



Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [PHP et HTML](#) ► [Introduction](#)

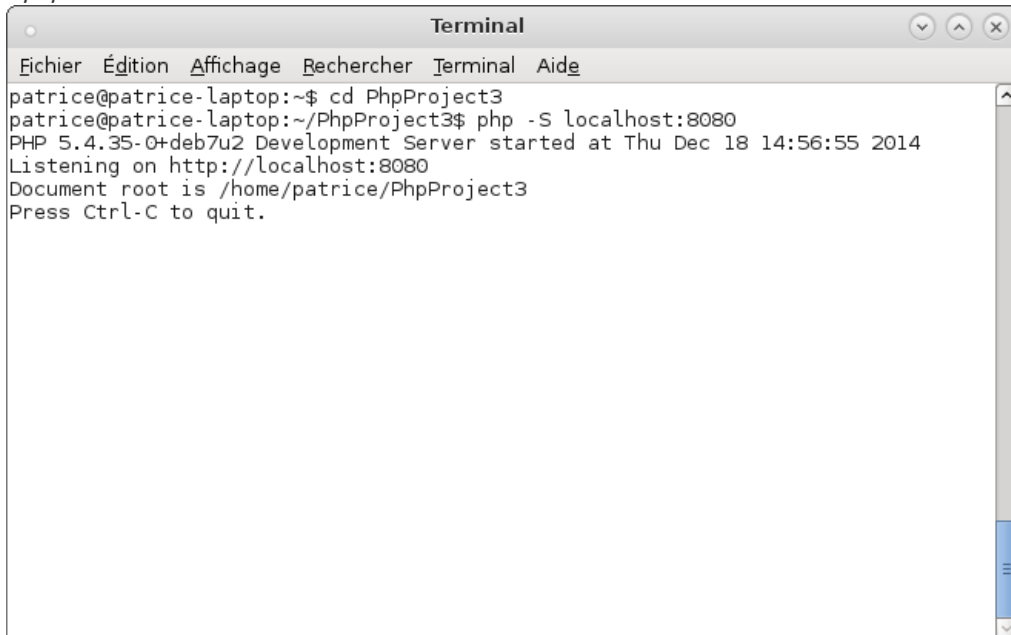
Introduction

PHP est un langage de script qui peut être "inséré" dans du code HTML. Les fichiers contenant du HTML et du code PHP porteront l'extension .php. Le code php pourra être inséré n'importe où dans la page et sera écrit entre tags : `<?php` pour l'ouverture et `?>` pour la fermeture :

```
1 <?php
2 // PHP_et_HTML_1.php
3 ?>
4 <!DOCTYPE html>
5 <html>
6   <head>
7     <meta charset="UTF-8">
8     <title>Cours PHP</title>
9   </head>
10  <body>
11    <?php
12      echo 'ici';
13    ?>
14  </body>
15 </html>
```

Pour visualiser ce fichier dans notre navigateur, nous allons utiliser le serveur web interne de PHP (built-in web server). Ce serveur web ne devra pas être utilisé en production.

Dans un terminal, nous nous plaçons dans le répertoire de notre projet, puis nous démarrons le serveur avec la commande `php -S localhost:8080`.



```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
patrice@patrice-laptop:~$ cd PhpProject3
patrice@patrice-laptop:~/PhpProject3$ php -S localhost:8080
PHP 5.4.35-0+deb7u2 Development Server started at Thu Dec 18 14:56:55 2014
Listening on http://localhost:8080
Document root is /home/patrice/PhpProject3
Press Ctrl-C to quit.
```

Pour accéder à notre script `PHP_et_HTML_1.php` nous saisissons l'url `http://localhost:8080/PHP_et_HTML_1.php` dans notre navigateur. L'url `http://localhost:8080` "pointe" sur le répertoire courant (ici le répertoire de notre projet).

Les structures de contrôle pourront bien-sûr être utilisées en faisant bien attention à séparer le code PHP du code HTML grâce aux tags d'ouverture et de fermeture.

```

1 <?php
2 // PHP_et_HTML_2.php
3 define('MAJORITE', 18);
4 $age = 18;
5 ?>
6 <!DOCTYPE html>
7 <html>
8     <head>
9         <meta charset="UTF-8">
10        <title>Cours PHP</title>
11    </head>
12    <body>
13        <?php if ($age >= MAJORITE) { ?>
14            <h1>Majeur</h1>
15        <?php } else { ?>
16            <h1>Mineur</h1>
17        <?php } ?>
18    </body>
19 </html>

```

Autre exemple avec une boucle for et une liste :

```

1 <?php
2 // PHP_et_HTML_3.php
3 ?>
4 <!DOCTYPE html>
5 <html>
6     <head>
7         <meta charset="UTF-8">
8         <title>Cours PHP</title>
9     </head>
10    <body>
11        <ul>
12            <?php for ($i = 0; $i < 10; $i++) { ?>
13                <li><?php echo $i ?></li>
14            <?php } ?>
15        </ul>
16    </body>
17 </html>

```

Une boucle foreach et un tableau associatif :

```

1 <?php
2 // PHP_et_HTML_4.php
3 $stab = ['pommes' => 10, 'poires' => 4, 'navets' => 12, 'carottes' => 28];
4 ?>
5 <!DOCTYPE html>
6 <html>
7     <head>
8         <meta charset="UTF-8">
9         <title>Cours PHP</title>
10    </head>
11    <body>
12        <table>
13            <thead>
14                <tr>
15                    <th>Aliment</th>
16                    <th>Quantité</th>
17                </tr>
18            </thead>
19            <tbody>
20                <?php foreach ($stab as $key => $value) { ?>
21                    <tr>
22                        <td><?php echo $key ?></td>

```

```
23         <td><?php echo $value ?></td>
24     </tr>
25     <?php } ?>
26 </tbody>
27 </table>
28 </body>
29 </html>
```

[Fin](#)

NAVIGATION



Accueil




■ [Ma page](#)

[Pages du site](#)[Mon profil](#)[Cours actuel](#)

[Intro PHP](#)

[Participants](#)[Le langage PHP : introduction](#)[Les types et les variables](#)[Les opérateurs](#)[Les structures de contrôle](#)[Les structure de données](#)[Les fonctions](#)[Les erreurs](#)[Les fichiers](#)[Les expressions rationnelles](#)[PHP et HTML](#)

Introduction

 [Syntaxe alternative](#) [T.P. tableau de personnes](#) [Les formulaires et les superglobales](#) [Validation des données de formulaire](#) [Détecter les modifications des formulaires](#) [Affichage des données issues d'un formulaire](#) [Les cookies](#) [Les sessions](#) [L'upload de fichier](#) [T.P. formulaire](#) [T.P. formulaire et session](#)[Petite application](#)

[Mes cours](#)

ADMINISTRATION

[Administration du cours](#)[Réglages de mon profil](#)

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))
[Intro PHP](#)



Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [PHP et HTML](#) ► [Syntaxe alternative](#)

Syntaxe alternative

Les structures de contrôle if, for, while, foreach et switch possèdent une syntaxe dite "alternative" souvent utilisée dans les pages HTML. Cette [syntaxe alternative](#) remplace l'accolade de début de bloc par : et l'accolade de fin par un mot clé suivi d'un point virgule. Par exemple la structure if/else devient :

```
if (condition) :  
    instructions;  
else :  
    instructions;  
endif;
```

Exemple :

```
1 <?php  
2 // syntaxe_alternative_1.php  
3 define('MAJORITE', 18);  
4 $age = 18;  
5 ?>  
6 <!DOCTYPE html>  
7 <html>  
8     <head>  
9         <meta charset="UTF-8">  
10        <title>Cours PHP</title>  
11    </head>  
12    <body>  
13        <?php if ($age >= MAJORITE) : ?>  
14            <h1>Majeur</h1>  
15        <?php else : ?>  
16            <h1>Mineur</h1>  
17        <?php endif; ?>  
18    </body>  
19 </html>
```

La boucle for devient :

```
for (initialisation; condition; incrémentation) :  
    instructions;  
endfor ;
```

Exemple :

```
1 <?php  
2 // syntaxe_alternative_2.php  
3 ?>  
4 <!DOCTYPE html>  
5 <html>  
6     <head>  
7         <meta charset="UTF-8">
```

```

8      <title>Cours PHP</title>
9      </head>
10     <body>
11         <ul>
12             <?php for ($i = 0; $i < 10; $i++) : ?>
13                 <li><?php echo $i ?></li>
14             <?php endfor; ?>
15         </ul>
16     </body>
17 </html>

```

La boucle while devient :

```

while (condition) :
    instructions;
endwhile ;

```

La boucle foreach devient :

```

foreach ($tab as $key => $value) :
    instructions ;
endforeach ;

```

Exemple :

```

1  <?php
2  // syntaxe_alternative_3.php
3  $tab = ['pommes' => 10, 'poires' => 4, 'navets' => 12, 'carottes' => 28];
4  ?>
5  <!DOCTYPE html>
6  <html>
7      <head>
8          <meta charset="UTF-8">
9          <title>Cours PHP</title>
10     </head>
11     <body>
12         <table>
13             <thead>
14                 <tr>
15                     <th>Aliment</th>
16                     <th>Quantité</th>
17                 </tr>
18             </thead>
19             <tbody>
20                 <?php foreach ($tab as $key => $value) : ?>
21                     <tr>
22                         <td><?php echo $key ?></td>
23                         <td><?php echo $value ?></td>
24                     </tr>
25                 <?php endforeach; ?>
26             </tbody>
27         </table>
28     </body>
29 </html>

```

Le switch devient :

```

switch (expression) :
    case value:
        instructions_1;
        break ;
    case value2 :

```

```
case valeur2 :  
    instructions_2 ;  
break ;  
default :  
    instructions_3 ;  
endswitch;
```

Exemple :

```
1 <?php  
2  
3 // syntaxe_alternative_4.php  
4 $nb = 10;  
5 switch ($nb) :  
6     case 5:  
7         echo 'petit';  
8         break;  
9     case 10:  
10        echo 'grand';  
11        break;  
12    default:  
13        echo 'inconnu';  
14        break;  
15 endswitch;
```

Fin

NAVIGATION



Accueil

■ Ma page

Pages du site

Mon profil

Cours actuel

Intro PHP

Participants

Le langage PHP : introduction

Les types et les variables

Les opérateurs

Les structures de contrôle

Les structure de données

Les fonctions

Les erreurs

Les fichiers

Les expressions rationnelles


PHP et HTML

 Introduction

 **Syntaxe alternative**

 T.P. tableau de personnes

 Les formulaires et les superglobales

 Validation des données de formulaire


 Détecter les modifications des formulaires

 Affichage des données issues d'un formulaire

 Les cookies

 Les sessions

 L'upload de fichier

 T.P. formulaire

 T.P. formulaire et session

Petite application

Mes cours

ADMINISTRATION

Administration du cours

Réglages de mon profil

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))
[Intro PHP](#)

Introduction au langage de programmation PHP

Accueil ▶ Mes cours ▶ Développement logiciel ▶ Intro PHP ▶ PHP et HTML ▶ T.P. tableau de personnes

T.P. tableau de personnes

Créez dans l'élément body un tableau HTML qui affichera les données du tableau. Les noms des colonnes seront : "Nom", "Prénom", "Age".

```
1 <?php
2
3 $personnages = [
4     ['Sparrow', 'Jack', 45],
5     ['Wayne', 'Bruce', 42],
6     ['Kirk', 'James T.', 57],
7     ['Kent', 'Clark', 40]
8 ];
9 ?>
10 <html>
11     <head>
12         <meta charset="UTF-8">
13         <title></title>
14     </head>
15     <body>
16     </body>
17 </html>
```

Le nom du script sera de la forme "tableau_personnes_<nom_prenom>.php". Vous remettrez une archive zip contenant ce fichier. Le nom de l'archive sera de la forme "tableau_personnes_<nom_prenom>.zip".

État du travail remis

Numéro de tentative	Ceci est la tentative 1.
Statut des travaux remis	Aucune tentative
Statut de l'évaluation	Pas évalué
Dernière modification	vendredi 27 mars 2015, 14:27

Ajouter un travail

Modifier votre travail remis

NAVIGATION

Accueil




■ Ma page

Pages du site

Mon profil

Cours actuel

Intro PHP



Participants
Le langage PHP : introduction
Les types et les variables
Les opérateurs
Les structures de contrôle
Les structure de données
Les fonctions
Les erreurs
Les fichiers
Les expressions rationnelles
PHP et HTML



[Introduction](#)



[Syntaxe alternative](#)



[T.P. tableau de personnes](#)



[Les formulaires et les superglobales](#)



[Validation des données de formulaire](#)



[Détecter les modifications des formulaires](#)



[Affichage des données issues d'un formulaire](#)



[Les cookies](#)



[Les sessions](#)



[L'upload de fichier](#)



[T.P. formulaire](#)



[T.P. formulaire et session](#)

Petite application

[Mes cours](#)

ADMINISTRATION



[Administration du cours](#)

[Réglages de mon profil](#)

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))
[Intro PHP](#)



Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [PHP et HTML](#) ► [Les formulaires et les superglobales](#)

Les formulaires et les superglobales

Les données recueillies dans un formulaire HTML sont envoyées au serveur sous forme d'une requête HTTP. Cette requête est adressée à un script précis et selon une "méthode" HTTP précise (généralement GET ou POST). Le script cible du formulaire est renseigné dans l'attribut action de l'élément form, la méthode est, elle, renseignée dans l'attribut method de l'élément. Le script PHP pourra récupérer et exploiter les données en utilisant des variables spécifiques appelées [superglobales](#). Ces superglobales sont des tableaux associatifs qui vont fournir beaucoup d'informations utiles. Voici les superglobales :

- [\\$_SERVER](#) : contient des informations sur le serveur et sur la requête HTTP qui lui a été adressée
- [\\$_GET](#) : contient les paramètres qui ont été passés en GET
- [\\$_POST](#) : contient les paramètres qui ont été passés en POST
- [\\$_COOKIE](#) : contient les informations des cookies
- [\\$_REQUEST](#) : contient des variables [\\$_GET](#), [\\$_POST](#) et [\\$_COOKIE](#)
- [\\$_SESSION](#) : contient les informations stockées dans la session PHP
- [\\$_FILES](#) : contient les informations sur les fichiers téléchargés
- [\\$_ENV](#) : contient les informations sur l'environnement d'exécution

Puisque les superglobales sont des tableaux associatifs, nous récupérerons les valeurs grâce à des clés. Ainsi la valeur d'un champ input ou textarea, pourra être récupérée, dans la superglobale, avec sa clé. Cette clé correspondra à la valeur de l'attribut name du champ input ou textarea. Par exemple voici un formulaire en get qui possède un champ de saisie login, dans une page saisir_login.html. Ce formulaire a pour cible le script afficher_login.php qui récupérera la valeur du champ dans une variable et qui l'affichera.

```
1 <!DOCTYPE html>
2 <!--formulaires_et_superglobales_1.html-->
3 <html>
4   <head>
5     <meta charset="UTF-8">
6     <title></title>
7   </head>
8   <body>
9     <form action="formulaires_et_superglobales_2.php">
10      <label>Login : <input type="text" id="login" name="login"></label><br>
11      <input type="submit" value="Valider">
12    </form>
13  </body>
14 </html>
```

```
1 <?php
2 //formulaires_et_superglobales_2.php
3 $login = $_GET['login'];
4 ?>
5 <!DOCTYPE html>
6 <html>
7   <head>
8     <meta charset="UTF-8">
9     <title></title>
10  </head>
11  <body>
12    <h4>Le login : <?php echo $login ?></h4>
13  </body>
14 </html>
```

Si le formulaire est envoyé en POST, il faudra bien-sûr utiliser la superglobale \$_POST.

Il est aussi possible de récupérer dans un tableau plusieurs valeurs associées au même attribut name (dans le cas de cases à cocher par exemple). Dans la superglobale, la valeur associée à la clé sera un tableau. Pour cela il suffit d'ajouter des [] à la fin de l'attribut name. Voici un exemple avec des checkboxes :

```

1 <!DOCTYPE html>
2 <!--formulaires_et_superglobales_3.html-->
3 <html>
4   <head>
5     <meta charset="UTF-8">
6     <title></title>
7   </head>
8   <body>
9     <form action="formulaires_et_superglobales_4.php" method="post">
10      Vos langages préférés :<br>
11      <label><input type="checkbox" id="php" name="lang_prog[]"
12        value="php" checked="checked">PHP</label><br>
13      <label><input type="checkbox" id="java" name="lang_prog[]"
14        value="java">Java</label><br>
15      <label><input type="checkbox" id="csharp" name="lang_prog[]"
16        value="csharp">C#</label><br>
17      <input type="submit" value="Valider">
18    </form>
19  </body>
20 </html>

```

Affichage des langages de programmation sélectionnés :

```

1 <?php
2 //formulaires_et_superglobales_4.php
3 $langages = $_POST['lang_prog'];
4 ?>
5 <!DOCTYPE html>
6 <html>
7   <head>
8     <meta charset="UTF-8">
9     <title></title>
10  </head>
11  <body>
12    <h4>Vos langages préférés :</h4>
13    <ul>
14      <?php foreach ($langages as $value) : ?>
15        <li><?php echo $value ?></li>
16      <?php endforeach; ?>
17    </ul>
18  </body>
19 </html>

```

Fin

NAVIGATION

[Accueil](#)

■ [Ma page](#)

Pages du site

Mon profil

Cours actuel

[Intro PHP](#)

Participants

Le langage PHP : introduction

Les types et les variables

Les opérateurs



Les structures de contrôle

Les structure de données

Les fonctions

Les erreurs

Les fichiers

Les expressions rationnelles

PHP et HTML

 [Introduction](#)

 [Syntaxe alternative](#)

 [T.P. tableau de personnes](#)

 **[Les formulaires et les superglobales](#)**

 [Validation des données de formulaire](#)


 [Détecter les modifications des formulaires](#)


 [Affichage des données issues d'un formulaire](#)

 [Les cookies](#)

 [Les sessions](#)

 [L'upload de fichier](#)

 [T.P. formulaire](#)

 [T.P. formulaire et session](#)

Petite application

[Mes cours](#)

ADMINISTRATION



Administration du cours

Réglages de mon profil

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))
[Intro PHP](#)



Introduction au langage de programmation PHP

[Accueil](#) ▶ [Mes cours](#) ▶ [Développement logiciel](#) ▶ [Intro PHP](#) ▶ [PHP et HTML](#) ▶ [Validation des données de formulaire](#)

Validation des données de formulaire

À chaque fois que nous traiterons un formulaire, il faudra s'assurer de plusieurs choses :

1. que les attributs name des champs n'ont pas été modifiés
2. que les attributs value n'ont pas été modifiés
3. que les champs ont bien été remplis
4. que la valeur de ces champs correspond à ce qui est attendu et ne sont pas "nocives"

PHP propose une fonction de "filtrage" `filter_input()` qui nous permettra de tester si un champ existe, si ce champ a bien été rempli et si la valeur de ce champ correspond à ce que l'on attend. La fonction `filter_input()` prendra en premier paramètre une constante qui correspond au type d'input de notre formulaire : `INPUT_GET`, `INPUT_POST`, `INPUT_COOKIE`, `INPUT_SERVER` ou `INPUT_ENV`. Le deuxième paramètre sera le nom du champ, c'est-à-dire la valeur de son attribut name. Le troisième paramètre sera le [type de filtre](#) exprimée par une constante. Puis un tableau associatif d'options. Cette fonction renverra la valeur du champ si tout va bien, null si le champ n'existe pas, false si le champ est vide ou incorrect. Voici un exemple qui vérifiera si le champ de saisi nb est bien un entier compris entre 0 et 10 inclus.

```
1 <!DOCTYPE html>
2 <!--validation_formulaire_1.html-->
3 <html>
4   <head>
5     <meta charset="UTF-8">
6     <title></title>
7   </head>
8   <body>
9     <form action="validation_formulaire_2.php" method="post">
10      <input type="text" id="nb" name="nb"><br>
11      <input type="submit" value="Valider">
12    </form>
13  </body>
14 </html>
```

```
1 <?php
2
3 // validation_formulaire_2.php
4 $nb = filter_input(INPUT_POST, 'nb', FILTER_VALIDATE_INT,
5   ['options' => ['min_range' => 0, 'max_range' => 10]]);
6 if (!isset($nb)) {
7   echo 'le formulaire a été modifié';
8 } elseif ($nb === false) {
9   echo '$nb est vide ou incorrecte';
10 } else {
11   echo "\$nb est correcte et a pour valeur $nb";
12 }
```

La fonction `filter_input()` permet de valider des inputs selon plusieurs filtres : booléens, d'entiers, de décimaux, d'emails, d'adresses IP, d'URL ou d'expressions rationnelles.

Nous allons créer un formulaire dans une page `formulaire.php`. Nous y saisisons un login et un password. Ce formulaire aura pour cible le script `check_form.php` qui vérifiera si :

- le login contient des lettres, des chiffres ou des underscores et possède entre 4 et 12 caractères
- le password contient des lettres, des chiffres et possède entre 4 et 8 caractères

Si les champs sont valides, le script affichera : "le formulaire est OK !". Si les champs sont vides ou incorrects, le script ré-affichera le formulaire de saisie avec une liste des erreurs.

```

1 <!DOCTYPE html>
2 <!--validation_formulaire_3.php-->
3 <html>
4   <head>
5     <meta charset="UTF-8">
6     <title></title>
7   </head>
8   <body>
9     <?php if (isset($messages) && !empty($messages)) : ?>
10      <ul>
11        <?php foreach ($messages as $item) : ?>
12          <li><?php echo $item ?></li>
13        <?php endforeach; ?>
14      </ul>
15    <?php endif; ?>
16    <form action="validation_formulaire_4.php" method="post">
17      <label>
18        Login : <input type="text" id="login" name="login">
19      </label>
20      <label>Password : <input type="password" id="password"
21                        name="password">
22      </label>
23      <input type="submit" value="Valider">
24    </form>
25  </body>
26 </html>

```

À la première exécution, la variables messages n'existera pas et le contenu du if ne sera donc pas exécuté.

```

1 <?php
2
3 // validation_formulaire_4.php
4 $login = filter_input(INPUT_POST, 'login', FILTER_VALIDATE_REGEXP,
5   ['options' => ['regexp' => '/^\w{4,12}$/']]);
6 $password = filter_input(INPUT_POST, 'password', FILTER_VALIDATE_REGEXP,
7   ['options' => ['regexp' => '/^[a-zA-Z0-9]{4,8}$/']]);
8 $messages = [];
9 if (!$login) {
10   $messages[] = 'Le login est incorrect';
11 }
12 if (!$password) {
13   $messages[] = 'Le password est incorrect';
14 }
15 if (empty($messages)) {
16   echo 'le formulaire est OK !';
17 } else {
18   include './validation_formulaire_3.php';
19 }

```

Détaillons ce script :

```

1 <?php
2
3 $login = filter_input(INPUT_POST, 'login', FILTER_VALIDATE_REGEXP,
4   ['options' => ['regexp' => '/^\w{4,12}$/']]);

```

Cette expression assignera à login la valeur du champ si la valeur du champ est validée, false si la valeur du champ est invalide ou vide, null si le champ n'existe pas. La validation se fera selon une expression rationnelle, le type de filtre est donc FILTER_VALIDATE_REGEXP. Cette expression rationnelle est passée en argument dans un tableau associatif.

```

1 <?php
2
3 $password = filter_input(INPUT_POST, 'password', FILTER_VALIDATE_REGEXP,
4     ['options' => ['regexp' => '/^[a-zA-Z0-9]{4,8}$/']]);

```

Même principe de validation pour le password. La variable password vaudra donc la valeur du champ, ou false ou null.

```

1 <?php
2
3 $messages = [];

```

Nous créons un tableau qui contiendra les messages d'erreurs.

```

1 <?php
2
3 if (!$login) {
4     $messages[] = 'Le login est incorrect';
5 }

```

Si la valeur de login peut être évaluée à false alors nous ajoutons un message au tableau. À ce moment de l'exécution la valeur de la variable login peut être de trois types :

- si le champ est valide alors la valeur de login est la chaîne de caractère saisie par l'utilisateur
- si le champ est invalide, login a pour valeur false
- si le champ est vide, login a pour valeur null

Une chaîne de caractère non-vide est considérée comme vraie, les valeurs null et false sont considérées comme false. Donc si le champ est invalide ou manquant, la valeur de login pourra être considérée comme false et nous entrerons dans le if. Si la valeur de login est une chaîne de caractère non-vide, elle sera considérée comme vraie et nous n'entrerons pas dans le if. Dans le if, un message est ajouté au tableau messages.

```

1 <?php
2
3 if (!$password) {
4     $messages[] = 'Le password est incorrect';
5 }

```

Nous appliquons le même traitement à la variable password.

```

1 <?php
2
3 if (empty($messages)) {
4     echo 'le formulaire est OK !';
5 } else {
6     include './validation_formulaire_3.php';
7 }

```

Si le tableau de message est vide, cela signifie que les champs sont valides et que le formulaire est correct. Sinon nous incluons le script formulaire.php. Cette fois, dans formulaire.php, la variable messages existe bien puisqu'elle a été définie dans le script "incluant". Si cette variable n'est pas vide alors nous créons une liste non-ordonnée et nous bouclons sur le tableau pour afficher chaque item.

Les messages d'erreur peuvent aussi être affichés après les champs concernés, ou après, selon le design choisi. Par exemple :

```

1 <!DOCTYPE html>
2 <!-- validation_formulaire_5.php-->
3 <html>
4     <head>
5         <meta charset="UTF-8">
6         <title></title>

```

```

7      </head>
8      <body>
9          <form action="validation_formulaire_6.php" method="post">
10             <label>
11                 Login : <input type="text" id="login" name="login">
12             </label>
13             <?php if (isset($login_message)) echo $login_message ?><br>
14             <label>Password : <input type="password" id="password"
15                                 name="password">
16             </label>
17             <?php if (isset($password_message)) echo $password_message ?><br>
18             <input type="submit" value="Valider">
19         </form>
20     </body>
21 </html>

```

```

1 <?php
2
3 // validation_formulaire_6.php
4 $login = filter_input(INPUT_POST, 'login', FILTER_VALIDATE_REGEXP,
5     ['options' => ['regexp' => '/^\w{4,12}$/']]);
6 $password = filter_input(INPUT_POST, 'password', FILTER_VALIDATE_REGEXP,
7     ['options' => ['regexp' => '/^[a-zA-Z0-9]{4,8}$/']]);
8 $is_valid = true;
9 if (!$login) {
10     $login_message = 'Le login est incorrect';
11     $is_valid = false;
12 }
13 if (!$password) {
14     $password_message = 'Le password est incorrect';
15     $is_valid = false;
16 }
17 if ($is_valid) {
18     echo 'le formulaire est OK !';
19 } else {
20     include './validation_formulaire_5.php';
21 }

```

Fin

NAVIGATION



Accueil

■ Ma page

Pages du site

Mon profil

Cours actuel

Intro PHP

Participants

Le langage PHP : introduction

Les types et les variables

Les opérateurs

Les structures de contrôle

Les structure de données

Les fonctions

Les erreurs


Les fichiers










Les expressions rationnelles

PHP et HTML

 Introduction

 Syntaxe alternative

 T.P. tableau de personnes

-  [Les formulaires et les superglobales](#)
-  **[Validation des données de formulaire](#)**
-  [Détecter les modifications des formulaires](#)
-  [Affichage des données issues d'un formulaire](#)
-  [Les cookies](#)
-  [Les sessions](#)
-  [L'upload de fichier](#)
-  [T.P. formulaire](#)
-  [T.P. formulaire et session](#)

[Petite application](#)

[Mes cours](#)

ADMINISTRATION



[Administration du cours](#)

[Réglages de mon profil](#)

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))
[Intro PHP](#)



Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [PHP et HTML](#) ► [Détecter les modifications des formulaires](#)

Détecter les modifications des formulaires


Comme le contenu HTML de votre page peut être modifié par le client, il est parfois nécessaire de vérifier si les valeurs des champs d'un formulaire n'ont pas été modifiées. Ceci s'applique aux listes, aux comboboxes, aux checkboxes. Une technique très simple consiste en la comparaison des valeurs issues du formulaire avec une liste de valeurs possibles : c'est la liste blanche.

```
1 <!DOCTYPE html>
2 <!--modification_formulaire_1.html-->
3 <html>
4   <head>
5     <meta charset="UTF-8">
6     <title></title>
7   </head>
8   <body>
9     <form action="modifications_formulaire_2.php" method="post">
10      Vos langages préférés :<br>
11      <label><input type="checkbox" id="php" name="lang_prog[]"
12        value="php" checked="checked">PHP</label><br>
13      <label><input type="checkbox" id="java" name="lang_prog[]"
14        value="java">Java</label><br>
15      <label><input type="checkbox" id="csharp" name="lang_prog[]"
16        value="csharp">C#</label><br>
17      <input type="submit" value="Valider">
18    </form>
19  </body>
20 </html>
```

```
1 <?php
2
3 // modifications_formulaire_2.php
4 $white_list = ['java', 'php', 'csharp'];
5 if (isset($_POST['lang_prog'])) {
6   $stab = $_POST['lang_prog'];
7   if (array_intersect($stab, $white_list) == $stab) {
8     echo 'c\'est glob';
9   } else {
10    echo 'c\'est pas glob';
11  }
12 } else {
13   echo 'rien...';
14 }
```

Nous avons créé une liste des valeurs possibles dans la variable `white_list`. La fonction `array_intersect()` renvoie dans un tableau les valeurs du tableau en premier argument qui sont aussi dans le tableau en deuxième argument. Nous récupérons ainsi un tableau des valeurs communes entre le tableau issu du formulaire et la `white_list`. Si, le tableau renvoyé par `array_intersect()` est équivalent au tableau issu du formulaire, les valeurs sont correctes.

Fin

[Ma page](#)[Pages du site](#)[Mon profil](#)[Cours actuel](#)[Intro PHP](#)[Participants](#)[Le langage PHP : introduction](#)[Les types et les variables](#)[Les opérateurs](#)[Les structures de contrôle](#)[Les structure de données](#)[Les fonctions](#)[Les erreurs](#)[Les fichiers](#)[Les expressions rationnelles](#)[PHP et HTML](#) [Introduction](#) [Syntaxe alternative](#) [T.P. tableau de personnes](#) [Les formulaires et les superglobales](#) [Validation des données de formulaire](#) [**Détecter les modifications des formulaires**](#) [Affichage des données issues d'un formulaire](#) [Les cookies](#) [Les sessions](#) [L'upload de fichier](#) [T.P. formulaire](#) [T.P. formulaire et session](#)[Petite application](#)[Mes cours](#)**ADMINISTRATION**[Administration du cours](#)[Réglages de mon profil](#)Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))[Intro PHP](#)



Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [PHP et HTML](#) ► [Affichage des données issues d'un formulaire](#)

Affichage des données issues d'un formulaire

L'affichage de données issues d'un formulaire ou d'une base de données ou d'une autre ressource peut être la source d'attaques, notamment l'injection de script. Pour éviter ces injections il suffit de demander à PHP de convertir certains caractères en entité HTML. Ainsi les caractères seront affichés correctement mais ne seront plus évalué comme les caractères d'un script. La fonction `htmlentities()` permet de convertir en entité HTML tous les caractères qui peuvent l'être. La fonction `htmlspecialchars()` permet de convertir en entité HTML tous les caractères spéciaux.

Voici un exemple simplissime d'injection de script :

```
1 <!DOCTYPE html>
2 <!--affichage_donnees_1.html-->
3 <html>
4   <head>
5     <meta charset="UTF-8">
6     <title></title>
7   </head>
8   <body>
9     <form action="affichage_donnees_2.php" method="post">
10      <label>Login : <input type="text" id="login" name="login"></label><br>
11      <input type="submit" value="Valider">
12    </form>
13  </body>
14 </html>
```

```
1 <?php
2
3 // affichage_donnees_2.php
4 echo $_POST['login'];
```

Nous affichons juste la valeur de la saisie... Sans aucune validation... Je peux réaliser une petite injection en écrivant `<script>alert('Injection !')</script>` dans le champ texte. Et lorsque je soumetts mon formulaire, une belle pop-up apparaît.

Pour empêcher cette injection je vais utiliser `htmlentities()` pour convertir les caractères.

```
1 <?php
2
3 // affichage_donnees_3.php
4 echo htmlentities($_POST['login']);
```

Cette fois la chaîne `<script>alert('ici')</script>` est écrite en toutes lettres mais le code n'a pas été exécuté.

Fin

NAVIGATION

[Accueil](#)

■ [Ma page](#)

Pages du site

Mon profil

Cours actuel



Intro PHP

Participants

Le langage PHP : introduction

Les types et les variables

Les opérateurs

Les structures de contrôle

Les structure de données

Les fonctions

Les erreurs

Les fichiers

Les expressions rationnelles

PHP et HTML

 [Introduction](#)

 [Syntaxe alternative](#)

 [T.P. tableau de personnes](#)

 [Les formulaires et les superglobales](#)

 [Validation des données de formulaire](#)


 [Détecter les modifications des formulaires](#)

 [Affichage des données issues d'un formulaire](#)

 [Les cookies](#)

 [Les sessions](#)

 [L'upload de fichier](#)

 [T.P. formulaire](#)

 [T.P. formulaire et session](#)

Petite application

[Mes cours](#)

ADMINISTRATION



[Administration du cours](#)

[Réglages de mon profil](#)

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))
[Intro PHP](#)



Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [PHP et HTML](#) ► [Les cookies](#)

Les cookies

Les cookies sont des informations sous forme de texte, qui sont stockées sur le navigateur du client sous forme d'ensembles clé=valeur, et qui permettent l'identification de ce dernier. Le cookie à conserver est envoyé dans l'entête de la réponse HTTP au client, puis à chaque requête du client, le cookie sera présent dans l'entête de la requête. Le client peut modifier et même supprimer le cookie qu'il conserve.

"En plus de la paire nom/valeur, un cookie peut aussi contenir une date d'expiration, un chemin, un nom de domaine et le type de connexion prévu, c'est-à-dire normale ou chiffrée." ([Wikipédia, article cookie](#)). La fonction `setcookie()` permet de créer un cookie à envoyer au client. Si aucune date d'expiration n'est indiquée, le cookie sera détruit à la fermeture du navigateur. La date est exprimée sous forme d'un timestamp unix, c'est à dire le nombre de secondes écoulées depuis le 1er janvier 1970 à minuit (epoch). L'appel de `setcookie()` avec un timestamp antérieur au timestamp du jour permet de supprimer un cookie. La date d'expiration peut être exprimée grâce à la fonction `time()` + nombre de secondes avant expiration, ou encore avec `mktime()`. L'appel à `setcookie()` doit se faire avant tout affichage, c'est-à-dire avant que les entêtes soient envoyées.

Les cookies peuvent être récupérés via les superglobales `$_COOKIE` ou `$_REQUEST`.

Voici un exemple d'une page qui envoie un cookie au client, et d'une autre qui affiche le cookie s'il y en a un et qui le supprime la cas échéant :

```
1 <?php
2 // cookies_1.php
3 setcookie('mon_cookie', 'machin');
4 ?>
5 <!DOCTYPE html>
6 <html>
7     <head>
8         <meta charset="UTF-8">
9         <title></title>
10    </head>
11    <body>
12        <h3>Votre cookie a été envoyé !</h3>
13        <a href="cookies_2.php">Vérifier votre cookie !</a>
14    </body>
15 </html>
```

```
1 <?php
2 // cookies_2.php
3 $cookie_value = 'pas de cookie';
4 if (isset($_COOKIE['mon_cookie'])) {
5     // récupération de la valeur du cookie
6     $cookie_value = $_COOKIE['mon_cookie'];
7     // destruction du cookie
8     setcookie('mon_cookie', '', time() - 3600);
9 }
10 ?>
11 <!DOCTYPE html>
12 <html>
13     <head>
14         <meta charset="UTF-8">
15         <title></title>
16    </head>
17    <body>
18        <?php echo "votre cookie est : $cookie_value" ?>
19    </body>
```

20 </html>

Fin

NAVIGATION



Accueil

■ Ma page

[Pages du site](#)[Mon profil](#)[Cours actuel](#)[Intro PHP](#)[Participants](#)[Le langage PHP : introduction](#)[Les types et les variables](#)[Les opérateurs](#)[Les structures de contrôle](#)[Les structure de données](#)[Les fonctions](#)[Les erreurs](#)[Les fichiers](#)[Les expressions rationnelles](#)[PHP et HTML](#) [Introduction](#) [Syntaxe alternative](#) [T.P. tableau de personnes](#) [Les formulaires et les superglobales](#) [Validation des données de formulaire](#) [Détecter les modifications des formulaires](#) [Affichage des données issues d'un formulaire](#) [Les cookies](#) [Les sessions](#) [L'upload de fichier](#) [T.P. formulaire](#) [T.P. formulaire et session](#)[Petite application](#)[Mes cours](#)

ADMINISTRATION

[Administration du cours](#)[Réglages de mon profil](#)

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))
[Intro PHP](#)



Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [PHP et HTML](#) ► [Les sessions](#)

Les sessions

Les [sessions](#) permettent de conserver des données sur le client, mais côté serveur. Un identifiant de session est attribué au client, souvent sous forme d'un cookie nommé par défaut PHPSESSID (défini dans la directive session.name du php.ini). À chaque fois que le client se connecte, ce cookie est récupéré par PHP et permet le chargement des données de session de l'utilisateur. Ces données peuvent être récupérées dans le tableau associatif \$_SESSION, qui est une superglobale.

Pour mettre une donnée en session, il suffit d'ajouter un couple clé/valeur à la superglobale \$_SESSION. Avant d'ajouter ou de récupérer des données en session, il faut appeler la fonction [session_start\(\)](#). Par défaut, les données de session sont sérialisées et conservées sous forme de fichier dans un répertoire indiqué par la directive session.save_path du fichier php.ini. Les [variables](#) de types resource ne peuvent donc être conservées en session. La durée de vie en secondes de la session est définie par la directive session.gc_maxlifetime du php.ini. La fonction [session_start\(\)](#) permet de démarrer la session. La fonction [session_status\(\)](#) renvoie le statut de la session.

Voici un script qui teste si une variable user a été insérée en session. Si cette variable existe, les valeurs associées aux clés "name" et "family_name" sont affichées, sinon un formulaire de connexion est affiché.

```
1  <?php
2  // sessions_1.php
3  session_start();
4  $user = null;
5  if (isset($_SESSION['user'])) {
6      $user = $_SESSION['user'];
7  }
8  ?>
9  <!DOCTYPE html>
10 <html>
11     <head>
12         <meta charset="UTF-8">
13         <title></title>
14     </head>
15     <body>
16         <header>
17             <?php if ($user) : ?>
18                 <form action="sessions_2.php">
19                     Bienvenue
20                     <?php
21                         echo "{$_user['name']} {$_user['family_name']}";
22                     ?>
23                     <input type="submit" value="se déconnecter">
24                 </form>
25             <?php else : ?>
26                 <form action="sessions_3.php" method="post">
27                     <label>
28                         Login : <input type="text" id="login" name="login">
29                     </label>
30                     <label>Password : <input type="password" id="password"
31                                     name="password">
32                     </label>
33                     <input type="submit" value="se connecter">
34                 </form>
35             <?php endif; ?>
36         </header>
37     </body>
```



```
38 </html>
```

```
1 <?php
2
3 // sessions_3.php
4 $login = $_POST['login'];
5 $password = $_POST['password'];
6 if ($login == 'Jack' && $password == 'Sparrow') {
7     session_start();
8     $_SESSION['user'] = ['name' => 'Jack', 'family_name' => 'Sparrow'];
9 }
10 header('location: sessions_1.php');
```

Le script de connexion récupère le login et le password transmis en post. Si le login et le password sont corrects, une session est démarrée et une variable user de type tableau associatif est insérée en session. Ensuite nous redirigeons vers la page sessions_1.php avec la fonction `header()`. La fonction `header()` permet de définir des en-têtes de la requête HTTP. et doit être appelée avant que les en-têtes aient été envoyées, c'est-à-dire avant tout affichage.

deconnexion.php

```
1 <?php
2
3 // sessions_2.php
4 session_start();
5 $user = null;
6 if (isset($_SESSION['user'])) {
7     unset($_SESSION['user']);
8 }
9 header('location: sessions_1.php');
```

Le script deconnexion.php supprime la clé user du tableau de session et redirige vers sessions_1.php.

Fin

NAVIGATION

Accueil

■ Ma page

Pages du site

Mon profil

Cours actuel

Intro PHP

Participants

Le langage PHP : introduction

Les types et les variables

Les opérateurs

Les structures de contrôle

Les structure de données

Les fonctions


Les erreurs

Les fichiers

Les expressions rationnelles


PHP et HTML

 Introduction


 Syntaxe alternative

 T.P. tableau de personnes

 Les formulaires et les superglobales

 Validation des données de formulaire

 Détecter les modifications des formulaires

-  [Affichage des données issues d'un formulaire](#)
-  [Les cookies](#)
-  **[Les sessions](#)**
-  [L'upload de fichier](#)
-  [T.P. formulaire](#)
-  [T.P. formulaire et session](#)

Petite application

[Mes cours](#)

ADMINISTRATION



[Administration du cours](#)

[Réglages de mon profil](#)

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))
[Intro PHP](#)



Introduction au langage de programmation PHP

[Accueil](#) ▶ [Mes cours](#) ▶ [Développement logiciel](#) ▶ [Intro PHP](#) ▶ [PHP et HTML](#) ▶ [L'upload de fichier](#)

L'upload de fichier

Pour réaliser un upload de fichiers sur le serveur il faut tout d'abord que le formulaire utilise la méthode POST et que l'encodage de ses données soit de type "multipart/form-data". Par défaut le type d'encodage des formulaires est "application/x-www-form-urlencoded". La sélection du fichier se fera via un champ input de type file.

```
1 <!DOCTYPE html>
2 <!--upload_1.html-->
3 <html>
4     <head>
5         <meta charset="UTF-8">
6         <title></title>
7     </head>
8     <body>
9         <form action="upload_2.php" method="post" enctype="multipart/form-data">
10            <label>Votre fichier :
11                <input type="file" id="my_file" name="my_file">
12            </label>
13            <input type="submit" value="Valider">
14        </form>
15    </body>
16 </html>
```

La récupération des informations sur le fichier uploadé se fera via la superglobe \$_FILES.

```
1 <?php
2
3 // upload_2.php
4 var_dump($_FILES['my_file']);
```

La superglobale nous permettra de récupérer les informations suivantes sur le fichier :

- son nom
- son type MIME
- le nom du fichier temporaire créé par PHP
- le code erreur
- sa taille

À partir de ces informations il est possible de filtrer les fichiers selon leur type, leur taille etc... Pour l'instant PHP n'a fait que créer un fichier temporaire, ce fichier peut être maintenant déplacé dans un répertoire de notre choix grâce à la fonction `move_uploaded_file()`. La fonction `move_uploaded_file()` prend en paramètre le chemin vers le fichier à déplacer et le chemin vers le fichier cible.

Nous créons à la racine de notre projet un répertoire uploads qui contiendra les fichiers "téléversés".

```
1 <?php
2
3 // upload_3.php
4 if (isset($_FILES['my_file']) && $_FILES['my_file']['error'] == UPLOAD_ERR_OK) {
5     $upload_file = './uploads/'
6         . $_FILES['my_file']['name'];
7     move_uploaded_file($_FILES['my_file']['tmp_name'], $upload_file);
8 }
```

Autre exemple avec une petite vérification de l'extension du fichier uploadé et un message d'erreur stocké en

session :

```

1  <?php
2  // upload_4.php
3  session_start();
4  ?>
5  <!DOCTYPE html>
6  <html>
7      <head>
8          <meta charset="UTF-8">
9          <title></title>
10     </head>
11     <body>
12         <h3>
13             <?php
14                 if (isset($_SESSION['message'])):
15                     echo $_SESSION['message'];
16                 endif;
17             ?>
18         </h3>
19         <form action="upload_5.php" method="post" enctype="multipart/form-data">
20             <label>Votre fichier :
21                 <input type="file" id="my_file" name="my_file">
22             </label>
23             <input type="submit" value="Valider">
24         </form>
25     </body>
26 </html>

```

```

1  <?php
2
3  // upload_5.php
4  session_start();
5  if (isset($_SESSION['message'])) {
6      unset($_SESSION['message']);
7  }
8
9  if (isset($_FILES['my_file']) && $_FILES['my_file']['error'] == UPLOAD_ERR_OK) {
10     $file_name = $_FILES['my_file']['name'];
11     $file_extension = strrchr($file_name, '.');
12     if (in_array($file_extension, ['.pdf', '.txt', '.odt', '.doc'])) {
13         $upload_file = './uploads/'
14             . $_FILES['my_file']['name'];
15         if (move_uploaded_file($_FILES['my_file']['tmp_name'], $upload_file)) {
16             echo 'upload effectué !';
17         } else {
18             echo 'un problème est survenu pendant l\'upload';
19         }
20     } else {
21         $_SESSION['message'] = 'Type de fichier incorrect';
22         header('location:upload_4.php');
23     }
24 }

```

Deux directives du php.ini permettent de gérer la taille maximum des fichiers que l'on peut uploader :

- post_max_size : "définit la taille maximale des données reçues par la méthode post"
- upload_max_filesize : définit la taille maximale en octet d'un fichier à charger

Fin

NAVIGATION

[Accueil](#)

■ [Ma page](#)



Pages du site

Mon profil

Cours actuel

[Intro PHP](#)

Participants

Le langage PHP : introduction

Les types et les variables

Les opérateurs

Les structures de contrôle

Les structure de données

Les fonctions

Les erreurs

Les fichiers

Les expressions rationnelles

PHP et HTML

 [Introduction](#)

 [Syntaxe alternative](#)

 [T.P. tableau de personnes](#)

 [Les formulaires et les superglobales](#)

 [Validation des données de formulaire](#)


 [Détecter les modifications des formulaires](#)


 [Affichage des données issues d'un formulaire](#)

 [Les cookies](#)

 [Les sessions](#)

 [L'upload de fichier](#)

 [T.P. formulaire](#)

 [T.P. formulaire et session](#)

Petite application

[Mes cours](#)

ADMINISTRATION



Administration du cours

Réglages de mon profil

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))

[Intro PHP](#)



Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [PHP et HTML](#) ► [T.P. formulaire](#)

T.P. formulaire

Créez dans une page formulaire.php un formulaire HTML avec :

- un champ de texte pour le nom
- un champ de texte pour le prénom
- un champ de texte pour l'age
- deux radio buttons pour le genre : féminin, masculin
- trois checkboxes pour les langages de programmation : PHP, Java, C#
- une combobox pour l'année de formation : 2012, 2013 ou 2014

Ce formulaire aura pour cible un script check_form.php qui validera le formulaire, et affichera les valeurs saisies dans le formulaire.

Vous implémenterez une validation des champs :

- le nom et le prénom ne devront contenir que des lettres majuscules, minuscules et des tirets
- l'age devra être un entier entre 0 et 150
- les valeurs des autres champs ne devront pas être modifiées par l'utilisateur.

Vous implémenterez un affichage des messages d'erreurs dans le formulaire.

Les scripts seront à remettre dans une archive zip dont le nom sera de la forme "formulaire_*nom_prenom*.zip".

État du travail remis

Numéro de tentative	Ceci est la tentative 1.
Statut des travaux remis	Aucune tentative
Statut de l'évaluation	Pas évalué
Dernière modification	vendredi 27 mars 2015, 14:28

Ajouter un travail

Modifier votre travail remis

NAVIGATION

[Accueil](#)

■ [Ma page](#)

Pages du site

Mon profil

Cours actuel

[Intro PHP](#)

Participants

Le langage PHP : introduction

Les types et les variables

Les opérateurs

Les structures de contrôle

Les structure de données



[Les fonctions](#)[Les erreurs](#)[Les fichiers](#)[Les expressions rationnelles](#)[PHP et HTML](#) [Introduction](#) [Syntaxe alternative](#) [T.P. tableau de personnes](#) [Les formulaires et les superglobales](#) [Validation des données de formulaire](#) [Détecter les modifications des formulaires](#) [Affichage des données issues d'un formulaire](#) [Les cookies](#) [Les sessions](#) [L'upload de fichier](#) **[T.P. formulaire](#)** [T.P. formulaire et session](#)[Petite application](#)[Mes cours](#)

ADMINISTRATION

[Administration du cours](#)[Réglages de mon profil](#)

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))

[Intro PHP](#)



Introduction au langage de programmation PHP

[Accueil](#) ▶ [Mes cours](#) ▶ [Développement logiciel](#) ▶ [Intro PHP](#) ▶ [PHP et HTML](#) ▶ [T.P. formulaire et session](#)

T.P. formulaire et session

Modifiez les scripts précédant pour que les messages d'erreurs soient stockés en session. En cas d'erreurs, le script `script check_form.php` redirigera vers la page `formulaire.php` en utilisant la fonction `header()`.

Les scripts seront à remettre dans une archive zip dont le nom sera de la forme "formulaire_session_*nom_prenom*.zip".

État du travail remis

Numéro de tentative	Ceci est la tentative 1.
Statut des travaux remis	Aucune tentative
Statut de l'évaluation	Pas évalué
Dernière modification	vendredi 27 mars 2015, 14:28

Ajouter un travail

Modifier votre travail remis

NAVIGATION



Accueil

■ [Ma page](#)

[Pages du site](#)

[Mon profil](#)

[Cours actuel](#)

[Intro PHP](#)

[Participants](#)

[Le langage PHP : introduction](#)

[Les types et les variables](#)

[Les opérateurs](#)

[Les structures de contrôle](#)

[Les structure de données](#)

[Les fonctions](#)

[Les erreurs](#)


[Les fichiers](#)

[Les expressions rationnelles](#)

[PHP et HTML](#)

 [Introduction](#)

 [Syntaxe alternative](#)





 [T.P. tableau de personnes](#)

 [Les formulaires et les superglobales](#)

 [Validation des données de formulaire](#)

 [Détecter les modifications des formulaires](#)

 [Affichage des données issues d'un formulaire](#)

-  [Les cookies](#)
-  [Les sessions](#)
-  [L'upload de fichier](#)
-  [T.P. formulaire](#)
-  **[T.P. formulaire et session](#)**

[Petite application](#)

[Mes cours](#)

ADMINISTRATION



[Administration du cours](#)

[Réglages de mon profil](#)

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))
[Intro PHP](#)