nombre.

← ↻ ⓘ localhost/Ex4.php

le nombre c'est :281

On a effectué 1639de tirages pour obtenir une autre fois le nbr:281

b)- Avec l'instruction 'for':

```
Ex4.php
1  <?php
2  $N= rand(100,1000);
3  echo " le nombre c'est :", $N,"<br>";
4  $M=0;
5  for($i=1; $N !=$M ; $i++){
6      $M= rand(100,1000);
7      if($N == $M)
8          break;
9  }
10 echo "On a effectuee ", $i,"de tirages pour obtenir une autre fois le nbr:", $M;
11 ?>
```

← ↻ ⓘ localhost/Ex4.php

le nombre c'est :529

On a effectuee 798de tirages pour obtenir une autre fois le nbr:529

Exercice 6 :

-Le but: Ecrire une application web qui permet de résoudre les équations du deuxième degré en PHP. L'utilisateur aura la possibilité de saisir les données nécessaires via un formulaire et l'application affiche la solution après la soumission de celui-ci.

-Expliquons le principe de cet exercice:

DÉBUT

Étape 1 → Si le formulaire est soumis, on récupère les coefficients a, b et c.

Étape 2 → Calculer le discriminant D.

Étape 6 → Afficher les solutions de l'équation.

Étape 7 → Sinon, afficher le formulaire pour saisir les coefficients a, b et c.

FIN

-En appliquant cet algorithme, on obtient:

```

Ex5.php
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Equation de 2eme degre</title>
5 </head>
6 <body>
7 <h3> Equation de 2eme degre</h3>
8 <?php
9 if ($_SERVER["REQUEST_METHOD"] == "POST") {
10     $a = $_POST['a'];
11     $b = $_POST['b'];
12     $c = $_POST['c'];
13     $D = pow($b,2) - 4*$a*$c;
14     if ($D > 0) {
15         $x1 = (-$b + sqrt($D)) / (2*$a);
16         $x2 = (-$b - sqrt($D)) / (2*$a);
17         echo "Les solutions de l'equation", $a, " x^2 +", ($b), "x +", ($c), " sont les s
18     } elseif ($D == 0) {
19         $x = -$b / (2*$a);
20         echo "L'equation:", $a, " x^2 +", ($b), "x +", ($c), " a une seule solution doubl
21     } else {
22         echo "L'equation:", $a, " x^2 +", ($b), "x +", ($c), " n'a pas de solution réelle
23     }

```

```

23 } else {
24     ?>
25     <form method="post">
26         <label for="a">Coeff a :</label>
27         <input type="number" name="a" required><br><br>
28
29         <label for="b">Coeff b :</label>
30         <input type="number" name="b" required><br><br>
31
32         <label for="c">Coeff c :</label>
33         <input type="number" name="c" required><br><br>
34
35         <input type="submit" value="Solution">
36     </form>
37     <?php
38     }
39     ?>
40 </body>
41 </html>

```

-Le resultat: On obtient au 1er lieu, une formulaire a remplir par les coefficients de l'equation , puis s'effectue sa resolution.



localhost/Ex5.php

Equation de 2eme degre

Coeff a :

Coeff b :

Coeff c :

Solution



localhost/Ex5.php

Equation de 2eme degre

Coeff a :

Coeff b :

Coeff c :

Solution



localhost/Ex5.php

Equation de 2eme degre

Les solutions de l'equation $x^2 + 0x - 16$ sont les suivantes: $x_1 = 4$ et $x_2 = -4$