**Le mot-clé static**

Le fait de déclarer des propriétés ou des méthodes comme statiques vous permet d'y accéder sans avoir besoin d'instancier la classe. On ne peut accéder à une propriété déclarée comme statique avec l'objet instancié d'une classe (bien que ce soit possible pour une méthode statique).

**Propriétés statiques**

On ne peut pas accéder à des propriétés statiques à travers l'objet en utilisant l'opérateur ->.

Comme n'importe quelle autre variable PHP statique, les propriétés statiques ne peuvent être initialisées qu'en utilisant un littéral ou une constante; les expressions ne sont pas permises. Ainsi, vous pouvez initialiser une propriété statique avec un entier ou un tableau, mais pas avec une autre variable, ni avec la valeur de retour d'une fonction, ni avec un objet.

<?php

class StaticA

{

const UNECONSTANTE = '300';

// Déclaration de propriétés statiques

public static $proprieteStatique1 = 10; // OK

public static $proprieteStatique2 = self::UNECONSTANTE; // OK

// Déclaration de propriétés d'instances

public $proprieteSimple1 = 100;

public $proprieteSimple2 = self::UNECONSTANTE;

// Mauvaise déclaration de propriétés statiques

public static $proprieteStatique3 = self::proprieteStatique1; // Génère une erreur

public static $proprieteStatique4 = rand(); // Génère une erreur

public static $proprieteStatique5 = new Personnage(); // Génère une erreur

}

echo StaticA::$proprieteStatique1."<br>"; // Affiche 10

echo StaticA::$proprieteStatique2."<br>"; // Affiche 300

$objetStatique = new StaticA();

echo StaticA::$proprieteSimple1."<br>"; // Affiche 100

echo StaticA::$proprieteSimple2."<br>"; // Affiche 300

echo StaticA::$proprieteStatique3."<br>"; // Erreur

echo StaticA::$proprieteStatique4."<br>"; // Erreur

echo StaticA::$proprieteStatique5."<br>"; // Erreur

?>

**Notez la présence de la pseudo-variable self qui permet d'accéder aux éléments statiques de la classe.**

**Méthodes statiques**

Comme les méthodes statiques peuvent être appelées sans qu'une instance d'objet n'ait été créée, la pseudo-variable $this n'est pas disponible dans les méthodes déclarées comme statiques. La pseudo-variable self y reste quant à elle utilisable.

<?php

class StaticB

{

public static $count = 0;

public static function methodeStatique(){

self::$count++;

echo "Test appel de la méthode statique n°".self::$count."<br>";

}

}

// Appels d'une méthode statique

StaticB::methodeStatique();

// ou

$className = "StaticB";

$className::methodeStatique();

// Appel possible depuis un objet instancié mais contraire aux bonnes pratiques

$objetStatiqueB = new StaticB();

$objetStatiqueB->methodeStatique();

?>