

Programmation WEB avec PHP

Matière : Programmation WEB avec PHP

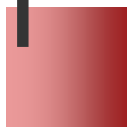
Date : 13.06.2022

Formateur : Malerba Sylvain

Auteur : Malerba Sylvain

SOMMAIRE

01



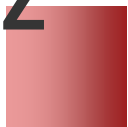
Les tableaux

03



Les constantes

02



Les fonctions

04



Les superglobales

01 Les tableaux

01

Les tableaux

Il existe plusieurs types de tableaux:

- **Tableaux indexés:** les clés sont des nombres
- **Tableaux associatifs:** les clés peuvent être des chaînes de caractères
- **Tableaux multidimensionnels:** un tableau contenant d'autres tableaux

Possibilité d'afficher le contenu d'un tableau avec la fonction `print_r()` et de calculer le nombre d'éléments avec `count()`

01

Les tableaux indexés

```
<?php  
  
    $tableau = array('PHP', 'HTML', 'JAVA');  
  
    echo "Le tableau contient";  
  
    print_r($tableau);  
  
?>
```

Le tableau contient Array ([0] => PHP [1] => HTML [2] => JAVA)

01

Les tableaux indexés

La récupération des données d'un tableau indexé peut se faire de 2 manières:

```
<?php

$tableau = array('PHP', 'HTML', 'JAVA');

echo "Le tableau contient<br>";

echo $tableau[0]."<br>";

echo $tableau[1]."<br>";

echo $tableau[2]."<br>";

?>
```

Récupération valeur par valeur

```
<?php

$tableau = array('PHP', 'HTML', 'JAVA');

echo "Le tableau contient<br>";

foreach ($tableau as $value) {

    echo $value."<br>";

}

?>
```

Parcourir le tableau dans une boucle

01

Exercice

- **A partir de l'exercice précédent, créez un tableau contenant le nom des pilotes.**
- **Afficher le nom de chaque pilote dans chaque TD.**
- **On diminuera le nombre de pilotes à 8**

01

Solution

```
<?php
$nb_pilote = 8;
$pilote = 0;

$liste_nom = array("Hamilton","Vettel","Kubica","Senna","Schumacher","Leclerc","Alonzo","Massa");

$pair = "red";
$impair = "blue";

echo "<table><tr>";

while ( $pilote< 8) {

    if($pilote % 2 != 0){

        echo "<td style='color: ".$pair."'>".$liste_nom[$pilote]."</td></tr>";

    }else{

        echo "<tr><td style='color: ".$impair."'> ".$liste_nom[$pilote]."</td>";

    }

    $pilote ++;

}

echo "</tr></table>";

?>
```

Hamilton	Vettel
Kubica	Senna
Schumacher	Leclerc
Alonzo	Massa

01

Solution

```
<?php
$nb_pilote = 8;
$pilote = 0;

$liste_nom = array("Hamilton", "Vettel", "Kubica", "Senna", "Schumacher", "Leclerc", "Alonzo", "Massa");

$pair = "red";
$impair = "blue";

echo "<table><tr>";

foreach ($liste_nom as $nom_pilote) {
    if($pilote % 2 != 0){
        echo "<td style='color: ".$pair."'>".$liste_nom[$pilote]."</td></tr>";
    }else{
        echo "<tr><td style='color: ".$impair."'> ".$liste_nom[$pilote]."</td>";
    }
    $pilote ++;
}

echo "</tr></table>";

?>
```

Hamilton	Vettel
Kubica	Senna
Schumacher	Leclerc
Alonzo	Massa

01

Les tableaux associatifs

```
<?php

$tableau = array('nom'=>'Malerba', 'prenom'=>'Sylvain','ville' => "Bastia");

echo "Le tableau contient<br>";

foreach ($tableau as $key => $value) {
    echo "<b>".$key."</b>".$value."<br>";
}

?>
```

Le tableau contient
nom: Malerba
prenom: Sylvain
ville: Bastia

01

Les tableaux multidimensionnels

```
<?php

$tableau = [
    "Fruits" =>[
        "fraise",
        "pomme",
        "banane"
    ],
    "Legumes" =>[
        "courgette",
        "aubergine"
    ]
];

print_r($tableau);

echo "<br>Le 2eme fruit est: <b>".$tableau["Fruits"][1]."<b>";

?>
```

Array ([Fruits] => Array ([0] => fraise [1] => pomme [2] => banane) [Legumes] => Array ([0] => courgette [1] => aubergine))
Le 2eme fruit est: **pomme**

01

Exercice

Equipe de Foot:

- **Créez un tableau multidimensionnel comportant le nom des joueurs d'une équipe classé par poste (gardien, défenseur, milieu, attaquant).**
- **Affichez l'ensemble de ces joueurs en les regroupant par poste.**

01

Solution

Gardien: Mandanda

Defenseur: Evra Rami Pavard Roberto Carlos

Milieu: Iniesta Xavi Rabiot

Attaquant: Mbappe Benzema Ronaldo

```
<?php

$list = [
    "Gardien" =>[
        "Mandanda"
    ],
    "Defenseur" =>[
        "Evra",
        "Rami",
        "Pavard",
        "Roberto Carlos"
    ],
    "Milieu" =>[
        "Iniesta",
        "Xavi",
        "Rabiot"
    ],
    "Attaquant" =>[
        "Mbappe",
        "Benzema",
        "Ronaldo"
    ]
];

echo '<table> ';
foreach ($list as $poste => $joueur) {
    echo '<tr><td><b>'. $poste. ': </b><td>';
    foreach ($joueur as $nom) {
        echo '<td>'. $nom. ' <td>';
    }
    echo '</tr>';
}
echo ' </table>';

?>
```

02 Les fonctions

Les fonctions : objectif

- Le principal but d'une fonction est de « regrouper ce qui est commun » .
- Exemple:

```
<?php
```

```
echo "Le cours de <b>PHP</b> va durer au total <b>40</b> heures <br/>";  
echo "Le cours de <b>HTML</b> va durer au total <b>10</b> heures <br/>";  
echo "Le cours de <b>CSS</b> va durer au total <b>5</b> heures <br/>";  
echo "Le cours de <b>JS</b> va durer au total <b>12</b> heures <br/>";  
echo "Le cours de <b>AJAX</b> va durer au total <b>2</b> heures <br/>";
```

```
?>
```

02

Les fonctions

- Une fonction est une série d'instructions que l'on peut appeler autant de fois que l'on souhaite.
- Une fonction prend en paramètre 0, 1, ou plusieurs arguments
- Renvoi ou non une valeur de retour

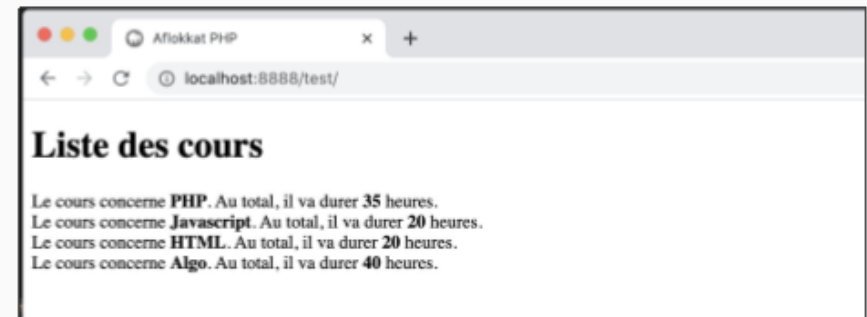
```
<?php  
  
function addition($chiffre1, $chiffre2){  
    $total = $chiffre1+$chiffre2;  
    return $total;  
}  
  
echo addition(1,1);  
  
?>
```


02

Exercice

- Implémenter une fonction `affiche_cours()` comportant 2 paramètres (l'intitulé du cours et le nombre d'heures)
- Exécuter le code ci-dessous pour afficher le résultat suivant :

```
<?php  
  
affiche_cours("PHP", 35);  
affiche_cours("Javascript", 20);  
affiche_cours("HTML", 20);  
affiche_cours("Algo", 40);  
  
?>
```

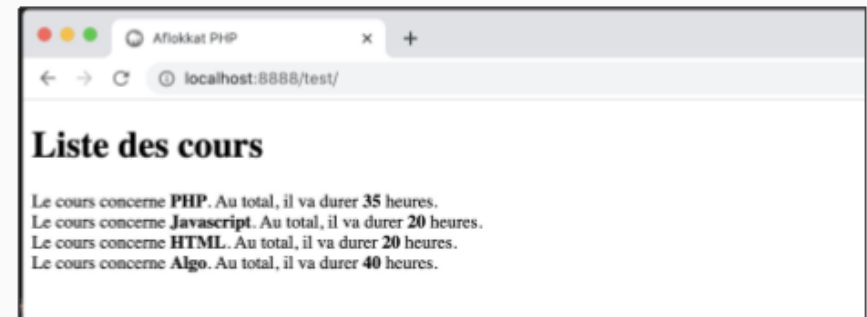


02

Exercice

- Implémenter une fonction `affiche_cours()` comportant 2 paramètres (l'intitulé du cours et le nombre d'heures)
- Exécuter le code ci-dessous pour afficher le résultat suivant :

```
<?php  
  
affiche_cours("PHP", 35);  
affiche_cours("Javascript", 20);  
affiche_cours("HTML", 20);  
affiche_cours("Algo", 40);  
  
?>
```



02

Correction

```
<?php

function affiche_cours($cours, $nb_heures) {

    echo "Le cours concerne <b>$cours</b>. Au total, il va durer <b>$nb_heures</b> heures.<br/>";

}

affiche_cours("PHP", 35);
affiche_cours("Javascript", 20);
affiche_cours("HTML", 20);
affiche_cours("Algo", 40);

?>
```

02

Les fonctions

- **Le nombre d'arguments d'une fonction déclarée doit être le même que lors de son appel**

02

Les fonctions

```
function affiche_cours($cours, $nb_heures) {  
    echo "Le cours concerne <b>$cours</b>. Au total, il va durer <b>$nb_heures</b> heures.<br/>";  
}  
  
affiche_cours("PHP");
```

[!] Fatal error: Uncaught ArgumentCountError: Too few arguments to function affiche_cours(), 1 passed in C:\wamp64\www\php\exo6.php on line 21 and exactly 2 expected in C:\wamp64\www\php\exo6.php on line 15

[!] ArgumentCountError: Too few arguments to function affiche_cours(), 1 passed in C:\wamp64\www\php\exo6.php on line 21 and exactly 2 expected in C:\wamp64\www\php\exo6.php on line 15

Call Stack

#	Time	Memory	Function	Location
1	0.0021	402744	{main}()	...\exo6.php:0
2	0.0021	402744	affiche_cours()	...\exo6.php:21

02

Les fonctions

On peut définir des arguments optionnels (en leur attribuant une valeur par défaut)

```
function affiche_cours($cours, $nb_heures = 35) {  
    echo "Le cours concerne <b>$cours</b>. Au total, il va durer <b>$nb_heures</b> heures.<br/>";  
}  
  
affiche_cours("PHP");
```

02

Les fonctions: la portée des variables

La portée d'un argument ou d'une variable ne dépasse pas le cadre de la fonction

```
$x = 5;

function carre($p) {

    $p = $p*$p;

    echo "Résultat : ".$p."<br>";

    $x = 10;

}

carre($x);
echo "Valeur de x en dehors de la fonction: ".$x."<br>";
echo "Valeur de p: ".$p."<br>";
```

Résultat :25

Valeur de x en dehors de la fonction: 5

(!) Notice: Undefined variable: p in C:\wamp64\www\php\exo6.php on line 27				
Call Stack				
#	Time	Memory	Function	Location
1	0.0078	402920	{main}()	...\exo6.php:0

Valeur de p:

02

Les fonctions: les variables globales

```
$x = 5;  
$message = "Coucou";  
  
function carre($p) {  
    global $message;  
  
    $p = $p*$p;  
  
    echo "Résultat :".$p."<br>";  
  
    $x = 10;  
  
    $message = "Salut";  
  
}  
  
carre($x);  
echo "Le message est le suivant: ".$message."<br>";
```

Résultat :25

Le message est le suivant: Salut

02

Les fonctions: passage par référence

Il est possible de modifier la valeur d'un paramètre de la fonction avec la syntaxe &

```
$x = 3;  
  
function carre(&$p) {  
    $p = $p * $p;  
  
    echo 'Valeur dans la fonction : ' . $p;  
}  
  
carre($x); |  
echo '<br>Valeur en dehors de la fonction : ' . $x;
```

Valeur dans la fonction : 9

Valeur en dehors de la fonction : 9

02

Les fonctions: les valeurs de retour

Une fonction peut également retourner une valeur

```
$x = 3;  
  
function carre($p) {  
    return $p * $p;  
}  
  
$x = carre($x);  
  
echo 'Valeur de $x : ' . $x;
```

Valeur de \$x : 9

02

Les fonctions: les valeurs de retour

Depuis PHP7, on peut déclarer un type de retour

```
$x = 3;  
  
function carre($p) : string {  
    return $p * $p;  
}  
  
$x = carre($x);  
  
echo 'Valeur de $x : ' . $x;
```

02

Exercice

- Définir 2 variables qui ont chacune une valeur au hasard entre 0 et 100
- Une valeur au hasard peut-être renvoyé par la fonction PHP rand(0, 100)
- Créez une fonction qui compare ces 2 valeurs
- Afficher le nombre le plus grands
- Répétez cette opération 10 fois

BONUS: afficher le plus grand nombre qui a été trouvé

02

Solution

```
<?php

function compare($a, $b) {

    $result = $a;

    if($a < $b){

        $result = $b;

    }

    return $result;

}

$max = 0;
for ($i = 0; $i < 10; $i++) {

    $a = rand(0, 100);

    $b = rand(0, 100);

    $biggest = compare($a, $b);

    $max = compare($biggest, $max);
    echo "Le plus grand nombre entre $a et $b est : $biggest <br>";

}
echo "Le plus grand nombre trouvé est " . $max . "<br>";

?>
```

Le plus grand nombre entre 45 et 40 est : 45
Le plus grand nombre entre 29 et 77 est : 77
Le plus grand nombre entre 6 et 3 est : 6
Le plus grand nombre entre 8 et 41 est : 41
Le plus grand nombre entre 67 et 45 est : 67
Le plus grand nombre entre 53 et 33 est : 53
Le plus grand nombre entre 41 et 16 est : 41
Le plus grand nombre entre 5 et 57 est : 57
Le plus grand nombre entre 39 et 34 est : 39
Le plus grand nombre entre 21 et 95 est : 95
Le plus grand nombre trouvé est 95


03 Les constantes

03

Les constantes

- Comme les variables, les constantes sont des conteneurs qui peuvent stocker des string, int, boolean, etc.
- La convention veut qu'on utilise des majuscules
- On ne peut pas modifier sa valeur (contrairement à une variable)
- Mot-clé `define`

```
<?php
define('COURS', 'PHP');
define('COURS', 'Javascript');
echo "<br>Cours: " . COURS;
?>
```

 Notice: Constant COURS already defined in C:\wamp64\www\php\exo_test.php on line 79				
Call Stack				
#	Time	Memory	Function	Location
1	0.0010	402608	{main}()	...\exo_test.php:0
2	0.0010	402640	define ()	...\exo_test.php:79

Cours: PHP

Les constantes: constantes de PHP

- **PHP nous fournit des constantes déjà définies**

```
<?php
    echo "PHP_VERSION : " . PHP_VERSION;
?>
```

PHP_VERSION : 7.2.14

Les constantes: constantes magiques

- Et également des constantes “magiques” qui prendront une valeur relative au contexte

```
<?php
echo 'Numéro de ligne : ' . __LINE__ . '<br>';
echo 'Chemin complet du fichier : ' . __FILE__ . '<br>';
echo 'Dossier contenant le fichier : ' . __DIR__ . '<br>';
echo 'Numéro de ligne : ' . __LINE__ . '<br>';
function test(){
    echo 'Constante appelée depuis la fonction ' . __FUNCTION__;
}
test();
?>
```

Numéro de ligne : 88
Chemin complet du fichier : C:\wamp64\www\php\exo_test.php
Dossier contenant le fichier : C:\wamp64\www\php
Numéro de ligne : 91
Constante appelée depuis la fonction test

04

Les variables superglobales

04

Les variables superglobales

Quelques exemples:

- **\$GLOBALS**
- **\$_SERVER**
- **\$_ENV**
- **\$_GET**
- **\$_POST**
- **\$_REQUEST**
- **\$_FILES**
- **\$_COOKIE**
- **\$_SESSION**


04

\$GLOBALS

```
<?php
$GLOBALS["variable_globale"] = "Coucou";

$variable_globale = "Hello";

function test() {
    echo '$variable_globale appelée sans $GLOBALS = ' . $variable_globale;
    echo '<br>$variable_globale = ' . $GLOBALS["variable_globale"];
}
test();
?>
```

 Notice: Undefined variable: variable_globale in C:\wamp64\www\php\exo_test.php on line 104

Call Stack

#	Time	Memory	Function	Location
1	0.0006	403504	{main}()	...\exo_test.php:0
2	0.0006	403504	test()	...\exo_test.php:110

\$variable_globale appelée sans \$GLOBALS =

\$variable_globale = Hello

\$_SERVER

Tableau qui contient les informations du serveur et de l'exécution du script

```
<?php  
  
    echo "<pre>";  
    print_r($_SERVER);  
    echo "</pre>";  
?>
```

\$ SERVER

```

Array
(
    [HTTP_HOST] => localhost
    [HTTP_CONNECTION] => keep-alive
    [HTTP_CACHE_CONTROL] => max-age=0
    [HTTP_SEC_CH_UA] => " Not A;Brand";v="99", "Chromium";v="102", "Google Chrome";v="102"
    [HTTP_SEC_CH_UA_MOBILE] => ?0
    [HTTP_SEC_CH_UA_PLATFORM] => "Windows"
    [HTTP_UPGRADE_INSECURE_REQUESTS] => 1
    [HTTP_USER_AGENT] => Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0 Safari/537.36
    [HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
    [HTTP_SEC_FETCH_SITE] => none
    [HTTP_SEC_FETCH_MODE] => navigate
    [HTTP_SEC_FETCH_USER] => ?1
    [HTTP_SEC_FETCH_DEST] => document
    [HTTP_ACCEPT_ENCODING] => gzip, deflate, br
    [HTTP_ACCEPT_LANGUAGE] => fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7
    [PATH] => C:\Program Files (x86)\Razer\ChromaBroadcast\bin;C:\Program Files\Razer\ChromaBroadcast\bin;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows
    [SystemRoot] => C:\Windows
    [COMSPEC] => C:\Windows\system32\cmd.exe
    [PATHEXT] => .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
    [WINDIR] => C:\Windows
    [SERVER_SIGNATURE] =>

Apache/2.4.37 (Win64) PHP/7.2.14 Server at localhost Port 80

[SERVER_SOFTWARE] => Apache/2.4.37 (Win64) PHP/7.2.14
[SERVER_NAME] => localhost
[SERVER_ADDR] => ::1
[SERVER_PORT] => 80
[REMOTE_ADDR] => ::1
[DOCUMENT_ROOT] => C:/wamp64/www
[REQUEST_SCHEME] => http
[CONTEXT_PREFIX] =>
[CONTEXT_DOCUMENT_ROOT] => C:/wamp64/www
[SERVER_ADMIN] => wampserver@wampserver.invalid
[SCRIPT_FILENAME] => C:/wamp64/www/php/exo_test.php
[REMOTE_PORT] => 51242
[GATEWAY_INTERFACE] => CGI/1.1
[SERVER_PROTOCOL] => HTTP/1.1
[REQUEST_METHOD] => GET
[QUERY_STRING] =>
[REQUEST_URI] => /php/exo_test.php
[SCRIPT_NAME] => /php/exo_test.php
[PHP_SELF] => /php/exo_test.php
[REQUEST_TIME_FLOAT] => 1655216039.814
[REQUEST_TIME] => 1655216039
)

```

04

`$_ENV`

Tableau qui contient les informations liées à l'environnement dans lequel s'exécute le script

\$_GET, \$_POST et \$_REQUEST

- **\$_GET:** contient les variables passées en méthode GET (affichées dans l'URL)
- **\$_POST:** contient les variables passées en méthode POST (ex: passées par un formulaire)
- **\$_REQUEST:** contient à la fois \$_GET, \$_POST et \$_COOKIE

04

Exercice

- **Créer un formulaire login/password avec bouton d'envoi en HTML**
- **Ce formulaire va renvoyer vers la même page PHP**
- **Comparer login/password avec 2 variables déclarées au préalable**
- **Si OK, afficher « Connexion OK »**
- **SI KO « Erreur email/ mot de passe» et laisser le formulaire affiché.**

04

Solution

```
<?php
$logged = false;
$login = "a@a.fr";
$password = "123456";

if (isset($_POST["login"]) && isset($_POST["password"])) {

    if ($login == $_POST["login"] && $password == $_POST["password"]) {

        $logged = true;

        echo "Connexion OK";

    } else {

        echo "<b>Erreur email/ mot de passe</b>";

    }
    echo "<br><br>";
}
?>

<?php
if (!$logged) {
?>
    <form action="exo8.php" method="post">
        <div>
            <label>Email:</label><br>
            <input type="email" name="login">
        </div>
        <div>
            <label>Mot de passe: </label><br>
            <input type="password" name="password">
        </div>
        <button type="submit">Connexion</button>
    </form>
<?php } ?>
```

04

Gestions des sessions et cookies

- **Permet de stocker des données différentes pour chaque utilisateur en utilisant un identifiant de session unique**
- **Informations stockées côté serveur**
- **Informations conservées lorsqu'un utilisateur navigue d'une page à une autre**
- **Données accessibles à partir de la variable `$_SESSION`**

- Un session se démarre avec la fonction `session_start()` (à appeler en début de chaque fichier qui utilise des sessions)
- On peut récupérer l'id de session unique avec `session_id()`
- On assigne une valeur de session ainsi : `$_SESSION["variable"] = $valeur`
- On détruit la valeur d'une session: `unset($_SESSION["variable"])`
- On détruit toutes les sessions: `session_destroy()`

04

Exercice

- **A partir de l'exercice précédent, stockez en session le login de l'utilisateur une fois connecté**
- **Créez une fonction `is_logged()` qui indique si l'utilisateur est connecté**
- **Afficher un message de bienvenu lorsque l'utilisateur est connecté**
- **Afficher le login de l'utilisateur dans une fonction**
- **L'utilisateur doit rester connecté même au rafraichissement de la page**
- **Mettre un lien de déconnexion (redirection avec paramètre en `$_GET`)**

04

Solution (Partie 1)

```
<head>

<title>Aflokkat PHP</title>

<meta charset="utf-8"> </head>

<body>

<h1>Login</h1>

<?php

$login = "a@a.fr";

$password = "123456";

if (isset($_POST["login"]) && isset($_POST["password"])) {

    if ($login == $_POST["login"] && $password == $_POST["password"]) {

        $_SESSION["login"] = $login;

    } else {

        $error = "<b>Erreur email/ mot de passe</b>";

    }

} elseif (isset($_GET["action"]) && $_GET["action"] == "logout") {

    session_destroy();

    unset($_SESSION);

}

function is_logged() {
```

04

Solution (Partie 2)

```
function is_logged() {  
    return isset($_SESSION["login"]);  
}  
  
function get_login() {  
    if (is_logged()) {  
        return $_SESSION["login"];  
    }  
}  
  
?>  
<?php  
if (!is_logged()) {  
    if (!empty($error)) {  
        echo $error."<br>br>";  
    }  
}  
  
?>
```


04

Solution (Partie 3)

```
<form action="" method="post">

  <div>

    <label>Email address</label><br>

    <input type="email" name="login">

  </div>

  <div>

    <label>Password</label><br>

    <input type="password" name="password">

  </div>

  <button type="submit">Submit</button>

</form>

<?php
}else {
?>
  Bienvenue <b>
    <?php echo get_login(); ?>

    </b> !

    <br>
    <a href="?action=logout">Deconnexion</a>

<?php
}
?>
```

- Fichier texte stocké par le navigateur côté client
- Attention, le cookie peut à tout moment être visualisé, modifié, supprimé par l'utilisateur
- En PHP, les cookies sont accessibles à partir de la variable `$_COOKIE`

- On crée un cookie avec la fonction `setcookie()` avec une date d'expiration dans le futur
- On modifie un cookie avec `setcookie()` en changeant sa valeur
- On supprime un cookie avec `setcookie()` en mettant une date d'expiration dans le passé

04

Les cookies (exemple)

```
setcookie(  
    string $name,  
    string $value = "",  
    int $expires_or_options = 0,  
    string $path = "",  
    string $domain = "",  
    bool $secure = false,  
    bool $httponly = false  
): bool
```

Source

<https://www.php.net/manual/fr/function.setcookie.php>

```
<?php setcookie('user_id', '1234', time()+3600*24, '', '', false, true);?>  
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Aflokkat PHP</title>  
        <meta charset="utf-8">  
    </head>  
    <body>  
        <h1>Cookies</h1>  
        <?php echo "L'id d'utilisateur est " . $_COOKIE["user_id"]; ?>  
    </body>  
</html>
```

Cookies

L'id d'utilisateur est 1234

- Toujours dans le même formulaire, stocker l'utilisateur qui vient de se connecter dans un cookie
 - la valeur du cookie est la concaténation du login et d'un hash sur le password :
 - \$valeur = \$login . "|" . password_hash(\$password, PASSWORD_BCRYPT)
- A l'affichage de la page, si le cookie existe, vérifier que ce qu'il contient est conforme
 - utiliser la fonction PHP password_verify(\$password, \$hash)
- Si c'est le cas, afficher l'utilisateur connecté
 - utiliser la fonction PHP explode() pour transformer une chaîne en tableau en fonction d'un délimiteur
 - cette fonction permettra d'obtenir le login de l'utilisateur à afficher à partir du cookie existant
- Si l'utilisateur se déconnecte, ne pas oublier de supprimer le cookie

04

Solution

```
$login = "a@a.fr";

$password = "123456";

if (isset($_COOKIE["user"])) {

    $tab = explode("|", $_COOKIE["user"]);

    if (password_verify($password, $tab[1])) {

        $_SESSION["login"] = $tab[0];

    }

}

if (isset($_GET["action"]) && $_GET["action"] == "logout") {

    session_destroy();
    setcookie("user", "", 0);
    unset($_SESSION);
    unset($_COOKIE);

}elseif (isset($_POST["login"]) && isset($_POST["password"])) {

    if ($login == $_POST["login"] && $password == $_POST["password"]) {

        $_SESSION["login"] = $login;

        setcookie("user", $login . "|" . password_hash($password, PASSWORD_BCRYPT), time() +
            3600 * 24);

    } else {

        $error = "<b>Erreur email/ mot de passe</b>";

    }

}
```