Les types de données en PHP

1. Objectifs

- Connaitre les types de données en PHP

2. Les types de données en PHP

- Les variables peuvent stocker des données de différents types, et peuvent faire différentes choses.
- PHP supporte les types de données suivants :**`String`, `Boolean`, `Integer`, `Float`, `Array`, `Object` et `NULL`**

- **`Integer` (entiers)**

```
<?php
  $a = 2;
  $b = 4;
  $somme = $a + $b;
  echo $somme;   //6
?>
```

- Le type `integer` est affecté aux variables qui contiennent des valeurs entières positives ou négatives en base 10 (décimal), en base 2 (binaire), en base 8 (octal) ou en base 16 (hexadécimal).
- Les nombres Integer ne contiennent que des **nombres entiers**, y compris les **nombres positifs** et **négatifs**, c'est-à-dire des nombres sans partie fractionnaire ni le point décimal. Ils peuvent être décimaux (base 10), octal (base 8) ou hexadécimaux (base 16). La base par défaut est décimale (base 10). La plage des entiers doit être comprise entre -2^31 et 2^31.
- Vous pouvez vérifier si une variable est de type Integer en utilisant la fonction PHP **`is_integer()`** ou la fonction PHP **`is_int()`**.

3. **`Float` (flottants)**

- **`Float`** est un nombre à virgule décimale ou un nombre sous forme exponentielle.

apcpedagogie.com

- Le type nombre à virgule flottante (**`float`**) permet de stocker un nombre décimal sur une plage de valeurs dépendante de la plate-forme (généralement de l'ordre de 10-308 à 10+308).
- Les entiers sont codés sur 32 bits sur la plupart des plates-formes, mais cela peut varier en fonction des serveurs. L'intervalle de valeur des entiers est donc dans ce cas de - 2 147 483 648, soit - 2^{31} à + 2 147 483 647, soit 2^{31} - 1 en base 10.
- Si une opération sur une variable de type integer l'amène à contenir une valeur en dehors de cet intervalle, elle est automatiquement convertie en type double et conserve sa nouvelle valeur.
- Vous pouvez vérifier si une variable est de type `Float` en utilisant la fonction PHP **`is_float()`**, la fonction **`is_double()`** ou **`is_real()`**.

```
<?php
  $x=22.41;
    echo $x;  //22.41
?>
```

4. **`String` (Chaîne de caractères)**

- Le type chaîne de caractères (`string`) permet de stocker toute séquence de caractères sur un octet (code ASCII compris entre 0 et 255), sans limite de taille.
- Une chaîne est une séquence de caractères. PHP travaille en ASCII soit 256 caractères. Il n'y a pas de limite théorique pour la taille de la chaîne.
- On distingue trois syntaxes pour utiliser un chaîne
- **Les guillemets simples** `'...'`(apostrophes) : Comme en shell, tous les caractères inclus dans la chaîne sont sortis tels quels sans interprétation. Si vous devez afficher un guillemet simple, il faudra l'échapper : `'`
- **Les guillemets doubles** `"..."`: Certaines séquences de caractères sont interprétées et les variables sont substituées (remplacées par leur valeur).
- **HereDoc**: Identique aux HereScripts en Shell. Le texte saisi jusqu'à un délimiteur spécifié est placé dans la variable.
- Voici le tableau issu de la documentation PHP des séquences pouvant être utilisés avec les guillemets doubles.

Séquence	Valeur
\n	Nouvelle ligne (linefeed, LF ou 0x0A (10) en ASCII)
\r	Retour à la ligne (carriage return, CR ou 0x0D (13) en ASCII)

apcpedagogie.com

Séquence	Valeur
\t	Tabulation horizontale (HT ou 0x09 (9) en ASCII)
\\	Antislash
\$	Caractère $
\"	Guillemets doubles
\[0-7]{1,3}	Une séquence de caractères qui permet de rechercher un nombre en notation octale.

apcpedagogie.com

Séquence	Valeur
\x[0-9A-Faf]{1,2}	Une séquence de caractères qui permet de rechercher un nombre en notation hexadécimale. **Exemple:**

```
<?php
  echo 'Attention l'abus d'alcool est dangereux';
  $var=2345;
   echo "la valeur de \$var est $var\n";
  $toto= <<<FIN
?>
```

- On peut facilement concaténer deux chaînes avec l'opérateur point ".". On peut ajouter du texte à une chaîne avec l'opérateur point égal ".=".

```
<?php
  $str="Salut les Amis !\n";
  $str.="Comment ça va ?"; // "Salut les Amis !\nComment ça va ?
  $str2=$str."\n"; // "Salut les Amis !\nComment ça va ?\n
?>
```

- Si vous devez utiliser une chaîne de caractères comme valeur numérique (dans une addition par exemple, attention à son contenu. La chaîne sera de type double (réel) si elle contient un 'e' ou un 'E'. Sinon ce sera un entier.
- La valeur numérique est ensuite définie par le début de la chaîne. Si la chaîne commence par une valeur numérique, elle sera utilisée, sinon elle sera égale à 0 zéro. Si la première expression est une chaîne, le type de variable dépend de la seconde expression.

```
<?php
  $val=10+"2.55"; // float, 12.55
  $val=1+"toto2"; // 1 + 0 = 1
  $val=2+"3 petits cochons"; // 2 + 3 = 5 (le 3 est en premier dans la chaîne)?
?>
```

5. **Boolean (Les booléens)**

- Un booléen peut prendre deux valeurs TRUE ou FALSE. Les deux constantes **TRUE** et **FALSE** peuvent être utilisées sans aucune distinction de casse (pas de différences entre

apcpedagogie.com

les majuscules et les minuscules).

- L’utilisation d’expressions booléennes est à la base de la création des instructions conditionnelles, qui permettent de gérer le déroulement d’un algorithme.

```
<?php
  $var=FALSE; // FALSE, False, false, ...
  $var2=True; // TRUE, True, true, ...
?>
```

6. **`Object`**

- Les objets sont définis comme des instances de classes définies par l’utilisateur qui peuvent contenir à la fois des valeurs et des fonctions. C’est un sujet un peu avancé qui sera discuté en détail dans d’autres articles.
- Les objets sont des instances de leur classe et contiennent leurs propriétés. Les objets stockent non seulement la valeur, mais également les informations qui indiquent comment traiter les données qui y sont stockées. Nous découvrirons le type de données des objets PHP dans son tutoriel dédié.
- Vous pouvez vérifier si une variable est un objet en utilisant la fonction PHP **`is_object()`**.

```
<?php
  $var=FALSE; // FALSE, False, false, ...
  $var2=True; // TRUE, True, true, ...
?>
```

7. **`NULL`**

- `NULL` est un type spécial de variables qui ne peuvent contenir qu’une seule valeur, c’est-à-dire NULL. Nous suivons la convention qui consiste à l’écrire en majuscules.
- La valeur vide `NULL` est utile dans les cas où l'on veut exprimer que quelque chose n'existe pas. Par exemple, un article n'a pas de catégorie. Parfois, null est incorrectement substitué à null (0) ou à des chaînes vides (''), mais ce n'est pas une bonne solution pour exprimer la non-existence.
- Vous pouvez vérifier si une variable est NULL en utilisant la fonction PHP **`is_null()`**.

```
<?php
  $x = NULL;
```

apcpedagogie.com

```
  echo $x;    // n'affiche rien
  $var = "" // n'est pas nul mais une chaîne vide
?>
```

8. Applications

1. **App01 :**
   - Créer une page nommée **application01_exphp.php**
   - Déclarer deux variables **$nbr1** et **$nbr2** contenant des nombres entiers.
   - Déclarer une variable **$somme**
   - Afficher le type de la variable **$somme**
   - Affecter à **$somme** la somme des nombres **$nbr1** et **$nbr2**.
   - Afficher a nouveau le type de **$somme**
   - Que remorquez-vous?
   - Reprendre les questions précédentes en utilisant des nombres réels a exposant.
2. **App02 :**
   - Créer une page nommée **application02_exphp.php**
     - Déclarer une constante **PI** contenant la valeur 3.14.
     - Déclarer une variable **$rayon** et une variable **$surface** et affecter leur des valeurs.
     - Ecrire le code permettant de calculer la surface du disque de rayon **$rayon** en utilisant la constante **PI**
     - Afficher cette surface.

9. **Références**:
   - https://www.php.net
   - Livre PHP5 Cours et exercices Jean Engels 2e édition
   - Technologies de l'Information et de la Communication 4 année secondaire

---

Sommaire du cours PHP

Source : https://apcpedagogie.com/les-types-de-donnees-en-php · téléchargé le 07/11/2026 · © APC Pédagogie · apcpedagogie.com