

Les fonctions internes de PHP

Table des matières

I. Contexte	3
II. Les fonctions internes de PHP	3
III. Exercice : Appliquez la notion	5
IV. Utiliser les fonctions internes de PHP	6
V. Exercice : Appliquez la notion	10
VI. Auto-évaluation	11
A. Exercice final	11
B. Exercice : Défi.....	13
Solutions des exercices	14

I. Contexte

Durée : 45 min

Environnement de travail : Local

Pré-requis : Aucun

Contexte

Les fonctions permettent d'exécuter, dans plusieurs parties d'un programme, une même série d'instructions.

PHP propose nativement de nombreuses fonctions pour faciliter votre quotidien. Au long de ce chapitre, nous allons découvrir comment les utiliser et en détailler quelques-unes que vous serez amené à utiliser au quotidien.

II. Les fonctions internes de PHP

Objectifs

- Comprendre ce qu'est une fonction
- Découvrir le périmètre couvert par les fonctions internes de PHP
- Identifier où retrouver ces fonctions et comment les utiliser

Mise en situation

PHP propose nativement de nombreuses fonctionnalités utilisées au quotidien par bon nombre de développeurs. Des opérations telles que le tri de tableaux ou la manipulation de dates sont facilitées par le langage grâce aux fonctions internes dont il dispose.

Avant de les découvrir plus en détails, il est important de comprendre ce que l'on appelle **fonction** lorsque l'on parle de programmation.

Définition

Fonction

On appelle **fonction** une portion de code, représentant un sous-programme, permettant d'effectuer un ensemble d'instructions par son simple appel dans le corps du programme principal.

Les fonctions permettent d'exécuter dans plusieurs parties du programme une même série d'instructions.

Les fonctions internes à PHP

Par défaut, PHP propose de nombreuses fonctions.

Certaines de ces fonctions vont être détaillées par la suite : nous ne pourrons pas toutes les voir, ce n'est pas possible et ce n'est pas le but de ce cours.

L'objectif étant de se rendre compte que ces fonctions existent et de savoir où les chercher.

Par exemple, PHP propose nativement des fonctions qui permettent :

- D'intervenir sur des chaînes de caractères : modifier la casse, rechercher ou remplacer des mots...
- De manipuler des dates : récupérer le jour, l'heure, de calculer des différences entre deux dates...
- De manipuler des tableaux : trier les éléments, les réordonner, filtrer les valeurs, y appliquer un traitement...
- D'intervenir sur le système de fichiers : créer, modifier, déplacer, supprimer un fichier...

- D'envoyer des e-mails...
- Et bien d'autres !

Il est possible de retrouver l'ensemble des fonctions existantes, triées par catégories, en suivant ce lien : <https://www.php.net/manual/fr/funcref.php>.

Méthode Lire la définition d'une fonction

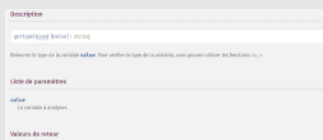
Lorsque l'on souhaite utiliser une fonction native de PHP, il est nécessaire de comprendre quels sont les paramètres attendus, quel est leur type, dans quel ordre les utiliser ou quels éléments cette fonction va retourner.

La bonne nouvelle, c'est que tous ces éléments sont documentés sur le site de PHP !

Si l'on prend l'exemple la fonction `gettype`, il est possible de retrouver la documentation ici : <https://www.php.net/manual/fr/function.gettype.php>. Cette fonction permet d'afficher le type d'une variable.

On retrouve systématiquement, dans la documentation :

- Le nom de la fonction
- Les versions de PHP avec lesquelles elle est compatible
- Son utilité
- Les paramètres qu'elle prendra en entrée et leur type
- Sa valeur de retour si elle en a une
- Des exemples d'utilisation



Ci-dessus, la page de référence de la fonction `gettype`.

Méthode Appeler une fonction

À ce stade, on sait désormais comment lire la définition d'une fonction. Mais comment y faire appel ? Il suffit de respecter la syntaxe suivante : `nom_de_la_fonction(argument1, ...) ;`.

Reprenons l'exemple de la fonction `echo`. On le sait, le but de cette fonction est d'afficher la valeur d'une variable. Voici comment procéder :

```
1 <?php
2
3 $addition = 2 + 2;
4
5 echo($addition);
6
```

Le mode opératoire sera le même, peu importe la fonction que vous utiliserez.

Méthode Récupérer la valeur de retour d'une fonction

Si l'on continue avec la fonction `echo`, on remarquera qu'on a seulement dû appeler cette fonction pour obtenir le résultat escompté.

En effet, certaines fonctions se contentent d'effectuer des actions, comme c'est le cas ici.

D'autres retourneront un résultat qu'il peut s'avérer nécessaire de récupérer pour le manipuler ou l'afficher.

Admettons que l'on souhaite calculer la racine carrée d'un nombre. Pour cela, il est possible d'utiliser la fonction `sqrt`, dont la documentation peut se trouver ici¹.

Grâce à la documentation de la fonction, on sait que celle-ci attend en entrée un nombre, et qu'elle nous retournera la racine carrée de ce nombre.

Pour récupérer le retour de cette fonction, il suffit de l'assigner à une variable.

```
1 <?php
2
3 $result = sqrt(9);
4
5 echo($result);
6
7
```

Dans l'exemple précédent, nous aurions également pu appeler la fonction `sqrt` directement au sein de la fonction `echo`.

Cela peut être une bonne façon de simplifier le code dans le cas où le résultat n'a pas besoin d'être manipulé.

```
1 <?php
2
3 echo(sqrt(9));
```

Attention cependant, il ne faut pas le faire au détriment de la lisibilité !

Syntaxe À retenir

- PHP propose nativement de nombreuses fonctionnalités grâce à des fonctions internes.
- On peut retrouver ces fonctions, triées par catégories, en suivant la documentation: <https://www.php.net/manual/fr/funcref.php>.
- Pour appeler une fonction, la syntaxe à respecter est la suivante : `nom_de_la_fonction(argument1, ...) ;`.
- Le résultat d'une fonction peut être assigné à une variable.

III. Exercice : Appliquez la notion

Pour réaliser cet exercice, vous aurez besoin de travailler sur l'environnement de travail :



Question 1

[solution n°1 p.15]

À l'aide de la documentation, déterminez les éléments suivants concernant la fonction `is_bool` :

- Son utilité
- Le paramètre attendu en entrée
- Le retour fourni par cette fonction

¹ <https://www.php.net/manual/fr/function.sqrt.php>

² <https://repl.it/>

Indice :

Vous pouvez trouver la documentation de la fonction ici : <https://www.php.net/manual/fr/function.is-bool.php>.

Question 2

[solution n°2 p.15]

Vous disposez des variables suivantes :

```
1 <?php
2
3 $var1 = 'foo';
4 $var2 = [];
5 $var3 = 0;
```

Utilisez la fonction `empty` et affichez le résultat pour ces variables.

IV. Utiliser les fonctions internes de PHP

Objectifs

- Obtenir un aperçu des possibilités offertes par les fonctions internes de PHP
- Découvrir des fonctions utilisées au quotidien

Mise en situation

Nous avons vu précédemment que PHP disposait d'un grand nombre de fonctions internes.

Nous savons désormais où chercher ces fonctions, et comment les utiliser. Nous allons découvrir quelques cas d'utilisation avec des fonctions qui sont utilisées au quotidien.

Méthode Contrôler l'intégrité d'une variable

Il est courant de devoir contrôler l'intégrité d'une variable. Est-elle déclarée ? Correspond-elle au type attendu ?

De nombreuses fonctions permettent d'effectuer ces contrôles simplement. Ces fonctions peuvent être retrouvées ici¹.

Vérifier la déclaration d'une variable

La fonction `isset`² permet de vérifier qu'une variable a bien été déclarée et qu'elle n'est pas nulle.

```
1 <?php
2
3 $name = 'Hugo';
4
5 // Affichera true
6 echo isset($name);
7
8 // Affichera false
9 echo isset($city);
```

1 <https://www.php.net/manual/fr/ref.var.php>

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

Détruire une variable

La fonction `unset`¹ permet de détruire une variable.

```
1 <?php
2
3 $name = 'Nathalie';
4
5 // Affichera true
6 echo isset($name);
7 // Destruction de la variable
8 unset($name);
9 // Affichera désormais false
10 echo isset($name);
```

Vérifier si une variable est vide

Il est possible de déterminer simplement si une variable est vide grâce à la fonction `empty`².

Les valeurs ci-dessous sont considérées comme étant vides :

- "" (une chaîne vide)
- 0 (0 en tant qu'entier)
- 0.0 (0 en tant que nombre à virgule flottante)
- "0" (0 en tant que chaîne de caractères)
- NULL
- FALSE
- `array()` (un tableau vide)

```
1 <?php
2
3 $array = [];
4
5 // Affichera "true"
6 var_dump(empty($array));
7
```

Vérifier le type d'une variable

Il est parfois nécessaire de vérifier le type d'une variable.

On peut par exemple vérifier qu'une variable est bien un tableau avec la fonction `is_array`³, ou qu'il s'agit bien d'un entier avec la fonction `is_int`⁴.

Ces fonctions retournent généralement un booléen.

```
1 <?php
2
3 $anArray = [1, 2 ,3];
4 $anInteger = 4;
5
6 // Affichera true
7 echo is_array($anArray);
8
```

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

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

3 <https://www.php.net/manual/fr/function.is-array.php>

4 <https://www.php.net/manual/fr/function.is-int.php>

```

9 // Affichera false
10 echo is_array($anInteger );
11
12 // Affichera false
13 echo is_int($anArray);
14
15 // Affichera true
16 echo is_int($anInteger );
17

```

Méthode Manipuler des chaînes de caractères

Ce type de fonctions permet de manipuler des chaînes de caractères selon différents cas d'utilisation. Il est courant d'avoir à déterminer la longueur d'une chaîne de caractères ou de devoir forcer la casse d'une chaîne de caractères.

Ces fonctions peuvent être retrouvées dans leur intégralité ici¹.

Modifier la casse d'une chaîne de caractères

Il est possible d'intervenir facilement sur la casse d'une chaîne de caractères. On peut par exemple transformer tous les caractères d'une chaîne en minuscules² ou en majuscules³.

Ces deux constantes du langage PHP sont méconnues et on se demande parfois à quoi elles peuvent servir.

`PHP_OS` renvoie le nom du système d'exploitation sur lequel l'exécutable PHP a été **compilé**. On a donc généralement `WINNT` sur une plateforme Windows, Linux ou FreeBSD sur des systèmes Unix par exemple. C'est donc utile pour modifier le comportement d'un programme en fonction du système d'exploitation où il fonctionne.

PHP est très portable, donc le cas se présente rarement, mais ça peut être utile parfois (voir l'exemple ci-dessous).

`PHP_EOL` renvoie une chaîne correspondant au saut de ligne sur la plateforme (LF, `\n` ou `#10` sur Unix, CRLF, `\n\r` ou `#13#10` sur Windows).

Dans l'exemple ci-dessous, on suppose que l'on est dans un script CLI (qui s'exécute sur la ligne de commande), et on veut afficher dans la console DOS ou le terminal UNIX un tableau de lignes contenant du texte encodé en UTF-8.

La constante `PHP_EOL` permet de rendre plus lisible le code source. Elle renvoie un saut de ligne correspondant au système d'exploitation actuel.

```

1 <?php
2
3 $var = 'Une Chaîne dE caractERes';
4
5 // Affichera "une chaîne de caracteres"
6 echo strtolower($var) . PHP_EOL;
7 // Affichera "UNE CHAÎNE DE CARACTERES"
8 echo strtoupper($var) . PHP_EOL;

```

1 <https://www.php.net/manual/fr/book.strings.php>

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

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

Déterminer la longueur d'une chaîne de caractères

Il est possible de déterminer simplement la longueur d'une chaîne de caractères avec la fonction `strlen`¹.

```
1 <?php
2
3 $var = 'Une Chaîne dE caractERes';
4 // Affichera "23"
5 echo strlen($var) . PHP_EOL;
```

Méthode Travailler avec les dates et les tableaux

Au quotidien, il n'est pas rare de devoir manipuler des dates² ou des tableaux³.

Récupérer la date du jour ? Transformer une chaîne de caractères en un tableau ou les éléments d'un tableau en une chaîne de caractères ? C'est tout à fait possible.

Récupérer la date du jour

Grâce à la fonction `date`⁴, il est possible de récupérer et de formater une date.

```
1 <?php
2
3 // Affiche la date au format jour/mois/année
4 echo date("d/m/Y") . PHP_EOL;
5
6 // Affiche le jour de la semaine
7 echo date("l") . PHP_EOL;
8
9 // Affiche l'heure
10 echo date("H:i:s") . PHP_EOL;
11
12
```

Convertir une chaîne de caractères en un tableau (et inversement)

Les fonctions `explode`⁵ et `implode`⁶ permettent de transformer une chaîne de caractères en un tableau, ou un tableau en une chaîne de caractères.

```
1 <?php
2
3 $array = ['mandarine', 'fraise', 'poire', 'pamplemousse'];
4
5 // Nous allons concaténer les valeurs de notre tableau en les séparant par une virgule
6 $arrayToString = implode(',', $array);
7
8
9 // Affichera "mandarine,fraise,poire,pamplemousseArray"
10 echo $arrayToString;
11
12 // On transforme une chaîne en tableau en considérant que les entrées seront les éléments
   séparés par une virgule
13 $stringToArray = explode(',', $arrayToString);
```

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

2 <https://www.php.net/manual/fr/ref.datetime.php>

3 <https://www.php.net/manual/fr/book.array.php>

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

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

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

```

14
15 print_r($stringToArray);
16 /* Affichera
17 Array
18 (
19     [0] => mandarine
20     [1] => fraise
21     [2] => poire
22     [3] => pamplemousse
23 )
24 */
25
26

```

Remarque

Avez-vous remarqué l'utilisation de la fonction `print_r`¹ ? Cette fonction permet d'afficher le contenu d'une variable de façon lisible.

Celle-ci peut par exemple être utile pour afficher rapidement le contenu d'un tableau pour vérifier ce qu'il contient.

Syntaxe

À retenir

- Le périmètre couvert par les fonctions internes de PHP est très vaste.
- De nombreuses fonctions répondent à des besoins quotidiens : avant d'implémenter un système complexe, mieux vaut vérifier qu'une fonction ne permet pas déjà cela.

V. Exercice : Appliquez la notion

Vous disposez du code suivant :

```

1 <?php
2
3 $languages = ['PHP', 'HTML', 'JavaScript', 'C++', 'Python'];
4
5 // Début du code à remplacer
6
7 $listOfLanguages = '';
8 // Pour chacun des éléments du tableau
9 foreach ($languages as $language) {
10     // On concatène le langage à la chaîne de caractères
11     $listOfLanguages .= $language . ', ';
12 }
13
14 // Fin du code à remplacer
15
16 // Affichera : PHP,HTML,JavaScript,C++,Python,
17 echo $listOfLanguages;
18

```

La finalité de ce code est d'afficher la liste des langages présents dans ce tableau sous forme d'une liste.

1 <https://www.php.net/manual/fr/function.print-r.php>

Pour réaliser cet exercice, vous aurez besoin de travailler sur l'environnement de travail :



Question

[solution n°3 p.15]

Simplifiez la portion de code délimitée, en utilisant une fonction native de php qui permet de convertir un tableau en une chaîne de caractères.

VI. Auto-évaluation

A. Exercice final

Exercice 1

[solution n°4 p.15]

Exercice

Le nom d'une fonction est sensible à la casse.

- ☐ Vrai
- ☐ Faux

Exercice

Selon la déclaration de code suivante, que retournera la fonction `isset` ?

```
1 <?php
2
3 $name = 'Hugo';
4
5 echo isset($city);
```

- ☐ true
- ☐ false

Exercice

La fonction `empty` permet...

- ☐ D'évaluer si une variable est considérée comme vide
- ☐ De vider le contenu d'une variable

Exercice

Grâce à quelle fonction est-il possible de détruire une variable ?

- ☐ `unlink()`
- ☐ `unset()`

Exercice

1 <https://repl.it/>

Selon la déclaration suivante, quelle valeur retournera la fonction `is_int` ?

```
1 <?php
2
3 $var = '1';
4
5 var_dump(is_int($var));
```

- ☐ true
- ☐ false

Exercice

Quelle fonction faudrait-il utiliser pour transformer le contenu d'un tableau en une chaîne de caractères ?

- ☐ implode
- ☐ explode
- ☐ is_array

Exercice

La fonction `strlen` permet de compter...

- ☐ Le nombre d'éléments dans un tableau
- ☐ Le nombre de caractères d'une chaîne de caractères
- ☐ Le nombre d'itérations d'une séquence donnée dans une chaîne

Exercice

Qu'affichera le code suivant ?

```
1 <?php
2
3 sqrt(9);
```

- ☐ 3
- ☐ 81
- ☐ Rien du tout
- ☐ true

Exercice

Parmi ces fonctions, laquelle faut-il utiliser pour transformer la casse d'une chaîne de caractères en majuscules ?

- ☐ strtolower
- ☐ strtoupper
- ☐ ucfirst

Exercice

Selon le code suivant, qu'affichera la fonction `print_r` ?

```

1 <?php
2
3 $array = ['mandarine', 'fraise', 'poire', 'pamplemousse'];
4 print_r(implode(',', $array));

```

- ☐ Array([0] => mandarine[1] => fraise[2] => poire[3] => pamplemousse)
- ☐ mandarine, fraise, poire, pamplemousse
- ☐ Il déclenchera une erreur, car il n'est pas possible d'utiliser une fonction dans une fonction
- ☐ Il déclenchera une erreur, car les paramètres de la fonction `implode` ne sont pas dans le bon ordre

B. Exercice : Défi

Vous disposez du code HTML et PHP permettant d'afficher la liste des utilisateurs d'un site Internet.

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4     <meta charset="utf-8"/>
5 </head>
6 <body>
7 <section>
8     <?php
9     $mainTitle = 'administration de votre site';
10    $pageTitle = 'liste des utilisateurs';
11    $users = ['JOHN', 'LILI', 'CLEMENCE']
12    ?>
13
14    <h1><?= $mainTitle ?></h1>
15
16    <h2><?= $pageTitle ?></h2>
17    <table>
18        <thead>
19            <tr>
20                <td>Nom</td>
21                <td>Action</td>
22            </tr>
23        </thead>
24        <tbody>
25            <?php
26                // On parcourt ici notre tableau d'utilisateurs pour les afficher un par un
27                foreach ($users as $user) {
28                    ?>
29                    <tr>
30                        <td><?= $user ?></td>
31                        <td><a href="#">Supprimer</a></td>
32                    </tr>
33                <?php
34                }
35            ?>
36        </tbody>
37    </table>
38    <p>Votre site dispose de 3 utilisateurs</p>
39 </section>
40 </body>
41 </html>

```

Voici le rendu actuel :

administration de votre site

liste des utilisateurs

Nom	Action
JOHN	Supprimer
LILI	Supprimer
CLEMENCE	Supprimer

Votre site dispose de 3 utilisateurs

Pour réaliser cet exercice, vous aurez besoin de travailler sur l'environnement de travail :



Question

[solution n°5 p.17]

À l'aide des fonctions prévues par PHP, effectuez les modifications suivantes :

- Le titre principal devra être écrit entièrement en majuscules,
- Le titre de la page devra être écrit avec sa première lettre en majuscule,
- Les noms de chaque utilisateur seront écrits en minuscules,
- À l'aide d'une fonction permettant de compter le nombre d'éléments dans un tableau, vous afficherez le nombre d'utilisateurs. L'ajout d'un nouvel élément dans le tableau permettra de vérifier que ce nombre est généré dynamiquement.

Voilà l'affichage attendu :

ADMINISTRATION DE VOTRE SITE

Liste des utilisateurs

Nom	Action
john	Supprimer
lili	Supprimer
clemence	Supprimer
anthony	Supprimer

Votre site dispose de 4 utilisateurs

Solutions des exercices

1 <https://repl.it/>

p. 5 Solution n°1

Voici les éléments que vous avez dû relever :

- La fonction détermine si une variable est booléenne
- Le paramètre attendu en entrée est la variable que l'on souhaite vérifier
- La fonction retourne un booléen

p. 6 Solution n°2

```
1 <?php
2
3 $var1 = 'foo';
4 $var2 = [];
5 $var3 = 0;
6
7 echo 'var 1 : ';
8 var_dump (empty($var1)) . PHP_EOL;
9 echo 'var 2 : ';
10 var_dump (empty($var2)) . PHP_EOL;
11 echo 'var 3 : ';
12 var_dump (empty($var3)) . PHP_EOL;
```

p. 11 Solution n°3

Voici le code que vous avez dû obtenir :

```
1 <?php
2
3 $languages = ['PHP', 'HTML', 'JavaScript', 'C++', 'Python'];
4
5 // La fonction implode permet nativement de remplacer le code à remplacer
6 $listOfLanguages = implode(' ', $languages);
7
8 // Affichera : PHP,HTML,JavaScript,C++,Python
9 echo $listOfLanguages;
10
```

Exercice p. 11 Solution n°4**Exercice**

Le nom d'une fonction est sensible à la casse.

- ☐ Vrai
- ☒ Faux

Exercice

Selon la déclaration de code suivante, que retournera la fonction `isset` ?

```
1 <?php
2
3 $name = 'Hugo';
4
5 echo isset($city);
```

- ☐ true
- ☒ false

Exercice

La fonction `empty` permet...

- ☒ D'évaluer si une variable est considérée comme vide
- ☐ De vider le contenu d'une variable

Exercice

Grâce à quelle fonction est-il possible de détruire une variable ?

- ☐ `unlink()`
- ☒ `unset()`
- ☐ La fonction est `unset()`

Exercice

Selon la déclaration suivante, quelle valeur retournera la fonction `is_int` ?

```
1 <?php
2
3 $var = '1';
4
5 var_dump(is_int($var));
```

- ☐ true
- ☒ false
- ☐ "1" n'est pas considéré comme un entier, mais comme une chaîne de caractères.

Exercice

Quelle fonction faudrait-il utiliser pour transformer le contenu d'un tableau en une chaîne de caractères ?

- ☒ `implode`
- ☐ `explode`
- ☐ `is_array`

Exercice


La fonction `strlen` permet de compter...

- ☐ Le nombre d'éléments dans un tableau
- ☒ Le nombre de caractères d'une chaîne de caractères
- ☐ Le nombre d'itérations d'une séquence donnée dans une chaîne

Exercice

Qu'affichera le code suivant ?

```
1 <?php
2
3 sqrt(9);
```

- ☐ 3
- ☐ 81
- ☒ Rien du tout
- ☐ true
-  Rien ne sera affiché : on a appelé la fonction `sqrt`, mais sans afficher son retour. Si on l'avait affiché, on aurait vu apparaître "3", puisque cette fonction retourne la racine carrée d'un nombre.

Exercice

Parmi ces fonctions, laquelle faut-il utiliser pour transformer la casse d'une chaîne de caractères en majuscules ?

- ☐ `strtolower`
- ☒ `strtoupper`
- ☐ `ucfirst`

Exercice

Selon le code suivant, qu'affichera la fonction `print_r` ?

```
1 <?php
2
3 $array = ['mandarine', 'fraise', 'poire', 'pamplemousse'];
4 print_r(implode(',', $array));
```

- ☐ `Array([0] => mandarine[1] => fraise[2] => poire[3] => pamplemousse)`
- ☒ `mandarine, fraise, poire, pamplemousse`
- ☐ Il déclenchera une erreur, car il n'est pas possible d'utiliser une fonction dans une fonction
- ☐ Il déclenchera une erreur, car les paramètres de la fonction `implode` ne sont pas dans le bon ordre

p. 14 Solution n°5

Ceci, « `< ?=` » est une balise dite « courte » d'ouverture php. C'est-à-dire que vous pouvez introduire du code php dans votre code HTML grâce à cette balise. Elle se ferme comme ceci : « `?>` ».

« `< ?=` » est un raccourci pour « `< ?echo` » et permet d'afficher du texte.

La différence avec la balise « `< ?php` » c'est que cette dernière peut contenir plusieurs lignes. La balise « `< ?=` » quant à elle doit être fermée juste après la variable qu'elle devait afficher.

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4     <meta charset="utf-8"/>
5 </head>
6 <body>
7 <section>
8     <?php
9         $mainTitle = 'administration de votre site';
10        $pageTitle = 'liste des utilisateurs';
11        $users = ['JOHN', 'LILI', 'CLEMENCE', 'Anthony']
12        ?>
13
14        <h1><?= strtoupper($mainTitle) ?></h1>
15
16        <h2><?= ucfirst($pageTitle) ?></h2>
17        <table>
18            <thead>
19                <tr>
20                    <td>Nom</td>
21                    <td>Action</td>
22                </tr>
23            </thead>
24            <tbody>
25                <?php
26                    // On parcourt ici notre tableau d'utilisateurs pour les afficher un par un
27                    foreach ($users as $user) {
28                        ?>
29                        <tr>
30                            <td><?= strtolower($user) ?></td>
31                            <td><a href="#">Supprimer</a></td>
32                        </tr>
33                    <?php
34                }
35            ?>
36        </table>
37        <p>Votre site dispose de <?= count($users) ?> utilisateurs</p>
38 </section>
39 </body>
40 </html>
41

```