

Initiation à la programmation web

Badr Benmammour

Plan

➤ Introduction

- ✓ L'architecture web
- ✓ LAMP : **L**inux **A**pache **M**ySQL **P**HP

➤ Partie 1 : Les bases de PHP

- ✓ Les variables
- ✓ Les fonctions
- ✓ Instruction conditionnelle
- ✓ Les boucles
- ✓ Les tableaux

➤ Partie 2 : La base de données

- ✓ Présentation de MySQL.
- ✓ Connexion à la BDD, le choix de la base, la déconnexion, créer une base de données, supprimer une base de données.
- ✓ La création de tables, la suppression, la modification.
- ✓ Les requête SQL

Introduction

HTML vs PHP

- Une page qui ne contient que du HTML possède l'extension ".html".
- Une page qui contient du code PHP, elle a l'extension ".php".
- Autres extensions .php3 ou .php5.
- La version actuelle de PHP est la v5.
- Le HTML est pratique, mais il est limité.
- A l'aide de PHP, vous pourrez réaliser bien plus de choses pour votre site web.
 - ✓ Un forum, où tout le monde peut discuter,
 - ✓ Un compteur de visiteurs,
 - ✓ ...
- Le PHP vous permet de créer **des pages web dynamiques**.

HTML vs PHP

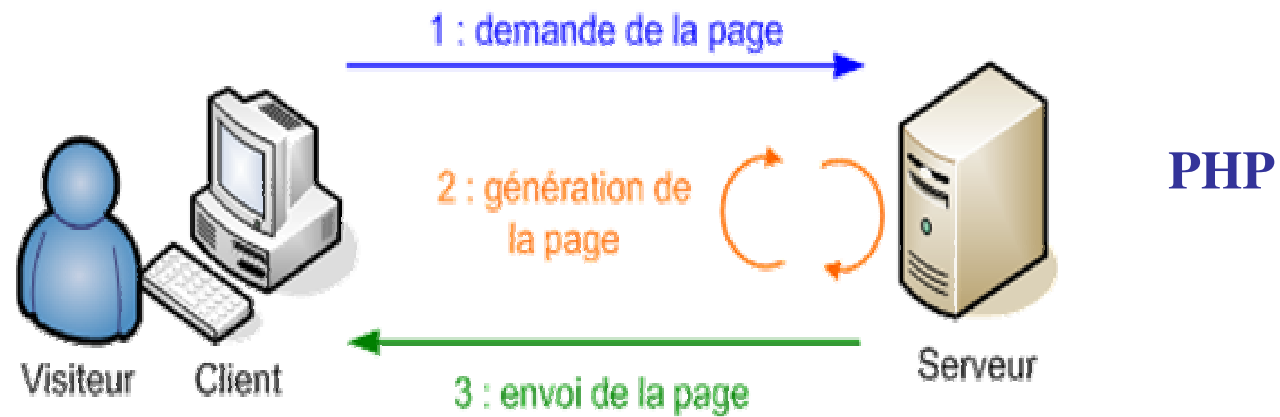
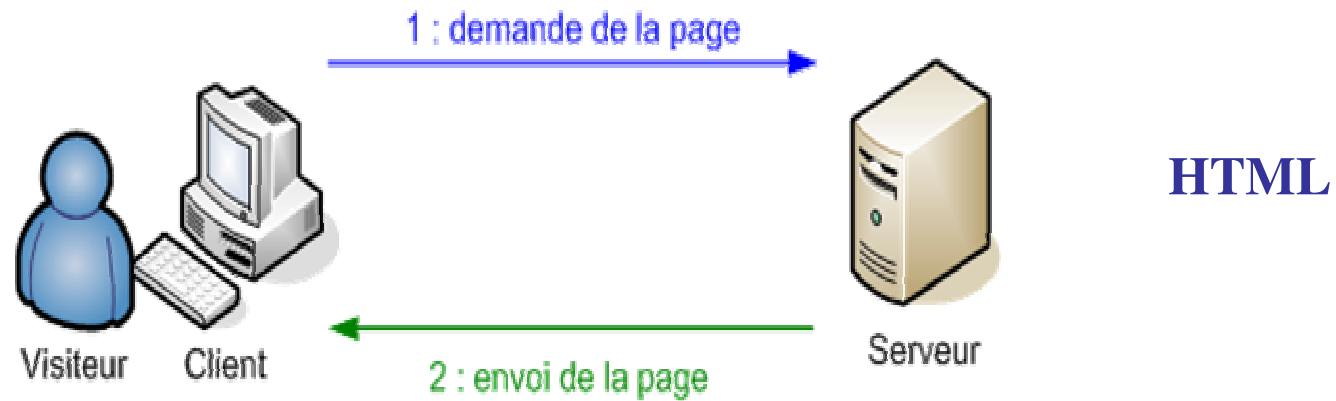
Code HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML 1.0 Strict//EN" "http://www.w3.org/
TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/
xhtml" xml:lang="fr" lang="fr">
  <head>
    <title>Bienvenue sur mon site !</title>
    <meta http-equiv="Content-Type"
    content="text/html; charset=iso-8859-1" />
  </head>
  <body>
    <p>
      Bienvenue sur mon site web !<br />
      Cliquez <a href="http://www.siteduzero.com/">
      ici</a> pour entrer !
    </p>
  </body>
</html>
```

Code PHP

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/
xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="fr" lang="fr">
  <head>
    <title>Bienvenue sur mon site !</title>
    <meta http-equiv="Content-Type"
    content="text/html; charset=iso-8859-1" />
  </head>
  <body>
    <p>
      Bienvenue sur mon site web !<br />
      <?php echo("Vous êtes le visiteur n°" . $nbre_visiteurs); ?>
      <br />
      Cliquez <a href="http://www.siteduzero.com">
      ici</a> pour entrer !
    </p>
  </body>
</html>
```

HTML vs PHP



HTML vs PHP

- Les ordinateurs des clients ne savent pas lire le code PHP : ils ne connaissent que le HTML.
- C'est donc au serveur de transformer le code PHP en HTML.
- Le PHP demande au serveur d'effectuer des actions (**des instructions**) :
 - ✓ Donner l'heure,
 - ✓ Le nombre de personnes connectées sur le site,
 - ✓ ...
- Ce genre de choses était impossible en HTML.

LAMP

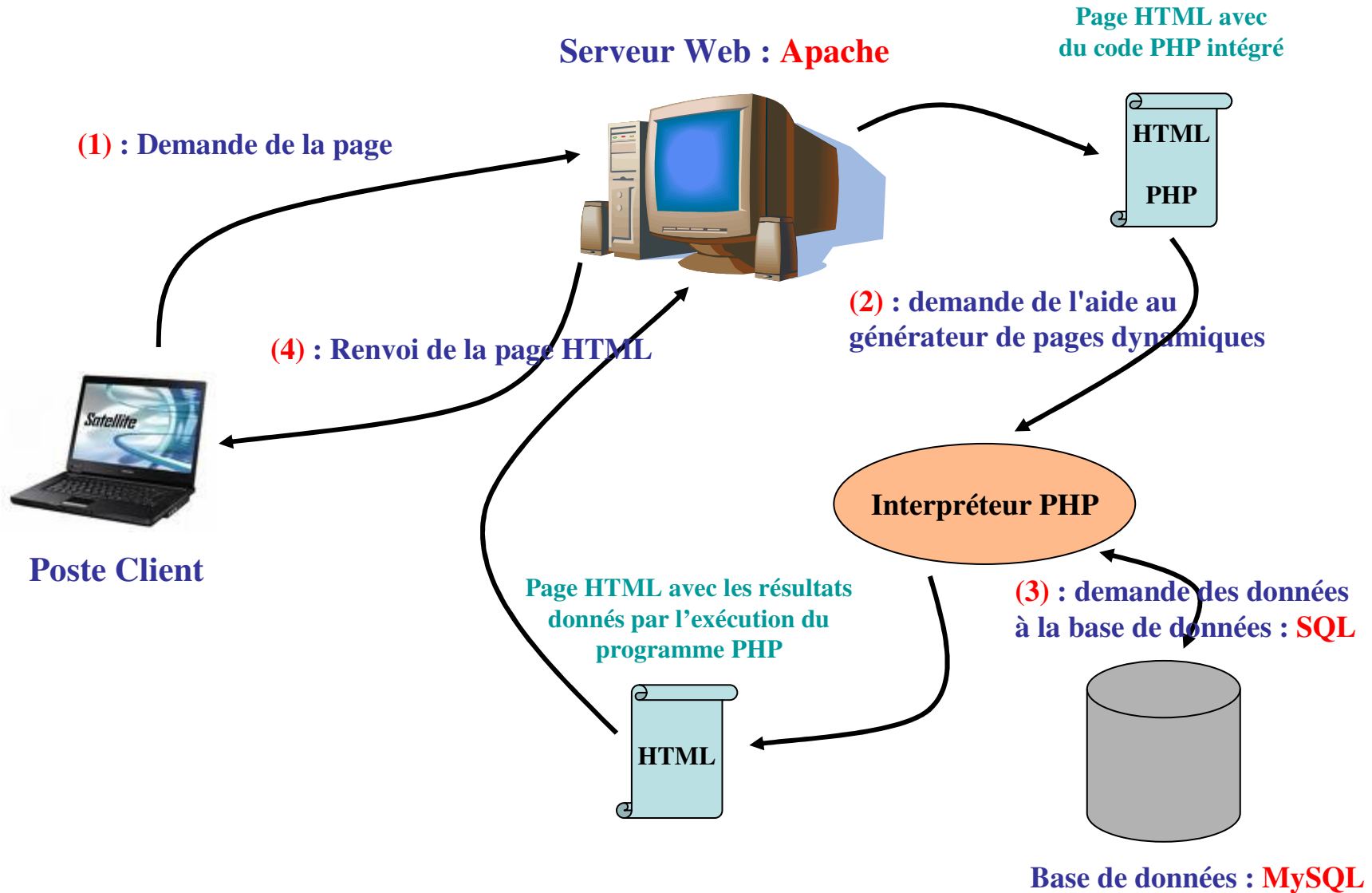
LAMP est un acronyme désignant un **ensemble de logiciels libres** permettant de construire **des serveurs de sites Web**.

- ✓ **Linux** : le système d'exploitation.
- ✓ **Apache** : le serveur Web (l'ensemble des matériels, logiciels et liaisons sur lequel sont implémentés les sites WEB et qui permettent de les rendre consultables sur le WEB).
- ✓ **MySQL** : le serveur de base de données.
- ✓ **PHP** : le langage de script.

➤ **Variantes :**

- ✓ **WAMP** : Windows Apache MySQL PHP.
- ✓ **MAMP** : Macintosh Apache MySQL PHP.
- ✓ **SAMP** : Solaris Apache MySQL PHP.

L'architecture web



Partie 1 : Les bases de PHP

Code PHP

```
<?php // Code PHP ligne 1  
// Code PHP ligne 2  
// Code PHP ligne 3  
// Code PHP ligne 4  
?>
```

<?php ?>

Pour indiquer que vous écrivez **un commentaire**, vous devez taper **2 slash** : **//**. Tapez ensuite votre commentaire.

Afficher du texte

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
    <title>Notre première instruction : echo</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
  </head>
  <body>
    <h2>Affichage de texte avec PHP</h2>

    <p>
      Cette ligne a été écrite entièrement en (x)HTML.<br />
      <?php echo "Celle-ci a été écrite entièrement en PHP."; ?>
    </p>
  </body>
</html>
```

Afficher du texte

```
<?php echo "Celle-ci a été écrite entièrement en PHP."; ?>
```

- ✓ Le code PHP est dans la balise **<?php ?>**
- ✓ **echo** : c'est l'instruction, le mot qui donne un ordre à l'ordinateur, ici, l'ordre est "Afficher le texte".
- ✓ Ensuite, entre guillemets, il y a le texte à afficher.
- ✓ Enfin, la ligne finit par un point-virgule ;.
- ✓ Ce signe doit être placé à la fin de chaque instruction.

Les variables

- Une information stockée en mémoire **temporairement**.
- En PHP, la variable (l'information) existe tant que la page est en cours de génération.
- Dès que la page PHP est générée, toutes les variables sont supprimées de la mémoire car elles ne servent plus à rien.
- Ce n'est donc pas un fichier qui reste stocké sur le disque dur mais une petite information temporaire.
- On peut mettre 3 sortes de données différentes dans une variable :
 - ✓ Texte,
 - ✓ Nombres,
 - ✓ Booléens.

Les variables

```
<?php  
$pseudo_du_visiteur = "Joseph"; // texte,  
  
?>
```

```
<?php  
$nombre_de_freres = 3; // nombres,  
  
?>
```

```
<?php  
$nombre_positif = true;  
$nombre_négatif = false; // booléens.  
  
?>
```

Le symbole Dollar (\$) : il précède toujours le nom d'une variable.

Afficher la valeur d'une variable

```
<?php
$pseudo_du_visiteur = "Joseph";
echo "Bonjour $pseudo_du_visiteur !";
?>
```

Des calculs simples sur les variables:

```
<?php
$nombre = 10;
$resultat = ($nombre + 5) * $nombre; // $resultat prend la valeur 150
?>
```

```
<?php
$nombre = 2 + 4; // $nombre prend la valeur 6
$nombre = 5 - 1; // $nombre prend la valeur 4
$nombre = 3 * 5; // $nombre prend la valeur 15
$nombre = 10 / 2; // $nombre prend la valeur 5
?>
```

Interaction avec un formulaire

➤ La méthode POST

source.php

```
<p>
    Veuillez taper votre prénom :
</p>
<form action="cible.php" method="post">
<p>
<input type="text" name="prenom" /> <input type="submit" value="Valider" />
</p>
</form>
```

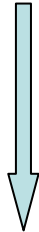
cible.php

```
<p>Bonjour !</p>
<p>Je sais comment tu t'appelles. Tu t'appelles <?php echo $_POST['prenom']; ?>
!</p>
<p>Si tu veux changer de prénom, <a href="source.php">clique ici</a> pour revenir
à source.php</p>
```

La méthode POST

Veillez taper votre prénom :

Valider



Bonjour !

Je sais comment tu t'appelles. Tu t'appelles Badr !

Si tu veux changer de prénom, [clique ici](#) pour revenir à source.php

Interaction avec un formulaire

➤ La méthode GET

source.php

```
<p>
  Veuillez taper votre prénom :
</p>
<form action="cible.php" method="get">
<p>
  <input type="text" name="prenom" /> <input type="submit" value="Valider" />
</p>
</form>
```

cible.php

```
<p>Bonjour !</p>
<p>Je sais comment tu t'appelles. Tu t'appelles <?php echo $_GET['prenom']; ?>
!</p>
<p>Si tu veux changer de prénom, <a href="source.php">clique ici</a> pour revenir
à source.php</p>
```

La méthode GET

- La différence avec la méthode POST est que GET passe les variables à la page web "cible.php" en les ajoutant à la fin de l'URL.

Après avoir cliqué sur Valider, l'URL aura ceci ajouté à la fin:

.../cible.php?prenom=valeur

Plusieurs variables:

http://www.monsite.com/infos.php?jour=27&mois=07&annee=2003&titre=Informations

- Le point d'interrogation "?" dit au navigateur que les **objets suivants sont des variables.**

Les fonctions

➤ Exemple:

```
<?php
```

```
// Ci-dessous, la fonction qui calcule la surface d'un cercle
```

```
function SurfaceCercle($rayon)
```

```
{
```

```
    $surface = $rayon * $rayon * 3.14 ; // calcul de la surface
```

```
    return $surface; // indique la valeur à renvoyer, ici la surface
```

```
}
```

```
$surface = SurfaceCercle (3);
```

```
echo "La surface d'un cercle de rayon 3 est de $surface";
```

```
?>
```

Instruction conditionnelle

- Afficher des choses différentes en fonction de certaines données.
 - ✓ Si c'est le matin, vous voudrez dire "bonjour" à votre visiteur
 - ✓ Si c'est le soir il vaudrait mieux dire "bonsoir".

- Les symboles à connaître :

Symbole	Signification
==	Est égal à
>	Est supérieur à
<	Est inférieur à
>=	Est supérieur ou égal à
<=	Est inférieur ou égal à
!=	Est différent de

Instruction conditionnelle

<?php

```
if (condition)
{
... (autant d'instructions que nécessaire)
}
else
{
... (autant d'instructions que nécessaire)
}
```

?>

Des conditions multiples:

AND ou &&

OR ou ||

Instruction conditionnelle

```
<?php
    if (condition1)
    {
        ... (autant d'instructions que nécessaire)
    }
    elseif (condition2)
    {
        ... (autant d'instructions que nécessaire)
    }
    else
    {
        ... (autant d'instructions que nécessaire)
    }
?>
```

Instruction conditionnelle: Le cas des booléens

<?php

```
if ($autorisation_entrer == true)
```

```
{
```

```
    echo "Bienvenue";
```

```
}
```

```
elseif ($autorisation_entrer == false)
```

```
{
```

```
    echo "T'as pas le droit d'entrer !";
```

```
}
```

?>

Astuce

Code 1:

```
<?php
    if ($variable == 23)
    {
        echo "<B>Bravo !</B>";
    }
?>
```

Code 2:

```
<?php
    if ($variable == 23)
    {
        ?>
        <B>Bravo !</B>
        <?php
    }
?>
```

L'instruction switch

<?php

switch (\$note) // la variable

{

case 5: // dans le cas où \$note vaut 5

echo "Tu es très mauvais";

break;

case 7:

echo "Tu es mauvais";

break;

case 12:

echo "Tu es assez bon";

break;

case 16:

echo "Tu te débrouilles très bien !";

break;

default:

echo "Désolé, je n'ai pas de message à afficher pour cette note";

}

?>

Les boucles

➤ While

<?php

while (condition)

{

// instructions à exécuter dans la boucle

}

?>

Les boucles

➤ For

<?php

```
for ($nombre_de_lignes = 1; $nombre_de_lignes <= 100;  
$nombre_de_lignes++)
```

```
{
```

```
echo "Ceci est la ligne n°$nombre_de_lignes<br />";
```

```
}
```

?>

Les tableaux (array)

➤ Tableaux numérotés :

Un array numéroté commence toujours à la case n°0 !

Le tableau, contenu de la variable `$prenoms`

Numéro	Valeur
0	François
1	Michel
2	Nicole
3	Véronique
4	Benoît

```
<?php
```

```
echo $prenoms[3];
```

```
?>
```



Affiche "Véronique"

Création d'un array

<?php

```
// La fonction array permet de créer un array  
$prenoms = array ("François", "Michel", "Nicole", "Véronique",  
"Benoît");
```

?>

<?php

```
for ($numero = 0; $numero < 5; $numero++)  
{  
    echo $prenoms[$numero];  
    echo "<br />";  
}
```

?>

Tableaux associatifs

- Dans un seul array, enregistrer les coordonnées de quelqu'un (nom, prénom, adresse, ville).
- Mettre une "étiquette" devant chaque information.

<?php

// On crée notre array associatif :

```
$coordonnees = array (  
    "Prénom" => "François",  
    "Nom" => "Dupont",  
    "Adresse" => "3, rue du Paradis",  
    "Ville" => "Marseille");
```

// Puis si je veux afficher la ville, je ferai :

```
echo $coordonnees['Ville'];
```

?>

Protéger le contenu d'une page par mot de passe

<?php

```
if (isset($_POST['mot_de_passe'])) // Si la variable existe
{
    // On se crée une variable $mot_de_passe avec le mot de passe entré
    $mot_de_passe = $_POST['mot_de_passe'];
}
else // La variable n'existe pas encore
{
    $mot_de_passe = ""; // On crée une variable $mot_de_passe vide
}
// J'ai choisi le mot de passe "Hbn4t0p"
if ($mot_de_passe == "Hbn4t0p") // Si le mot de passe est bon
{
    // On affiche la page cachée.
}
else // le mot de passe n'est pas bon
{
```

protéger le contenu d'une page par mot de passe

// On affiche la zone de texte pour rentrer le mot de passe.

?>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
```

```
  <head>
```

```
    <title>Page protégée par mot de passe</title>
```

```
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

```
  </head>
```

```
  <body>
```

```
    <p>Veuillez entrer le mot de passe:</p>
```

```
    <form action="protection.php" method="post">
```

```
      <p>
```

```
        <input type="text" name="mot_de_passe" /> <input type="submit" value="Valider" />
```

```
      </p>
```

```
    </form>
```

```
  </body>
```

```
</html>
```

```
<?php
```

```
}
```

```
?>
```