Les types de données

Le langage PHP de base

El hadji Mamadou NGUER Enseignant chercheur en Informatique à l'UVS

Chapitre 2 : Les types de données

Objectifs spécifiques : A la suite de ce chapitre, l'étudiant doit être capable de:

1. Décrire les types de données du langage,
2. D'utiliser les fonctions sur les types de données

Chapitre 2 : Les types de données

Plan de la séquence :

1. Généralités
2. Les chaines de caractères (String)
3. Les entiers (Integer)
4. Les nombres à virgule flottante
5. Les booléens (Boolean)
6. Les tableaux (Array)
7. Les objets (Objects)
8. Les valeurs nulles (NULL)
9. Les constantes

Généralités

- En programmation informatique, un type de donnée, ou simplement un type, définit la nature des valeurs que peut prendre une donnée, ainsi que les opérateurs qui peuvent lui être appliqués.
- Les types de données en PHP sont :
  - les chaines de caractères (String)
  - Les entiers (Integer)
  - Les nombres à virgule flottante (Floating Point Numbers)
  - Les booléens (Boolean)
  - Les tableaux (Array),
  - Les objets (Object)
  - Les valeurs nulles (NULL)
- La fonction **var_dump** permet de connaître le type d'une variable

Les chaines de caractères (String)

- Une chaîne est une séquence de caractères, comme "Comment vas-tu?".
- Une chaîne peut être n'importe quel texte entre guillemets. Vous pouvez utiliser des guillemets simples ou doubles:
- **Exemple**

```
<?php
    $x = "Bonjour tout le monde!";
    echo $x;
    echo "<br>";
    $x = "Bonjour tout le monde!";
    echo $x;
    var_dump($x);   //Donne le type de la variable $x
?>
```

Les chaines de caractères (String)

Il existe dans PHP plusieurs fonctions pour manipuler des chaînes.

- **strlen($str)** : retourne le nombre de caractères d'une chaîne
- **str_word_count ($str):** retourne le nombre de mots d'une chaîne
- **strrev ($str):** inverse une chaîne de caractères
- **strpos($str,$str1)** : retourne la 1e occ de $str1 dans $str et null sinon.
- **str_replace ($str1,$str2,$str)** : remplace $str1 par $str2 dans $str
- **strtolower($str)** : conversion en minuscules
- **strtoupper($str)** : conversion en majuscules
- **trim($str)** : suppression des espaces de début et de fin de chaîne
- **substr($str,$i,$j)** : retourne une sous chaîne de **$str** de taille **$j** à partir de la position **$i**
- **strnatcmp($str1,$str2)** : comparaison de 2 chaînes
- **addslashes($str)** : déspécialise les caractères spéciaux (', '', \)
- **ord($char)** : retourne la valeur ASCII du caractère **$char**

Pour une référence complète des fonctions sur les chaînes de caractères :
https://www.w3schools.com/php/php_ref_string.asp

Les entiers (Integer)

- Un nombre entier est un nombre sans décimales.
- **Règles pour les nombres entiers:**
  - Un entier doit avoir au moins un chiffre (0-9) et peut être positif ou négatif
  - Un entier ne peut contenir ni virgule, blancs, ni un point décimal
  - Un entiers peut être spécifié dans trois formats: décimal (base 10), hexadécimale (base-16 - avec le préfixe 0x) ou octal (base-8 - le préfixe 0)
- **Exemple**

```
<?php
    $x = 5985;
    $x = -345; // negative number
    $x = 0x8C; // hexadecimal number
    $x = 047; // octal number
$?>
```

Les nombres à virgule flottante

- Un nombre à virgule flottante est un nombre avec un point décimal ou un nombre sous forme exponentielle.

- **Exemple**

```
<?php
 $x = 10.365;
 echo "$x <br>";

 $x = 2.4e3;
 echo " $x <br>";

 $x = 8E-5;
 echo " $x <br>";
?>
```

Les booléens (Boolean)

- Les booléens ne peuvent avoir que les deux valeurs : TRUE ou FALSE.
- Ils sont souvent utilisés dans des tests conditionnels qui seront dans la suite du cours.
- **Exemple**

```
<?php
 $x=true;
 $y=false;

 echo " $x et $y<br>";
?>
```

Les tableaux (Array)

- Un tableau permet de stocker plusieurs valeurs dans une seule variable.
- L'exemple suivant crée un tableau et affiche ses valeurs en utilisant un boucle for
- Les boucles seront étudiés dans la suite du cours
- **Exemple**

```
<?php
 $etu=array("Moussa","Fatou","Marianne");

 for ($i=0;$i<count($etu);$i++){
    echo $etu[$i];
  }

//count($etu) donne le nombre de valeurs du tableau $etu.
?>
```

Les objets (Objects)

- Un objet est un type de données qui stocke des données et des informations sur la façon de traiter ces données.
- En PHP, un objet doit être explicitement déclaré comme suit :
  - Premièrement, on doit déclarer la classe objet correspondant en utilisant le mot-clé class. Une classe est une structure qui peut contenir des propriétés et des méthodes.
  - On définit ensuite le type de données de la classe d'objet, puis on utilise le type de données dans les instances de cette classe.
- L'exemple suivant montre comment déclarer et utiliser un objet en PHP
- Les détails sur les objets seront étudiés dans la suite du cours

Les objets (Objects)

- **Exemple :**

```php
<?php
    //Déclaration de la classe
    class Etu{
        //Déclaration des propriétés
        var $nom;

        //Déclaration d'un méthode spéciale appelée constructeur
        function __construct($nm="Sénégal"){
            $this->nom = $nm;
        }
        //Déclaration d'une méthode
        function afficheNom(){
            echo "<br>".$this->nom;;
        }
    }
//Déclaration d'une instance de la classe Etu. Sans paramètre
$etu1 = new Etu();
$etu1->afficheNom();//Appel de la méthode afficheNom

//Déclaration d'une autre instance de la classe Etu. Avec Paramètre
$etu2 = new Etu("Saliou");
$etu2->afficheNom();
?>
```

Remarque : Pour avoir plusieurs constructeurs, on les nomme __construct1, __construct2, etc.

Les valeurs nulles (NULL)

- La valeur spéciale NULL indique qu'une variable n'a pas de valeur.
- NULL est la seule valeur possible du type de données NULL.
- Elle indique si une variable est vide ou non. Il est utile de distinguer entre la chaîne vide et les valeurs nulles de bases de données.
- Une variable peut être vide en lui affectant la valeur NULL:
- **Exemple**

```
<?php
    $x="Hello!";
    $y=null;
    $z="world";
    echo $x,$y,$z;
?>
```

Les constantes

- Une constante est comme une variable, sauf qu'une fois définie, elle ne pourra être modifiée ou indéfinie.
- Une constante contient une valeur simple
- Le nom d'une constante valide commence par une lettre ou un trait de soulignement (pas de signe $ avant le nom d'une constante).
- Contrairement aux variables, les constantes sont automatiquement globales dans tout le script.
- Le diapositive suivant montre comment définir et utiliser une constante dans un programme PHP.

Les constantes

Définir une constante PHP

- La fonction **define** permet de définir une constante
- Elle prend trois paramètres:
  - Le premier définit le nom de la constante,
  - le second définit la valeur de la constante, et
  - le troisième spécifie si le nom de la constante est sensible à la casse ou pas, elle est à false par défaut.
- **Exemple**

```php
<?php
define("PI", 3.14); //Sensible à la casse
define("GREETING", "Bonjour", true); //Non sensible à la casse
echo PI;
echo greeting;
?>
```

Conclusion

- Au cours de cette séquence, nous avons :
  - Décrit les types de données du langage,
  - Et mettre en pratique les fonctions sur les types de données

- Pour consolider les connaissances acquises dans cette séquence, nous vous conseillons de faire :
  - Les tests de connaissance de la séquence
  - Et les exercices de la fiche de TP