



Programmation Web en PHP

1. Qu'est-ce que PHP ?

PHP est un langage de script **Open Source**, interprété et exécuté côté serveur Web, spécialement conçu pour développer des sites Web dynamiques.

- ✓ Sa syntaxe est proche de celle du langage C;
- ✓ Ses instructions doivent être mises, obligatoirement, à l'intérieur d'un fichier texte portant l'extension **".php"** sous forme d'une pseudo-balise:

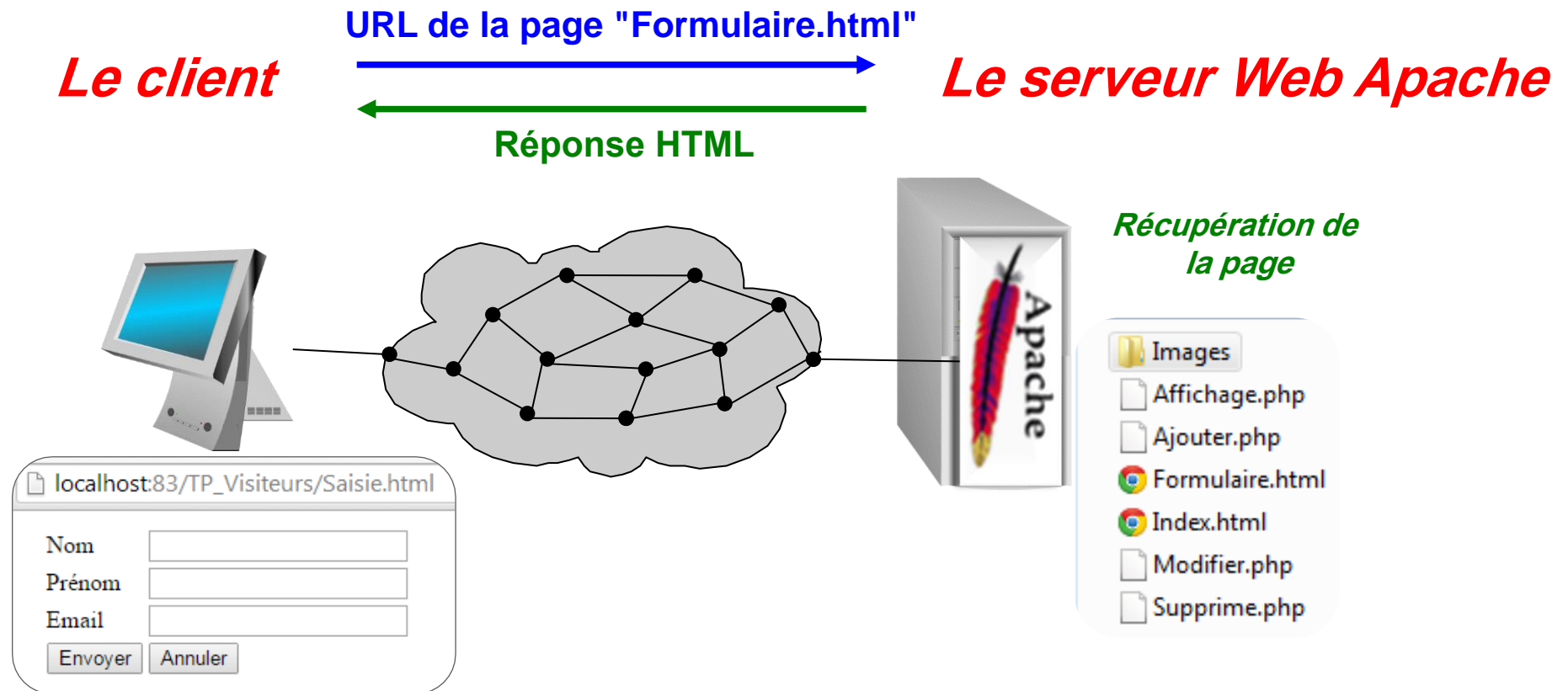
```
<?PHP //code php    ?>  
<?php    //code php    ?>
```

```
<html >  
  <head > <title>Test de page PHP</title>  </head >  
  <body >  
    <h2>Page de test</h2>  
    <?php  
      echo "Bonjour tout le monde <br>";  
    ?>  
  </body >  
</html>
```



Programmation Web en PHP

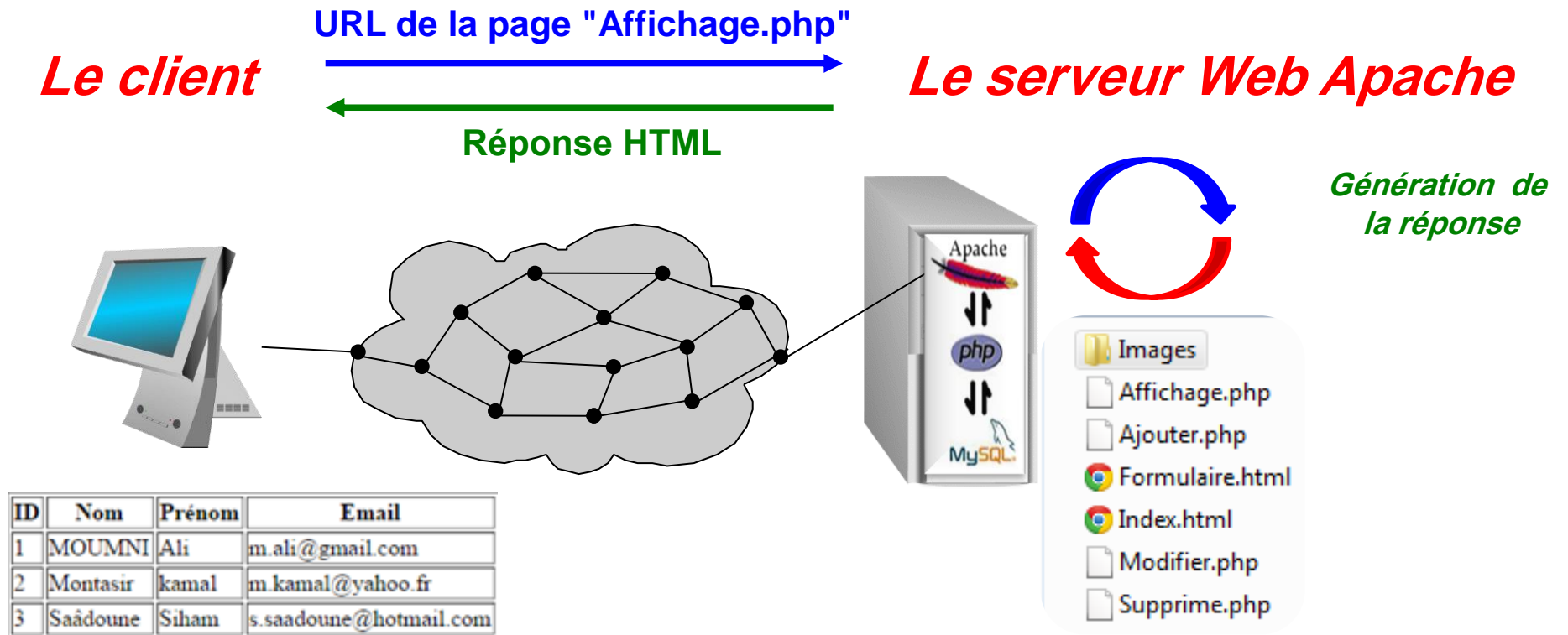
2. Cycle de vie d'une page HTML





Programmation Web en PHP

3. Cycle de vie d'une page PHP



Attention:

- ✓ le client ne sait pas lire une page PHP ! C'est le serveur qui génère une réponse HTML et l'envoie au client. Il n'a aucun **moyen d'accéder au code qui a produit la réponse**.
- ✓ La page qui porte l'extension PHP peut contenir : **i)** que du code PHP; **ii)** que du code HTML; **iii)** les deux à la fois.



Programmation Web en PHP

3. Cycle de vie d'une page PHP: **Exemple 1** (pas d'accès à la BD)

```
<html >
  <head >
<title>Test de page PHP</title>
  </head >
  <body >
    <h2>Page de test</h2>
    <?php
      // Ceci est du code PHP
      $date = date("d/m/Y");    # Variable PHP + fonction
      $heure = date("H:i:s");
      /* Affichage */
      echo "Nous sommes le <strong>$date</strong> et il est
        <strong>$heure</strong>";
    ?>
    <hr />
  </body >
</html>
```

Commentaires



http://localhost/script.php

Page de test

Nous sommes le **01/11/2024** et il est **04:52:59**

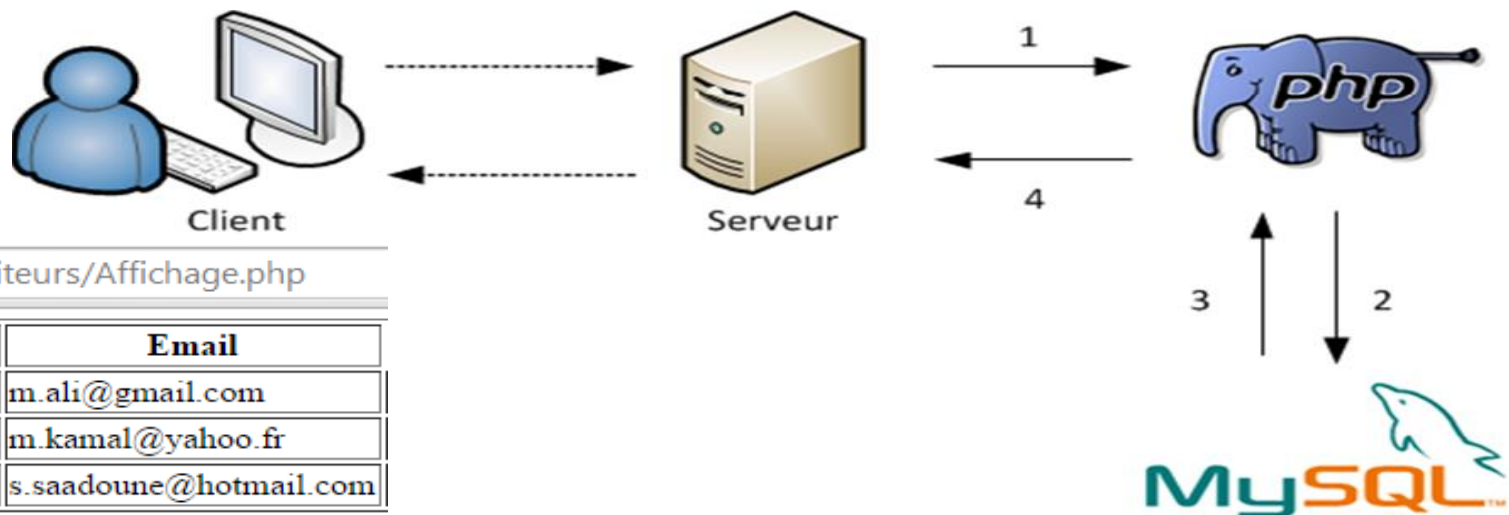


Programmation Web en PHP

3. Cycle de vie d'une page PHP: Exemple 2 (Accès à une BD MySQL)

```
<?php
$dbh = new PDO('mysql:host=localhost;dbname=tpvisiteurs','root','');
$req="select * from visiteurs;";
$res= $dbh->query($req); ?>

<center> <table border="1">
<tr> <th> ID </th> <th> Nom</th><th> Prénom </th> <th> Email </th>
<?php while($V = $res->fetch()) {
?>
<tr> <td> <?php echo $V['id']; ?> </td>
<td> <?php echo $V['nom']; ?> </td>
<td> <?php echo $V['prenom']; ?> </td>
<td> <?php echo $V['email']; ?> </td>
<?php } $dbh = NULL;
?>
</table> </center>
```





Séparation des instructions

- Comme en C, PHP requiert que les instructions soient terminées par un point-virgule
- La balise fermante d'un bloc de code PHP implique automatiquement un point-virgule. Il est recommandé de systématiquement terminer vos instructions par un point-virgule.

```
<? php  
echo 'ceci est un test';  
?>  
  
<? php echo 'ceci est un test' ?>
```



Commentaires

PHP supporte les commentaires 'C', 'C++' et Unix shell

```
<? php
    echo 'ceci est un test'; //Ceci est un commentaire sur une ligne, style C++
/* Ceci est un commentaire sur
   plusieurs lignes */
echo "ceci est un autre test";

echo 'Et un test final'; # Ceci est un commentaire style shell

?>
```



Les variables

- Elles sont représentées par le signe dollar "\$" suivi du nom de la variable;
- On peut créer des variables n'importe où;
- La déclaration de leur type n'est pas obligatoire, c'est leur valeur qui détermine leur type;
- Leur initialisation est aussi non obligatoire;
- Le nom de la variable est sensible à la casse. **\$x != \$X**

Assignment simple

```
$a=10;  
$b=$a;
```

Assignment par référence

```
$a=10;  
$b=&$a;
```

Une « **référence** » en PHP ou plus précisément un alias est un moyen d'accéder au contenu d'une même variable en utilisant un autre nom.



Structures conditionnelles (1)

```
if (conditions) instruction;
```

```
If (conditions)
{
    // bloc
}
```

```
if (conditions)
{
    // bloc1
}
else
{
    // bloc2
}
```



Structures conditionnelles (2)

```
<?php
    $prix = 35;
    if ($prix > 100)
    {
        echo " pour un achat de $prix DHs, la remise est de <b>15%</b><br/>";
        $pnet = $prix * 0.85;
        echo " le prix net est de $pnet";
    }
    elseif ($prix > 50)
    {
        echo " pour un achat de $prix DHs, la remise est de <b>10%</b><br/>";
        $pnet = $prix * 0.90;
        echo " le prix net est de $pnet";
    }
    else
    {
        echo " pour un achat de $prix DHs, la remise est de <b>05%</b><br/>";
        $pnet = $prix * 0.95;
        echo " le prix net est de $pnet";
    }
?>
```

← → ↻ 📱 localhost:83/structures_cond/TP_if_else.php

pour un achat de 35 DHs, la remise est de **05%**
le prix net est de 33.25



Structures conditionnelles (3)

Opérateur "?"

```
<?php
    $ch=" bonjour ";
    $sexe="M";
    $ch .= ($sexe=="F") ? "Madame" : "Monsieur";
    echo "<h2>$ch</h2> <br/>";
?>
```

switch

```
<?php
    $dept=2;
    switch ($dept)
    {
    case 3: echo "<h2>Mathématiques</h2> <br/>"; break;
    case 2: echo "<h2>Informatique</h2> <br/>"; break;
    default : echo "departement inconnu";
    }
?>
```



Les boucles

La boucle "for"

```
for ($i=0 ; $i < $n ; $i++) {  
    // instructions  
}
```

```
<?php  
for ($i=1;$i<7;$i++)  
{  
    ...  
    echo "<h$i>titre de niveau $i</h$i> ";  
}
```

La boucle " while"

```
while (condition) {  
    // instructions  
}
```

```
<?php  
$n=1;  
while ($n<7)  
{  
    ...  
    echo $n."&nbsp;";  
    $n++;  
}
```

La boucle " do while "

```
do {  
    // instructions  
} while (condition);
```

```
<?php  
$n=1;  
do  
{  
    ...  
    echo $n."&nbsp;";  
    $n++;  
} while ($n<7);
```



Les boucles: *Exercice 1*

Créer une page PHP permettant d'afficher, sous forme de liste non ordonnée, les carrés des nombres de 1 à 30 selon le format :

- $1^2 = 1$
- $2^2 = 4$
- ...
- $30^2 = 900$

Utiliser d'abord l'instruction `echo` avec des apostrophes puis avec des guillemets.



Les boucles: *Exercice 2*

Créer un script PHP permettant d'afficher la table de multiplication sous la forme suivante :

	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

#5FFBF8

#F60ABF



Programmation Web en PHP

Premiers pas avec PHP

```
<table border=1 width="400" height="250" align="center">
<tr bgcolor="#5FFBF8">
<th> </th> <th>2 </th> <th>3 </th> <th> 4</th> <th>5 </th>
  <th>6 </th> <th>7 </th> <th>8 </th> <th> 9</th>
<?php
for($i=2;$i<10;$i++)
{
echo "<tr> <th bgcolor=\"\#5FFBF8\">\".$i.\" </th>\";
for($j=2;$j<10;$j++)
if($i%2==0)
    if($i==$j)
        echo "<th>\".$i*$j.\" </th>\";
    else
        echo "<td align=center>\".$i*$j.\" </td>\";
else
    if($i==$j)
        echo "<th bgcolor=\"\#F60ABF\">\".$i*$j.\" </th>\";
    else
        echo "<td align=center bgcolor=\"\#F60ABF\">\".$i*$j.\" </td>\";
}
?>
</table>
```

	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81



Les tableaux: *Définition*

Les tableaux (arrays en anglais), sont des types de données structurés permettant de stocker plusieurs valeurs de même type ou de types différents.

- ✓ Lors de la déclaration d'un tableau, il est inutile de préciser sa dimension et le type de données qu'il va contenir. PHP s'en charge automatiquement.
- ✓ Les tableaux sont dits dynamiques. A chaque nouvelle entrée enregistrée dans le tableau, PHP agrandit sa taille de 1 élément.
- ✓ Le langage PHP propose également deux types distincts de tableaux : les **tableaux à index numériques** et les **tableaux associatifs**.
- ✓ Chaque élément peut être identifié par :
 - un indice numérique (le 1^{er} est ZERO) → tableaux avec les indices
exemple: `$tab[0]` `$tab[2]` `$tab[$i + 1]`
 - une étiquette (de type string) appelée CLE (Key)
exemple: `$tab["nom"]` → tableaux associatifs



Les tableaux: *Création*

Tableau indexé numériquement:

❑ *Usage des indices:*

```
$tab[0] = 25;  
$tab[1] = " php " ;  
$tab[20] = 10.50; // les éléments 2 à 19 n'existent pas  
$tab[21] = TRUE;  
$tab[ ] = " exemple "; // si indice non précisé → indice suivant (ici 22)  
$indice = 8; $tab[$indice] = 50; // l'indice est une variable entière
```

❑ *Usage du mot clé "array":*

```
$tab = array(val0,val1,val2,.....,valN);
```



Les tableaux: *Création*

Tableau associatif :

□ *Usage des étiquettes:*

```
$tab[" nom " ] = " MOUMNI";  
$prenom= " Ali";  
$tab[" prenom "] = $prenom;
```

ATTENTION : Les clés des tableaux associatifs sont sensibles à la casse

□ *Usage du mot clé "array":*

```
$tab = array("cleA " =>valA, " cleB " => valB, ... " cleZ " =>valZ);
```



Les tableaux: *Lecture*

Boucle FOR :

```
for ($i=0;$i < count($tab);$i++)  
    { echo $tab[$i]; }
```

Boucle WHILE :

```
$i=0;  
While (isset($tab[$i])) // isset() retourne TRUE si l'élément existe  
    { echo $tab[$i];  
      $i++;  
    }
```



Les tableaux: *Lecture*

La fonction **FOREACH()** :

Par rapport à la boucle FOR, FOREACH() ne doit pas connaître à l'avance le nombre d'éléments d'un tableau

Tableaux avec les indices

```
foreach($tab as $valeur) { echo " $valeur <br>" ; }  
foreach($tab as $indice=>$valeur)  
{ echo " l'élément de clé $indice contient $valeur <br>" ; }
```

Tableaux associatifs

```
foreach($tab as $cle=>$valeur)  
{ echo " l'élément de clé $cle contient $valeur <br>" ; }
```



Les tableaux: *Exercice 1*

1. Créer un tableau ayant pour valeurs les noms des douze mois de l'année. Quelles sont alors les clés ? Parcourir ensuite le tableau pour afficher les mois de l'année sous forme de liste ordonnée.
2. Créer un tableau associant à chaque mois de l'année le nombre de jours du mois. (On supposera que l'année n'est pas bissextile.) Quelles sont alors les clés ? Les valeurs ? Afficher, sous forme de tableau, le nombre de jours de chaque mois.



Les tableaux: *Exercice 2*

1. Donnez le résultat d'exécution du script suivant :

```
<?php
```

```
$A=10; $T=array($A++=>"Lundi",$A++=>"Mardi","Mercredi");  
foreach ($T as $cle=>$val) echo "$cle: $val<br>";    ?>
```

2. On considère le code PHP suivant :

```
<?php
```

```
$Clients=array( array("code"=>"123","Nom"=>"Alami","Prenom"=>"Salima"),  
                array("code"=>"124","Nom"=>"MOHTADI","Prenom"=>"Samir"),  
                array("code"=>"125","Nom"=>"ANDALOUSSI","Prenom"=>"Karim")    );?>
```

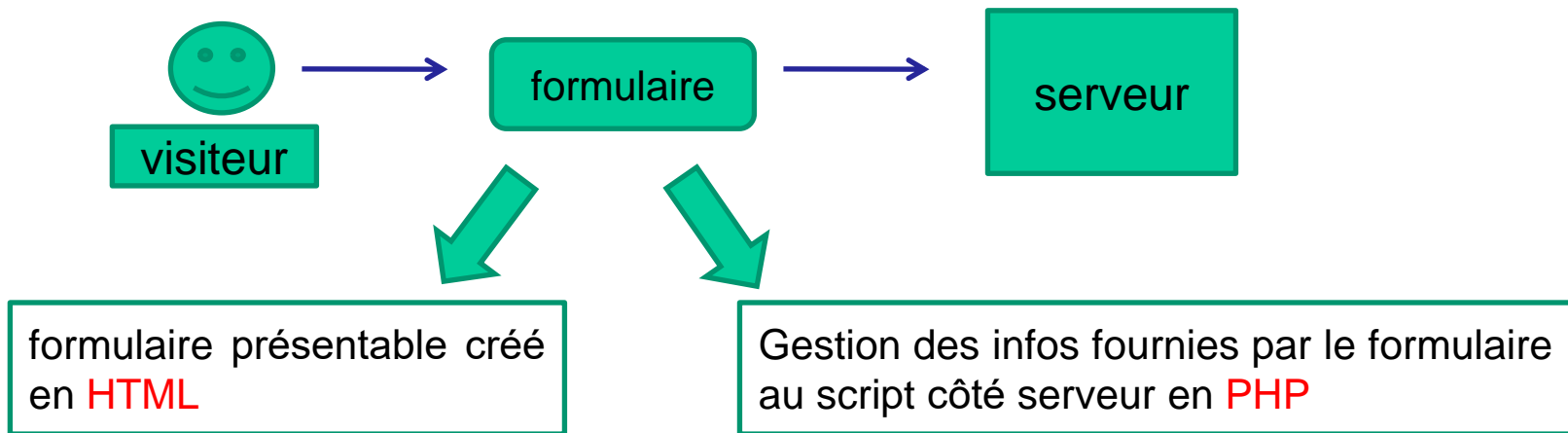
Donnez le code PHP nécessaire à la production d'un tableau HTML dynamique à partir des données stockées dans le tableau \$Clients.



Les tableaux: *Exercice 3*

Donnez le code PHP nécessaire à la production d'un tableau HTML dynamique à partir des données stockées dans un fichier texte ayant le contenu suivant:

```
10 # ALAMI # Ahmed  
20 # SLIMANI # Sanae  
30 # MOUHTADI # Louay  
40 # AMAMI # Lamiae
```



```
<form method="....." action="cible.php">  
  On mettra ici les éléments de notre formulaire.<br />  
</form>
```

- **action** lien vers la page PHP qui va traiter le contenu du formulaire
- **method** manière de transmettre l'information
 - **get** → le contenu des formulaires est ajouté à l'URL spécifié dans action (cible.php?nom=Moumni)
 - **post** → le contenu est posté dans le corps de la requête HTTP



Programmation Web en PHP

Gestion des formulaires

HTML

```
<form action="cible.php"
  method="get">
  Nom:
  <input type="text" name="nom">
  Prénom:
  <input type="text" name="prenom">
  <input type="submit"
name="action" value="OK">
</form>
```

PHP

```
<html>
...
<div>
  Bonjour
  <?php
    echo $_GET["prenom"] . ' ' .
      $_GET["nom"];
  ?>
</div>
</html>
```

localhost:83/TP_Form/form.html

Nom Prénom

localhost:83/TP_Form/cible.php?nom=MOUMNI&prenom=Ali&action=OK

...
Bonjour Ali MOUMNI



Programmation Web en PHP

Gestion des formulaires

HTML

```
<form action="cible.php"
  method="POST">
  Nom:
  <input type="text" name="nom">
  Prénom:
  <input type="text" name="prenom">
  <input type="submit"
name="action" value="OK">
</form>
```

PHP

```
<html>
...
<div>
  Bonjour
  <?php
    echo $_POST["prenom"].' '.
         $_POST["nom"];
  ?>
</div>
</html>
```

Nom MOUMNI Prenom Ali OK

