

# PHP Numbers

Dans ce chapitre, nous examinerons en profondeur les entiers, les flottants et les chaînes de nombres.

# Numéros PHP

Une chose à noter à propos de PHP est qu'il fournit une conversion automatique du type de données.

Ainsi, si vous affectez une valeur entière à une variable, le type de cette variable sera automatiquement un entier. Ensuite, si vous affectez une chaîne à la même variable, le type deviendra une chaîne.

Cette conversion automatique peut parfois casser votre code.

# Entiers PHP

2, 256, -256, 10358, -179567 sont tous des entiers.

Un entier est un nombre sans partie décimale.

Un type de données entier est un nombre non décimal entre -2147483648 et 2147483647 dans les systèmes 32 bits, et entre -9223372036854775808 et 9223372036854775807 dans les systèmes 64 bits. Une valeur supérieure (ou inférieure) à celle-ci sera stockée en tant que **float**, car elle dépasse la limite d'un entier.

**Remarque** : Une autre chose importante à savoir est que même si  $4 * 2,5$  vaut 10, le résultat est stocké en tant que **float**, car l'un des opérandes est un **float** (2,5).

Voici quelques règles pour les entiers :

- Un entier doit avoir au moins un chiffre
- Un entier ne doit PAS avoir de point décimal
- Un entier peut être positif ou négatif

Les nombres entiers peuvent être spécifiés dans trois formats : décimal (base 10), hexadécimal (base 16 - préfixé par 0x) ou octal (base 8 - préfixé par 0)

PHP a les constantes prédéfinies suivantes pour les entiers :

- PHP\_INT\_MAX - Le plus grand entier pris en charge
- PHP\_INT\_MIN - Le plus petit entier pris en charge
- PHP\_INT\_SIZE - La taille d'un entier en octets

PHP a les fonctions suivantes pour vérifier si le type d'une variable est un entier :

- is\_int()
- is\_integer() - alias de is\_int()
- is\_long() - alias de is\_int()

### Exemple

Vérifiez si le type d'une variable est entier :

```
<?php
$x = 5985;
var_dump(is_int($x));

$x = 59.85;
var_dump(is_int($x));
?>
```

# Flottants PHP

Un flottant est un nombre avec un point décimal ou un nombre sous forme exponentielle.

2.0, 256.4, 10.358, 7.64E+5, 5.56E-5 sont tous des flottants.

Le type de données float peut généralement stocker une valeur jusqu'à 1,7976931348623E+308 (en fonction de la plate-forme) et avoir une précision maximale de 14 chiffres.

PHP a les constantes prédéfinies suivantes pour les flottants (à partir de PHP 7.2) :

- PHP\_FLOAT\_MAX - Le plus grand nombre à virgule flottante représentable
- PHP\_FLOAT\_MIN - Le plus petit nombre à virgule flottante positif représentable
- - PHP\_FLOAT\_MAX - Le plus petit nombre à virgule flottante négatif représentable
- PHP\_FLOAT\_DIG - Le nombre de chiffres décimaux qui peuvent être arrondis en un flottant et inversement sans perte de précision
- PHP\_FLOAT\_EPSILON - Le plus petit nombre positif représentable x, de sorte que  $x + 1.0 \neq 1.0$

PHP a les fonctions suivantes pour vérifier si le type d'une variable est float :

- is\_float()
- is\_double() - alias de is\_float()

Exemple

Vérifiez si le type d'une variable est flottant :

```
<?php
$x = 10.365;
var_dump(is_float($x));
?>
```

# PHP Infini

Une valeur numérique supérieure à PHP\_FLOAT\_MAX est considérée comme infinie.

PHP a les fonctions suivantes pour vérifier si une valeur numérique est finie ou infinie :

- is\_finite()
- is\_infinite()

Cependant, la fonction PHP `var_dump()` renvoie le type de données et la valeur :

Exemple

Vérifiez si une valeur numérique est finie ou infinie :

```
<?php
$x = 1.9e411;
var_dump($x);
?>
```

# PHP NaN

NaN signifie pas un nombre. NaN est utilisé pour des opérations mathématiques impossibles.

PHP a les fonctions suivantes pour vérifier si une valeur n'est pas un nombre :

- `is_nan()`

Cependant, la fonction PHP `var_dump()` renvoie le type de données et la valeur :

Exemple

Un calcul non valide renverra une valeur NaN

```
<?php
$x = acos(8);
var_dump($x);
?>
```



# Chaînes numériques PHP

La fonction PHP `is_numeric()` peut être utilisée pour savoir si une variable est numérique. La fonction renvoie `true` si la variable est un nombre ou une chaîne numérique, `false` sinon.

Exemple

Vérifiez si la variable est numérique :

```
<?php
$x = 5985;
var_dump(is_numeric($x)); //true

$x = "5985";
var_dump(is_numeric($x)); //true

$x = "59.85" + 100;
var_dump(is_numeric($x)); //true

$x = "Hello";
var_dump(is_numeric($x)); //false
?>
```

# Conversion des chaînes de caractères et des flottants en nombres entiers en PHP

Parfois, vous devez convertir une valeur numérique en un autre type de données.

Les fonctions (int), (integer) ou intval() sont souvent utilisées pour convertir une valeur en entier.

Exemple

Transformez float et string en entier :

```
<?php
// Convertir float en int
$x = 23465.768;
$int_cast = (int)$x;
echo $int_cast;

echo "<br>";

// Convertir string en int
$x = "23465.768";
$int_cast = (int)$x;
echo $int_cast;
?>
```