



Self souscription

Orange Money devient plus simple avec Self souscription. Profit services moins chers

Orange

Les types de données en PHP

Télécharger le PDF
du 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 variables PHP vont pouvoir stocker différents types de valeurs, comme du texte ou un nombre par exemple. Par abus de langage, nous parlerons souvent de « types de variables » PHP.

En PHP, contrairement à d'autres langages de programmation, nous n'avons pas besoin de préciser à priori le type de valeur qu'une variable va pouvoir stocker. Le PHP va en effet automatiquement détecter quel est le type de la valeur stockée dans telle ou telle variable, et nous allons ensuite pouvoir performer différentes opérations selon le type de la variable, ce qui va s'avérer très pratique pour nous !

Une conséquence directe de cela est qu'on va pouvoir stocker différents types de valeurs dans une variable au fil du temps sans se préoccuper d'une quelconque compatibilité. Par exemple, une variable va pouvoir stocker une valeur textuelle à un moment dans un script puis un nombre à un autre moment.

Les variables en PHP vont pouvoir stocker 8 grands types de données différents :

- Le type « chaîne de caractères » ou **String** en anglais ;
- Le type « nombre entier » ou **Integer** en anglais ;
- Le type « nombre décimal » ou **Float** en anglais ;
- Le type « booléen » ou **Boolean** en anglais ;
- Le type « tableau » ou **Array** en anglais ;
- Le type « objet » ou **Object** en anglais ;
- Le type « NULL » qui se dit également **NULL** en anglais ;
- Le type « ressource » ou **Resource** en anglais ;

Nous allons pour le moment nous concentrer sur les types simples de valeurs. Les autres types feront l'objet de leçons ou de parties dédiées dans ce cours.

Le type chaîne de caractères ou String

Le premier type de données qu'une variable va pouvoir stocker est le type **String** ou chaîne de caractères. Une chaîne de caractères est une séquence de caractères, ou ce qu'on appelle communément un texte.

Notez que toute valeur stockée dans une variable en utilisant des guillemets ou des apostrophes sera considérée comme une chaîne de caractères, et ceci même dans le cas où nos caractères sont à priori des chiffres comme « 28 » par exemple.

8. Opérateurs et concaténation en PHP

LES STRUCTURES DE CONTRÔLE EN PHP

9. Présentation des conditions et des opérateurs de comparaison

10. Les conditions if, if...else et if...elseif...else

11. Créer des conditions robustes avec les opérateurs logiques

12. Ecrire des conditions condensées avec les opérateurs ternaire et fusion null

13. L'instruction switch en PHP

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 TABLEAUX 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

Ici, notre première variable `$prez` stocke la chaîne de caractère « Je m'appelle Pierre ». Notre deuxième variable `$age`, quant à elle, stocke le nombre 28. En revanche, notre troisième variable `$age2` stocke la chaîne de caractères « 28 » et non pas un nombre.

En effet, l'utilisation de guillemets ou d'apostrophe fait qu'une valeur est immédiatement considérée comme une chaîne de caractères, quelle que soit cette valeur.



Pour s'en convaincre, on peut utiliser la fonction `gettype()` qui nous permet de connaître le type d'une variable (en anglais). Nous verrons plus en détail ce que sont les fonctions plus tard dans ce cours.

Pour le moment, il vous suffit de savoir que la fonction `gettype()` va renvoyer en résultat le type de la valeur stockée dans une variable. Nous allons ensuite utiliser une instruction `echo` pour afficher ce résultat renvoyé.

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <title>Cours PHP & MySQL</title>
5.     <meta charset="utf-8">
6.     <link rel="stylesheet" href="cours.css">
7.   </head>
8.
9.   <body>
10.    <h1>Titre principal</h1>
11.    <?php
12.      $prez = "Je m'appelle Pierre";
13.      $age = 28; //Stocke le nombre 28
14.      $age2 = "28"; //Stocke la chaîne de caractères "28"
15.
16.      echo "La variable \$age contient une valeur de type ";
17.      echo gettype($age);
18.
19.      echo "<br>La variable \$age2 contient une valeur de type ";
```

Ce site utilise des cookies pour vous fournir la meilleure expérience de navigation possible. En continuant sur ce site, vous acceptez l'utilisation des

cookies. [Réglages](#) [ACCEPTER](#)

27. Les variables superglobales PHP	24. <code></html></code>
28. Création et gestion des cookies en PHP	
29. Définir et utiliser les sessions en PHP	
MANIPULER DES FICHIERS EN PHP	
30. Introduction à la manipulation de fichiers en PHP	
31. Ouvrir, lire et fermer un fichier en PHP	
32. Créer et écrire dans un fichier en PHP	
33. Autres opérations sur les fichiers en PHP	
UTILISER LES EXPRESSIONS RÉGULIÈRES OU RATIONNELLES EN PHP	
34. Introduction aux expressions rationnelles ou expressions régulières	
35. Les fonctions PCRE PHP	
36. Les classes de caractères des regex	
37. Les métacaractères des regex PHP	
38. Les options des expressions régulières disponibles en PHP	
PROGRAMMATION ORIENTÉE OBJET (POO) PHP : CONCEPTS DE BASE	
39. Introduction à la programmation orientée objet PHP : classes, instances et objets	
40. Propriétés et méthodes en PHP orienté objet	
41. Les méthodes PHP constructeur et destructeur	
42. Encapsulation et visibilité des propriétés et méthodes PHP	
43. Classes étendues et héritage en PHP orienté objet	
44. Surcharge d'éléments et opérateur de résolution de portée en PHP	
45. Les constantes de classe en PHP	

Le code ci-dessus peut vous sembler complexe à votre niveau car nous effectuons deux opérations sur la même ligne : tout d'abord, on demande à `gettype()` va retourner le type de la valeur contenue dans notre variable puis on `echo` le résultat renvoyé par `gettype()`.

Ne cherchez pas forcément à tout comprendre immédiatement. Une nouvelle fois, nous reparlerons des fonctions plus tard dans ce cours. Pour le moment, nous voulions juste démontrer que `$age2` contient bien une valeur de type « chaine de caractères » ou `String` en anglais, ce qui est bien le cas.

Les types de données nombre entier (Integer) et nombre décimal (Float ou Double)

En PHP, on va pouvoir stocker deux types différents de donnée numéraires dans nos variables : le type `Integer`, qui contient tous les nombres entiers positifs ou négatifs et le type `Float` ou `Double`, qui contient les nombres décimaux (nombres à virgule) positifs ou négatifs.

On va donc pouvoir stocker un entier ou un nombre décimal dans une variable. Pour cela, il suffit d'affecter le nombre à stocker à notre variable, sans guillemet ni apostrophe.

Attention cependant : lorsque l'on code, on utilise toujours les notations anglo-saxonnes. Ainsi, il faudra préciser des points à la place de nos virgules pour les nombres relatifs. Voyons immédiatement un exemple ensemble :

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <title>Cours PHP & MySQL</title>
5.     <meta charset="utf-8">
6.     <link rel="stylesheet" href="cours.css">
7.   </head>
8.
9.   <body>
```

47. Les méthodes et les classes abstraites en PHP objet	14. <code>\$age2 = "28"; //Stocke la chaîne de caractères "28"</code>
48. Les interfaces en PHP orienté objet	15. <code>\$distance = 2.84;</code>
49. Les méthodes magiques en orienté objet PHP	16.
PROGRAMMATION ORIENTÉE OBJET PHP : NOTIONS AVANCÉES	17. <code>echo "La variable \ \$age contient une valeur de type ";</code>
	18. <code>echo gettype(\$age);</code>
	19.
	20. <code>echo "
La variable \ \$distance contient une valeur de type ";</code>
	21. <code>echo gettype(\$distance);</code>
50. Le chaînage de méthodes en PHP	22. <code>?></code>
51. Les closures et les classes anonymes en PHP objet	23. <code><p>Un paragraphe</p></code>
52. L'auto chargement des classes en PHP	24. <code></body></code>
53. Le mot clef final en PHP objet	25. <code></html></code>
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	
61. Présentation des filtres PHP	
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	

Le type de données booléen (Boolean)

Une variable en PHP peut encore stocker une valeur de type booléen (**Boolean** en anglais).

Le type booléen est un type qui ne contient que deux valeurs : les valeurs **true** (vrai) et **false** (faux). Ce type n'est pas courant dans la vie de tous les jours mais est très (très) utilisé en informatique.

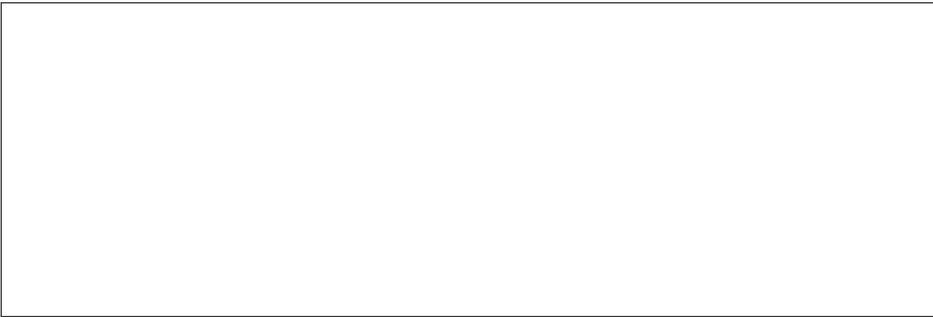
Nous aurons souvent recours aux booléens dans ce cours car ils vont être à la base de nombreux mécanismes en PHP, et aurons donc largement le temps de comprendre tout l'intérêt de ces valeurs.

Pour stocker une valeur de type booléen dans une variable, pensez bien à ne pas entourer **true** ou **false** par des guillemets ou des apostrophes. Si vous utilisez des guillemets ou des apostrophes, en effet, les valeurs seront considérées comme étant de type chaîne de caractères ou **String** et nous n'allons pas pouvoir les utiliser pour réaliser certaines opérations.

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <title>Cours PHP & MySQL</title>
5.     <meta charset="utf-8">
```

68. Se connecter à une base de données MySQL en PHP
69. Créer une base de données MySQL et une table dans la base
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

```
10. <h1>Titre principal</h1>
11. <?php
12.     $prez = "Je m'appelle Pierre";
13.     $age = 28; //Stocke le nombre 28
14.     $age2 = "28"; //Stocke la chaine de caractères "28"
15.     $distance = 2.84;
16.     $vrai = true;
17.     $faux = false;
18.
19.     echo "La variable \$vrai contient une valeur de type ";
20.     echo gettype($vrai);
21.
22.     echo "<br>La variable \$faux contient une valeur de type ";
23.     echo gettype($faux);
24.
25.     ?>
26. <p>Un paragraphe</p>
27. </body>
    </html>
```



Le type de données Null

Le type de données **Null** est un type un peu particulier puisqu'il correspond à l'absence de valeur et sert donc à représenter des variables vides en PHP.

Ce type de valeur ne contient qu'une seule valeur : la valeur **NULL** qui correspond elle-même à l'absence de valeur. Il est un peu tôt pour vous faire comprendre l'intérêt de ce type de valeurs ; nous en reparlerons plus tard dans ce cours lorsque nous aurons à l'utiliser.

Notez que si vous déclarez une nouvelle variable sans lui affecter de valeur (ce qui est déconseillé de manière générale), cette variable sera automatiquement de type **Null**.

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Cours PHP & MySQL</title>
5.         <meta charset="utf-8">
6.         <link rel="stylesheet" href="cours.css">
```

```
11. <?php
12.     $prez = "Je m'appelle Pierre";
13.     $age = 28; //Stocke le nombre 28
14.     $age2 = "28"; //Stocke la chaîne de caractères "28"
15.     $distance = 2.84;
16.     $vrai = true;
17.     $faux = false;
18.     $vide = NULL;
19.     $vide2;
20.
21.     echo "La variable \$vide contient une valeur de type ";
22.     echo gettype($vide);
23.
24.     echo "<br>La variable \$vide2 contient une valeur de type ";
25.     echo gettype($vide2);
26.     ?>
27.     <p>Un paragraphe</p>
28. </body>
29. </html>
```



Les types de données PHP tableau (Array) et objet (Object)

Les types de données **Array** et **Object** sont des types de données complexes particuliers qui méritent de faire chacun l'objet de chapitres séparés.

Nous n'étudierons donc pas ces deux types pour le moment car n'avons pas les connaissances suffisantes pour bien les comprendre.

Sachez simplement que l'on va pouvoir stocker plusieurs valeurs d'un coup à l'intérieur d'une variable en lui assignant des valeurs de type **Array** (tableau) ou **Object** (objet).

Le type de données ressource (Resource)

Une ressource est une variable particulière qui contient une référence vers une ressource externe au PHP, comme dans le cas d'une variable qui représente la connexion vers une base de données par exemple.

Là encore, ce type de données est complexe et nécessite d'avoir une bonne vision d'ensemble du langage pour être bien compris. Nous l'étudierons donc plus tard.

[Précédent](#)

[Suivant](#)

Laisser un commentaire

[Connexion](#)

[Confidentialité](#)

[CGV](#)

[Sitemap](#)

© Pierre Giraud - Toute reproduction interdite - Mentions légales