Intro PHP: Les variables: 27/03/2015

fyligrane



Introduction au langage de programmation PHP

```
Accueil ▶ Mes cours ▶ Développement logiciel ▶ Intro PHP ▶ Les types et les variables ▶ Les variables
```

Les variables

En PHP, une variable se déclare en utilisant le signe \$. Ce \$ doit précéder le nom de la variable. De plus, le nom d'une variable doit :

- commencer par une lettre ou un caractère '_' et non par un chiffre
- ne contenir que des caractères alphanumériques et/ou des underscores

Le nom des variables est sensible à la casse, c'est-à-dire qu'une lettre majuscule n'est pas équivalente à la même lettre en minuscule.

Quelques exemples de variables possédant un nom valide :

- \$variable
- \$_variable
- \$ 2variable
- \$ma variable
- \$Variable
- \$Ma Variable
- \$maVariable
- \$MaClasse

Notez que les variables \$variable et \$Variable sont différentes, tout comme \$ma_variable et \$Ma_Variable.

Pour assigner une valeur à une variable nous utiliserons l'opérateur "=".

Par exemple pour déclarer une variable nb, lui assigner la valeur 456 et l'afficher nous écrirons :

Notez que nous avons ici déclaré nb puis nous lui avons assigné une valeur. Nous aurions pu faire cela en une seule ligne, c'est-à-dire déclarer et assigner tout de suite une valeur à la variable. Il est d'ailleurs de bon ton d'initialiser la valeur des variables que l'on crée.

Comme vous l'aurez sûrement remarqué, le mot clé echo permet de réaliser un affichage. Lorsque j'écris echo \$nb; j'affiche en fait la valeur de \$nb.

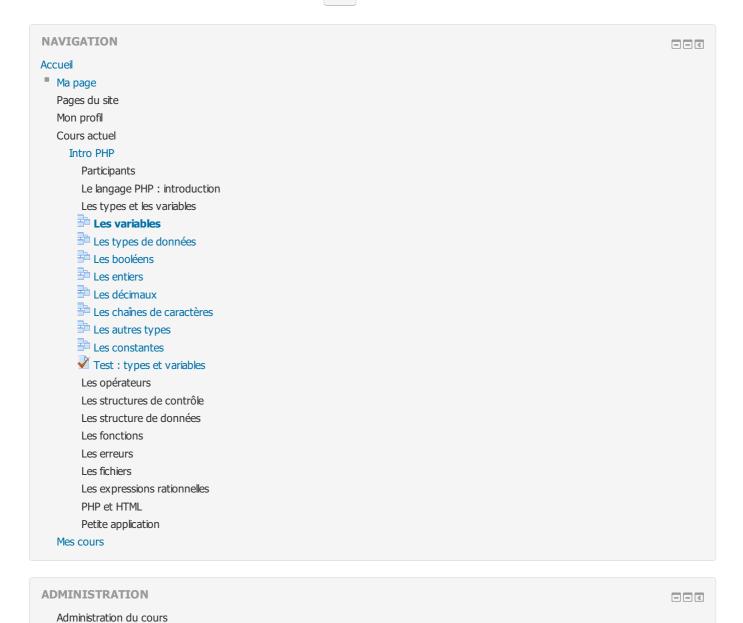
Le nom d'une variable doit être choisi avec soin : plus ce nom sera auto-référentiel, c'est-à-dire plus il donnera d'indication sur l'utilité de cette variable, plus il facilitera la production du code et sa maintenance. Une variable \$user_account est préférable à une variable \$ua, par exemple. Vous avez noté que le code PHP est précédé d'une balise ouvrante : <?php. Dans une page HTML, le code PHP sera

<u>Intro PHP: Les variables:</u> 27/03/2015

systématiquement encadré par une balise ouvrante PHP : <?php et une balise fermante : ?>. Dans un simple script, la balise ouvrante suffit. Un script sera contenu dans un fichier qui portera l'extension .php (par exemple : mon_script.php).

Dernier point : les instructions en PHP se terminent par un point-virgule (;) !

Fin



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

Réglages de mon profil

Intro PHP: Les types de données: 27/03/2015

fyligrane



Introduction au langage de programmation PHP

```
Accueil ▶ Mes cours ▶ Développement logiciel ▶ Intro PHP ▶ Les types et les variables ▶ Les types de données
```

Les types de données

En PHP, nous trouvons 8 types de données basiques :

- boolean : les valeurs booléennes
- integer : les nombres entiers, équivalent au type NOMBRE d'AlgoBox
- double (ou float): les nombres décimaux
- string : les chaînes de caractères, équivalent au type CHAINE d'AlgoBox
- array : les tableaux, ressemblant au type LISTE d'AlgoBox
- · object : les objets
- resource : les ressources (fichiers...)
- NULL : qui représente l'absence de valeur

Revenons à notre exemple précédent :

Ici nous avons déclarer une variable et nous lui avons assigné immédiatement une valeur, puis nous avons affiché cette valeur. Avec AlgoBox cela donnerait cela :

En PHP, nous ne déclarerons pas explicitement le type des variables. Lors de l'assignation, PHP donnera automatiquement le bon type à la variable. Lorsque j'écris

```
$nb = 456;
```

PHP crée une variable nb, déclare que cette variable est de type integer puisque 456 est de type integer, et affecte la valeur 456 à la variable nb. Le type de la variable est détecté et "attribué" automatiquement par PHP lorsqu'on affecte une valeur à la variable : c'est un typage dynamique.

Au cours de mon programme, je peux très bien affecter à ma variable une autre valeur. Mais si j'affecte une valeur d'un type différent de celui de la variable, je peux m'exposer à quelques problèmes, en tout cas avec AlgoBox...

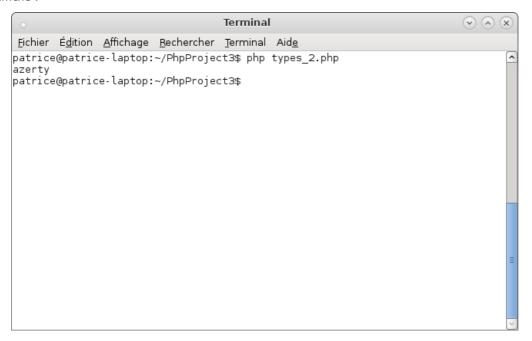
À l'exécution cela nous donne :

```
***Algorithme lancé***
***Algorithme interrompu ligne 5 : erreur de calcul***
```

<u>Intro PHP: Les types de données:</u> 27/03/2015

Testons maintenant avec PHP.

Et cela affiche:



Non seulement cela fonctionne, mais PHP a modifié dynamiquement le type de nb lors de l'affectation de "azerty". La variable nb est maintenant de type string. PHP permet donc à une variable de changer de type selon la valeur qu'on lui assigne : c'est un typage faible.

AlgoBox est donc, par opposition à PHP, un langage au typage statique et fort.



```
NAVIGATION
                                                                                                                 Accueil
Ma page
  Pages du site
  Mon profil
  Cours actuel
     Intro PHP
       Participants
       Le langage PHP: introduction
       Les types et les variables
      Les variables
      Les types de données
      Les booléens
      Les entiers
      Les décimaux
      Les chaînes de caractères
      Les autres types
      Les constantes
```

Intro PHP: Les types de données: 27/03/2015



Les opérateurs

Les structures de contrôle

Les structure de données

Les fonctions

Les erreurs

Les fichiers

Les expressions rationnelles

PHP et HTML

Petite application

Mes cours

ADMINISTRATION

-- <

Administration du cours

Réglages de mon profil

Intro PHP: Les booléens: 27/03/2015

fyligrane



Introduction au langage de programmation PHP

```
Accueil ▶ Mes cours ▶ Développement logiciel ▶ Intro PHP ▶ Les types et les variables ▶ Les booléens
```

Les booléens

Une donnée de type boolean peut prendre deux valeurs : true ou false. Les mots clés true et false sont insensibles à la casse. L'affichage d'une valeur true sera 1, l'affiche d'une valeur false sera... rien du tout. Dans certains cas, des valeurs d'autres types peuvent être considérées comme true ou false. Sont considérées comme false les valeurs :

Туре	Valeur
integer	0
double	0.0
string	"" et " (chaîne vide) et "0"
array	[] (tableau vide)
object	null

Les chaînes précédées par "//" sont des commentaires sur une ligne. Ces chaînes ne seront pas exécutées par l'interpréteur. Il est aussi possible de créer des commentaires sur plusieurs lignes en insérant ces commentaires entre "/*" et "*/".

```
1
   <?php
2
3 // les booleens_2.php
4 /*
   * ces lignes
5
   * ne seront pas
6
7
   * exécutées
8
9 | $var = true;
10
    // cette ligne ne sera pas exécutée
11
   echo $var;
12
   echo PHP EOL;
```

Fin

```
NAVIGATION

Accueil

Ma page
Pages du site
Mon profil
Cours actuel
```

<u>Intro PHP: Les booléens:</u> 27/03/2015

Intro PHP Participants Le langage PHP: introduction Les types et les variables Les variables Les types de données Les booléens Les entiers Les décimaux Les chaînes de caractères Les autres types Les constantes Test: types et variables Les opérateurs Les structures de contrôle Les structure de données Les fonctions Les erreurs Les fichiers Les expressions rationnelles PHP et HTML Petite application Mes cours

ADMINISTRATION	
Administration du cours	
Réglages de mon profil	

<u>Intro PHP: Les entiers:</u> 27/03/2015

fyligrane



Introduction au langage de programmation PHP

```
Accueil ▶ Mes cours ▶ Développement logiciel ▶ Intro PHP ▶ Les types et les variables ▶ Les entiers
```

Les entiers

Le type entier (integer) comprend les valeurs entières positives et négatives. Les valeurs maximales des entiers sont :

Plateforme	Valeur
32 bits	2 147 483 647
64 bits	9 223 372 036 854 775 807

Un entier peut être écrit en décimal (en base 10), en octal (en base 8) ou en hexadécimal (en base 16), ou encore en binaire (base 2). Les nombres doivent être précédés par :

- 0 en octal
- 0x ou 0X en hexadécimal
- 0b ou 0B en binaire.

Notation	Valeur
décimal	126
octal	0176
hexadécimal	0X7E
binaire	0b1111110

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 types de données
Les types de données
Les booléens
Les entiers
```

<u>Intro PHP: Les entiers:</u> 27/03/2015

Les décimaux

Les chaînes de caractères

Les autres types

Les constantes

Test: types et variables

Les opérateurs

Les structures de contrôle

Les structure de données

Les fonctions

Les erreurs

Les fichiers

Les expressions rationnelles

PHP et HTML

Petite application

Mes cours

ADMINISTRATION

Administration du cours

Réglages de mon profil

Intro PHP: Les décimaux: 27/03/2015

fyligrane



Introduction au langage de programmation PHP

```
Accueil ▶ Mes cours ▶ Développement logiciel ▶ Intro PHP ▶ Les types et les variables ▶ Les décimaux
```

Les décimaux

Les décimaux ou nombres à virgule flottantes (double ou float) ont une capacité beaucoup plus importante que les entiers. Ils permettent donc de représenter des nombres décimaux mais aussi des nombres entiers importants. Ils répondent à une norme spécifique : la norme IEEE754. En PHP, la virgule des nombres décimaux est remplacée par un point.

Exemple:

affiche:

```
Terminal

Fichier Édition Affichage Rechercher Terminal Aide

patrice@patrice-laptop:~/PhpProject3$ php les_decimaux_1.php
12.5
-0.023
patrice@patrice-laptop:~/PhpProject3$
```

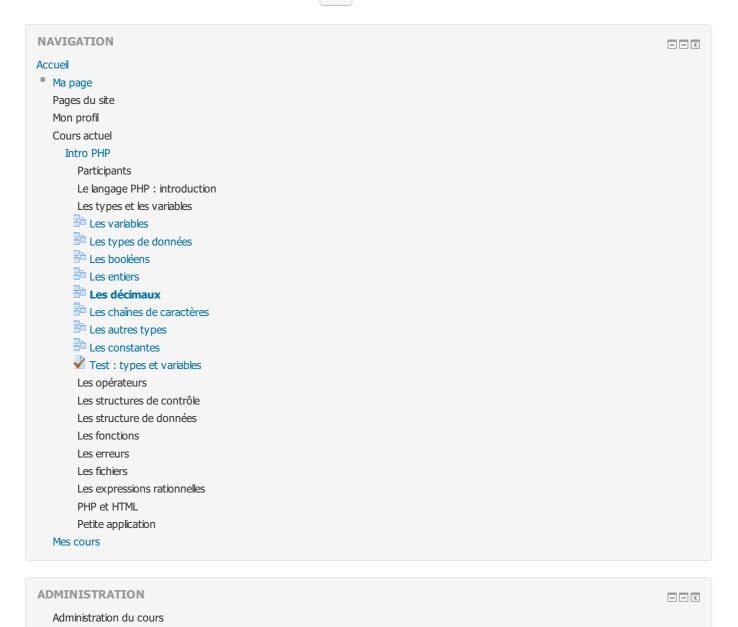
Un nombre décimal peut aussi être exprimé sous forme d'une puissance de 10 en utilisant l'opérateur e ou E :

Une remarque importante : certains nombres n'ont pas de représentation exacte en binaire, comme 0.1 ou 0.7 par exemple. Les calculs effectués avec ces valeurs pourront donc être faux à la marge...

<u>Intro PHP: Les décimaux:</u> 27/03/2015

Il sera donc nécessaire d'arrondir les valeurs décimales avant d'effectuer des calculs ou des comparaisons. Heureusement, PHP nous propose des fonctions pour arrondir des valeurs ou encore effectuer des calculs sur des grands nombres. Voici un petit exemple que vous pourrez tester plus tard : pour php l'expression 0.1 + 0.2 n'est pas exactement égale à 0.3!

Fin



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

Réglages de mon profil

fyligrane



Introduction au langage de programmation PHP

```
Accueil ▶ Mes cours ▶ Développement logiciel ▶ Intro PHP ▶ Les types et les variables ▶ Les chaînes de caractères
```

Les chaînes de caractères

En PHP, les chaînes de caractères (string) peuvent être délimitées par des simples quotes ('), par exemple : 'une chaîne de caractères', ou par des doubles quotes ("), par exemple : "une autre chaîne de caractères". Il y a une différence majeure entre la chaîne entre simples quotes et celle entre doubles quotes : dans une chaîne entre doubles quotes, les variables contenues dans la chaîne seront évaluées, c'est-à-dire remplacées par leurs valeurs. Par exemple, le code :

affiche:

```
Terminal

Fichier Édition Affichage Rechercher Terminal Aide

patrice@patrice-laptop:~/PhpProject3$ php les_chaines_1.php
la variable nb a pour valeur 180
la variable nb a pour valeur $nb
patrice@patrice-laptop:~/PhpProject3$
```

Dans la chaîne contenue dans la variable \$str, \$nb a bien été remplacée par sa valeur, c'est-à-dire 180. Dans la variable \$str2, la variable \$nb n'a pas été remplacée par sa valeur, puisque la chaîne est délimitée par des simples quotes.

Voici un petit problème : je souhaite afficher une chaîne qui contiendra : mot de passe : 'secret'. Avec des simples quotes je devrais écrire cela :

Mon éditeur me signale une erreur de syntaxe à la ligne 4. En PHP, le caractère simple quote (') a une signification précise : il indique le début ou la fin d'une chaîne de caractères entre simples quotes. Ici mon éditeur comprend que la chaîne contient 'le mot de passe : ' et s'arrête après le caractère espace. Pour empêcher l'évaluation du caractère simple quote dans la chaîne je vais l'échapper avec un caractère anti-slash (\). Ce qui me donnera :

Et cela affichera:

```
Terminal

Fichier Édition Affichage Rechercher Terminal Aide

patrice@patrice-laptop:~/PhpProject3$ php les_chaines_3.php

le mot de passe : 'secret'

patrice@patrice-laptop:~/PhpProject3$
```

L'anti-slash passe en fait la consigne suivante à PHP: "Ne tiens pas compte de la signification de ce caractère et affiche le comme un caractère normal". L'anti-slash lui même ne sera pas affiché. Dans une chaînes de caractères, les caractères précédés par un anti-slash (\) ne sont donc pas évalués et sont affichés comme des caractères normaux. Le caractère antislash est un caractère d'échappement.

Une autre solution consiste à écrire la chaîne complète entre doubles quotes, dans ce cas les simples quotes n'ont plus de signification particulière et sont considérées comme de simples caractères.

```
1
2
3
// les_chaines_4.php
$str = "le mot de passe : 'secret'";
echo $str;
echo PHP_EOL;
```

Dans une chaîne entre doubles quotes, certains caractères ont une signification précise, par exemple :

Caractère	Signification
"	délimitation de chaîne
\n	saut de ligne
\t	tabulation
\	échappement
\$	variable

Si je souhaite afficher un de ces caractères dans une chaînes entre doubles quotes, je devrai échapper ces caractères c'est-à-dire les faire précéder d'un anti-slash.

```
<?php
2
3
   // les_chaines_5.php
4
   $str = "une chaîne entre doubles quotes est délimitée par des \".";
5 echo $str;
6 echo PHP EOL;
7
   $str2 = "\$nb est une variable PHP.";
8
   echo $str2;
9
   echo PHP EOL;
10 | $str3 = "Dans une chaîne entre doubles quotes, je peux afficher \\n, \\r, \$ "
11
           . "en échappant ces caractères avec un \\.";
12 echo $str3;
13
   echo PHP_EOL;
```

Ce qui affichera:

```
Terminal

Fichier Édition Affichage Rechercher Terminal Aide

patrice@patrice-laptop:~/PhpProject3$ php les_chaines_5.php

une chaîne entre doubles quotes est délimitée par des ".

$$\frac{1}{2}$$ best une variable PHP.

Dans une chaîne entre doubles quotes, je peux afficher \n, \r, $ en échappant ce s caractères avec un \.

patrice@patrice-laptop:~/PhpProject3$
```

Par contre, dans une chaîne entre simples quotes, nul besoin d'échapper ces caractères, ils seront affichés comme de simples caractères :

Ce qui affichera:

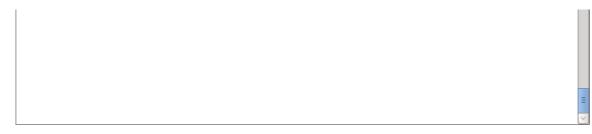
```
Terminal

Fichier Édition Affichage Rechercher Terminal Aide

patrice@patrice-laptop:~/PhpProject3$ php les_chaines_6.php

Dans une chaîne entre simples quotes, je peux afficher \n, \t, $ sans échappement.

patrice@patrice-laptop:~/PhpProject3$
```



La syntaxe Heredoc

Une chaîne Heredoc se comporte comme une chaîne entre doubles quotes. Une chaîne Heredoc débute par l'opérateur <<< suivi d'un identifiant puis d'un saut de ligne. La chaîne se termine par l'identifiant suivi d'un point-virgule. L'identifiant de fin doit être écrit au début de la ligne.

Cela affichera:

```
Terminal

Fichier Édition Affichage Rechercher Terminal Aide

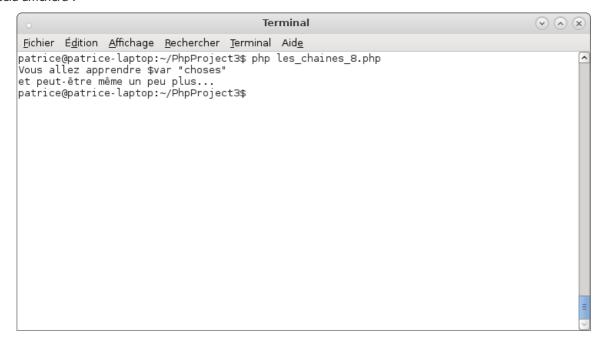
patrice@patrice-laptop:~/PhpProject3$ php les_chaines_7.php

Vous allez apprendre 123 "choses"
et peut-être même un peu plus...
patrice@patrice-laptop:~/PhpProject3$
```

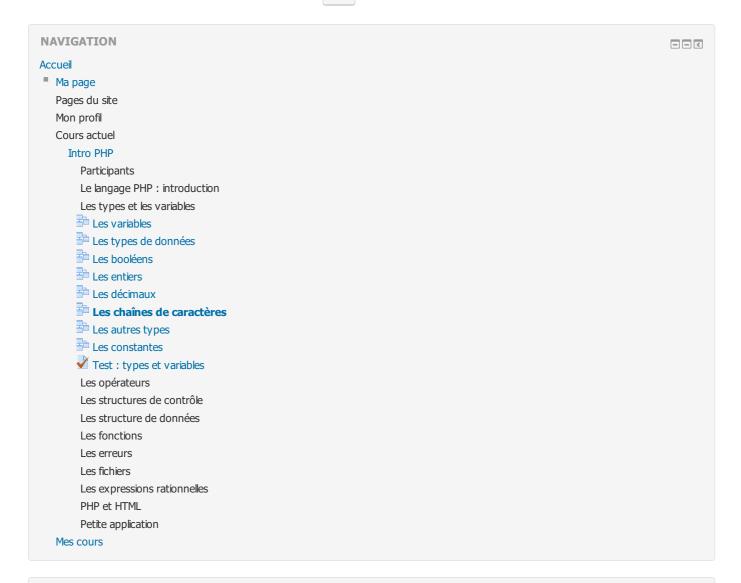
La syntaxe Nowdoc

Une chaîne Nowdoc se comporte, elle, comme une chaîne entre simples quotes. Une chaîne Nowdoc débute aussi par l'opérateur <<< suivi d'un identifiant entre simples quotes, puis d'un saut de ligne. La chaîne se termine par l'identifiant suivi d'un point-virgule. L'identifiant de fin doit être écrit au début de la ligne. Même exemple que précédemment mais cette fois avec une syntaxe Nowdoc :

Cela affichera:



Fin



ADMINISTRATION

Administration du cours

Réglages de mon profil

Intro PHP: Les autres types: 27/03/2015

fyligrane



Introduction au langage de programmation PHP

Accueil ▶ Mes cours ▶ Développement logiciel ▶ Intro PHP ▶ Les types et les variables ▶ Les autres types

Les autres types

Les autres types, à savoir les tableaux (array), les ressources (resource) et les objets (object) feront l'objet de leçons spécifiques.

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 variables	
Les types de données	
Les booléens	
Les entiers	
Les décimaux	
Les chaînes de caractères	
Les autres types	
Les constantes	
Test: types et variables	
Les opérateurs	
Les structures de contrôle	
Les structure de données	
Les fonctions	
Les erreurs	
Les fichiers	
Les expressions rationnelles	
PHP et HTML	
Petite application	
Mes cours	

ADMINISTRATION

Administration du cours

Réglages de mon profil

Intro PHP: Les constantes: 27/03/2015

fyligrane



Introduction au langage de programmation PHP

```
Accueil ▶ Mes cours ▶ Développement logiciel ▶ Intro PHP ▶ Les types et les variables ▶ Les constantes
```

Les constantes

Une constante est une valeur qui ne changera pas au cours du programme et qui portera un nom. Une constante pourra être de type boolean, integer, double, string. Pour définir une constante, nous allons utiliser la fonction define(). Cette fonction prendra en argument une chaîne de caractères qui sera le nom de la constante, puis la valeur que nous souhaitons assigner à cette constante. Pour utiliser la constante, nous écrirons simplement son nom. Il existe une convention pour nommer les constantes : le nom de la constante doit être écrit en majuscule et les espaces remplacés par des underscores (_).

Cela affichera:

```
Terminal

Pichier Édition Affichage Rechercher Terminal Aide

patrice@patrice-laptop:~/PhpProject3$ php les_constantes_1.php

18

patrice@patrice-laptop:~/PhpProject3$
```

PHP propose un certain nombre de constantes pré-définies, par exemple :

- PHP_EOL : saut de ligne
- PHP_INT_SIZE : taille en octet d'un entier
- PHP_INT_MAX : valeur maximale d'un entier

Fin

```
NAVIGATION

Accueil

Ma page
Pages du site
Mon profil
Cours actuel
Intro PHP
```

Intro PHP: Les constantes: 27/03/2015

Participants

Le langage PHP: introduction

Les types et les variables

Les variables

Les types de données

Les booléens

Les entiers

E Les décimaux

Les chaînes de caractères

Les autres types

Les constantes

Test: types et variables

Les opérateurs

Les structures de contrôle

Les structure de données

Les fonctions

Les erreurs

Les fichiers

Les expressions rationnelles

PHP et HTML

Petite application

Mes cours

ADMINISTRATION

Administration du cours

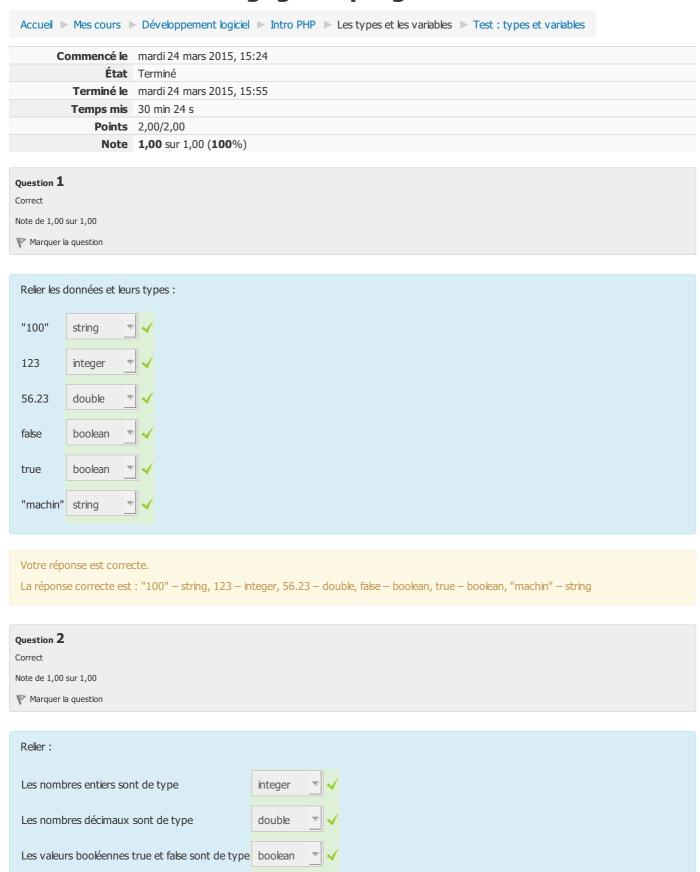
Réglages de mon profil

<u>Test: types et variables</u> 27/03/2015

fyligrane



Introduction au langage de programmation PHP



string

Les chaînes de caractères sont de type

<u>Test: types et variables</u> 27/03/2015

Votre réponse est correcte.

La réponse correcte est : Les nombres entiers sont de type – integer, Les nombres décimaux sont de type – double, Les valeurs booléennes true et false sont de type – boolean, Les chaînes de caractères sont de type – string

Terminer la relecture

NAVIGATION DU TEST



Terminer la relecture