Le langage PHP

04 - Utilisation des tableaux

## Question 1 - PHP et HTML

### Indications

L’idéal est de consulter le manuel officiel du langage PHP (disponible en partie en langue française) :

* Partie 1 : [PHP: Séquences d'échappement](https://www.php.net/manual/fr/regexp.reference.escape.php)
* Partie 2 : [PHP: echo](https://www.php.net/manual/fr/function.echo.php) et [PHP: print](https://www.php.net/manual/fr/function.print.php)
* Partie 3 : [PHP: printf](https://www.php.net/manual/fr/function.printf.php) et [PHP: vprintf](https://www.php.net/manual/fr/function.vprintf.php)
* Partie 5 : [Balises PHP](https://www.php.net/manual/fr/language.basic-syntax.phptags.php)

### Partie 1

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Partie 1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

*Une variable « $message » est déclaré contenant une chaine de caractères.*

### Partie 2

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Partie 2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

*Trois variables sont déclarées, « $firstname », « $lastname » et « $age ». Les deux premières sont typés comme des chaines de caractères et « $age » comme un nombre entier positif.*

*Trois nouvelles variables sont ensuite déclarées. Elles contiennent toutes trois une chaine de caractères. Cependant elles sont concaténées avec les deux variables « $firstname » et « $lastname » déclarés juste avant. « $string1 » utilise de simple « ‘’ » pour déclarer la chaine de caractère, ils faut donc cassé la chaine de caractères en la fermant, ajouter un « . » pour ensuite ajouter la variable en concaténation. Alors que « $string2 » utilise des « ‘’ ‘’ » ce qui permet à la chaine de caractère d’être concaténé juste en ajoutant la variable dans la chaine de caractère.*

*Ensuite on affiche les différentes variables « $string1 », « $string2 » et « $string3 ». En premier lieu avec la fonction « echo » puis avec « print ». La différence entre les deux sont dans leur valeur de retour, « echo » ne renverra pas de valeur, tandis que « print » enverra un toujours « 1 ». L’autre différence sont les arguments que l’on peut donner à ces fonctions. Pour « echo » on peut lui donner autant d’arguments que l’on veut tant qu’elles sont séparées par des virgules. Tandis que « print » n’accepte qu’un seul argument.*

### Partie 3

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Partie 3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

*En premier lieu un tableau est déclaré, « $customerData », il contient trois clés, « firstname », « lastname », et « age ». Pour chaque clé, une valeur est donnée. Dans l’ordre des clés, « John », « Doe », et « 38 ».*

*Une seconde variable est déclarée, « $string4 », celle-ci contient une chaine de caractères. Elle nous servira pour l’affichage des variables juste en dessous.*

*Deux fonctions pour afficher les variables sont appelés. « printf » et « vprintf ». Ces deux fonctions ont une particularité en commun, elles permettent d’afficher une suite de chaine de caractères en fonction d’une chaine de caractère que l’ont a formatés juste avant pour afficher les éléments que l’on souhaite. Par exemple, la variable « $string4 » contient deux fois les caractères « %s » et une fois les caractères « %d ». Ces caractères vont permettre de formater la chaine et d’indiquer aux fonctions ce qu’il doit s’y trouver comme type de variable. Ici « %s » indique qu’une chaine de caractère de type « string » est attendu, tandis que « %d » indique qu’une chaine de caractères de type « integer » est attendu.*

*Dans la fonction « printf », en indiquant en premier paramètre « $string4 », on spécifie la chaine de caractères à afficher au final, tout en indiquant les endroits formatés, puis les arguments indique les valeurs à mettre à la place des caractères formatés, dans l’ordre de placement des arguments.*

*Pour « vprintf » c’est exactement le même principe, sauf qu’on utilise un tableau au lieu d’une suite de variable pour remplacer les caractères formatés.*

### Partie 4

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Partie 4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

*Pour cette partie on commence par déclaré une boucle qui part de la valeur de l’indice à 0, et s’arrêtera quand la boucle aura atteint la valeur de l’indice 9. Tout simplement ici on vient afficher une chaine de caractères grâce à la fonction « echo », en s’aidant de la valeur de l’indice et en ajoutant la valeur 1 à celle-ci.*

### Partie 5

<!--

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Partie 5

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-->

*Pour la dernière partie, on vient afficher des chaines de caractères, mais en s’aidant de balise HTML. En ouvrant la possibilité d’écrire du PHP avec les balises « <?php » dans les balises HTML, il est possible d’utiliser la fonction « echo ». « <?= » fais exactement la même chose que « < ?php echo ». C’est simplement une façon plus simple et rapide de l’écrire.*