OFPPT
La voie de l'avenir

DÉVELOPPEMENT DIGITAL

Développer des Sites Web Dynamiques

Sommaire

CHAPITRE 2

Programmer en PHP

1- Maitriser le langage PHP

2- Traiter les données en PHP

3- Utiliser l'orientée objet en PHP

Maitriser le langage PHP

Ce que vous allez apprendre dans ce chapitre :

- Structure générale d'un script PHP
- Manipulation des Variables/constantes/Affectation
- Manipulation des types de données
- Instructions de sortie
- Contrôles de flux et boucles
- Fonctions sur les chaines de caractères et les dates
- Les Fonctions
- Les Formulaires simples
- Transmission de variables (GET, POST)
- Variables d'environnement
- Redirection entre pages

Maitriser le langage PHP

1- Structure générale d'un script PHP

Balises PHP

- Lorsque PHP traite un fichier, il cherche les balises d'ouverture et de fermeture (**<?php** et **?>**) qui délimitent le code qu'il doit interpréter.
- En version 7, PHP accepte deux syntaxes pour les balises :

  <?php ... ?>

  <? ... ?>
- Avant la version 7, PHP acceptait deux syntaxes supplémentaires pour les balises :

  <script language="php"> ... </script>

  <% ... %>
- Tout ce qui est à l'extérieur de la balise PHP est transmis tel quel au navigateur; seul les instructions PHP sont interprétées par le moteur PHP.

```
<?php
//Commentaire sur une seule ligne, style C++

/*
   Ceci est un
      commentaire
         sur
            plusieurs lignes
*/


# Ceci est un commentaire style shell sur une seule ligne
?>
```

Commentaires

- **//** : permet d'ajouter un commentaire sur une seule ligne
- **#** : permet d'ajouter un commentaire sur une seule ligne
- **/* ... */** : permet d'ajouter un commentaire sur plusieurs lignes

Maitriser le langage PHP

1- Structure générale d'un script PHP

La fonction echo

echo - Affiche une chaîne de caractères
PHP inclus une balise ouvrante echo courte **<?=** qui est un raccourci au code plus verbeux **<?php echo**

```
<?php

 echo 'Ceci est un test';
 echo 'Ceci est un autre test';

?>

<?=

'Et un test final';

?>
```

Séparation des instructions

PHP requiert que les instructions soient terminées par un point-virgule à la fin de chaque instruction.

Maitriser le langage PHP

2- Manipulation des variables/Constantes/Affectation

Les variables essentielles

En PHP, les variables sont représentées par un signe dollar "$" suivi du nom de la variable.

Un nom de variable valide doit respecter les règles suivantes (après le symbole **$**) :

- Le nom est sensible à la casse : $a et $A sont deux variables distinctes.
- Le premier caractère doit être une lettre ou un souligné _
- À partir du deuxième caractère seul les lettres, chiffres ou soulignés sont acceptés.

Exemples :

```
<? php
$var; //nom de variable valide
$Var; //nom de variable valide
$4vars; //nom de variable invalide: commencer avec un nombre n'est pas autorisé
$_var; //nom de variable valide
$état; //nom de variable valide
$éta+; //nom de variable invalide: les caractères spéciaux comme + ne sont pas autorisés
&var Vat; //nom de variable invalide: l'espace n'est pas autorisé
?>
```

Maitriser le langage PHP

2- Manipulation des variables/Constantes/Affectation

Variables prédéfinies

Les variables "**superglobales**" sont disponibles quel que soit le contexte du script.

	$GLOBALS Référence toutes les variables disponibles dans un contexte global	
$_SERVER Variables de serveur et d'exécution	**$_GET** Variables HTTP GET	**$_POST** Variables HTTP POST
$_FILES Variable de téléchargement de fichier via HTTP	**$_REQUEST** Variables de requête HTTP	**$_SESSION** Variables de session
$_ENV Variables d'environnement	**$_COOKIE** Cookies HTTP	**$php_errormsg** Le dernier message d'erreur
$http_response_header En-têtes de réponse HTTP	**$argc** Le nombre d'arguments passés au script	**$argv** Tableau d'arguments passés au script

Maitriser le langage PHP

2- Manipulation des variables/Constantes/Affectation

Portée des variables

- Une variable globale doit être déclarée à l'intérieur de chaque fonction afin de pouvoir être utilisée dans cette fonction.
- Il suffit d'utiliser le mot clé **global** avant la variable pour l'utiliser.
- Une variable statique a une portée locale uniquement, mais elle ne perd pas sa valeur lorsque le script appelle la fonction.
- Il suffit d'utiliser le mot clé **static** avant la variable ou l'attribut ou la méthode pour l'utiliser.

```php
<?php
function test() {
   $var = "variable locale";

   echo '$var dans le contexte global : ' . $GLOBALS[« var"] .
"\n";
   echo '$var dans le contexte courant : ' . $var . "\n";
}

$var = "Exemple de contenu";
test();
?>
```

```php
<?php
function test()
{
  //la variable $i est intialisée unitquemet lors du premier appel à la function
   static $i = 0;
   echo $i;
  // à chaque fois la function est appelée, elle affichera une valeur de $i + 1
   $i++;
}
?>
```

Maitriser le langage PHP

2- Manipulation des variables/Constantes/Affectation

Exemples fonctions de gestion des variables

get_defined_vars
Liste toutes les variables définies

floatval
Convertit une chaîne en nombre à virgule flottante

get_resource_type
Retourne le type de ressource

is_array
Détermine si une variable est un tableau

empty
Détermine si une variable est vide

is_object
Détermine si une variable est de type objet

isset
Détermine si une variable est déclarée et est différente de null

get_resource_id
Retourne un entier identifiant une ressource

unset
Détruit une variable

var_dump
Affiche les informations d'une variable

is_null
Indique si une variable vaut null

print_r
Affiche des informations lisibles pour une variable

var_export
Retourne le code PHP utilisé pour générer une variable

Maitriser le langage PHP

2- Manipulation des variables/Constantes/Affectation

Les constantes

- Une constante est un identifiant (un nom) qui représente une valeur simple.
- Les constantes sont sensibles à la casse.
- Par convention, les constantes sont toujours en majuscule.
- Les constantes peuvent être définies en utilisant le mot clé **const** ou en utilisant la fonction **define()**.
- une constante n'est pas préfixée d'un $
- Pour vérifier qu'une constante est définie, utiliser la fonction **defined()**.
- Si une constante indéfinie est utilisée une Error est renvoyée lancée.
- Contrairement aux constantes définies en utilisant l'instruction define(), les constantes définies en utilisant le mot-clé const doivent être déclarées au plus haut niveau du contexte, car elles seront définies au moment de la compilation.

```
<? Php

const test = 'Bonjour les étudiants !';
echo test; //Affiche Bonjour les étudiants !


?>
```

2- Manipulation des variables/Constantes/Affectation

Les opérateurs d'affectation

- L'opérateur d'affectation le plus simple est le **signe =**
- Il signifie que l'opérande de gauche se voit affecter la valeur de l'expression qui est à droite du signe égal.
- L'affectation par référence se fait grâce au **signe &**

Exemples :

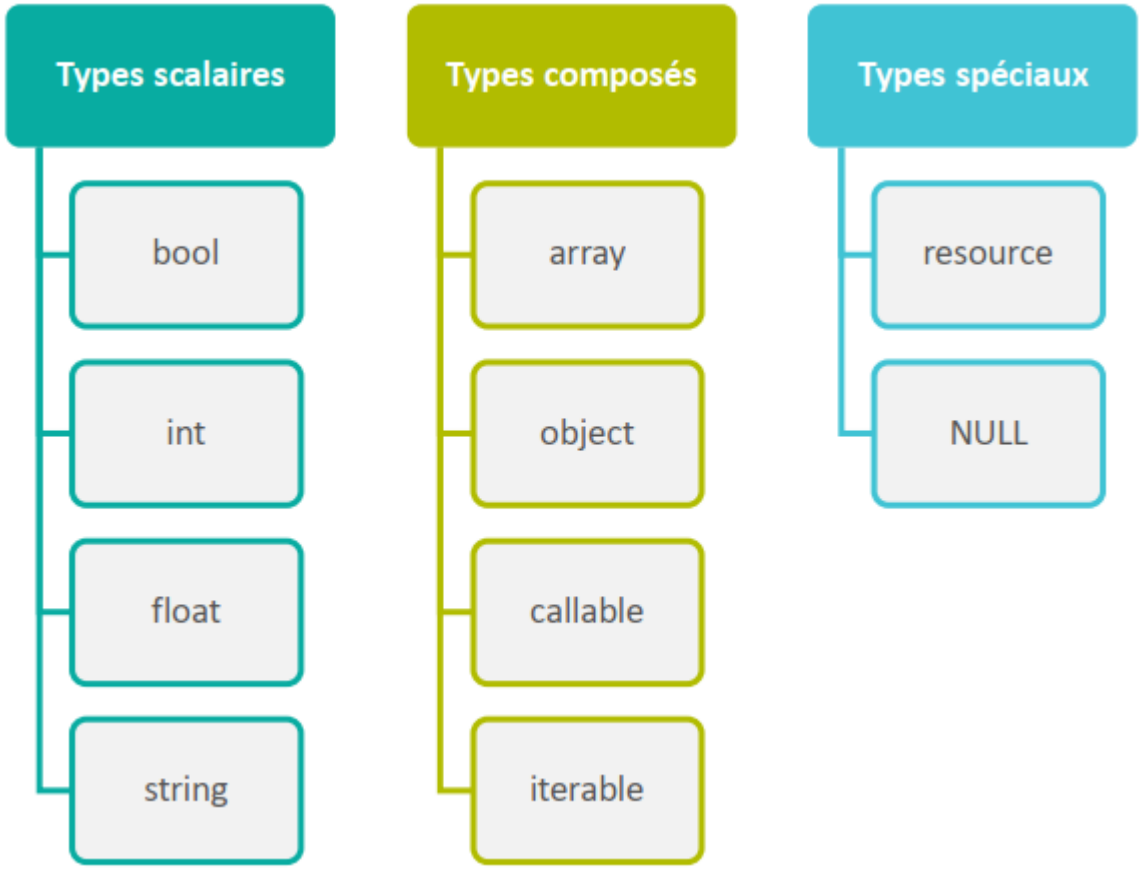
```
<? php
$a = 2; //$a est maintenant égal à 2
$b = 3; //$b est maintenant égal à 3
$c = $a + $b; //$c est maintenant égal à 5
$d = $$c; //$d est maintenant égal à 5

$e = "Bonjour ";
$e .= " les étudiants";  // affecte la valeur "Bonjour les étudiants" à la variable e

$c += $a*2 // $c est maintenant égal à 9 de même que $d est maintenant égal à 9
?>
```

Maitriser le langage PHP

3- Manipulation des types des variables

Introduction

• Les variables PHP pourront stocker différents types de valeurs comme des nombres ou des tableaux. Par abus de langage, on parle des "types variables" de PHP.
• PHP supporte 10 types basiques :



Maitriser le langage PHP

3- Manipulation des types des variables

Type booléen

- Il peut avoir **true** ou **false**.
- **==** est un opérateur qui teste l'égalité et retourne un booléen.
- Pour convertir explicitement une valeur en booléen, utilisez (bool) ou (boolean).
- Lors d'une conversion en booléen, les valeurs suivantes sont considérées comme false :
- false
- l'entier 0, les nombres à virgule flottante 0.0 et -0.0
- la chaîne vide, et la chaîne "0"
- un tableau avec aucun élément
- le type spécial NULL (incluant les variables non définies)

```php
<? php
 $bool_val = (bool)true;
 echo $bool_val;   // Affiche : 1

 $bool_val2 = (bool)false;
 echo $bool_val2;   // N'affiche rien

 $bool_exp1 = (bool)false;
 echo $bool_exp1 ? 'true' : 'false';   // Affiche : false

 $bool_exp2 = (bool)false;
 echo json_encode($bool_exp2);   // Affiche : false

 $bool_exp3 = (bool)false;
 echo var_export($bool_exp3);   // Affiche : false

 $bool_exp4 = (bool)false;
 echo (int)$bool_exp4; // Affiche : 0

 $bool_exp5 = (bool)false;
 var_dump($bool_exp5); // Affiche : bool(false)
```

Maitriser le langage PHP

3- Manipulation des types des variables

Type entier

- Un entier est un nombre appartenant à l'ensemble $\mathbb{Z}$ = {..., -2, -1, 0, 1, 2, ...}.
- Les entiers peuvent être spécifiés en notation:
  - Décimale (base 10)
  - Hexadécimale (base 16) : Pour utiliser la notation hexadécimale, précédez le nombre de 0x
  - Octale (base 8) : Pour utiliser cette notation, précédez le nombre d'un 0 (zéro). À partir PHP 8.1.0, la notation octale peut être précédé avec 0o ou 0O
  - Binaire (base 2) : Pour utiliser la notation binaire, précédez le nombre de 0b.
- L'opérateur de négation peut être utilisé pour désigner un entier négatif.
- À partir de PHP 7.4.0, les entiers littéraux peuvent contenir des underscores (_) entre les chiffres, pour une meilleure lisibilité des littéraux. Ces underscores sont supprimés par le scanner de PHP. Exemple : $a = 1_203_489 // c'est la même chose que $a = 1203489
- Pour convertir explicitement une valeur en un entier, utiliser soit le mot-clé (int), soit (integer).
  - Depuis un booléen : false correspond à 0, et true correspond à 1.
  - Depuis un nombre décimal : le nombre sera arrondi vers zéro.

```php
<?php
// un nombre décimal
$a = 1234;

// un nombre octal (équivalent à 83 en décimal)
$a = 0123;

// un nombre octal (à partir de PHP 8.1.0)
$a = 0o123;

// un nombre hexadecimal (équivalent à 26 en décimal)
$a = 0x1A;

// un nombre binaire (équivalent à 255 en decimal)
$a = 0b11111111;

// un nombre décimal (à partir de PHP 7.4.0)
$a = 1_234_567;
?>
```

Maitriser le langage PHP

3- Manipulation des types des variables

Type nombre à virgules

- Le type « nombre décimal » ou Float en anglais.
- Pour convertir explicitement une valeur en un float, utiliser le mot-clé (float).
- Depuis une chaine de caractères : Si une chaîne est numérique ou numérique
de tête alors elle sera transformée en sa valeur flottante correspondante, sinon elle sera convertie à zéro.
- Depuis un booléen : false correspond à 0, et true correspond à 1
- Depuis d'autres types : la conversion est effectuée d'abord en int puis en float

```php
<?php
$a = 1.234;

$b = 1.2e3;

$c = 7E-10;

$d = 1_234.567; // à partir de PHP 7.4.0

$epsilon = 0.00001;

$f = 1.23456780;
?>
```

Maitriser le langage PHP

3- Manipulation des types des variables

Type chaine de caractères

La façon la plus simple de spécifier une chaîne de caractères est de l'entourer de guillemets simples (le caractère ').

- Pour spécifier un guillemet simple littéral, il doit être échappé à l'aide d'un antislash (\)
- Pour spécifier un antislash littéral, doublez-le (\\)
- Les séquences d'échappement pour les caractères spéciaux ne seront pas interprétées
- Un retour à la ligne dans la chaine de caractère permet d'avoir une nouvelle ligne.

Si la chaîne de caractères est entourée de guillemets doubles (le caractère ") :

- Les noms de variables seront interprétés.
- \n : Fin de ligne (LF ou 0x0A (10) en ASCII)
- \r : Retour à la ligne (CR ou 0x0D (13) en ASCII)
- \t : Tabulation horizontale (HT or 0x09 (9) en ASCII)
- \v : Tabulation verticale (VT ou 0x0B (11) en ASCII)
- \e : échappement (ESC or 0x1B (27) en ASCII)
- \f : Saut de page (FF ou 0x0C (12) en ASCII)
- \\ : Antislash
- \$: Signe dollar
- \" : Guillemet double

```php
<?php
echo 'ceci est une chaîne simple';
echo ‘’une autre chaîne simple ’’;
echo 'Vous pouvez également ajouter des nouvelles lignes
dans vos chaînes
de cette façon';
echo « Des nouvelles lignes \ndans vos chaînes ";
// Affiche : Ahmed a dit : "J'ai un autre exemple ici"
echo 'Ahmed a dit : "J\'ai un autre exemple ici"';
// Affiche : Voulez-vous supprimer C:\*.*?
echo 'Voulez-vous supprimer C:\\*.*?';
// Affiche : Voulez-vous supprimer C:\*.*?
echo 'Voulez-vous supprimer C:\*.*?';
// Affiche : Ceci n'affichera pas \n de nouvelle ligne
echo 'Ceci n\'affichera pas \n de nouvelle ligne';
$traitees = "égales à";
$ici = 2;
// Affiche : Les variables ne seront pas $traitees $ici
echo 'Les variables ne seront pas $traitees $ici';
// Affiche : Les variables ne seront pas égales à 2
echo "Les variables ne seront pas $traitees $ici";
?>
```

Maitriser le langage PHP

3- Manipulation des types des variables

Type tableau

Syntaxe d'un tableau est :

- **clé => valeur**
- La **clé** peut être soit un int, soit une chaîne de caractères. Les nombres à virgule flottante seront aussi modifiés en entier.
- La **valeur** peut être de n'importe quel type.
- Structure de langage **array()**. La syntaxe courte emplace array() par **[]**
- Le tableau prend un nombre illimité de paramètres, chacun séparé par une virgule

```php
<?php
// signifie la 0 => Salut,  1 => Aloha, 2=>Hello
$tableau1 = array("Salut", "Aloha", "Hello");
Echo $tableau1[0]; // Afficher : Salut
// signifie la 0 => Salut,  1 => Aloha, 2=>Hello
$tableau2 = ["Salut", "Aloha", "Hello"];
Echo $tableau2[0]; // Afficher : Salut

$tableau3 = array(
    "0" => "Salut" ,
    "1" => "Aloha" ,
    "2" => "Hello" ,
);
Echo $tableau3[2]; // Afficher : Hello
$tableau4 = [
    "0" => "Salut" ,
    "1" => "Aloha" ,
    "2" => "Hello" ,
];
Echo $tableau4[2]; // Afficher : Hello
?>
```

Maitriser le langage PHP

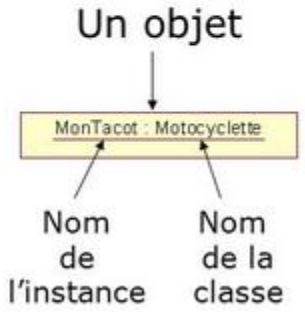
3- Manipulation des types des variables

- **Type objet**

Pour créer un nouvel objet, le mot clé **new** est utilisé afin d'instancier une class.

- Une **class** définit le comportement d'un objet. La classe ne contient aucune donnée.
- Un **objet** est une instance d'une classe qui contient des données.
- Un **membre** est une variable qui appartient à un objet.
- Une **méthode** est une fonction qui appartient à un objet et a accès à ses membres.
- Un **constructeur** est une méthode spécifique qui est exécutée lorsqu'un objet est créé.

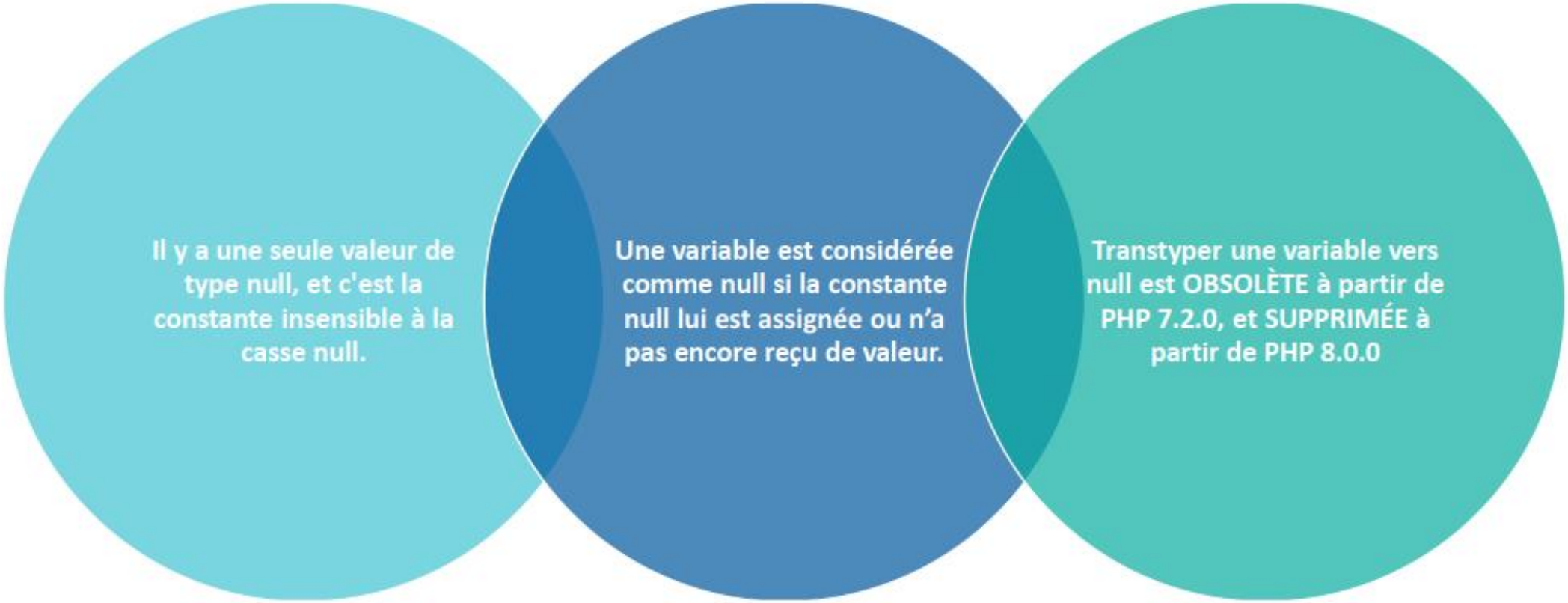
✓En PHP, la déclaration d'un objet revient à instancier une classe avec **new** :




```php
//Déclaration d'une instance :
$MonTacot = new Motocylette;

//Utilisation :
$MonTacot->setCouleur("rouge");
```

3- Manipulation des types des variables

Type NULL



Maitriser le langage PHP

3- Manipulation des types des variables

Transtypage : Modification des types

- Le transtypage permet de convertir le type d'une variable dans un autre type explicite.

```
$variable = (operateur_typage) valeur;
```

- La conversion de type en PHP fonctionne de la même manière qu'en C : le nom du type désiré est écrit entre parenthèses devant la variable à transtyper (" `cast` ").

```php
<?php
$foo = 10;   // $foo est un entier
$bar = (double) $foo;   // $bar est un double
?>
```

Maitriser le langage PHP

3- Manipulation des types des variables

Transtypage : Modification des types

(int), (integer) :
modification en int

(bool), (boolean) :
modification en bool

(float), (double), (real) :
modification en float

(string) :
modification en string

(array) :
modification en array

(object) :
modification en object

4- Instructions de sortie

Envoi vers le navigateur

- Trois fonctions permettent d'envoyer du texte vers le navigateur :
- **echo(string ...$expressions) : void →** Affiche une chaîne de caractères.
- **print(string $expression) : int →** Affiche une chaîne de caractères.
- **printf(string $format, mixed ...$values) : int →** Affiche une chaîne de caractères formatée.

- **Echo :**
- N'a pas de valeur de retour.
- Peut prendre plusieurs arguments (bien que cette utilisation soit rare).
- Est légèrement plus rapide que print.

- **Print :**
- Renvoie la valeur 1, il peut donc être utilisé dans des expressions.
- Peut prendre un seul argument.

- **Le type mixed :**
- La valeur peut être de n'importe quelle valeur

Maitriser le langage PHP

5-Contrôle de flux et boucles

L'instruction if…else

```
if (expression)
  commandes
else
  commandes
```

• Si l'expression vaut true, PHP exécutera l'instruction et si elle vaut false, l'instruction sera ignorée.
• Les instructions après le else ne sont exécutées que si l'expression du if est false.

L'instruction if else

```
if (expression)
  commandes
elseif (expression)
  commandes
elseif (expression)
  commandes
elseif (expression)
  commandes
else
  commandes
```

• L'expression elseif est exécutée seulement si le if précédent et tout autre elseif précédent sont évalués comme false, et que votre elseif est évalué à true.

• Le « elseif » et « else if » sont traités de la même façon seulement quand des accolades sont utilisées. (Exemple : **else if ($a == $b){ echo $a." égal ".$b;}** est une synthèse correcte, tandis que **else if ($a == $b) : echo $a." égal ".$b;** est une synthèse qui génère une erreur PHP. De la même façon, les deux syntaxes **elseif ($a ==$b){ echo $a." égal ".$b;}** et **elseif ($a == $b) : echo $a." égal ".$b;** sont toutes les deux correctes.

L’instruction switch

- L'instruction switch équivaut à une série d'instructions if.
- Un cas spécial est default. Ce cas est utilisé lorsque tous les autres cas ont échoué.
- PHP continue d'exécuter les instructions jusqu'à la fin du bloc d'instructions du switch, ou bien dès qu'il trouve l'instruction break.
- Dans une commande switch, une condition n'est évaluée qu'une fois, et le résultat est comparé à chaque case

```php
<?php
switch ($i) {
  case 0:
    echo "i égal 0";
    break;
  case 1:
    echo "i égal 1";
    break;
  case 2:
    echo "i égal 2";
    break;
   default:
     echo "i n'est ni égal à 2, ni à 1, ni à 0.";
}
?>
```

L’instruction match

• De la même manière qu'une instruction switch, l’instruction match a une expression de sujet qui est comparée à plusieurs alternatives.
• Différences entre match et switch :
• match évaluera une valeur un peu comme les expressions ternaires.
• la comparaison match est un contrôle d'identité (===) plutôt qu'un contrôle d'égalité faible (==) comme switch.
• match renvoie une valeur

```php
<?php
$valeur_retour = match (expression) {
    expression_conditionelle_unique => expression_retour,
    expression1_conditionnelle, expression2_conditionnelle => expression_retour,
};
?>
```

5-Contrôle de flux et boucles

L’instruction while

• PHP exécute l'instruction tant que l'expression de la boucle while est évaluée comme true. Si l'expression du while est false avant la première itération, l'instruction ne sera jamais exécutée.

```
while (expression) :
  commandes
endwhile;
```

L’instruction do-while

• La principale différence par rapport à la boucle while est que la première itération de la boucle do-while est toujours exécutée.

```
do
  commandes
while (expression);
```

Maitriser le langage PHP

5-Contrôle de flux et boucles

Opérateurs d'incrémentation et décrémentation

PHP supporte les opérateurs de pre- et post-incrémentation et décrémentation, comme en langage C.

Note: Les opérateurs d'incrémentation/décrémentation n'affectent que les nombres et les chaînes de caractères. Les tableaux, objets, booléen et ressources ne sont pas affectées.
La décrémentation des valeurs **null** n'a également aucun effet, mais leur incrémentation donnera comme résultat 1.

Opérateurs d'incrémentation et décrémentation

Exemple	Nom	Résultat
++$a	Pre-incrémente	Incrémente *$a* de 1, puis retourne *$a*.
$a++	Post-incrémente	Retourne *$a*, puis incrémente *$a* de 1.
--$a	Pré-décrémente	Décrémente *$a* de 1, puis retourne *$a*.
$a--	Post-décrémente	Retourne *$a*, puis décrémente *$a* de 1.

Maitriser le langage PHP

5-Contrôle de flux et boucles

Opérateurs d'incrémentation et décrémentation

```php
<?php
echo '<h3>Post-incrémentation</h3>';
$a = 5;
echo "Devrait valoir  5: " . $a++ . "<br />\n";
echo "Devrait valoir  6: " . $a . "<br />\n";
echo '<h3>Pre-incrémentation</h3>';
$a = 5;
echo "Devrait valoir  6: " . ++$a . "<br />\n";
echo "Devrait valoir  6: " . $a . "<br />\n";
echo '<h3>Post-décrémentation</h3>';
$a = 5;
echo "Devrait valoir  5: " . $a-- . "<br />\n";
echo "Devrait valoir  4: " . $a . "<br />\n";
echo '<h3>Pre-décrémentation</h3>';
$a = 5;
echo "Devrait valoir  4: " . --$a . "<br />\n";
echo "Devrait valoir  4: " . $a . "<br />\n";
?>
```

Maitriser le langage PHP

5-Contrôle de flux et boucles

L'instruction for

- **expr1 :** est évaluée (exécutée), au début de la boucle.
- **expr2 :** est évaluée, au début de chaque itération. Si l'évaluation vaut true, la boucle continue et les commandes sont exécutées. Si l'évaluation vaut false, l'exécution de la boucle s'arrête.
- **expr3 :** est évaluée (exécutée), à la fin de chaque itération.
- Les expressions peuvent éventuellement être laissées vides ou peuvent contenir plusieurs expressions séparées par des virgules.

```
for (expr1; expr2; expr3)
   commandes
```

L'instruction foreach

- foreach ne fonctionne que pour les tableaux et les objets.
- La forme suivante passe en revue le tableau iterable_expression. À chaque itération, la valeur de l'élément courant est assignée à $value.

```
foreach (iterable_expression as $value){
   //commandes
}
```

- La forme suivante assignera en plus la clé de l'élément courant à la variable $key à chaque itération.

```
foreach (iterable_expression as $key =>
$value){
   //commandes
}
```

Maitriser le langage PHP

6 – Les fonctions des chaines des caractères et Dates

addcslashes	addslashes	chop - Alias de rtrim	ltrim
Ajoute des slash dans une chaîne, à la mode du langage C	Ajoute des antislashs dans une chaîne	Supprime les espaces (ou d'autres caractères) de fin de chaîne	Supprime les espaces (ou d'autres caractères) de début de chaîne

count_chars	similar_text	strlen
Retourne des statistiques sur les caractères utilisés dans une chaîne	Calcule la similarité de deux chaînes	Calcule la taille d'une chaîne

strtoupper	strtr	mb_convert_case	mb_strtolower
Renvoie une chaîne en majuscules	Remplace des caractères dans une chaîne	Modifie la casse d'une chaîne	Met tous les caractères en minuscules

Maitriser le langage PHP

6 – Les fonctions des chaines des caractères et Dates

Fonctions sur les Date/Heure

- **date =** Formate une date/heure locale

Caractères pour le paramètre format	Description	Exemple de valeurs retournées
Jour	---	---
d	Jour du mois, sur deux chiffres (avec un zéro initial)	01 à 31
D	Jour de la semaine, en trois lettres (et en anglais - par défaut : en anglais, ou sinon, dans la langue locale du serveur)	Mon à Sun
j	Jour du mois sans les zéros initiaux	1 à 31
l ('L' minuscule)	Jour de la semaine, textuel, version longue, en anglais	Sunday à Saturday
N	Représentation numérique ISO 8601 du jour de la semaine	1 (pour Lundi) à 7 (pour Dimanche)
S	Suffixe ordinal d'un nombre pour le jour du mois, en anglais, sur deux lettres	st, nd, rd ou th. Fonctionne bien avec j
w	Jour de la semaine au format numérique	0 (pour dimanche) à 6 (pour samedi)
z	Jour de l'année	0 à 365
Semaine	---	---
W	Numéro de semaine dans l'année ISO 8601, les semaines commencent le lundi	Exemple : 42 (la 42ème semaine de l'année)

Tab : Exemple de caractère reconnus dans le paramètre format

Maitriser le langage PHP

7- les Fonctions

Une fonction peut être définie en utilisant la syntaxe suivante :

Exemple #1 Pseudo code pour illustrer l'usage d'une fonction

```php
<?php
function foo($arg_1, $arg_2, /* ..., */ $arg_n)
{
    echo "Exemple de fonction.\n";
    return $retval;
}
?>
```

Les noms de fonctions suivent les mêmes règles que les autres labels en PHP. Un nom de fonction valide commence par une lettre ou un souligné, suivi par un nombre quelconque de lettres, de nombres ou de soulignés. Ces règles peuvent être représentées par l'expression rationnelle suivante : ^[a-zA-Z_\x80-\xff][a-zA-Z0-9_\x80-\xff]*$

7- les Fonctions

En PHP et même dans plusieurs langages de programmation il existe plusieurs types des fonctions :

- **Fonctions Variables**
- **Fonctions anonymes**
- **Fonctions fléchées**

Maitriser le langage PHP

7- les Fonctions Variables

PHP supporte le concept de fonctions variables. Cela signifie que si le nom d'une variable est suivi de parenthèses, PHP recherchera une fonction de même nom et essaiera de l'exécuter. Cela peut servir, entre autres, pour faire des fonctions de rappel, des tables de fonctions...

Les fonctions variables ne peuvent pas fonctionner avec les éléments de langage comme les **echo**, **print**, **unset()**, **isset()**, **empty()**, **include**, **require** etc. Vous devez utiliser votre propre gestion de fonctions pour utiliser un de ces éléments de langage comme fonctions variables

Exemple #1 Exemple de fonction variable

```php
<?php
function foo() {
    echo "dans foo()<br />\n";
}

function bar($arg = '')
{
    echo "Dans bar(); l'argument était '$arg'.<br />\n";
}
```

```php
$func = 'foo';
$func();        // Appel foo()

$func = 'bar';
$func('test');  // Appel bar()
```

7- les Fonctions Anonymes

Les fonctions anonymes, aussi appelées fermetures ou closures permettent la création de fonctions sans préciser leur nom. Elles sont particulièrement utiles comme fonctions de rappel callable, mais leur utilisation n'est pas limitée à ce seul usage.

Les fonctions anonymes sont implémentées en utilisant la classe Closure.

```php
<?php
$greet = function($name)
{
    printf("Bonjour %s\r\n", $name);
};

$greet('World');
$greet('PHP');
?>
```

Maitriser le langage PHP

7- les Fonctions Fléchées

Les fonctions fléchées ont été introduites en PHP 7.4 en tant que syntaxe plus concise pour les .
Les fonctions anonymes comme les fonctions fléchées sont implémentées en utilisant la classe Closure.
Les fonctions fléchées ont la forme basique

fn (argument_list) => expr.

Les fonctions fléchées supportent les mêmes fonctionnalités que les fonctions anonymes, à l'exception que l'utilisation des variables de la portée parente est automatique.
Quand une variable utilisée dans l'expression est définie dans la portée parente, elle sera implicitement capturée par valeur. Dans l'exemple suivant, les fonctions *$fn1* et *$fn2* se comportent de façon identique.

```php
<?php

$y = 1;

$fn1 = fn($x) => $x + $y;
// equivalent to using $y by value:
$fn2 = function ($x) use ($y) {
    return $x + $y;
};

var_export($fn1(3));
?>
```

8- Formulaires simples et Variables Superglobals

$_GET

• Elle donne les valeurs des informations indiquées dans l'url.
• Les informations après le point d'interrogation ? d'une URL sont en réalité des données que l'on fait transiter d'une page à une autre.

Exemple :
• Le site s'appel : lesite.com
• Ma page PHP de utilisateur : action.php
• Pour accéder à la page de l'utilisateur, l'URL sera :
http://www.lesite.com/action.php
• Pour envoyer les informations de l'utilisateur à la page php nous utiliserons :
http://www.lesite.com/action.php?nom=Rasmus&prenom=Lerdorf
• Les paramètres sont séparé par le symbole **&**

Maitriser le langage PHP

8- Formulaires simples et Variables Superglobals

$_GET

Exemple de formulaire :

```
<form action="action.php" method="get">
 <p>Votre nom : <input type="text" name="nom" /></p>
 <p>Votre prénom : <input type="text" name="prenom" /></p>
 <p><input type="submit" value="OK"></p>
</form>
```

• Quand l'utilisateur clique sur le bouton « OK », l'URL envoyé au serveur sera visible par le visiteur dans la barre d'adresse comme ceci:
http://www.lesite.com/action.php?nom=Rasmus&prenom=Lerdorf
• Le fichier « action.php » peut maintenant employer la variable super globale $_GET pour récupérer les données du formulaire

```
<?php
   echo "Bonjour $_GET["nom"]  $_GET[“prenom"] et bienvenue";
?>
```

8- Formulaires simples et Variables Superglobals

$_POST

• PHP $_POST est une variable super globale PHP qui est utilisée pour collecter des données de formulaire après avoir soumis un formulaire HTML avec method="post".
• $_POST est également utilisé pour passer des variables.
• Lorsque nous envoyons des variables via un formulaire, la variable est récupérée de cette façon $_POST['Variable'] et non plus $Variable

Exemple de formulaire envoyé :

```
<form action="action.php" method="post">
 <p>Votre nom : <input type="text" name="nom" /></p>
 <p>Votre prénom : <input type="text" name="prenom" /></p>
 <p><input type="submit" value="OK"></p>
</form>
```

```
<?php
echo $_POST['nom'];
```

CHAPITRE 2

Programmer en PHP

1- Maitriser le langage PHP

2- Traiter les données en PHP

3- Utiliser l'orientée objet en PHP

Traiter les données en PHP

Ce que vous allez apprendre dans ce chapitre :

- Traitement des tableaux (simple, Associatif)
- Manipulation de fichier (chargement, Suppression, téléchargement)

Traiter les données en PHP

1- Traitement des tableaux : les tableaux multidimensionnels

•Un tableau multidimensionnel est un tableau contenant lui même un ou plusieurs autres tableaux en valeurs.

•Un tableau multidimensionnel est un tableau qui stocke un autre tableau à chaque index au lieu d'un seul élément.

•Le tableau multidimensionnel est également connu sous le nom de tableau PHP de tableaux .

•Comme son nom l'indique, chaque élément de ce tableau peut être un tableau et ils peuvent également contenir d'autres sous-tableaux.

•Le tableau multidimensionnel est un tableau dans lequel chaque élément peut également être un tableau et chaque élément du sous-tableau peut être un tableau ou contenir en outre un tableau en lui-même et ainsi de suite.

Traiter les données en PHP

1- Traitement des tableaux : les tableaux multidimensionnels

- Un tableau peut aussi contenir des tableaux (c.à.d. que chaque élément d'un tableau est aussi un tableau). On parle alors de tableau "deux dimensions" (2D), ou plus.
- Pour un tableau à deux dimensions, comme tout tableau, est une **variable** possédant un nom unique, dans laquelle chaque élément est non plus repéré par un seul indice, mais par deux indices (numéro de ligne et numéro de colonne) permettant d'indiquer sa position.
- La dimension d'un tableau correspond au nombre d'informations nécessaires à la localisation d'un élément.
- Les tableaux PHP bidimensionnels contiennent des tableaux dans lesquels les éléments sont repérées par des colonnes et des lignes.
- Voici un tableau à 2 dimensions , qui contient 3 lignes et 4 colonnes :

	Colonne 0	Colonne 1	Colonne 2	Colonne 3
Ligne 0	tab [0][0]	tab [0][1]	tab [0][2]	tab [0][3]
Ligne 1	tab [1][0]	tab [1][1]	tab [1][2]	tab [1][3]
Ligne 2	tab [2][0]	tab [2][1]	tab [2][2]	tab [2][3]

Traiter les données en PHP

1- Traitement des tableaux : les tableaux multidimensionnels

- Soit le tableau suivant:

Prénom et nom	Moyenne	Rang
Salah MAJDOUB	10.45	12
Taysir ALLANI	12.21	5
Rafika HARBAOUI	09.18	13
Monia SELLITI	14.11	1

- Nous pouvons stocker les données du tableau ci-dessus dans un tableau à deux dimensions, comme ceci:

```php
$apprenants = array (
      array("Salah MAJDOUB",10.45,12),
      array("Taysir ALLANI",12.21,5),
      array("Rafika HARBAOUI",09.18,13),
      array("Monia SELLITI",14.11,1)
    );
```

Traiter les données en PHP

1- Traitement des tableaux : les tableaux multidimensionnels

- Maintenant, le tableau $apprenants en deux dimensions contient quatre tableaux, et il a deux indices: ligne et colonne.
- Pour accéder aux éléments du tableau **$apprenants**, nous devons pointer sur les deux indices (ligne et colonne):

```php
echo "Prénom et nom: ".$apprenants[0][0].": Moyenne: ".$apprenants[0][1].", Rang: ".$apprenants[0][2].".<br>";
    echo "Prénom et nom: ".$apprenants[1][0].": Moyenne: ".$apprenants[1][1].", Rang: ".$apprenants[1][2].".<br>";
    echo "Prénom et nom: ".$apprenants[2][0].": Moyenne: ".$apprenants[2][1].", Rang: ".$apprenants[2][2].".<br>";
    echo "Prénom et nom: ".$apprenants[3][0].": Moyenne: ".$apprenants[3][1].", Rang: ".$apprenants[3][2].".<br>";
```

```php
// une première boucle pour parcourir l'ensemble des clés
foreach ($apprenants as $cle => $valeurs) {
  //une deuxième boucle pour parcourir toutes les valeurs associées à une clé
  foreach ($apprenants[$cle] as $nom){
      echo($nom."</br>");
```

Traiter les données en PHP

1- Traitement des tableaux : les tableaux multidimensionnels

Création d'un tableau bidimensionnel:

- Un tableau à deux dimensions est un tableau dont chaque ligne contient un autre tableau. Ce type de tableau est aussi appelé tableau **multidimensionnel**.

```php
$apprenants = array(
    array('prenom'=>'Mohamed', 'nom'=>'HAJJI','age'=>20,'groupe'=>'MDI21'),
    array('prenom'=>'Samir', 'nom'=>'BLOUM','age'=>21,'groupe'=>'MDI21'),
    array('prenom'=>'Samira', 'nom'=>'MEJBRI','age'=>19,'groupe'=>'MDI21'),
    array('prenom'=>'Seddik', 'nom'=>'TAYEB','age'=>19,'groupe'=>'MDI21'),
    array('prenom'=>'Rafira', 'nom'=>'HAJJEJ','age'=>20,'groupe'=>'MDI21'),
    );
```

Traiter les données en PHP

1- Traitement des tableaux : les tableaux multidimensionnels

Parcourir un tableau bidimensionnel :

•La boucle **for** peut permettre la lecture de tableaux multidimensionnels, à condition d'écrire autant de niveaux de boucles qu'il y a de dimensions dans le tableau.

•La boucle **foreach()** nous permet de lire et afficher les données d'un tableau multidimensionnel, à condition d'écrire autant de niveaux de boucles qu'il y a de dimensions dans le tableau.

•Le premier **foreach()** lit chaque ligne du tableau

•Le deuxième foreach() lit chaque tableau de chaque ligne. On accède à ses propriétés et à ses valeurs avec les variables **$cle** et **$valeur**.

```php
<?php
$apprenants = array(
    array('prenom'=>'Mohamed', 'nom'=>'HAJJI'),
    array('prenom'=>'Samir', 'nom'=>'BLOUM')
    );
// Lecture de chaque ligne du tableau
foreach($apprenants as $ligne){
        // Lecture de chaque tableau de chaque ligne
    foreach($ligne as $cle=>$valeur){
        // Affichage
        echo $cle.': '.$valeur;
        echo '<br>';
    }
}
?>
```

Traiter les données en PHP

1- Traitement des tableaux : les tableaux multidimensionnels

Operations sur les tableaux : Diffèrents Fonctions

Ajout d'élément au début du tableau

- La fonction **array_unshift() de PHP ()** ajoute de nouveaux éléments au début d'un tableau et renvoie la nouvelle longueur.

Syntaxe

```
1 | array_unshift(valeur1, valeur2, ..., valeurX)
```

- Pour ajouter un élément au tableau utilisez le mot clé **array_unshift**

```
1 | $apprenants = array( "Thamer", "Mohamed", "Mounira", "Samira", "Tarek" );
```

- Ajoutez le nom "Marwen" à votre tableau

```
1 | array_unshift($apprenants,"Marwen");
```

Traiter les données en PHP

1- Traitement des tableaux : les tableaux multidimensionnels

Operations sur les tableaux : Diffèrents Fonctions

Ajout d'élément à la fin du tableau

> - La fonction **array_push ()** ajoute de nouveaux éléments à la fin d'un tableau et renvoie la nouvelle longueur.

Syntaxe

```
1 | array_push(valeur1, valeur2, ..., valeurX)
```

- Pour ajouter un élément au tableau utilisez le mot clé **array_push**

```
1 | $apprenants = array( "Thamer", "Mohamed", "Mounira", "Samira", "Tarek" );
```

- Ajoutez le nom "Marwen" à la fin de votre tableau

```
1 | array_push($apprenants,"Marwen");
```

Traiter les données en PHP

1- Traitement des tableaux : les tableaux multidimensionnels

Operations sur les tableaux : Diffèrents Fonctions

Supprimer le premier élément d'un tableau

> - La fonction **array_shift ()** supprime le premier élément d'un tableau et renvoie cet élément.

Syntaxe

```
1 | array_shift()
```

- Pour supprimer le premier élément d'un tableau utilisez le mot clé **array_shift**

```
1 | $apprenants = array( "Thamer", "Mohamed", "Mounira", "Samira", "Tarek" );
```

- Supprimez le le dernier élément du tableau

```
1 | array_shift($apprenants);
```

Traiter les données en PHP

1- Traitement des tableaux : les tableaux multidimensionnels

Operations sur les tableaux : Diffèrents Fonctions

Supprimer le dernier élément d'un tableau

- La méthode **array_pop ()** supprime le dernier élément d'un tableau et renvoie cet élément.

Syntaxe

```
1 | array_pop()
```

- Pour supprimer le dernier élément d'un tableau utilisez le mot clé **pop**

```
1 | $apprenants = array( "Thamer", "Mohamed", "Mounira", "Samira", "Tarek" );
```

- Supprimez le le dernier élément du tableau

```
1 | array_pop($apprenants);
```

Traiter les données en PHP

1- Traitement des tableaux : les tableaux multidimensionnels

Operations sur les tableaux : Diffèrents Fonctions

count() et sizeof()

- retournent toutes les deux la taille du tableau passé en paramètre.

sort()

- trie les éléments d'un tableau du plus petit au plus grand.

rsort()

- trie les éléments d'un tableau du plus grand au plus petit.

in_array()

- permet de vérifier qu'une valeur est présente dans un tableau.

array_rand()

- extrait une ou plusieurs valeurs du tableau au hasard.

current()

- retourne la valeur de l'élément courant du tableau (où se trouve le pointeur)

Traiter les données en PHP

1- Traitement des tableaux : les tableaux multidimensionnels

Operations sur les tableaux : Diffèrents Fonctions

explode()	array explode (string separator, string string [, int limit]) Retourne un tableau qui contient les éléments d'une chaîne contenant des délimiteurs
implode()	string implode (string glue, array pieces) Retourne une chaîne constituée de tous les éléments du tableau, pris dans l'ordre, transformés en chaîne, et séparés par le séparateur
split()	array split (string pattern, string string [, int limit]) Scinde une chaîne en un tableau, grâce à une expression régulière.
join()	string join (string glue, array pieces) Regroupe tous les éléments d'un tableau dans une chaîne, avec une chaîne de jointure.

Traiter les données en PHP

1- Traitement des tableaux : les tableaux multidimensionnels

Opérateurs de tableaux

Exemple	Nom	Résultat
$a + $b	Union	Union de *$a* et *$b*.
$a == $b	Égalité	**true** si *$a* et *$b* contiennent les mêmes paires clés/valeurs.
$a === $b	Identique	**true** si *$a* et *$b* contiennent les mêmes paires clés/valeurs dans le même ordre et du même type.
$a != $b	Inégalité	**true** si *$a* n'est pas égal à *$b*.
$a <> $b	Inégalité	**true** si *$a* n'est pas égal à *$b*.
$a !== $b	Non-identique	**true** si *$a* n'est pas identique à *$b*.

Traiter les données en PHP

2- Manipulation des fichiers

Ouvrir un fichier

- fopen = Ouvre un fichier ou une URL

Expression :

```
fopen(
    string $filename,
    string $mode,
    bool $use_include_path = false,
    resource $context = ?
): resource
```

- Filename : est de la forme "protocole://"
- mode : spécifie le type d'accès désiré au flux
- use_include_path : paramètre optionnel peut être défini à 1 ou à true pour chercher le fichier dans l'include_path.

Traiter les données en PHP

2- Manipulation des fichiers

- Liste des protocoles et des gestionnaires supportés :

- file:// — Accès au système de fichiers local
- http:// — Accès aux URLs HTTP(s)
- ftp:// — Accès aux URLs FTP(s)
- php:// — Accès aux divers flux I/O
- zlib:// — Flux de compression
- data:// — Données (RFC 2397)
- glob:// — Trouve des noms de fichiers correspondant à un masque donné
- phar:// — Archive PHP
- ssh2:// — Shell sécurisé 2
- rar:// — RAR
- ogg:// — Flux Audio
- expect:// — Flux d'interactions de processus

Traiter les données en PHP

2- Manipulation des fichiers

Paramètre mode de fopen()

Liste des modes possibles pour la fonction fopen() en utilisant le paramètre mode

Mode	Description
'r'	Ouvre en lecture seule, et place le pointeur de fichier au début du fichier.
'r+'	Ouvre en lecture et écriture, et place le pointeur de fichier au début du fichier.
'w'	Ouvre en écriture seule ; place le pointeur de fichier au début du fichier et réduit la taille du fichier à 0. **Si le fichier n'existe pas, on tente de le créer.**
'w+'	Ouvre en lecture et écriture ; le comportement est le même que pour 'w'.
'a'	Ouvre en écriture seule ; place le pointeur de fichier à la fin du fichier. **Si le fichier n'existe pas, on tente de le créer.**
'a+'	Ouvre en lecture et écriture ; place le pointeur de fichier à la fin du fichier. **Si le fichier n'existe pas, on tente de le créer.**
'x'	Crée et ouvre le fichier en écriture seulement ; place le pointeur de fichier au début du fichier. Si le fichier existe déjà, fopen() va échouer, en retournant false et en générant une erreur de niveau E_WARNING. **Si le fichier n'existe pas, fopen() tente de le créer.**
'x+'	Crée et ouvre le fichier pour lecture et écriture; le comportement est le même que pour 'x'.
'c'	Ouvre le fichier pour écriture seulement. **Si le fichier n'existe pas, il sera crée** et s'il existe, il n'est pas tronqué et l'appel à la fonction n'échoue pas. Le pointeur du fichier est positionné au début.
'c+'	Ouvre le fichier pour lecture et écriture, le comportement est le même que pour le mode 'c'.
'e'	Défini l'indicateur close-on-exec sur le descripteur de fichier ouvert. Disponible uniquement en PHP compilé sur les systèmes conforme POSIX.1-2008.

Traiter les données en PHP

2- Manipulation des fichiers

Lire un fichier

fread()
Lecture du fichier en mode binaire

file()
Lit le fichier et renvoie le résultat dans un tableau

file_get_contents
Lit tout un fichier dans une chaîne

fstat
Lit les informations sur un fichier à partir d'un pointeur de fichier

fgets()
Récupère la ligne courante à partir de l'emplacement du pointeur sur fichier

fgetc
Lit un caractère dans un fichier

readfile
Affiche un fichier

fclose
Ferme un fichier

fwrite
Écrit un fichier en mode binaire

fputs
Alias de fwrite

file_put_contents
Écrit des données dans un fichier

Traiter les données en PHP

2- Manipulation des fichiers : Exercice

•Pour gérer une bibliothèque les informations suivants doivent être prises en compte:
•Les informations à enregistrer pour les livres sont :
les titres, le (ou les) auteur(s),la langue, la date de parution, l'éditeur, la date d'impression et le nombre de pages.
•La gestion des livres seront dans une page web formée d'un tableau de la forme :

Titre	Auteur(s)	Langue	La date de parution	L'éditeur	La date d'impression	Le nombre de pages
PHP 5 Cours et exercices	Jean Engels	Français	2005	eBook	27-juin-05	228
L'art de développement android	Grant Allen	Français	2004	Reynald Goulet	06-déc-12	380
Programmer en Python	Luciano RAMALHO	Anglais	2010	O'Reilly Broché	03-janv-19	580
Développer un site web en Php, Mysql et Javascript, Jquery, CSS3 et HTML5	Robin Nixon	Français	2019	Reynald Goulet	03-janv-19	790
Python pour le data scientist	Emmanuel Jakobowicz	Français	2018	Dunod	24-oct-18	304
Data science : fondamentaux et études de cas	Michel Lutz et Eric Biernat	Anglais	2015	Eyrolles	01-oct-15	296

•**Travail à faire:**

•Créer la page "**saisie_livre.php**" qui contient le formulaire d'entrée d'un livre
•Créer la page "**liste_livre.php**" qui sert à afficher la liste des livres saisies selon le modèle cité ce dessus
•Créer un script PHP pour la vérification et l'envoi des informations saisies vers le fichier "**bibliotheque.txt**" pour être stockées

CHAPITRE 2

Programmer en PHP

1- Maitriser le langage PHP

2- Traiter les données en PHP

3- Utiliser l'orientée objet en PHP