

Introduction au langage PHP

Première partie



XAMPP



Plan



- Introduction
- Pré-requis
- Installation d'Xampp
- PHP : premier pas
- Les variables et les types
- Les opérateurs arithmétiques et logiques
- Les structures de contrôles
- Les structures itératives
- Les fonctions

Introduction

The slide features a dark blue background. A horizontal bar with a teal-to-white gradient spans the width of the slide, positioned below the title. Below this bar, there are several thin, parallel horizontal lines in a light blue-grey color, extending from the right side of the slide.

Définition

PHP (Hypertext Preprocessor), créé en 1994 par Rasmus Lerdorf, est un langage de **scripts libre** principalement utilisé pour être **exécuté par un serveur HTTP**, mais il peut fonctionner comme n'importe quel langage interprété de façon locale, en exécutant les programmes en ligne de commande.

Rappel des propriétés

- Simple : KISS & RAD principes , communautaires ...
- Portable : PHP est un langage interprété.
- Open Source : Free, Faster bugs fixes ...
- Extensible : Built-in useful modules, pear ...

Schéma d'exécution d'un script PHP

Client Side



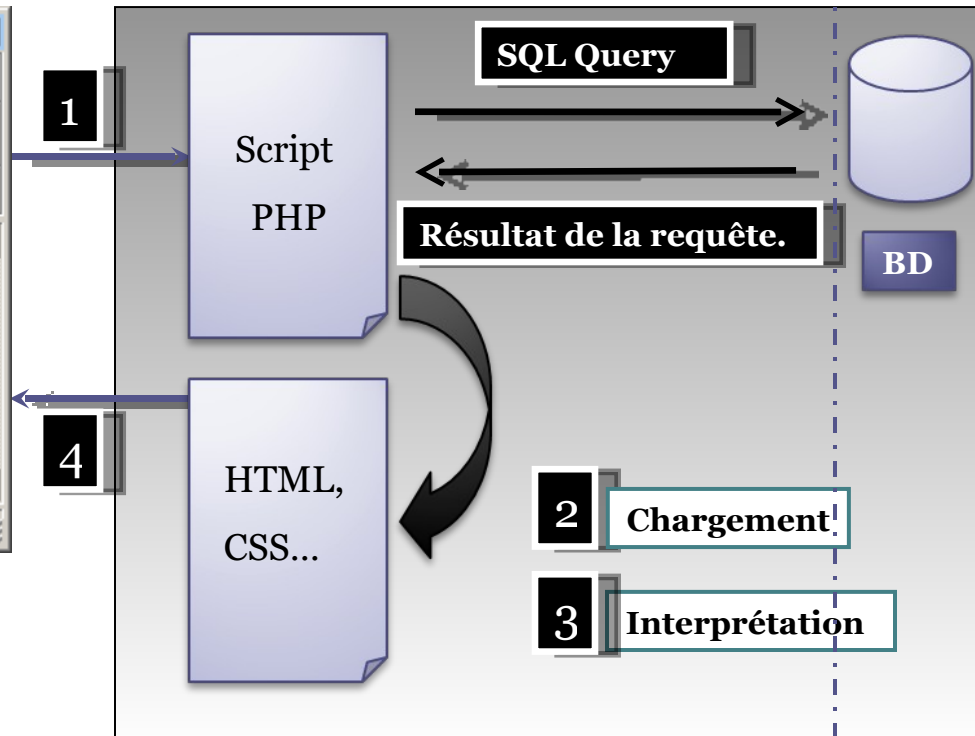
1

Demande helloworld.php.

4

Réponse avec du code statique.

Server Side



Pré-requis

The slide features a dark blue header. Below the title, there is a thick teal horizontal bar. Underneath this bar, the background is white. On the right side of the white area, there are several thin, light blue horizontal lines of varying lengths, creating a stepped or layered effect.

Pré-requis

- Nous avons besoin d'un **serveur web Apache**, un moteur **PHP** et d'un serveur de base de donnée **MySQL** qui soient installés sur nos machines.
- Plusieurs personnes savent par expérience qu'il n'est pas facile d'installer un serveur web Apache et que ça se **complique** si vous voulez y ajouter MySQL, PHP et Perl.



Pré-requis

Heureusement des solutions « all-in-one » existent ;-)



- Easy PHP 
 - <http://www.easyphp.org>
- WAMP 
WAMP SERVER
 - <http://www.wampserver.com/>
- Xampp  XAMPP
 - <http://www.apachefriends.org/fr/xampp.html>

Pré-requis

Manuels et documentations

- **Version en ligne**
 - <http://www.php.net/manual/fr/preface.php>
- **Version hors ligne (téléchargement)**
 - <http://fr3.php.net/download-docs.php>
- **Autre**
 - <http://www.nexen.net/> (recommandée)

Editeurs & IDE

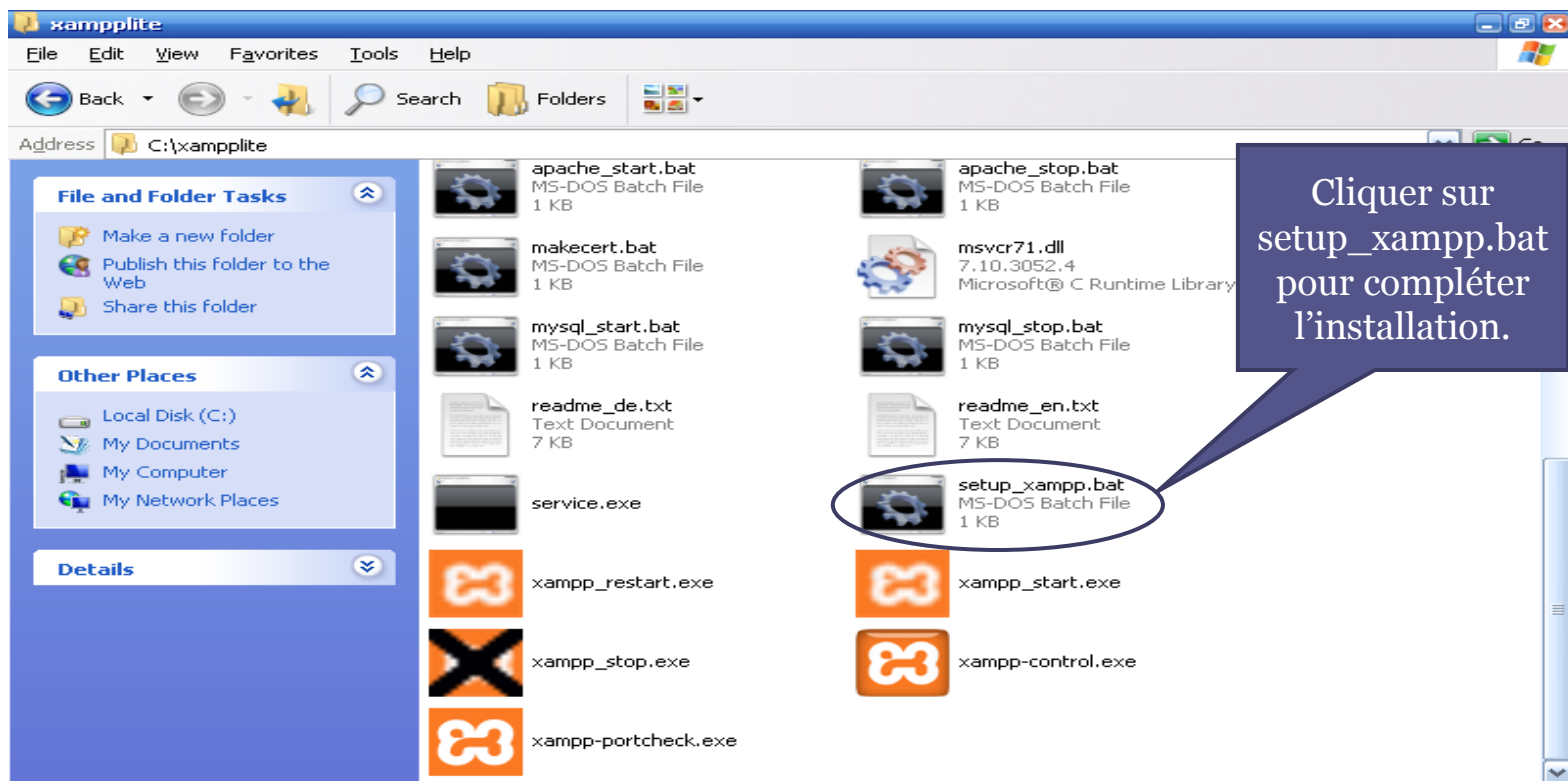
- NotePad, Notepad2, Notepad++, Emacs, Vim ... (free).
- NuSphere PhpED (IDE payant).
- Zend Development Environment (IDE payant).
- Dreamweaver (IDE payant).
- Etc.

Installation du XAMPP

The slide features a dark blue header with the page number '12' in the top right corner. Below the header, the title 'Installation du XAMPP' is written in a large, white, sans-serif font. Underneath the title, there are several horizontal decorative lines: a thick teal line, followed by a thin light blue line, and then two more thin light blue lines.

Premier pas avec XAMPP

- Télécharger et installer XAMPP (Xampplite).



<http://www.apachefriends.org/fr/xampp.html>

PHP : premier pas

The slide features a dark blue header with the title 'PHP : premier pas' in white. Below the header, there is a white area with decorative horizontal lines in teal and light blue.

Syntaxe générale du langage

```
<?php
```

```
//Corps du script
```

```
?>
```

Il faut enregistrer notre document sous l'extension **.php** pour qu'il sera interprété par le serveur comme étant un document php.

Bien que les balises `<?` et `?>` puissent être également utilisées sur une configuration par défaut, elles sont peu recommandées car elles dépendent de la configuration du serveur.

Le Hello World

```
<html>
<head>
<title> Test page </title>
</head>
<body>
<?php
    echo "<h2>Hello World :)</h2>";
?>
</body>
</html>
```

Remarque : PHP peut être imbriqué avec du HTML.

Les commentaires

*Un commentaire permet de vous y retrouver dans votre code.
Ce texte est ignoré durant la génération de la page.*

- *Single line comment : // Texte ou # Texte*
- *Multi line comment : /* Texte */*

```
<?php
echo "Ceci est un test"; // Ceci est un commentaire sur une ligne comme en C.
                        /* Ceci est un commentaire sur plusieurs lignes,
                           comme en C .                               */
echo "Ceci est encore un test";
echo "Enfin, le test final"; # Ceci est un commentaire comme en Shell Unix
?>
```

Les variables et les types

The title is positioned above a series of horizontal lines. The first line is a solid teal color. Below it are three thin, light blue lines that are slightly offset to the right, creating a stepped or layered effect.

Les variables

- En PHP, les variables sont représentées par le caractère dollar "\$" suivi du nom de la variable.

\$Nom_de_la_variable

- Le nom est **sensible** à la casse (i.e. \$x != \$X).
- Un nom de variable valide **doit** commencer par une lettre ou un souligné (_), suivi de lettres, chiffres ou soulignés.

Exemples

- \$name **Valide**
- \$_name **Valide**
- \$1name **Non Valide**
- \$_1name **Valide**
- \$NaMe3 **Valide**

Les types

- **PHP ne nécessite pas de déclaration explicite** du type d'une variable.
- Le type d'une variable est déterminé par **le contexte d'utilisation**.

Par exemple, si vous assignez une chaîne de caractères à la variable \$var , \$var devient une chaîne de caractère. Si vous assignez un nombre entier à \$var ,elle devient un entier.

Le nom d'une variable en PHP est simplement une étiquette.

Les types

1 - Booléens

C'est le type le plus simple. Un booléen exprime **une valeur de vérité**. Il peut prendre comme valeur soit **TRUE** soit **FALSE**.

Exemple

```
<?php
```

```
$foo = True; // assigne la valeur TRUE à la variable $foo
```

```
?>
```

Les types

2 – Entiers

- Les entiers peuvent être spécifiés en base **décimale** (base 10), en **hexadécimale** (base 16) ou **octale** (base 8).
- Les entiers peuvent être **optionnellement** précédés par le signe plus ou moins (- ou +).
- Pour utiliser la notation octale, vous devez **préfixer le nombre avec un zéro**; pour utiliser la notation hexadécimale, vous devez **préfixer le nombre avec 0x**.

Les types

<?php

`$a = 1234; // nombre entier en base 10`

`$a = -123; // nombre entier négatif`

`$a = 0123; // nombre entier en base 8, octale
// (équivalent à 83 en base 10)`

`$a = 0x1A; // nombre entier en base 16, hexa.
// (équivalent à 26 en base 10)`

?>

Les types

3 – Les chaînes de caractère

Les chaînes de caractères sont des séquences de caractères. En PHP, un caractère est un octet et il y en a 256 de possibles.

4 – NULL

La valeur spéciale NULL représente l'absence de valeur. Une variable avec la valeur NULL n'a pas de valeur.

Exemple :

```
<?php
    $firstName = "Jonathan";
    $lastName = "Miller";
    $fullName = $firstName." ".$lastName;
    echo strtoupper($fullName),"<br>";
    // JONATHAN MILLER
    echo substr($firstName, 0, 3),"<br>";
    //Jon
?>
```

Les types

5 - Les tableaux

- Un tableau PHP est en fait une association ordonnée .
- Une association est un type qui fait correspondre des valeurs à des clés .
- Un tableau array peut être créé avec la fonction array .
- Cette fonction prend en argument des structures **key => value** , séparées par des virgules. On parle alors des **tableaux associatifs** .

Les types

Exemple : Tableau classique

```
<?php  
$array = array(1, 2, 3, 4, 5);  
echo $array[0]; // 1  
?>
```

Exemple : Tableau associatif

```
<?php  
$arr = array("foo" => "bar", 12 => true);  
echo $arr["foo"]; // bar  
echo $arr[12]; // 1  
?>
```

Les types

```
<?php
// Ce tableau est identique à
array(5 => 43, 32, 56, "b" => 12);
// Celui ci
array(5 => 43, 6 => 32, 7 => 56, "b" => 12);
?>
```

Les types

```
<?php
$arr = array(5 => 1, 12 => 2);
$arr[] = 56;           // Ceci revient à $arr[13] = 56;

$arr["x"] = 42;        // Ceci ajoute un nouvel élément avec
                        // l'index "x"

unset($arr[5]);         // Ceci efface un élément du tableau
unset($arr);            // Ceci efface tout le tableau
?>
```

Les conversions des types

```
$a = 5.5; // $a contient 5.5 (float).  
$b = (int) 5.5; // $a contient 5 (int).
```

```
<?php  
var_dump(25/7); // float(3.5714285714286)  
var_dump((int) (25/7)); // int(3)  
var_dump(round(25/7)); // float(4)  
?>
```

var_dump(): renvoie le type et le contenu de la variable.

Conversion automatique

\$a = 5 + "5" ;

\$a = 5 + "5abc« ;

\$a = 026;

\$a = 059638;

\$a = 5 + "1.5" ;

•\$a contient 10

•\$a contient 10

•\$a contient 22

•\$a contient 5

•\$a contient 6.5

La fonction echo

La fonction **echo** permet d'afficher un texte qui se trouve entre “ “ (avec interpolation) ou ' ' (sans interpolation).

```
<?php
echo 'Ceci est une chaîne simple';
// affiche : Ceci est une chaîne simple
echo 'Arnold a coutume de dire : "I\'ll be back"';
// affiche : 'Arnold a coutume de dire "I'll be back"
echo 'Les variables ne seront pas $afficher $ici';
// affiche : Les variable ne seront pas $afficher $ici
?>
```

Les opérateurs arithmétiques et logiques

The slide features a dark blue header with the title in white. Below the header, there are several horizontal decorative bars: a thick teal bar, followed by a thinner light blue bar, and then two thin white lines.

Les opérateurs arithmétiques et logiques

• ++, --	incrémentation/décrémentation
• !	Négation
• *, /, %	arithmétique
• +, -	arithmétique
• .	concaténation des chaînes de caractères
• <, <=, >, >=	comparaison
• ==, !=	égalité et différence
• AND ou &&	Et logique
• OR ou	Ou logique

Les structures de contrôles

The title is positioned above a series of horizontal lines. The first line is a solid teal color. Below it are three thin, parallel light blue lines. The bottom half of the slide is a solid white background.

Les structures de contrôles

1 – L'instruction IF

Syntaxe

```
<?php
    if (conditional test)
    {
        do this;
    }
?>
```

```
<?php
    if (conditional test)
    {
        do this;
    }
    else
    {
        do this;
    }
?>
```

Les structures de contrôles

Exemple 1:

```
<?php
if ($a > $b) {
    print "a est plus grand que b";
}
?>
```

Exemple 2:

```
<?php
if ($a > $b) {
    echo "a est plus grand que b";
} else {
    echo "a est plus petit que b";
}
?>
```

Les structures de contrôles

2 – L'instruction SWITCH

Syntaxe

```
<?php
    switch (condition variable)
    {
        case possible result #1:
            do this;

        ...
        case possible result #n:
            do this;
        case default;
            do this;
    }
?>
```

Les structures de contrôles

Exemple 1

```
if ($i == 0) {  
    echo "i égale 0";  
} elseif ($i == 1) {  
    echo "i égale 1";  
} elseif ($i == 2) {  
    echo "i égale 2";  
}
```



```
switch ($i) {  
    case 0:  
        echo "i égale 0";  
        break;  
    case 1:  
        echo "i égale 1";  
        break;  
    case 2:  
        echo "i égale 2";  
        break;  
}
```


Les structures de contrôles

Remarque

Le Switch peut également fonctionner avec les chaînes de caractère (contrairement à certains langages comme le langage C).

```
<?php
switch ($i) {
case "tarte":
    echo "i est une tarte";
    break;
case "glace":
    echo "i est une glace";
    break;
case "gateau":
    echo "i est un gateau";
    break;
}
?>
```

Les structures itératives

The slide features a dark blue header with the page number '42' in the top right corner. The title 'Les structures itératives' is centered in a large, white, sans-serif font. Below the title, there are several horizontal decorative lines: a thick teal line, followed by a thin light blue line, and then two thin white lines.

Les structures itératives

1 – La boucle While()

Syntaxe

```
<?php
    while (condition is true)
    {
        do this;
    }
?>
```

Les structures itératives

Exemple

```
<?php
// define number and limits for multiplication tables
$num = 11;
$upperLimit = 10;
$lowerLimit = 1;
// loop and multiply to create table
while ($lowerLimit <= $upperLimit)
{
    echo "$num x $lowerLimit = " . ($num * $lowerLimit);
    $lowerLimit++;
}
?>
```

Les structures itératives

2 – La boucle do()

Syntaxe

```
<?php
    do
    {
        do this;
    }
    while (condition is true)
?>
```

Rq : Avec la boucle do, on est sûr que les instructions sont exécutées, au moins, une fois.

Les structures itératives

3 – La boucle For()

Syntaxe

```
<?php
    for (initialize counter; conditional test; update counter)
    {
        do this;
    }
?>
```

Les structures itératives

Exemple typique

```
<?php  
    for ($x = 2; $x <= 100; $x++)  
    {  
        echo "$x <br>";  
    }  
?>
```

Les structures itératives

Sachez que ça existe !

```
/* exemple 1 */  
for ($i = 1; ; $i++)  
{  
    if ($i > 10) {  
        break;  
    }  
    echo $i;  
}
```

```
/* exemple 2 */  
$i = 1;  
for (; ; ) {  
    if ($i > 10) {  
        break;  
    }  
    echo $i;  
    $i++;  
}
```

```
/* exemple 3 */  
for ($i = 1; $i <= 10; print $i, $i++);
```


Les structures itératives

4 – La boucle Foreach()

Syntaxe

```
<?php
    foreach (array_expression as $value)
    {
        do this;
    }
?>
```

```
<?php
    foreach (array_expression as $key => $value)
    {
        do this;
    }
?>
```

Les structures itératives

Exemples

```
$a = array (  
  "un" => 1,  
  "deux" => 2,  
  "trois" => 3,  
  "dix-sept" => 17  
);  
foreach ($a as $k => $v) {  
  echo "\$a[$k] => $v.\n";  
}
```

```
$arr = array(1, 2, 3, 4);  
foreach ($arr as $value) {  
  echo " $value <br> ";  
}
```

Les fonctions

The slide features a dark blue header with the title 'Les fonctions' in white. Below the header, there is a white area with decorative horizontal lines in teal and light blue.

Les fonctions

Syntaxe

```
<?php  
function nom_de_la_fonction($arg_1, $arg_2, /* ..., */ $arg_n)  
{  
    do this;  
    return $retval;  
}  
?>
```

Les fonctions

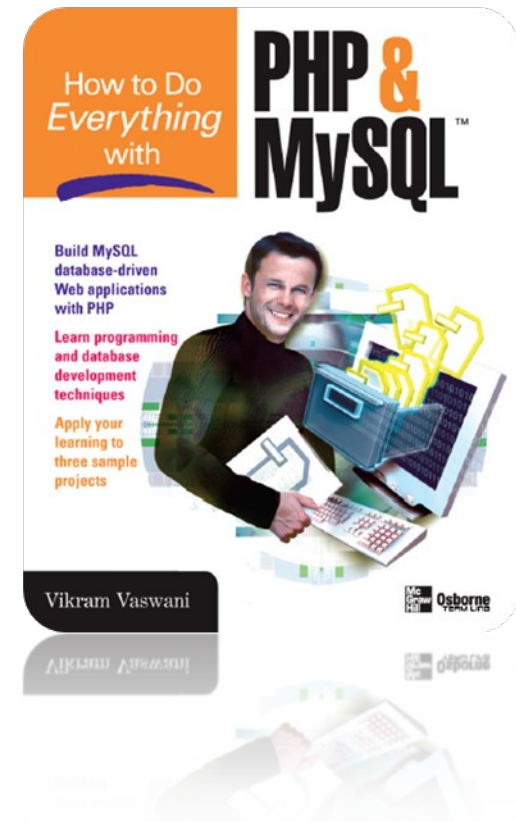
```
<?php
// define a function
function getTriangleArea($base, $height)
{
    $area = $base * $height * 0.5;
    return $area;
}
// invoke a function
echo 'The area of a triangle with base 10 and height 50 is ' .
getTriangleArea(10, 50);
?>
```

Les fonctions

```
<?php
function affiche_bjr()
{
    for ($i = 0; $i < 5; $i++) echo "Bonjour! <br>";
    //Afficher 5 fois « Bonjour! ».
    //Rq : la fonction ne possède ni arguments ni valeur de
    retour.
}
affiche_bjr(); //Appel à la fonction
?>
```

Références

- <http://www.wikipedia.com/>
- <http://www.siteduzero.com/>
- Manuel PHP 4.3.8 – Version Française





Questions ?