

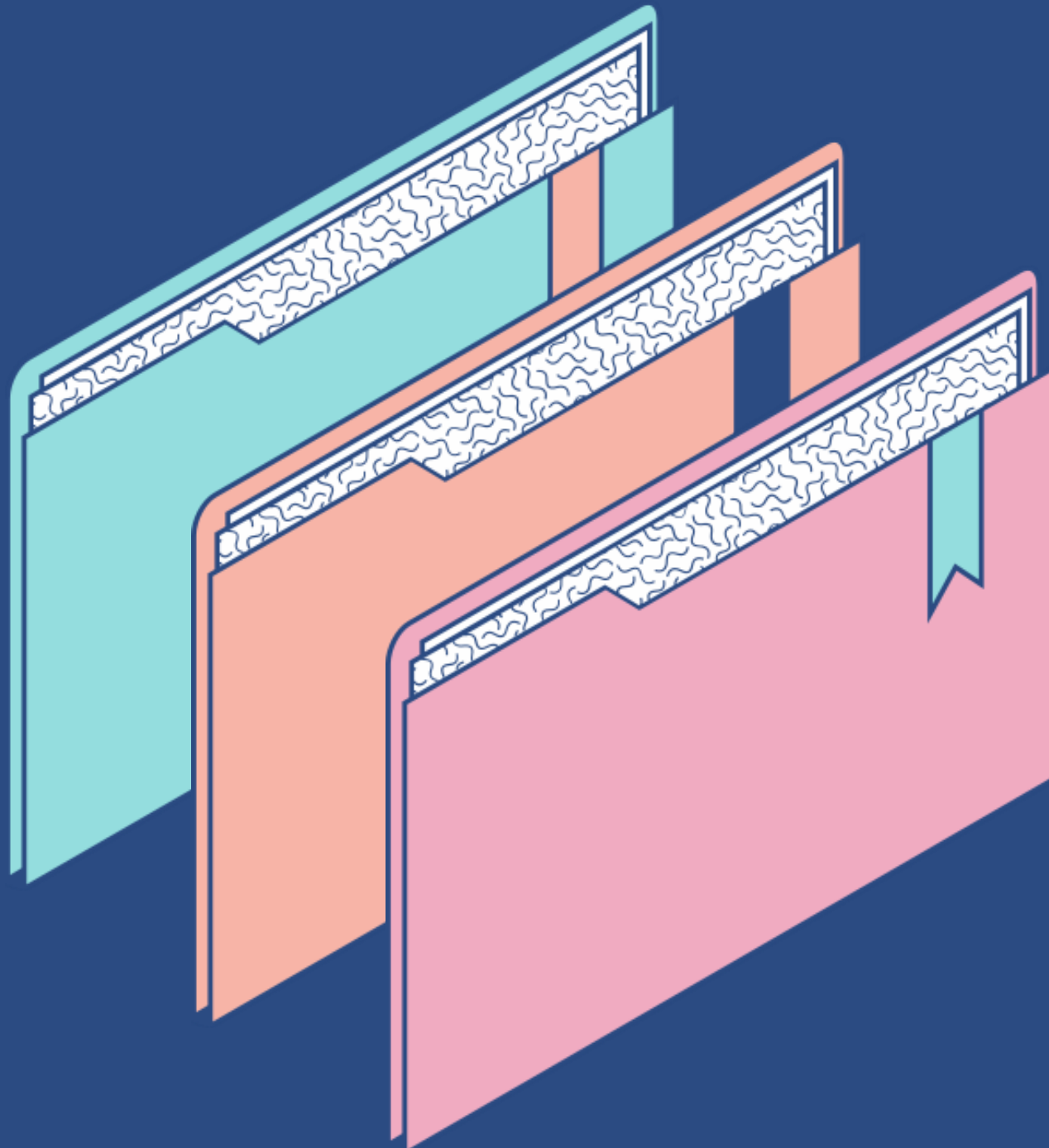
PHP

Apprendre PHP Introduction

```
1 <?php
2 require_once 'class/Message.php';
3 require_once 'class/GuestBook.php';
4 $errors = null;
5 $success = false;
6 $guestbook = new GuestBook(__DIR__ . DIRECTORY_SEPARATOR . 'data' . DIRECTORY_SEPARATOR . 'messages');
7 if (isset($_POST['username'], $_POST['message'])) {
8     $message = new Message($_POST['username'], $_POST['message']);
9     if ($message->isValid()) {
10         $guestbook->addMessage($message);
11         $success = true;
12         $_POST = [];
13     } else {
14         $errors = $message->getErrors();
15     }
16 }
17 $messages = $guestbook->getMessages();
18 $title = 'Livres d'or';
19 require 'elements/header.php';
20 ?>
21
22 <div class="container">
23     <h1>Livres d'or</h1>
24
25     <?php if (!empty($errors)): ?>
26     <div class="alert alert-danger">
27         Formulaire invalide
28     </div>
29     <?php endif ?>
30
31     <?php if ($success): ?>
32     <div class="alert alert-success">
33         Merci pour votre message
34     </div>
```

SOMMAIRE

- Présentation du PHP
- Environnement de travail
- Syntaxe du langage
- Variables
- Opérateurs
- Concaténation
- Fonctions
- Conditions
- Boucles
- Tableaux
- Super globales
- Importation de fichiers
- Interaction avec BDD
- MVC



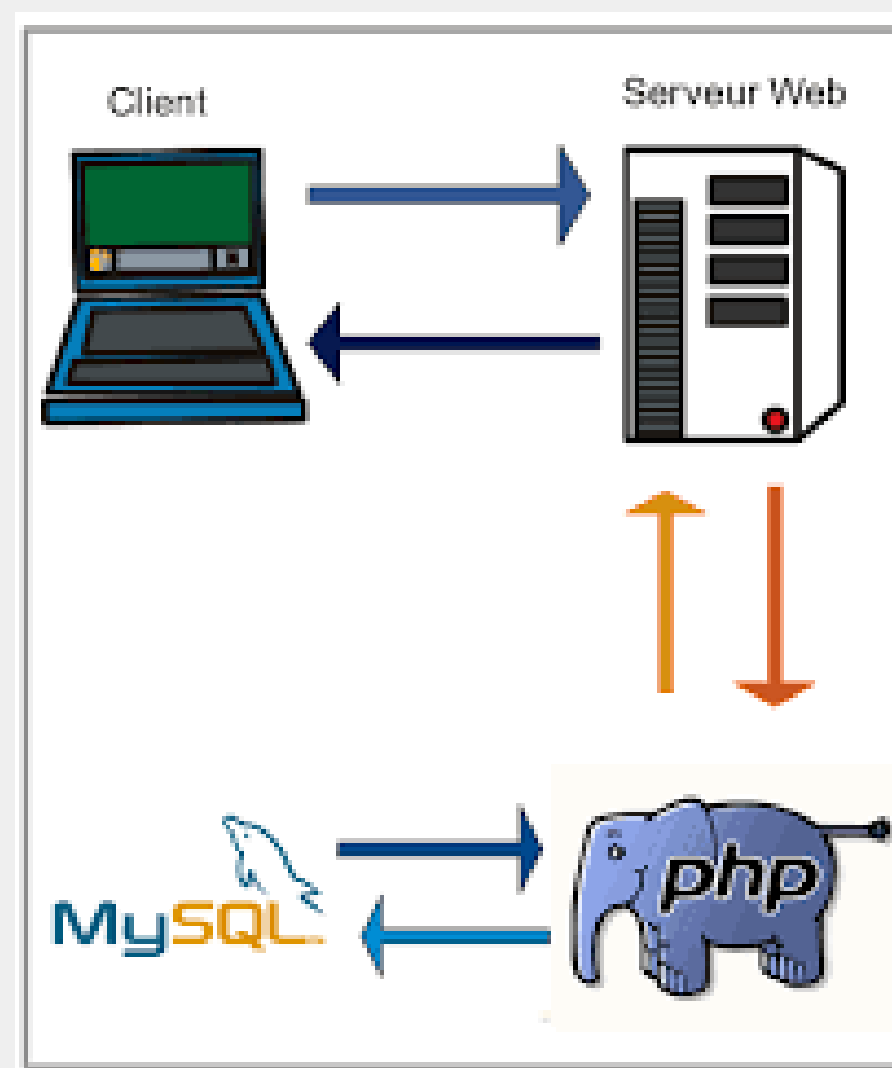
Présentation du PHP

```
1  <?php
2  require_once 'class/Message.php';
3  require_once 'class/GuestBook.php';
4  $errors = null;
5  $success = false;
6  $guestbook = new GuestBook(__DIR__ . DIRECTORY_SEPARATOR . 'data' . DIRECTORY_SEPARATOR . 'messages');
7  if (isset($_POST['username'], $_POST['message'])) {
8      $message = new Message($_POST['username'], $_POST['message']);
9      if ($message->isValid()) {
10         $guestbook->addMessage($message);
11         $success = true;
12         $_POST = [];
13     } else {
14         $errors = $message->getErrors();
15     }
16 }
17 $messages = $guestbook->getMessages();
18 $title = 'Livres d'or';
19 require 'elements/header.php';
20 ?>
21
22 <div class="container">
23     <h1>Livres d'or</h1>
24
25     <?php if (empty($errors)): ?>
26     <div class="alert alert-danger">
27         Formulaire invalide
28     </div>
29     <?php endif ?>
30
31     <?php if ($success): ?>
32     <div class="alert alert-success">
33         Merci pour votre message
34     </div>
```

Apprendre PHP Introduction



Présentation du PHP





Présentation du PHP

- PHP est un langage multiplateforme (Windows, Linux, Mac ...)
- Pour fonctionner, PHP nécessite un serveur web (Apache/IIS)
 - PHP est un langage interprété (conversion html)



Environnement de travail

- Pour le développement PHP, nous avons besoin d'un serveur web (xampp,...) contenant :
 - Apache (Serveur web pour héberger les fichiers)
 - MySQL (Serveur d'hébergement des BDD)

Syntaxe du langage

7

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>tp01php</title>
</head>
  <body>
    <H3>votre première page en PHP</H3>

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

  </body>
</html>
```

HTML

code PHP



Syntaxe du langage

- L'intégration de code PHP se fait au sein de fichiers ".php" (par exemple, index.php)
 - Les scripts php s'intégreront entre balises `<?php ?>`

Exemple de code PHP

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ma première page php</title>
</head>
<body>
  <h1>mon premier programme</h1>
  <?php
//le script php se trouvera entre ces balises
  ?>
</body>
</html>
Chaque ligne de nos scripts devra se terminer par un ;
<?php
//script php;
?>
```

Commentaires

```
<?php
//commentaire sur une ligne
/*
-----
Commentaire sur plusieurs lignes
-----
*/
?>
```



Exercice d'application

Création du premier programme PHP

Créez un programme permettant d'afficher le code "Hello World"

1) Créez une page index.php au sein de votre serveur web (htdocs/index.php)

2) Créez une page HTML complète, et affichez le texte "Hello World" à l'aide de la commande php "echo"

tips : le code `echo "Pierre";` permettra d'afficher Pierre

Variables in PHP

```
$area_of_triangle;
```

```
//integer value
```

```
// printing area
```

```
//float value
```

```
($height * $base) / 2;
```

```
$5input = 'Demo';
```





Types de variables

- Le type « chaîne de caractères » ou String en anglais,
 - Le type « nombre entier » ou Integer en anglais,
 - Le type « nombre décimal » ou Float en anglais,
 - Le type « booléen » ou Boolean en anglais,
 - Le type « tableau » ou Array en anglais,
 - Le type « objet » ou Object en anglais,
- Le type « NULL » qui se dit également NULL en anglais.



Déclaration de variables

En PHP, une variable s'écrit :

```
$nomVariable = valeur;
```

Exemple :

```
$variable = 10;  
$total = $variable + 10; //total vaut 20 (10 de variable + 10 en numérique).  
$variable2 = "Pierre";
```


Afficher le contenu d'une variable

Affichage du contenu d'une variable : echo var

```
//initialisation d'une variable  
$nbr =2 ;  
//la fonction php echo permet d'afficher le contenu de la variable nbr  
echo $nbr ;
```



Afficher le type d'une variable

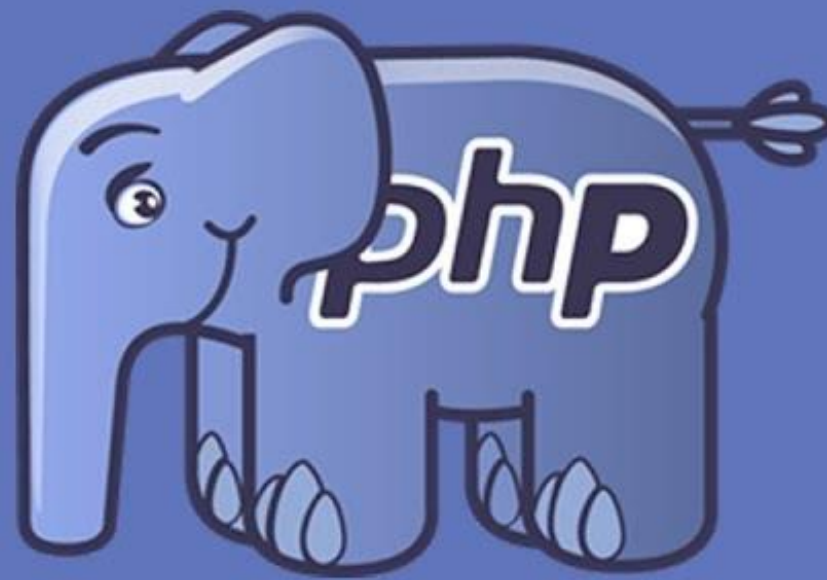
Affichage du type d'une variable : `gettype(var)`

```
//initialisation d'une variable  
$nbr =2 ;  
//affichage dans la page web avec la fonction echo  
echo $nbr ;  
//utilisation de la fonction gettype pour afficher le type de la variable  
echo gettype($nbr);
```



Exercice d'application variables PHP

- Créer une variable de type int avec pour valeur 5,
 - Afficher le contenu de la variable
 - Afficher son type
- Créer une variable de type String avec pour valeur votre prénom,
 - Afficher le contenu de la variable
- Créer une variable de type booléen avec pour valeur false,
 - Afficher son type



8

OPÉRATEURS



Opérateurs arithmétiques

- Addition :
 - $a+b$
- Soustraction :
 - $a-b$
- Multiplication :
 - $a*b$
- Division :
 - a/b
- Modulo :
 - $a \% b$ (reste de la division de a divisé par b)
- Exponentielle :
 - $a^{**}b$ (Résultat de l'élévation de a à la puissance b)



Exercice d'application opérateurs arithmétiques n°1

- Créer 2 variables \$a et \$b qui ont pour valeur 12 et 10,
- Stocker le résultat de l'addition de \$a et \$b dans une variable \$total,
 - Afficher le résultat



Exercice d'application opérateurs arithmétiques n°2

- Créer 3 variables \$a, \$b et \$c qui ont pour valeur $\$a = 5$, $\$b = 3$ et $\$c = \$a + \$b$,
- Afficher la valeur de chaque variable
 - passer la valeur de \$a à 2,
 - Afficher la valeur de \$a,
 - passer la valeur de \$c à $\$b - \a ,
- Afficher la valeur de chaque variable



Exercice d'application opérateurs arithmétiques n°3

- Créer 2 variables \$a et \$b qui ont pour valeur 15 et 23,
- Afficher la valeur de chaque variable
- Intervertissez les valeurs de \$a et \$b,
- Afficher la valeur de \$a et \$b

Exercice d'application opérateurs arithmétiques n°3

```
$a = 15;  
$b = 23;  
echo "Début -- a vaut : $a et b vaut : $b";  
$c = $a;  
$a = $b;  
$b = $c;  
echo "Fin -- a vaut : $a et b vaut : $b";
```



Exercice d'application opérateurs arithmétiques n°4

- Ecrire un programme qui prend le prix HT d'un article, le nombre d'articles et le taux de TVA, et qui fournit le prix total TTC correspondant.
- Afficher le prix HT, le nbr d'articles et le taux de TVA
- Afficher le résultat

Exercice d'application opérateurs arithmétiques n°4

```
$prixHt = 80;  
$nbArticles = 4;  
$tauxTva = 20;  
$total = ($prixHt+$prixHt*$tauxTva/100)*$nbArticles;  
echo "prix ht : $prixHt, nb Articles : $nbArticles, taux tva : $tauxTva";  
echo "prix total : $total";
```

```
39 <?php  
40 maChaine = "Hello"  
41         + "World";
```

Qu'est-ce que
la concaténation ?

Concaténation

- Ecrire le nom d'une variable au sein d'une page web : `\$var`

```
<?php
$nom = " test " ;
//on va utiliser le symbole \devant le nom de la variable,
//ce caractère permet d'annuler l'interprétation du caractère qui va suivre,
//dans ce cas il va afficher le nom de la variable et ensuite son contenu.
echo "affichage de la variable s'appelant \$test $test" ;
?>
```



Concaténation

- Concaténer des chaînes de caractère : `$var1 . $var2`

```
<?php
    $a = " test " ;
    $b = " azerty " ;
    $c = $a.$b;
    echo $c;
?>
```




Exercice d'application concaténation n°1

- Créer une variable \$a qui a pour valeur « bonjour »,
- Afficher le nom de la variable et sa valeur.



Exercice d'application concaténation n°2

- Créer 1 variable \$a qui a pour valeur « bon »,
- Créer 1 variable \$b qui a pour valeur « jour »,
 - Créer 1 variable \$c qui a pour valeur 10,
 - Concaténer \$a, \$b et \$c +1,
- Afficher le résultat de la concaténation.

Exercice d'application concaténation n°2

```
<?php
    $a = " bon " ;
    $b = " jour " ;
    $c = 10 ;
    $total = $a.$b.$c+1 ;
    echo $total ;
?>
```




Fonctions PHP

- Création d'une fonction :

```
<?php  
function nom_de_la_fonction()  
{  
    // Code de la fonction  
}  
?>
```

Fonctions PHP

- Appel d'une fonction :

```
<?php
function ma_fonction()
{
    //Code de la fonction
}
ma_fonction(); // Appel de la fonction
?>
```

Fonctions PHP

- Création d'une fonction avec paramètres

```
<?php

function ma_fonction($a,$b){ // Fonction à 2 paramètres
    $result= $a+$b ;
    return $result ; // Retour de la fonction
}

$value = ma_fonction(10,5); // Appel de la fonction avec 2 paramètres

echo $value; // Affichage du retour de la fonction

?>
```



Exercice fonctions n°1

- Créer une fonction qui soustrait à \$a la variable \$b (2 paramètres en entrée),
- la fonction doit retourner le résultat (return).

Exercice fonctions n°1

```
<?php

function ma_fonction($a,$b){ // Fonction à 2 paramètres
    $result= $a-$b ;
    return $result ; // Retour de la fonction
}

$value = ma_fonction(10,5); // Appel de la fonction avec 2 paramètres

echo $value; // Affichage du retour de la fonction

?>
```



Exercice fonctions n°2

- Créer une fonction qui prend en entrée un nombre à virgule (float),
- la fonction doit retourner l'arrondi (return) du nombre en entrée.

Exercice fonctions n°2

```
<?php

function ma_fonction($a){ // Fonction à 2 paramètres
    $result= intval($a); // Fonction intval() permettant de convertir en int
    return $result ; // Retour de la fonction
}

$value = ma_fonction(10.25); // Appel de la fonction avec 2 paramètres

echo $value; // Affichage du retour de la fonction

?>
```



Exercice fonctions n°3

-Créer une fonction qui prend en entrée 3 valeurs et retourne somme des 3 valeurs.



Exercice fonctions n°4

-Créer une fonction qui prend en entrée 3 valeurs et retourne la valeur moyenne des 3 valeurs (saisies en paramètre).

Exercice fonctions n°4

```
<?php

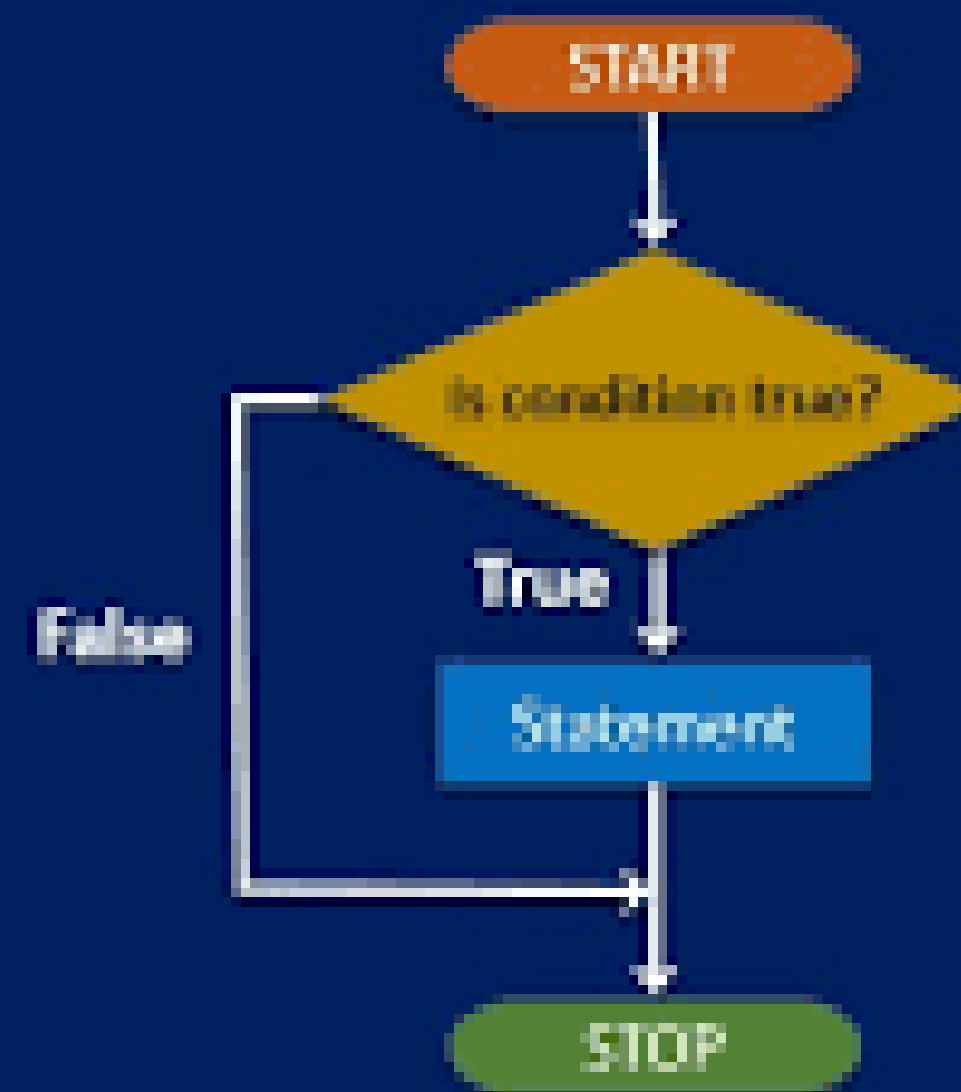
function ma_fonction($a,$b,$c){ // Fonction à 2 paramètres
    $result = ($a+$b+$c)/3;
    return $result ; // Retour de la fonction
}

$value = ma_fonction(2,4,6); // Appel de la fonction avec 2 paramètres

echo $value; // Affichage du retour de la fonction

?>
```

if else Statement in PHP



Opérateurs de comparaison

Exemple	Nom	Résultat
$\$a == \b	Egal	true si $\$a$ est égal à $\$b$ après le transtypage.
$\$a === \b	Identique	true si $\$a$ est égal à $\$b$ et qu'ils sont de même type.
$\$a \neq \b	Différent	true si $\$a$ est différent de $\$b$ après le transtypage.
$\$a <> \b	Différent	true si $\$a$ est différent de $\$b$ après le transtypage.
$\$a \neq \b	Différent	true si $\$a$ est différent de $\$b$ ou bien s'ils ne sont pas du même type.
$\$a < \b	Plus petit que	true si $\$a$ est strictement plus petit que $\$b$.
$\$a > \b	Plus grand que	true si $\$a$ est strictement plus grand que $\$b$.
$\$a \leq \b	Inférieur ou égal	true si $\$a$ est plus petit ou égal à $\$b$.
$\$a \geq \b	Supérieur ou égal	true si $\$a$ est plus grand ou égal à $\$b$.





Opérateurs logiques

Exemple	Nom	Résultat
\$a and \$b	And (Et)	true si \$a ET \$b valent true.
\$a or \$b	Or (Ou)	true si \$a OU \$b valent true.
\$a xor \$b	XOR	true si \$a OU \$b est true, mais pas les deux en même temps.
! \$a	Not (Non)	true si \$a n'est pas true.
\$a && \$b	And (Et)	true si \$a ET \$b sont true.
\$a \$b	Or (Ou)	true si \$a OU \$b est true.

Exemple opérateur

```
<?php
$a = 6;
if($a<=3 and $a >0)
{ //test si $a est plus petit que 3 et est supérieur à 0
  echo "la valeur de la variable \"$a\" est plus petite que 3";
}
else if($a>=3 && $a <5)
{ //test si $a est plus grand ou égal et 3 et inférieur à 5
  echo "la valeur de la variable \"$a\" est comprise entre 3 et 5";
}
else
{ //test autre cas
  echo "la valeur de la variable \"$a\" est supérieur à 5";
}
?>
```



Exercice conditions n°1

-Créer une fonction qui teste si un nombre est positif ou négatif



Exercice conditions n°2

-Créer une fonction qui prend en entrée 3 valeurs et retourne le nombre le plus grand

Exercice conditions n°2

```
function getMax($a,$b,$c)
{
    if($a > $b && $a > $c)
    {
        $max = $a;
    }else if($b > $c)
    {
        $max = $b;
    }else{
        $max = $c;
    }
    return $max;
}
echo getMax(10,4.26,60.5);
```



Exercice conditions n°3

-Créer une fonction qui prend en entrée 3 valeurs et retourne le nombre le plus petit

Exercice conditions n°3

```
function getMin($a,$b,$c)
{
    if($a < $b && $a < $c)
    {
        $min = $a;
    }else if($b < $c)
    {
        $min = $b;
    }else{
        $min = $c;
    }
    return $min;
}
echo getMin(10,4.26,60.5);
```




Boucles : while, for
do... while, foreach



Boucle for

```
<?php
// for (valeur initiale ; condition ; opération)
for ($i=0; $i<10; $i++) //boucle for
{
    echo 'Ceci est une boucle for en PHP';
    echo '<br>';
}
?>
```

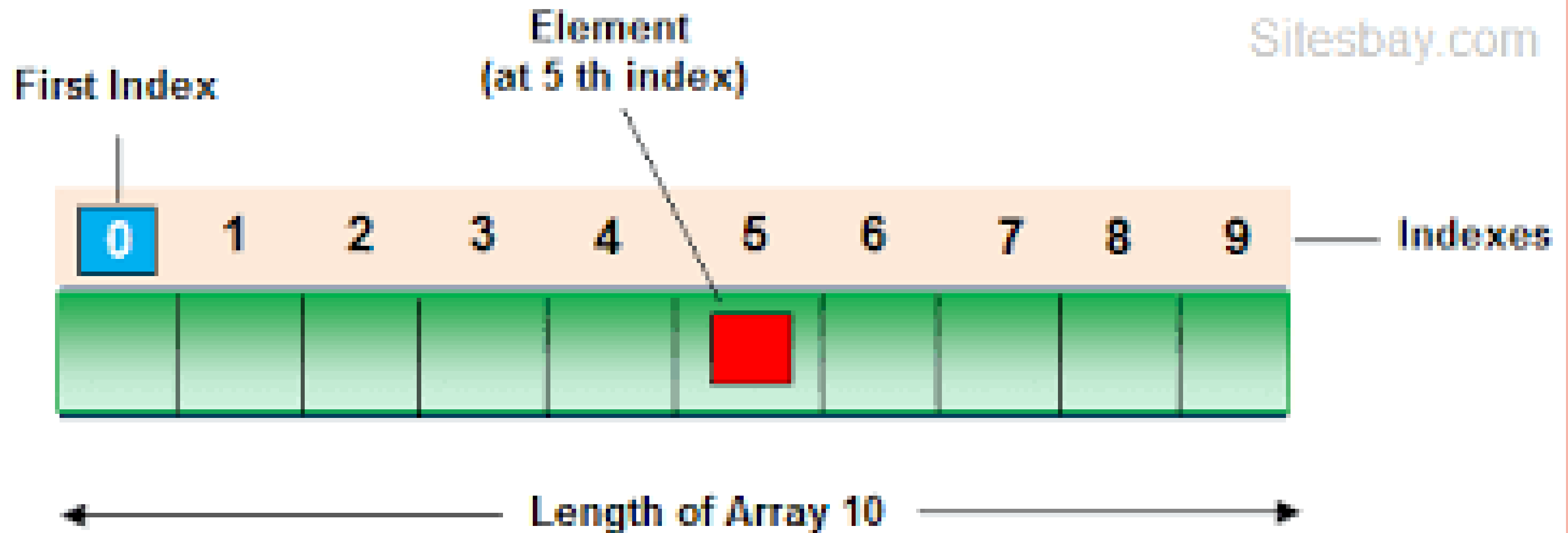
Boucle while

```
<?php
    $i = 0; //variable compteur
    while ($i < 10) //boucle while
    {
        //j'affiche la valeur de $i
        echo $i;
        //à chaque tour j'incrémente $i (+1)
        $i++;
        //je saute une ligne
        echo '<br>';
    }
?>
```

Boucle foreach

```
<?php
    foreach ($tableau as $valeur)
    {
        |   echo $valeur.'<br />'. "\"\n";
    }
?>
```

Tableaux





Création de tableau

- Les tableaux (Array) permettent de grouper des informations ensemble.
- Les tableaux peuvent stocker une multitude de valeurs de type différent.

Il n'est pas utile de préciser la taille du tableau, à chaque nouvel ajout PHP agrandit de 1 la taille du tableau.



Déclaration de tableau

- La déclaration d'un tableau se fait de la même manière qu'une variable, avec le signe \$ et un nom.
- Il faudra ensuite utiliser la structure de langage `array()`, permettant de déclarer notre variable en tant que tableau.
- Si aucun paramètre n'est ajouté, le tableau sera initialisé vide. Si une (ou un ensemble) de valeur est précisé, le tableau sera initialisé avec ces valeurs



Déclaration de tableau

- La déclaration d'un tableau se fait de la même manière qu'une variable, avec le signe \$ et un nom.
- Il faudra ensuite utiliser la structure de langage `array()`, permettant de déclarer notre variable en tant que tableau.
- Si aucun paramètre n'est ajouté, le tableau sera initialisé vide. Si une (ou un ensemble) de valeur est précisé, le tableau sera initialisé avec ces valeurs

Déclaration de tableau

```
<?php
//déclaration d'un tableau vide (tab) :
$tab = array() ;
//déclaration d'un tableau indexé numériquement :
$tab1 = array(1,8,7,11) ;

//déclaration d'un tableau associatif :
$identite = array(
    'nom' => "tucat",
    'prenom' => "jean",
    'age' => 246,
    'estFormateur' => false
);
?>
```


Ajout de valeur à un tableau

```
<?php
// Ajout d'un élément a un tableau indexé numériquement il sera ajouté à la dernière position.
$legumes[] = 'salade';
// Ajout d'un élément a un tableau indexé numériquement en dernière position
array_push($legumes,"choux");
// Ajout d'un élément a un tableau indexé numériquement à une position (2° position).
$legumes[1] = 'tomate';
// Ajout de la taille de la personne dans le tableau associatif
$identite['taille'] = 180;
// Affichage d'un tableau
print_r($legumes);
?>
```

Parcourir un tableau

```
<?php
    //création d'un tableau $prenoms
    $prenoms[0] = 'Mathieu';
    $prenoms[1] = 'Sophie';
    $prenoms[2] = 'Florence';
    //ou
    $prenoms = array('Mathieu', 'Sophie', 'Florence');
    //parcours de tout le tableau
    foreach ($prenoms as $key => $value) {
        |         echo '<br>';
        //Affiche le contenu de la case à chaque tour.
        |         print_r($value);
    }
?>
```



Exercice tableaux n°1

-Créer une fonction qui affiche la valeur la plus grande du tableau.

Exercice tableaux n°1

```
<?php

function maxArray($tab)
{
    $max = $tab[0];
    foreach($tab as $value)
    {
        if($value > $max)
        {
            $max = $value;
        }
    }
    return $max;
}

$tab = array(1,5,9,2,3,4,8,7,6,10,4);
echo maxArray($tab);

?>
```




Exercice tableaux n°2

-Créer une fonction qui affiche la moyenne du tableau.

Exercice tableaux n°2

```
<?php

function moyArray($tab)
{
    $moy = 0;
    foreach($tab as $value)
    {
        $moy += $value; // equivaut à $moy = $moy + $value
    }
    $moy = $moy/sizeof($tab); // Permet de retourner le nombre d'éléments du tableau
    return $moy;
}

$tab = array(1,5,9,2,3,4,8,7,6,10,4);
echo moyArray($tab);

?>
```

PHP : \$_GET & \$_POST

Name

Age

Gender ☐ Male ☐ Female

File1.php



- Database
- **Print**
- Conditional Page Content

File2.php

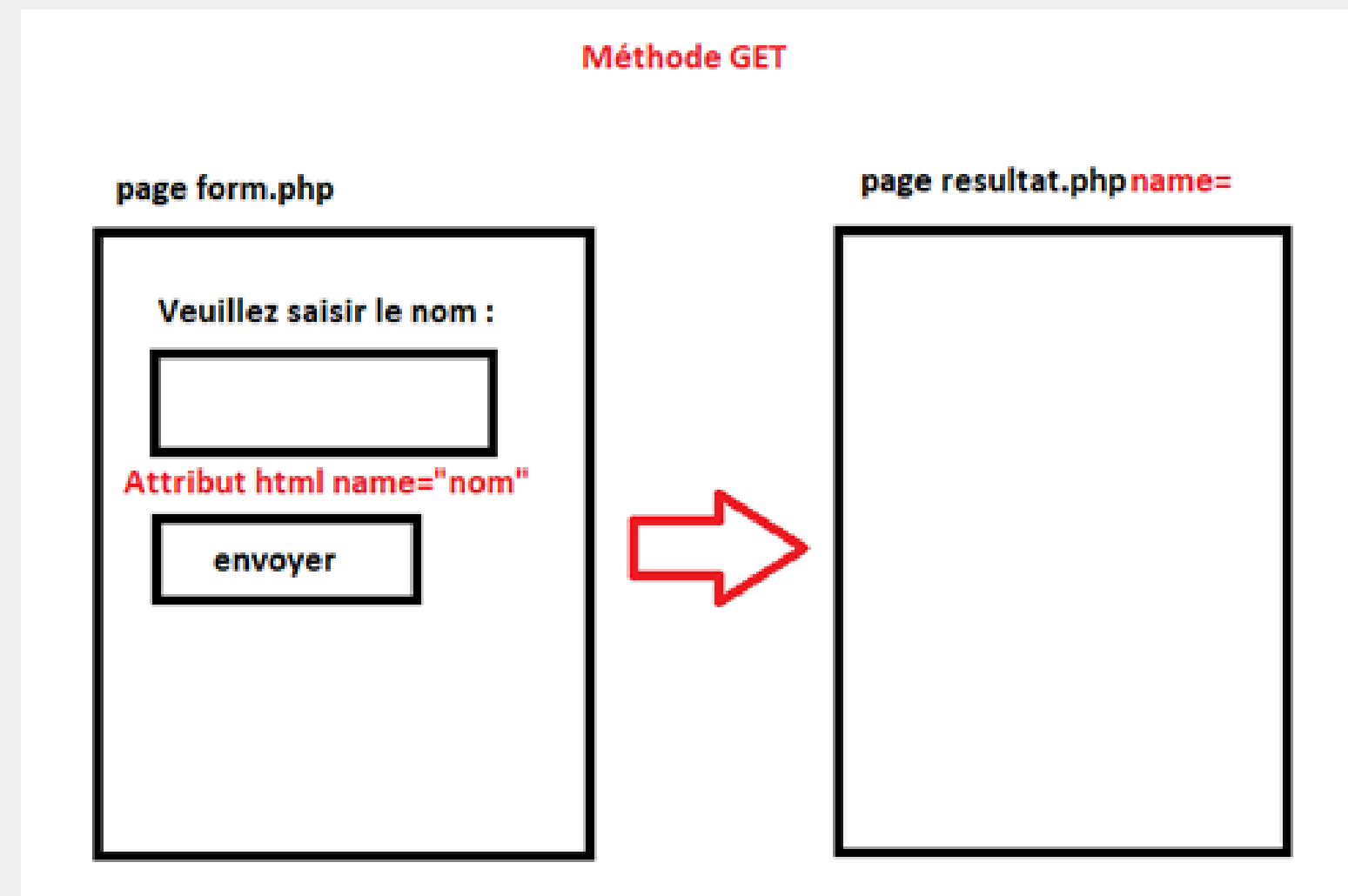


Présentation des super globales

- Le transfert de données entre pages PHP est géré par des variables spéciales, nommées super globales
 - Les formulaires possèdent 2 méthodes d'envoi : *get* et *post*
 - *GET* : passage des informations par URL (limité à 255 caractères)
- *POST* : passage des informations par le body de la page (plus sécurisée, et sans limite de taille)

Super Globale GET

- Le contenu du formulaire va transiter dans l'URL en récupérant les attributs "name" du formulaire



Super Globale GET

```
<html lang="fr">
  <head>
    <meta charset="UTF-8">
    <title>formulaire</title>
  </head>
  <body>
    <form action="resultat.php" method="get">
      <p>veuillez saisir votre nom :</p>
      <input type="text" name="nom">
      <br>
      <input type="text" name="prenom">
      <br>
      <input type="submit" value="Envoyer">
    </form>
  </body>
</html>
```

- envoi des données nom et prenom à la page resultat.php sous la forme :

<http://resultat.php?name=valeur&prenom=valeur>



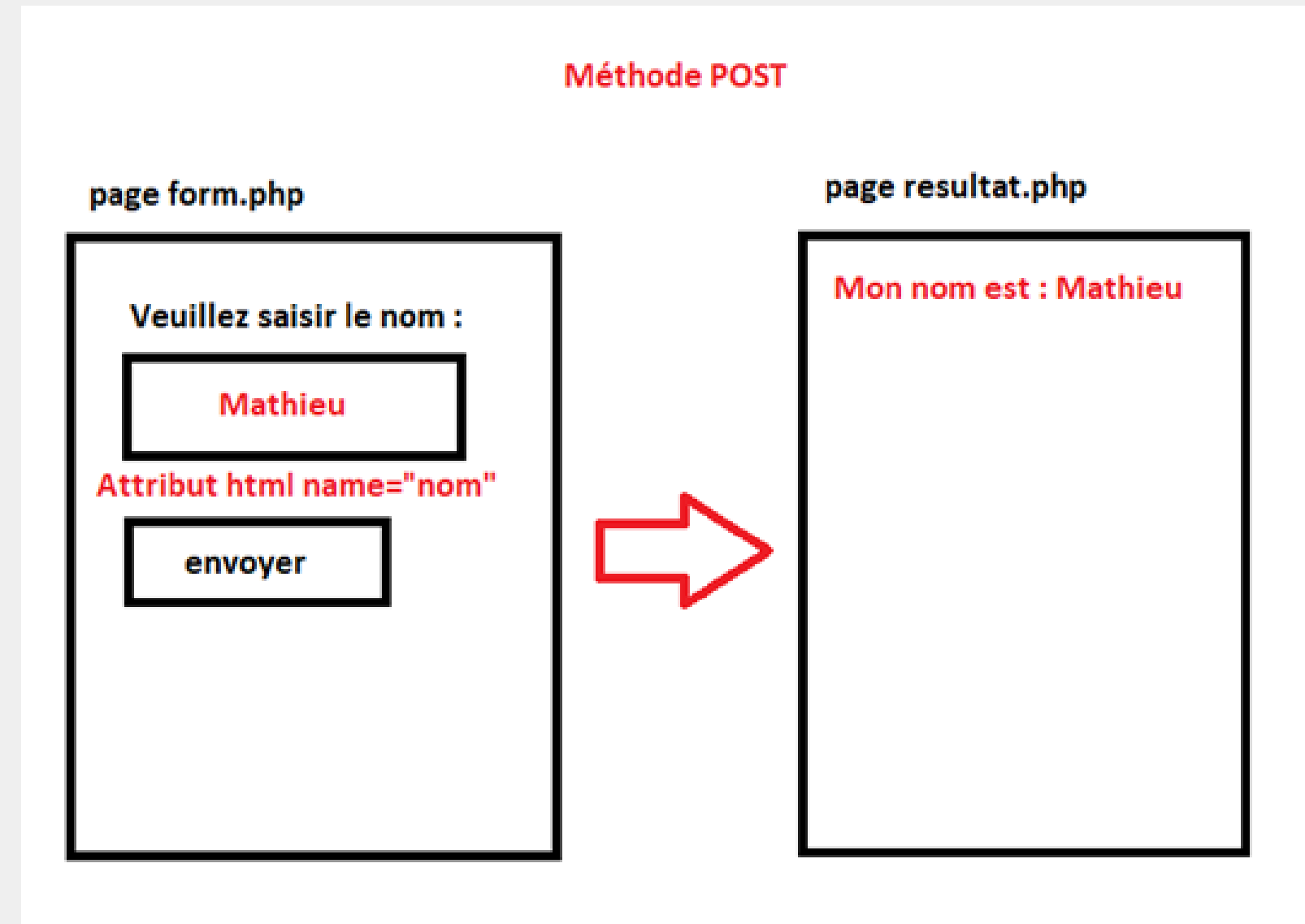
Super Globale GET

```
<?php
//test de l'existence de la super globale $_GET
if(isset($_GET['nom'])){
    $nom = $_GET['nom'];
    $prenom = $_GET['prenom'];
    echo "mon nom est : $nom et mon prenom est $prenom";
}
?>
```

- Récupération des variables nom et prenom sur la page resultat.php à l'aide de la commande : `$_GET['name']`

Super Globale POST

- Le contenu du formulaire va transiter dans l'URL en récupérant les attributs "name" du formulaire



Super Globale POST

```
<html lang="fr">
  <head>
    <meta charset="UTF-8">
    <title>formulaire</title>
  </head>
  <body>
    <form action="resultat.php" method="post">
      <p>veuillez saisir votre nom :</p>
      <input type="text" name="nom">
      <br>
      <input type="text" name="prenom">
      <br>
      <input type="submit" value="Envoyer">
    </form>
  </body>
</html>
```

- envoi des données nom et prenom à la page resultat.php au sein du body de la page

Super Globale POST

```
<?php
//test de l'existence de la super globale $_POST
if(isset($_POST['nom']) && isset($_POST['prenom'])){
    $nom = $_POST['nom'];
    $nom = $_POST['prenom'];
    echo "mon nom est : $nom et mon prenom est $prenom";
}
?>
```

- Récupération des variables nom et prenom sur la page resultat.php à l'aide de la commande : `$_POST['name']`



Exercice super globales n°1

- Créer une page de formulaire dans laquelle on aura 2 champs de formulaire de type nombre.
- Afficher dans cette même page la somme des 2 champs avec un affichage du style :
La somme est égale à : valeur

Correction exercice super globales n°1

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <form action = "index.php" method="POST">
    <input type="number" name="nombre1">
    <br>
    <input type="number" name="nombre2">
    <br>
    <input type="submit" value="Envoyer">
  </form>
</body>
</html>
```

```
<?php
//test de l'existence de la super globale $_POST
if(isset($_POST['nombre1']) && isset($_POST['nombre2'])) {
    $somme = $_POST['nombre1'] + $_POST['nombre2'];
    echo "la somme est $somme";
}
?>
```




Exercice super globales n°2

- Créer une page de formulaire dans laquelle on aura 3 champs de formulaire de type nombre :
 - 1 champ de formulaire qui demande un prix HT d'un article,
 - 1 champ de formulaire qui demande le nombre d'article,
 - 1 champ de formulaire qui demande le taux de TVA,

- Afficher dans cette même page le prix TTC (prix HT+prix HT*taux TVA*quantité) avec un affichage du style :

Le prix TTC est égal à : valeur €.

Correction exercice super globales n°2

```
<body>
  <form action = "index.php" method="POST">
    <input type="number" name="prixHt">
    <br>
    <!-- step permet l'ajout de nombres décimaux -->
    <input type="number" step="0.01" name="tva">
    <br>
    <input type="number" name="quantite">
    <br>
    <input type="submit" value="Envoyer">
  </form>
</body>
```

```
<?php
//test de l'existence de la super globale $_POST
if(isset($_POST['prixHt']) && isset($_POST['tva'])&& isset($_POST['quantite'])){
    echo "le prix ttc est egal à :".(($_POST['prixHt']+$_POST['prixHt']*$_POST['tva'])*$_POST['quantite'])."€"
}
?>
```

Importation des fichiers

79





Importation de fichiers

- Il est possible en PHP d'importer des fichiers à l'intérieur du répertoire du projet
Pour cela, il faudra utiliser la super globale `$_FILES`
- Lors de l'import, le fichier sera importé avec un nom temporaire (ex: tmp_123469324.jpg) à la racine de votre serveur apache

Importation de fichiers

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Importer un fichier </title>
</head>
<body>
  <form action="index.php" method="POST" enctype="multipart/form-data">
    <h2>importer une image</h2>
    <input type="file" name="file">
    <p><button type="submit">importer</button></p>
  </form>
</html>
```



Importation de fichiers

- Nous allons maintenant voir comment traiter et déplacer notre fichier dans le bon dossier :
 - Créez un nouveau projet à la racine de votre serveur apache (htdocs/"nom_du_projet")
 - Créez un nouveau répertoire "import" à la racine de votre projet
 - créez un dossier "image" à l'intérieur de celui ci
 - Créez un fichier index.php à la racine de votre projet, et copiez collez y le code html de la page précédente

Importation de fichiers

- Ajoutez le code suivant à la suite de votre de votre fichier index.php, et testez !

```
<?php
//test si le fichier importé existe
if(isset($_FILES['file'])) {
    //stocke le chemin et le nom temporaire du fichier importé (ex /tmp/125423.pdf)
    $tmpName = $_FILES['file']['tmp_name'];
    //stocke le nom du fichier (nom du fichier et son extension importé ex : test.jpg)
    $name = $_FILES['file']['name'];
    //stocke la taille du fichier en octets
    $size = $_FILES['file']['size'];
    //stocke les erreurs (pb d'import, pb de droits etc...)
    $error = $_FILES['file']['error'];
    //déplacer le fichier importé dans le dossier image à la racine du projet
    $fichier = move_uploaded_file($tmpName, "./import/image/$name");
}
?>
```


Interaction avec une BDD

84





Introduction PHP/BDD

- PHP permet d'interagir de façon simple, et possiblement sécurisé avec une BDD.
- L'interaction se fait à travers plusieurs étapes :
 - Connexion à la BDD
 - Exécution de la requête
 - Récupération du résultat (en cas de SELECT)

Connexion à la BDD

```
//connexion à la base de données  
  
$bdd = new PDO('mysql:host=localhost;dbname=nom_de_la_bdd', 'root','',  
array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));  
$bdd->exec("set names utf8");
```

- mysql:host : adresse de la BDD (ip, dns)
 - dbname : nom de la base de donnée
 - root : utilisateur de la BDD
 - " : mot de passe de l'utilisateur
- array(...) : mode de l'objet PDO (ici, mode erreurs avancé)



Execution d'une requête SQL

Il existe 2 méthodes pour exécuter les requêtes SQL :

- Les requêtes classiques, NON SECURISEES
 - Les requêtes préparées, SECURISEES



Requête classique

```
//Exécution de la requête SQL avec un try catch pour la gestion des exceptions (messages d'erreurs)
try
{
    //requête pour stocker le contenu de toute la table le contenu est stocké dans le tableau $reponse
    $reponse = $bdd->query('SELECT * FROM utilisateur');
    //boucle pour parcourir et afficher le contenu de chaque ligne de la table
    while ($donnees = $reponse->fetch())
    {
        //affichage des données d'une colonne de la bdd par son nom d'attribut
        echo '<p>'.$donnees['nom_attribut'].'</p>';
    }
}
catch(Exception $e)
{
    //affichage d'une exception en cas d'erreur
    die('Erreur : '.$e->getMessage());
}
```

***Mauvaise
pratique***

Requête préparée

```
//Exécution de la requête SQL avec un try catch pour la gestion des exceptions (messages d'erreurs)
try
{
    //Préparation de la requête SQL nous stockons dans une variable $req la requête à exécuter
    $req = $bdd->prepare('SELECT * FROM utilisateur where nom_utilisateur = :nom_utilisateur');
    //Exécution de la requête SQL création à l'aide d'un tableau qui va contenir le ou les paramètres à affecter à la requête SQL
    $req->execute(array(
        'nom_utilisateur' => $nom_utilisateur
    ));
    //boucle pour parcourir et afficher le contenu de chaque ligne de la table
    while ($donnees = $reponse->fetch())
    {
        //affichage des données d'une colonne du résultat de la requête par son nom d'attribut
        echo '<p>'.$donnees['nom_attribut'].'</p>';
        {
            //fermeture de la connexion à la bdd
            $req->closeCursor();
        }
    }
}
catch(Exception $e)
{
    //affichage d'une exception en cas d'erreur
    die('Erreur : '.$e->getMessage());
}
```

**Bonne
pratique**



Exercice BDD n°1

a) Créer un BDD "articles"

- Créer un table "article" : id_article, nom_article, contenu_article

b) Créer une page PHP contenant un formulaire HTML POST

- Champs : nom_article, contenu_article

c) Créer une page de traitement PHP

- Créer 2 variables \$name et \$content, récupérant le contenu du formulaire

- Ajouter la connexion à la BDD

- d) Insérer dans la BDD

- methode 1 : Ajouter une requête SIMPLE insérant les 2 champs dans la BDD

- methode 2 : Ajouter une requête PREPAREE insérant les 2 champs dans la BDD

- e) Créer une page d'affichage des résultats

- Créer une page permettant d'afficher dans un tableau le contenu de la table article

Correction Exercice BDD n°1

```
// Connexion à la BDD
$bdd = new PDO('mysql:host=localhost;dbname=articles', 'root','', array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
$bdd->exec("set names utf8");

// Test de l'existence des champs du formulaire
if(isset($_POST['nom_article']) && isset($_POST['contenu_article']))
{
    // Récupération des données du formulaire
    $nom = $_POST['nom_article'];
    $contenu = $_POST['contenu_article'];
}
```


Correction Exercice BDD n°1

```
// METHODE 1 : REQUETE SIMPLE
```

```
try
{
    // Insertion à l'aide de la methode exec
    $reponse = $bdd->exec("INSERT INTO article (nom_article,contenu_article) VALUES ('$nom','$contenu')");
}
catch(Exception $e)
{
    //affichage d'une exception en cas d'erreur
    die('Erreur : '.$e->getMessage());
}
```


Correction Exercice BDD n°1

```
// METHODE 2 : REQUETE PREPAREE
```

```
try
{
    $requete = $bdd->prepare("
        INSERT INTO article (nom_article, contenu_article)
        VALUES (:nom, :contenu)
    ");
    $requete->execute(array(
        ':nom' => $nom,
        ':contenu' => $contenu
    ));
}
```

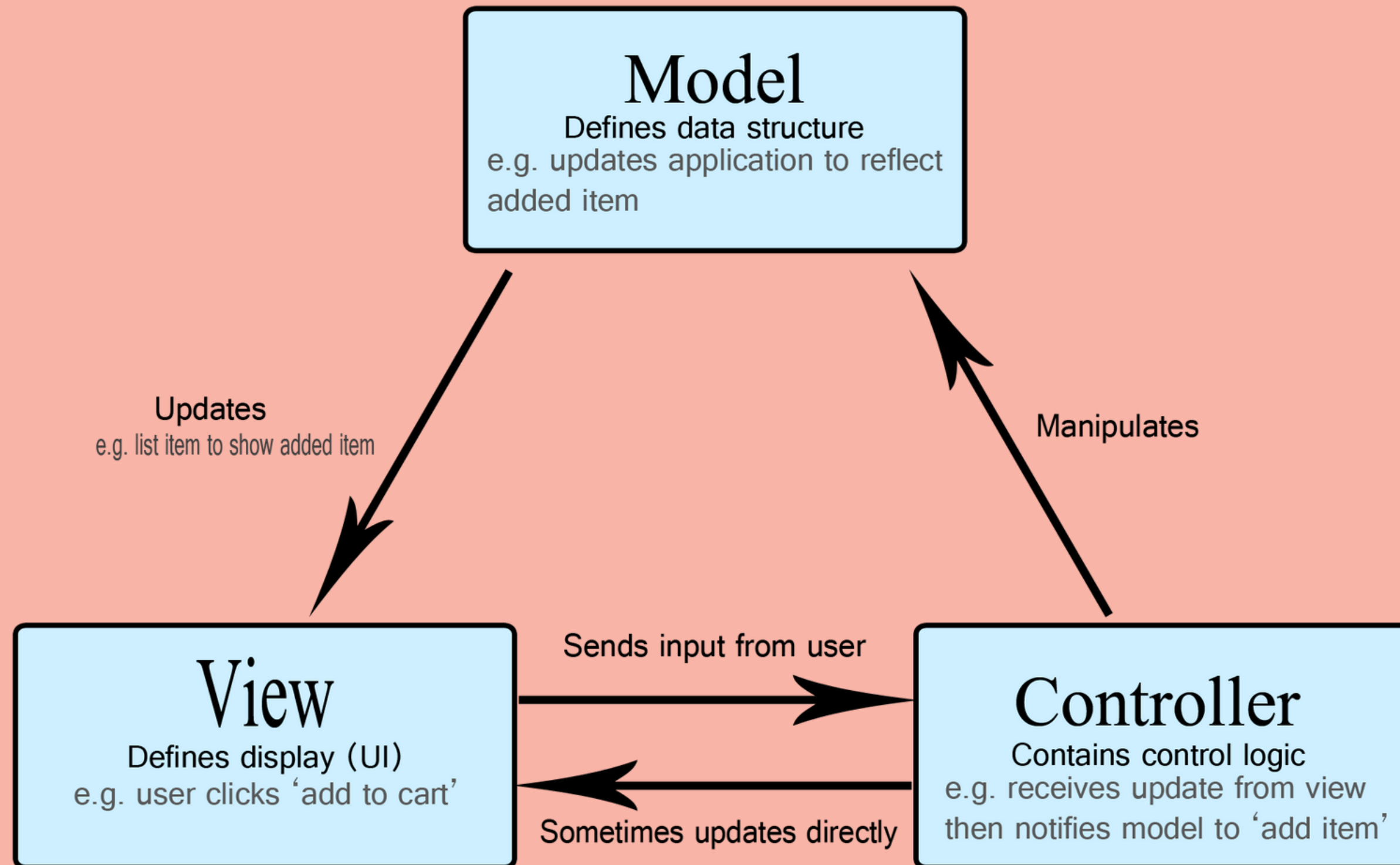
```
catch(Exception $e)
{
    //affichage d'une exception en cas d'erreur
    die('Erreur : '.$e->getMessage());
}
```

Correction Exercice BDD n°1

```
<table>
  <thead>
    <tr>
      <th>Nom</th>
      <th>Contenu</th>
    </tr>
  </thead>
  <tbody>
    <?php
      // Connexion à la BDD
      $bdd = new PDO('mysql:host=localhost;dbname=articles', 'root', '', array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
      $bdd->exec("set names utf8");
      $reponse = $bdd->query('SELECT * FROM article');
      //boucle pour parcourir et afficher le contenu de chaque ligne de la table
      while ($donnees = $reponse->fetch()) {
        //affichage des données d'une colonne du résultat de la requête par son nom d'attribut
        echo "<tr>
          <td>" . $donnees['nom_article'] . "</td>
          <td>" . $donnees['contenu_article'] . "</td>
        </tr>"; {
      }
    <?>
  </tbody>
</table>
```

Modèle MVC

95





Introduction MVC

Jusqu'à présent, nous avons inclus nos codes PHP de façon non structurée.

Afin de mieux organiser notre code, et faciliter les maintenances et mäj, nous allons l'organiser selon le design pattern MVC

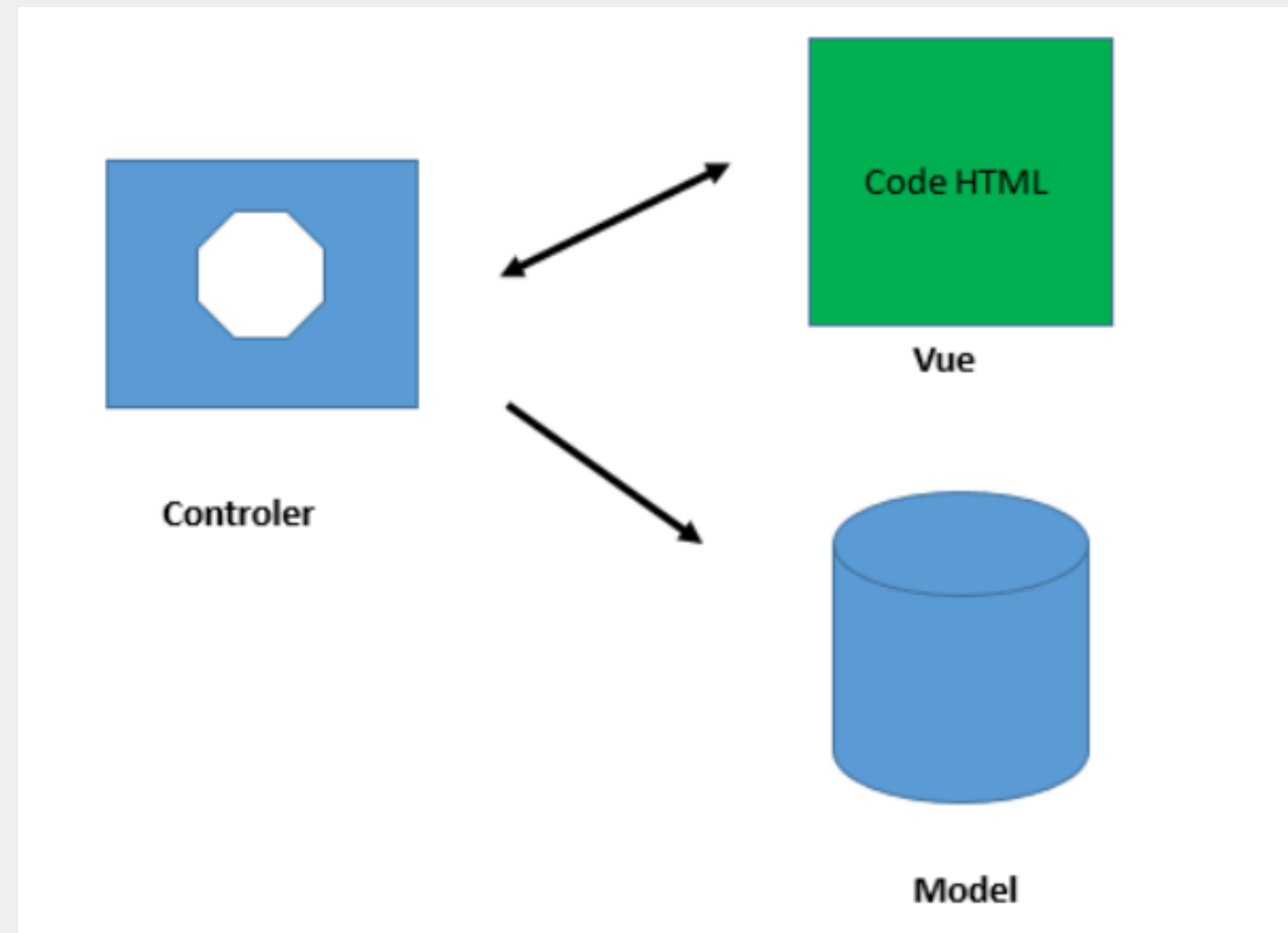
Au sein de ce modèle, chaque fichier aura un rôle précis



Présentation MVC

- **Modèle** : Ensemble du code permettant l'accès aux données, afin de les préparer pour le contrôleur.
(Ensemble du code concernant le SQL)
- **Vue** : Ensemble du code concernant la partie d'affichage
(Ensemble du code contenant du HTML)
- **Contrôleur** : Ensemble du code gérant la logique de notre page, calculs, traitements de données.
Le contrôleur demandera les données au modèle, les traite, et les envoie à la vue.
(Exclusivement du code PHP)

Echanges entre les couches MVC



Exemple MVC : Exercice 1 BDD

Architecture MVC

```
✓ controller
  php controller_article.php
  model
    php connect.php
    php model_article.php
  vue
    php vue_article.php
```


Exemple MVC : Exercice 1 BDD

Le formulaire sera déplacé au sein d'une vue : `vue_article.php`

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ajouter un article</title>
</head>
<body>
  <form action="" method="post">
    <p>saisir le nom de l'article :</p>
    <input type="text" name="nom_article">
    <p>saisir le contenu de l'article :</p>
    <input type="text" name="contenu_article">
    <input type="submit" value="Ajouter">
  </form>
</body>
</html>
```

Exemple MVC : Exercice 1 BDD

La connexion à la BDD se fera dans un modele spécifique : connect.php

```
<?php
//connexion à la bdd
$bdd = new PDO('mysql:host=localhost;dbname=articles', 'root','',
array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
?>
```

Exemple MVC : Exercice 1 BDD

L'insertion des articles dans la BDD se fera dans un modèle : `modele_article.php`

```
try {  
    //Exécution de la requête SQL insert  
    $reponse = $bdd->query('insert into article(nom_article,contenu_article) values("'" . $name . "'", "'" . $content . "'')');  
    echo "ajout de l'article : $name qui a comme contenu : $content";  
} catch (Exception $e) {  
    //affichage d'une exception en cas d'erreur  
    die('Erreur : ' . $e->getMessage());  
}
```


Exemple MVC : Exercice 1 BDD

La partie logique sera déplacé au sein d'un controller : `controller_article.php`

```
<?php
//ajout de la vue
include('../vue/vue_Article.php');
//connexion à la BDD
include('../model/connect.php');
//test existence des champs nom_article et contenu article
if (isset($_POST['nom_article']) and isset($_POST['contenu_article'])) {
    //création des 2 variables qui vont récupérer le contenu des super globales POST
    $name = $_POST['nom_article'];
    $content = $_POST['contenu_article'];
    //ajout du model
    include('../model/model_Article.php');
} else {
    //affichage dans la page html de ce que l'on a enregistré en bdd
    echo '<p>veuillez remplir les champs de formulaire</p>';
}
?>
```



Exercice MVC n°1

Reprendre l'exercice 1 BDD, l'adapter en MVC à l'aide de l'exemple précédent, et y ajouter la partie requêtes préparées, et les parties affichage des articles

Correction Exercice 1 MVC

Architecture MVC

```
✓ controller
  ├── controller_displayArticle.php
  └── controller_setArticle.php
✓ model
  ├── connect.php
  ├── model_displayArticle.php
  └── model_setArticle.php
✓ vue
  ├── vue_displayArticle.php
  └── vue_setArticle.php
```


Correction Exercice 1 MVC

model : connect.php

```
<?php
//connexion à la bdd
$bdd = new PDO('mysql:host=localhost;dbname=articles', 'root','',
array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
?>
```

Correction Exercice 1 MVC

model : displayArticle.php

```
<?php

$reponse = $bdd->query('SELECT * FROM article');
$datas = [];
//boucle pour parcourir et afficher le contenu de chaque ligne de la table
while ($donnees = $reponse->fetch()) {
    //affichage des données d'une colonne du résultat de la requête par son nom d'attribut
    array_push($datas,$donnees['nom_article']);
    array_push($datas,$donnees['contenu_article']);
}

?>
```

Correction Exercice 1 MVC

```
<?php

//METHODE 1
/*try {
    //Exécution de la requête SQL insert
    $reponse = $bdd->query('insert into article(nom_article,contenu_article) values("'" . $name . "'", "'" . $content . "'')');
    echo "ajout de l'article : $name qui a comme contenu : $content";
} catch (Exception $e) {
    //affichage d'une exception en cas d'erreur
    die('Erreur : ' . $e->getMessage());
}*/

//METHODE 2
try
{
    $requete = $bdd->prepare("
        INSERT INTO article (nom_article,contenu_article)
        VALUES (:nom, :contenu)
    ");
    $requete->execute(array(
        ':nom' => $nom,
        ':contenu' => $contenu
    ));
}

catch(Exception $e)
{
    //affichage d'une exception en cas d'erreur
    die('Erreur : ' . $e->getMessage());
}

?>
```

model : setArticle.php

Correction Exercice 1 MVC

vue : `vue_displayArticle.php`

```
<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
</head>

<body>

<table>
<thead>
<tr>
<th>Nom</th>
<th>Contenu</th>
</tr>
</thead>
<tbody>
<?php
    $i=0;
    while(isset($datas[$i+1])){
        echo "
        <tr>
            <td>" . $datas[$i] . "</td>
            <td>" . $datas[$i+1] . "</td>
        </tr>";
        $i+=2;
    }
    ?>
</tbody>
</table>

</body>

</html>
```

Correction Exercice 1 MVC

vue : vue_setArticle.php

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ajouter un article</title>
</head>
<body>
  <form action="" method="post">
    <p>saisir le nom de l'article :</p>
    <input type="text" name="nom_article">
    <p>saisir le contenu de l'article :</p>
    <input type="text" name="contenu_article">
    <input type="submit" value="Ajouter">
  </form>
</body>
</html>
```

Correction Exercice 1 MVC

controller : controller_displayArticle.php

```
<?php

//connexion à la BDD
include('../model/connect.php');

//ajout du model
include('../model/model_displayArticle.php');

//ajout de la vue
include('../vue/vue_displayArticle.php');

?>
```


Correction Exercice 1 MVC

controller : controller_setArticle.php

```
<?php
//ajout de la vue
include('../vue/vue_setArticle.php');
//connexion à la BDD
include('../model/connect.php');
//test existence des champs nom_article et contenu article
if (isset($_POST['nom_article']) and isset($_POST['contenu_article'])) {
    //création des 2 variables qui vont récupérer le contenu des super globales POST
    $name = $_POST['nom_article'];
    $content = $_POST['contenu_article'];
    //ajout du model
    include('../model/model_setArticle.php');
} else {
    //affichage dans la page html de ce que l'on a enregistré en bdd
    echo '<p>veuillez remplir les champs de formulaire</p>';
}
?>
```