

Atelier de Développement Web Côté Serveur

Institut Supérieur des Études Technologiques de Mahdia
Département: Technologie de l'Informatique



Année universitaire: 2024/2025

Atelier de Développement Web Côté Serveur

1



Atelier 2 Les bases du langage PHP

Atelier de Développement Web Côté Serveur

2

Atelier de Développement Web Côté Serveur Atelier 2: Les bases du langage PHP

Plan

- 1 Les structures de contrôle
- 2 Les tableaux
- 3 Les chaînes de caractères
- 4 Les fonctions



Atelier de Développement Web Côté Serveur

39

3

Atelier de Développement Web Côté Serveur Atelier 2: Les bases du langage PHP

Les structures conditionnelles et répétitives

Les structures de contrôle



▪ L'instruction if

```
if (condition realisee ){
    liste d'instructions}
```

▪ L'instruction if ... else

```
if (condition realisee ){
    liste d'instructions}
else{
    autre liste d'instructions}
```

▪ L'instruction if ... elseif ... else

```
if (condition realisee ){
    liste d'instructions}
elseif ( autre condition ){
    autre liste d'instructions}
else( derniere condition realisee){
    liste d'instructions}
```

Exemple : Instruction if... elseif ... else

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

Atelier de Développement Web Côté Serveur

39

4

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les structures conditionnelles et répétitives

Les structures de contrôle

▪ L'instruction switch

```
switch( variable ){
case valeur_1 : liste d'instructions ; break ;
case valeur_2 : liste d'instructions ; break ;
case valeur_n : liste d'instructions ; break ;
default : liste d'instructions ; break ;
}
```

Exemple : Instruction switch

```
<?php
switch($i){
case 0:
    echo "i egal 0";
    break;
case 1:
    echo "i egal 1";
    break;
case 2:
    echo "i egal 2";
    break;
}
?>
```

Atelier de Développement Web Côté Serveur 39

5

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les structures conditionnelles et répétitives

Les structures de contrôle

▪ La boucle for

```
for(expression1 ; condition ; expression2)
    expression3 ;
```

Exemple : Boucle for

```
for( $i = 1 ; $i <= 10 ; $i++ ){
    echo $i ;}
```

▪ La boucle while

```
while(condition){
    bloc d'instructions ;}
```

Exemple : Boucle while

```
$i = 1 ;
while( $i <= 10 ) { echo $i ++ ;}
```

Atelier de Développement Web Côté Serveur 39

6

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les structures conditionnelles et répétitives

Les structures de contrôle

▪ La boucle do..while

```
do{
    bloc d'instructions ; }
while(condition) ;
```

Exemple : Boucle do..while

```
$i = 0 ; do{
    echo $i ;
}while( $i > 0 ) ;
```

Atelier de Développement Web Côté Serveur 39

7

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les chaînes de caractères

Les tableaux

Atelier de Développement Web Côté Serveur 39

8

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les tableaux

Principe

- La création se fait à l'aide de la fonction **array()**
- Les éléments d'un tableau peuvent appartenir à des types distincts

```

graph LR
    A[Types de tableau] --> B[Tableau numéroté]
    A --> C[Tableau associatif]
  
```

Atelier de Développement Web Côté Serveur 39

9

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les tableaux

Les tableaux numérotés

Principe

- Construire un tableau numéroté

```
$tableau = array(valeur0, valeur1,..., valeurN);
```

- Accéder aux éléments par l'intermédiaire de numéros
- L'index d'un tableau en PHP commence de 0

Exemple : Création d'un tableau

```
<?php
$jours = array("Dimanche", "Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi");
?>
```

Le premier élément ("Dimanche") aura le numéro 0, ensuite "Lundi" le numéro 1, etc.

Atelier de Développement Web Côté Serveur 39

10

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les tableaux

Les tableaux numérotés

- Affichage**: Il faut donner la position de l'élément entre crochets après le nom du tableau.

Exemple : Affichage d'un élément d'un tableau

```
echo $jours[1];
```

- Modification**

Exemple : Modification d'un élément d'un tableau

```
$jours[1] = 'Monday';
```

- Ajout**

Exemple : Ajout d'un élément d'un tableau

```
$jours[6] = "Samedi";
```

Atelier de Développement Web Côté Serveur 39

11

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les tableaux

Les tableaux numérotés

Le contenu d'un tableau numéroté peut être affiché plus facilement

- Avec une **boucle for**

```
for($i = 0; $i < count($tab); $i++){
    echo $tab[$i], "<br/>";
}
```

- Avec une **boucle while**

```
$i = 0;
while($i < count($tab)){
    echo $tab[$i], "<br/>"; $i++;
}
```

- Avec la boucle **foreach** (utilisée spécialement avec les tableaux)

```
foreach($tab as $elem){
    echo $elem, "<br/>";
}
```

Atelier de Développement Web Côté Serveur 39

12

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les tableaux

Les tableaux associatifs

Principe

- Les tableaux associatifs fonctionnent sur le même principe, sauf qu'au lieu de numéroté les cases, on va les étiqueter en leur donnant à chacune un nom différent.
- Construire un tableau associatif
- Utiliser la fonction **array** en mettant "l'étiquette" devant chaque information.
\$tableau = array(ind0 => val0,..., indN => valN);

Exemple : Création d'un tableau associatif

```
<?php
$coordonnes = array(
    'prenom' => 'Aymen ',
    'nom' => 'Sfar ',
    'ville' => 'Mahdia' );
?>
```

Atelier de Développement Web Côté Serveur 39

13

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les tableaux

Les tableaux associatifs

- Affichage**
Exemple : Affichage d'un élément d'un tableau associatif
echo \$coordonnes['ville'];
- Modification**
Exemple : Modification d'un élément d'un tableau associatif
\$coordonnes['ville'] = 'Mahdia';
- Ajout**
Exemple : Ajout d'un élément d'un tableau associatif
\$coordonnes['adresse'] = 'Route tunis';

Atelier de Développement Web Côté Serveur 39

14

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les tableaux

Parcours d'un tableau associatif

Parcourir un tableau associatif avec

- boucle **foreach()**

Atelier de Développement Web Côté Serveur 39

15

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les tableaux

Parcours d'un tableau associatif

1 La boucle **foreach** (Il existe 2 syntaxes)

- Première syntaxe**
foreach (array_expression as \$value)
{ commandes }
- Exemple : Boucle foreach**
foreach(\$coordonnes as \$element){
 echo \$element . '
 ' ;
}
foreach met tour à tour dans la variable \$element le prénom, le nom, l'adresse et la ville contenus dans l'array \$coordonnes.
- Deuxième syntaxe**
foreach (array_expression as \$key => \$value)
{ commandes }
La seconde forme assignera en plus la clé de l'élément courant à la variable \$key à chaque itération
- Exemple : Boucle foreach**
foreach(\$coordonnes as \$cle => \$element)
{ echo '[' . \$cle . ']' vaut ' . \$element . '
 ' ; }
À chaque tour de boucle, on dispose de 2 variables :
 - \$cle contient la clé de l'élément courant
 - \$element contient la valeur de l'élément courant

Atelier de Développement Web Côté Serveur 39

16

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les tableaux

Fonctions sur les tableaux

Il existe d'autres fonctions qui permettent de se déplacer dans un tableau :

- **print_r** : affiche les éléments d'un tableau avec leur clé.
- **reset** : remet le pointeur au début du tableau.
- **end** : positionne le pointeur en fin du tableau.
- **pos** : retourne la valeur de l'élément courant du tableau.
- **next** : avance le pointeur.
- **prev** : recule le pointeur.
- **sizeof** : retourne le nombre d'éléments du tableau.
- **shuffle(tab)** : trie le tableau tab de manière aléatoire.
- **count(tab)** : donne le nombre d'éléments du tableau tab.
- **array_merge(tab1, tab2)** : concatène les tableaux tab1 et tab2 en un seul autre tableau.

Atelier de Développement Web Côté Serveur 39

17

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les tableaux

Fonctions de tri des tableaux

Tri selon les valeurs

- La fonction **sort()** effectue un tri sur les valeurs des éléments d'un tableau selon un critère alphanumérique (selon les codes ASCII). Pour les tableaux associatifs les clés seront perdues et remplacées par un indice créé après le tri et commençant à 0.
- La fonction **rsort()** effectue la même action mais en ordre inverse des codes ASCII.
- La fonction **asort()** trie également les valeurs selon le critère des codes ASCII, mais en préservant les clés pour les tableaux associatifs.
- La fonction **arsort()** la même action mais en ordre inverse des codes ASCII.

Atelier de Développement Web Côté Serveur 39

18

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les tableaux

Fonctions de tri des tableaux

Tri sur les clés

- La fonction **ksort()** trie les clés du tableau selon le critère des codes ASCII, et préserve les associations clé/valeur.
- La fonction **krsort()** effectue la même action mais en ordre inverse des codes ASCII.

Exemple : Tri d'un tableau

```
<?php
$tab=array ( "1622"=>"Moliere ", " 1920"=>"Vian ", " 1802"=>"Hugo" );
ksort ( $tab );
echo "<h3> Tri sur les clés </h3>";
foreach( $tab as $cle=>$valeur){
    echo " l'element a pour clé: $cle et pour valeur : $valeur <br/>";
}
?>
```

Atelier de Développement Web Côté Serveur 39

19

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les tableaux

Fonctions de recherche dans un tableau

- **array_key_exists** : pour vérifier si une clé existe dans l'array.

Exemple : Fonction array_key_exists

```
<?php
if( array_key_exists ( 'nom' , $coordonnees )){
    echo ' La clef "nom" existe ' ; }
?>
```

- **in_array** : pour vérifier si une valeur existe dans l'array.

Exemple : Fonction in_array

```
<?php
$fruits = array ( 'Banane ', 'Pomme ', 'Poire ', 'Cerise ' );
if( in_array ( 'Framboise ', $fruits)){
    echo ' La valeur "Framboise " se trouve dans les fruits ' ; }
?>
```

Atelier de Développement Web Côté Serveur 39

20

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les tableaux

Fonctions de recherche dans un tableau

▪ **array_search** : pour récupérer la clé d'une valeur dans l'array. (le numéro si c'est un array numéroté, ou le nom de la clé si c'est un array associatif)

Exemple : Fonction array_search

```
<?php
$position = array_search( 'Fraise', $fruits );
echo " 'Fraise' se trouve en position " . $position . "<br/> ";
$position = array_search( 'Banane', $fruits );
echo " 'Banane' se trouve en position " . $position . " ";
?>
```

Atelier de Développement Web Côté Serveur 39

21

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les chaînes de caractères

Les chaînes de caractères

Atelier de Développement Web Côté Serveur 39

22

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les chaînes de caractères

Définition

- Il s'agit d'un ensemble de caractères entre guillemets simples ou doubles.
- En PHP, ce type de données a un nom : **string**.
- On peut stocker des textes courts comme très longs au besoin.

Exemple

```
$classe = "DSI2.1";
$classe = 'DSI2.1';
```

Atelier de Développement Web Côté Serveur 39

23

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les chaînes de caractères

La concaténation

Concaténer avec des guillemets **doubles**

Concaténation avec des guillemets doubles

```
<?php
$age = 20;
echo " Vous avez $age ans ";
?>
```

=> Affiche: Vous avez 20 ans

Concaténer avec des guillemets **simples**

Concaténation avec des guillemets simples

```
<?php
$age = 20;
echo ' Vous avez $age ans ';
?>
```

=> Affiche: Vous avez \ \$age ans

```
<?php
$age = 20;
echo ' Vous avez ' . $age . ' ans ';
?>
```


=> Affiche: Vous avez 20 ans

Atelier de Développement Web Côté Serveur 39

24

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les chaînes de caractères

Les fonctions prédéfinies




- **string strtolower(string \$chaine)** : retourne la chaîne de caractères après avoir converti tous les caractères alphabétiques en minuscules.
- **string strtoupper(string \$chaine)** : retourne la chaîne de caractères après avoir converti tous les caractères alphabétiques en majuscules.
- **mixed strpos(string \$chaine, mixed \$str)** : cherche la position numérique de la première occurrence de `$str` dans la chaîne de caractères `$chaine`. Retourne `FALSE` si l'occurrence n'a pas été trouvée.
- **string substr(string \$source, int \$début, int \$taille)** : retourne une portion de la source à partir de `$début` et de longueur `$taille`, si la variable `$taille` ou `$début` a une valeur négative, le nombre sera calculé à partir de la fin de la chaîne source.

Atelier de Développement Web Côté Serveur 39

25

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les chaînes de caractères

Les fonctions prédéfinies




- **string trim(string \$str)** : retourne la chaîne nettoyée de blancs au début et à la fin de la chaîne.
- **int strlen(string \$str)** : retourne la longueur de la chaîne.
- **string implode(string \$separateur, array \$tab)** : retourne une chaîne constituée de tous les éléments du tableau séparés par le séparateur donné.
- **array explode(string \$separateur, string \$str)** : retourne un tableau qui contient les éléments de la chaîne séparés par le séparateur.
- **string str_replace(string \$modèle, string \$remplacement, string \$chaine)** : retourne une chaîne ou un tableau, dont toutes les occurrences de `$modèle` dans `$chaine` ont été remplacées par `$remplacement`.
- **caractère chr(int \$nombre)** : retourne un caractère en mode ASCII.

Atelier de Développement Web Côté Serveur 39

26

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les chaînes de caractères




LES FONCTIONS

Atelier de Développement Web Côté Serveur 39

27

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les fonctions

Déclaration et appel d'une fonction



- La syntaxe de déclaration s'appuie sur le mot clé **function** suivi du nom de la fonction par lequel on va l'appeler depuis n'importe quel endroit du code PHP.
- Les fonctions peuvent ou non renvoyer un résultat.
- Utiliser l'instruction **return** renvoyant une variable pouvant être de type quelconque.

```

Déclaration d'une fonction
Function nom_fonction ( $ arg 1 , $ arg 2 , . . $argn ){
    declaration des variables ;
    bloc d'instructions;
    // fin du corps de la fonction
    return $resultat ;
}

```

Atelier de Développement Web Côté Serveur 39

28

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les fonctions

Déclaration d'une fonction

- **Casse et noms des fonctions**
 - Les appels de fonctions ne sont pas sensibles à la casse.
 - Les appels de fonction nom(), Fonction Nom() et FONCTION NOM sont donc tous valides et conduisent au même résultat.
- **Restrictions du choix des noms de fonctions**
 - Ne donnez pas à une fonction un nom déjà attribué à une autre fonction.
 - Un nom de fonction ne peut contenir que des lettres, des chiffres et des blancs soulignés.
 - Un nom de fonction ne doit pas commencer par un chiffre.
 - PHP interdit la surcharge des fonctions : il ne permet pas de donner à une fonction personnalisée le même nom qu'une fonction prédéfinie ou qu'une autre fonction personnalisée.

Atelier de Développement Web Côté Serveur 39

29

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les fonctions

Variables locales et variables globales

- Les variables en PHP :
 - global
 - static
 - local
- Toute variable déclarée en dehors d'une fonction est globale.
- Pour utiliser une variable globale dans une fonction, l'instruction global est suivie du nom de la variable.
- Pour conserver la valeur acquise par une variable entre deux appels de la même fonction, l'instruction static est suivie du nom de la variable.
- Les variables statiques restent locales à la fonction et ne sont pas réutilisables à l'extérieur

Atelier de Développement Web Côté Serveur 39

30

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les fonctions

Variables statiques

Exemple : Variables statiques

```
<?php
function cumul( $prix ) {
    static $cumul = 0 ;
    static $i = 1 ;
    echo " Total des achats $i = " ;
    $cumul += $prix ; $i++ ; return $cumul ; }

echo cumul ( 175 ) , "<br />" ; //Total des achats1=175
echo cumul ( 65 ) , "<br />" ; //Total des achats2=240
echo cumul ( 69 ) , "<br />" ; //Total des achats3=309
?>
```

Atelier de Développement Web Côté Serveur 39

31

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les fonctions

Fonctions utiles

- Fonctions avec nombre d'arguments inconnu
 - **func_num_args()** : fournit le nombre d'arguments qui ont été passés lors de l'appel de la fonction.
 - **func_get_arg(\$i)** : retourne la valeur de la variable située à la position \$i dans la liste des arguments passés en paramètres. (Ces arguments sont numérotés à partir de 0)

Exemple : Fonction avec nombre d'arguments inconnu

```
<?php
function produit () {
    $nbarg = func_num_args() ;
    $prod=1 ;
    //la fonction produit a ici $nbarg arguments
    for($i=0; $i<$nbarg ; $i++)
        $prod*= func_get_arg ( $i ) ;
    return $prod;
}
echo " le produit est: " , produit ( 3 ,77,10,5,81,9),"<br />";
?>
//affiche le produit 8 419 950
```

Atelier de Développement Web Côté Serveur 39

32

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les fonctions

Importation des fichiers

- Les instructions **require()** ou **include()** permettent de charger un fichier dans un script PHP.
- Ces deux instructions sont identiques d'un point de vue fonctionnel.
- La différence se situe au niveau du type d'erreur qui sera renvoyé en cas d'échec.
 - `include` produit simplement une alerte de type `E_WARNING`.
 - `require` provoque un `E_ERROR` suivi d'un arrêt total du script.

Remarque : Utilisation de include et require
L'instruction `require` est utilisée si le fichier que vous souhaitez inclure est essentiel pour votre script et qu'il ne peut plus rien effectuer sans avoir exécuté son contenu.
L'instruction `include` est utilisée si le fichier auquel vous souhaitez faire appel n'a aucun aspect fondamental dans votre script.

Atelier de Développement Web Côté Serveur 39

33

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les fonctions

Importation des fichiers

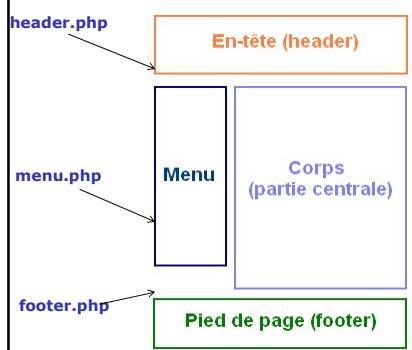
- Quand utiliser `include`/`require`?
- Pour toutes les parties qui se répètent sur un site !
 - En-tête de page
 - Pied de page
 - Menu de navigation
 - Barre contextuelle
 - Paramètres communs PHP (par exemple connexion à MySQL, variables de configuration, chaînes de texte...)

Atelier de Développement Web Côté Serveur 39

34

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les fonctions

Importation des fichiers



```
<html>
<head> </head>
<body>
  <?php include("header.php"); ?>
  <?php include("menu.php"); ?>
  <!-- Le corps -->

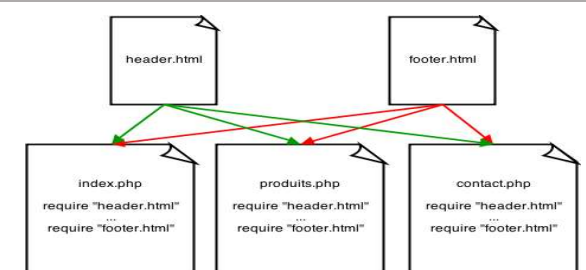
  <div id="corps">
    <h1>Page Accueil </h1>
    <p>
      Bienvenue sur mon super site ! <br />
    </p>
  </div>
  <!-- Le pied de page -->
  <?php include("footer.php"); ?>
</body>
</html>
```

Atelier de Développement Web Côté Serveur 39

35

Atelier de Développement Web Côté Serveur
Atelier 2: Les bases du langage PHP
Les fonctions

Importation des fichiers



Grand intérêt:
plusieurs fichiers (ici `index.php`, `produits.php` et `contact.php`) font appel aux mêmes sources (`header.html` et `footer.html`)

Atelier de Développement Web Côté Serveur 39

36

Atelier de Développement Web Côté Serveur

Atelier 2: Les bases du langage PHP

Les fonctions

Importation des fichiers

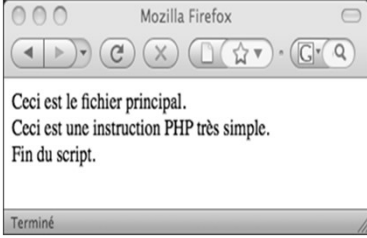
Exemple : Utilisation de l'instruction require()

code stocké dans le fichier reutilisable.php

```
<?php
echo " Ceci est une instruction PHP tres simple .
< br />" ;
?>
```

code stocké dans le fichier principal.php

```
<?php
echo " Ceci est le fichier principal . < br />" ;
require ( 'reutilisable.php ' ) ;
echo " Fin du script . < br />" ;
?>
```



Atelier de Développement Web Côté Serveur 39