

Les finalistes compétissent pour une part des 1.5 r

Les finalistes compétissent pour une part des 1.5 million dollars en subventions

Ou

Africa's Business Heroes

Introduction aux variables PHP

Télécharger le PDF du cours

Dans cette partie, nous allons découvrir ce que sont les variables en PHP et apprendre à les manipuler.

Qu'est-ce qu'une variable ?

Une variable est un conteneur servant à stocker des informations de manière temporaire, comme une chaîne de caractères (un texte) ou un nombre par exemple.

Le propre d'une variable est de pouvoir varier, c'est-à-dire de pouvoir stocker différentes valeurs au fil du temps.

En PHP, les variables ne servent à stocker une information que temporairement. Plus précisément, une variable ne va exister que durant le temps de l'exécution du script l'utilisant.

Ainsi, on ne va pas pouvoir stocker d'informations durablement avec les variables (pour cela, nous pourrions par exemple utiliser les fichiers, cookies ou les bases de données dont nous parlerons plus tard dans ce cours).



Cours complet PHP et MySQL

INTRODUCTION AU COURS PHP ET MYSQL

1. Introduction au cours : définitions et rôles du PHP et du MySQL

2. Client et serveur : définitions et interactions

3. Mise en place de notre environnement de travail

4. Créer, enregistrer et exécuter un script PHP

5. Afficher un résultat en PHP avec une instruction echo ou print

DÉCOUVERTE DES VARIABLES EN PHP

Les finalistes compétissent pour une part des 1.5 million dollars en subventions

Africa's Business Heroes

Note : Dans le début de ce cours, nous allons définir nous-mêmes les valeurs qui vont être stockées dans nos variables, ce qui n'a pas beaucoup d'intérêt en pratique. C'est donc tout à fait normal si vous ne voyez pas immédiatement le but d'utiliser des variables. Ici, vous pouvez retenir que les variables vont être vraiment intéressantes lorsqu'elles vont nous servir à stocker des données envoyées par les utilisateurs (via des formulaires par exemple) puisqu'on va ensuite pouvoir manipuler ces données.

Les finalistes compétissent pour une part des 1.5 r

9. Présentation des conditions et des opérateurs de comparaison	dit également qu'on « déclare » une variable.
10. Les conditions if, if...else et if...elseif...else	On va pouvoir choisir le nom qu'on souhaite donner à chacune de nos variables. Cependant, il y a quelques règles à respecter et à connaître lors de la déclaration d'une nouvelle variable :
11. Créer des conditions robustes avec les opérateurs logiques	<ul style="list-style-type: none">• Toute variable en PHP doit commencer par le signe \$ qui sera suivi du nom de la variable ;• Le nom d'une variable doit obligatoirement commencer par une lettre ou un underscore (_) et ne doit pas commencer par un chiffre ;• Le nom d'une variable ne doit contenir que des lettres, des chiffres et des underscores mais pas de caractères spéciaux ;• Le nom d'une variable ne doit pas contenir d'espace.
12. Ecrire des conditions condensées avec les opérateurs ternaire et fusion null	De plus, notez que le nom des variables est sensible à la casse en PHP. Cela signifie que l'usage de majuscules ou de minuscules va créer des variables différentes. Par exemple, les variables \$texte , \$TEXTE et \$tEXTe vont être des variables différentes.
13. L'instruction switch en PHP	Enfin, sachez qu'il existe des noms « réservés » en PHP. Vous ne pouvez pas utiliser ces noms comme noms pour vos variables, tout simplement car le langage PHP les utilise déjà pour désigner différents objets intégrés au langage. Nous verrons ces différents noms au fil de ce cours.
14. Les boucles PHP et les opérateurs d'incrément et de décrémentation	
15. Inclure des fichiers dans un autre en PHP avec include et require	
DÉCOUVERTE DES FONCTIONS EN PHP	
16. Introduction aux fonctions PHP	
17. Contrôler le passage des arguments	
18. Contrôler les valeurs de retour d'une fonction	
19. La portée des variables en PHP	
20. Constantes et constantes magiques en PHP	
LES VARIABLES TABLEUX EN PHP	
21. Présentation des tableaux et tableaux numérotés en PHP	
22. Les tableaux associatifs en PHP	
23. Les tableaux multidimensionnels en PHP	
MANIPULER DES DATES EN PHP	
24. Le timestamp UNIX et la date en PHP	
25. Obtenir et formater une date en PHP	
26. Comparer des dates et tester la validité d'une date en PHP	

Déclarer une variable PHP en pratique

Pratiquons immédiatement et créons nos premières variables ensemble. Ici, nous allons créer deux variables **\$prenom** et **\$age** qui vont stocker respectivement une chaîne de caractères et un nombre.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours PHP & MySQL</title>
    <meta charset="utf-8">
    <meta name="viewport"
      content="width=device-width, initial-scale=1, user-scalable=no">
    <link rel="stylesheet" href="cours.css">
  </head>
  <body>
    <h1>Titre principal</h1>
    <?php
      $prenom = "Pierre";
      $age = 28;
    ?>
    <p>Un paragraphe</p>
  </body>
</html>
```

Dans le script ci-dessus, nous déclarons donc nos deux variables **\$prenom** et **\$age**. On assigne ou on affecte la valeur **Pierre** à la variable **\$prenom**. Cette variable va donc stocker la valeur **Pierre**. De la même façon, on assigne la valeur **28** à la variable **\$age**.

Notez qu'il va falloir utiliser des guillemets ou des apostrophes pour stocker une chaîne de caractères dans une variable. En revanche, nous n'en utiliserons pas pour assigner un nombre à une variable.

**pour une part des
1.5 million dollars
en subventions**

Il y a une chose que vous devez bien comprendre avec les variables : le signe = est dans ce cadre un opérateur d'affectation ou d'assignation et non pas un opérateur d'égalité comme cela est le cas au sens mathématiques du terme.

En effet, vous devez bien comprendre que le propre d'une variable est de pouvoir varier, ce qui signifie qu'on va pouvoir assigner différentes valeurs à une variable au cours du script. Notre variable stockera toujours la dernière valeur qui lui a été assignée. Cette valeur écrasera la valeur précédente stockée par la variable.

Nous allons pouvoir réaliser toutes sortes d'opérations avec nos variables. La plus basique d'entre elles consiste à afficher le contenu d'une variable PHP. Pour cela, nous allons généralement utiliser une instruction **echo** comme ceci :

[illegible][cookies.](#) [Réglages](#) [ACCEPTER](#)

Les finalistes compétissent pour une part des 1.5 r

<div>⬆</div> <div>49. Les méthodes magiques en orienté objet PHP</div> <div>PROGRAMMATION ORIENTÉE OBJET PHP : NOTIONS AVANCÉES</div>	<div><div>1. <!DOCTYPE html></div><div>2. <html></div><div>3. <head></div><div>4. <title>Cours PHP & MySQL</title></div><div>5. <meta charset="utf-8"></div><div>6. <link rel="stylesheet" href="cours.css"></div><div>7. </head></div><div>8.</div><div>9. <body></div><div>10. <h1>Titre principal</h1></div><div>11. <?php</div><div>12. \$prenom = "Pierre";</div><div>13. \$age = 28; // \$age stocke le nombre 28</div><div>14.</div><div>15. echo "La variable \ \$age contient : ";</div><div>16. echo \$age;</div><div>17. echo "
";</div><div>18.</div><div>19. \$age = 29; // \$age stocke le nombre 29</div><div>20. echo "La variable \ \$age contient : ";</div><div>21. echo \$age;</div><div>22. ?></div><div>23. <p>Un paragraphe</p></div><div>24. </body></div><div>25. </html></div></div>
50. Le chainage de méthodes en PHP	
51. Les closures et les classes anonymes en PHP objet	
52. L'auto chargement des classes en PHP	
53. Le mot clef final en PHP objet	
54. La résolution statique à la volée ou late static bindings en PHP	
55. Utiliser les traits en orienté objet PHP	
56. L'interface Iterator et le parcours d'objets en PHP	
57. Le passage d'objets en PHP : identifiants et références	
58. Le clonage d'objets et la méthode magique PHP __clone()	
59. La comparaison d'objets PHP	
ESPACES DE NOMS, FILTRES ET GESTION DES ERREURS EN PHP	
60. Les espaces de noms PHP	<p>Pour bien comprendre l'exemple précédent, vous devez savoir qu'ici notre script PHP est lu ou interprété linéairement, c'est-à-dire ligne par ligne dans l'ordre d'écriture.</p>
61. Présentation des filtres PHP	<p>Ici, notre variable \$age commence par stocker le nombre 28. La première série d'instructions echo va afficher un texte, le contenu de la variable et créer un retour à la ligne.</p>
62. Filtres de validation, de nettoyage et drapeaux de l'extension PHP Filter	
63. Utilisation pratique des filtres en PHP	
64. Définition et gestion des erreurs en PHP	
65. Déclenchement, capture et gestion des exceptions PHP : try, throw, catch	
INTRODUCTION AUX BASES DE DONNÉES, AU SQL ET À MYSQL	
66. Introduction aux bases de données, au SQL et au MySQL	<p>Ensuite, on affecte une nouvelle valeur à notre variable \$age. Notre nouvelle valeur, 29, va écraser l'ancienne. La variable \$age va donc désormais stocker le nombre 29. La dernière instruction echo affiche la nouvelle valeur de la variable.</p>

Les finalistes compétissent pour une part des 1.5 r

MANIPULER DES DONNÉES
DANS DES BASES MYSQL AVEC
PDO

70. Insérer des données dans une
table MySQL

71. Les requêtes MySQL préparées
avec PDO PHP

72. Modifier les données d'une
table MySQL ou sa structure

73. Supprimer des données, une
table ou une base de données
MySQL

74. Sélection simple de données
dans une table MySQL en PHP

75. Utiliser des critères de
sélection pour sélectionner des
données dans une table MySQL

76. Utiliser les fonctions
d'agrégation et les fonctions
scalaires SQL

JOINTURES, UNION ET SOUS
REQUÊTES

77. Présentation des jointures SQL

78. Création de jointures SQL

79. L'opérateur SQL UNION

80. Les opérateurs de sous
requête SQL

GESTION DES FORMULAIRES
HTML AVEC PHP

81. Rappels sur les formulaires
HTML

82. Récupérer et manipuler les
données des formulaires HTML en
PHP

83. Sécurisation et validation des
formulaires en PHP

CONCLUSION DU COURS PHP
ET MYSQL

84. Conclusion du cours complet
PHP et MySQL

Les opérateurs d'affectation et de comparaison

Une nouvelle fois (et j'insiste car c'est très important), vous devez bien comprendre que le signe égal simple utilisé ci-dessus n'est pas un opérateur de comparaison mais bien un opérateur d'affectation (ou d'assignation) : il sert à affecter une valeur à une variable.

Cela signifie que l'opérateur = ne représente pas l'égalité d'un point de vue mathématique.

L'égalité en termes de valeurs simples est symbolisée en PHP par le double signe égal : ==. L'égalité en termes de valeurs et de types de données, c'est-à-dire l'identité, va être représentée en PHP par le triple signe égal : ===.

En effet, nous allons voir plus tard dans ce cours que les variables peuvent stocker différents types de données : des chaînes de caractères, des nombres entiers, des nombres décimaux, etc. En utilisant des guillemets ou des apostrophes, on indique que la valeur stockée par la variable est une chaîne de caractères.

Ainsi, si j'écris \$age = "28" par exemple, la variable \$age stockera la chaîne de caractères 28. Cette chaîne de caractères va être égale en valeur au nombre 28 mais les types de ces deux valeurs vont être différents (la première valeur est une chaîne de caractères, la seconde est un nombre entier). Les deux variables ne vont pas être égales d'un point de vue de l'identité. Nous aurons l'occasion de réexpliquer tout cela dans les prochains chapitres.

A quoi servent de manière concrète les variables PHP ?

Les variables vont être extrêmement utiles en PHP et cela dans de nombreuses situations. Par exemple, elles vont nous permettre de manipuler des données non connues a priori. En effet, imaginons par exemple que vous souhaitiez manipuler (afficher, stocker, etc.) des données récoltées via un formulaire qui sera rempli par vos visiteurs.

Nous ne connaissons pas les valeurs qui vont être envoyées par nos visiteurs a priori. Pour manipuler ces données nous allons utiliser les variables. Ici, nous allons donc créer un script qui va traiter les données envoyées par les utilisateurs. Les valeurs envoyées vont être stockées dans des variables et nous allons manipuler ces variables.

Ainsi, à chaque fois qu'un visiteur enverra le formulaire avec ses données, notre script se déclenchera et les données seront placées dans des variables prédéfinies. Le script en question sera créé de manière à manipuler ces variables et leur contenu pour les afficher, les stocker, etc.

Les variables vont également être un outil privilégié pour dynamiser notre site grâce à leur faculté de pouvoir stocker différentes valeurs. Imaginons ici que nous souhaitions afficher une horloge sur notre site.

Nous allons alors créer un script qui va recalculer toutes les secondes par exemple l'heure actuelle. Cette heure sera placée dans une variable \$heure par exemple. Le script sera fait de telle sorte que le contenu de cette variable s'actualisera toutes les secondes (chaque seconde, la variable stockera une valeur actualisée qui sera l'heure actuelle) et affichera le contenu de cette variable.

Nous allons bien évidemment pouvoir utiliser les variables dans de nombreuses autres

Les finalistes compétissent pour une part des 1.5 r



Laisser un commentaire

Vous devez vous connecter pour publier un commentaire.

[Connexion](#)

[Confidentialité](#)

[CGV](#)

[Sitemap](#)

© Pierre Giraud - Toute reproduction interdite - Mentions légales