Programmation orientée objet en PHP

Le constructeur et this

Lors de la création de notre objet de type Humain nous avons utilisé l'opérateur new et une "méthode" spéciale nommée constructeur. Par défaut chaque classe possède un constructeur, appelé constructeur par défaut ou encore constructeur "vide". Pour appeler ce constructeur nous utilisons le nom de la classe suivi de parenthèses comme pour une fonction classique et sans argument.
Le rôle du constructeur est d'initialiser les attributs de l'objet, c'est à dire leur donner une valeur lors de la création de l'objet. Mais le constructeur par défaut, lui, ne modifie pas les valeurs des attributs. Si je souhaite initialiser les attributs lors de l'appel au constructeur je vais devoir créer un constructeur personnalisé.
Le constructeur personnalisé possède un nom particulier : __construct. Comme n'importe quelle fonction, il peut posséder des paramètres, y compris des paramètres facultatifs. Je pourrai donc appeler ce constructeur en lui donnant en paramètre un age, une taille et un poids. La signature de mon constructeur sera donc __construct($page, $ptaille, $ppoids). La lettre p devant chaque nom de variable signifie "paramètre", et nous permettra d'éviter les confusions entre les attributs et les paramètres du constructeur.
Dans le corps de la fonction il va falloir que j'assigne à l'attribut age la valeur de page, à l'attribut taille la valeur de ptaille, et à l'attribut poids la valeur de ppoids. Pour effectuer cela je vais avoir besoin du mot clé "this" qui signifie l'instance courante, l'objet que je suis en train de créer. Dans le constructeur, j'écrirai que :

- la valeur de l'attribut age de l'objet que je suis en train de créer (l'instance courante) est égal à la valeur de page
- la valeur de l'attribut taille de l'objet que je suis en train de créer (l'instance courante) est égal à la valeur de ptaille
- la valeur de l'attribut poids de l'objet que je suis en train de créer (l'instance courante) est égal à la valeur de ppoids

Ce qui nous donne en PHP :

```php
<?php

// Humain.php
class Humain {

    public $age;
    public $taille;
    public $poids;

    public function __construct($page, $ptaille, $ppoids) {
        $this->age = $page;
        $this->taille = $ptaille;
        $this->poids = $ppoids;
    }

}

$sparrow = new Humain(42, 1.84, 75);
echo "age = $sparrow->age" . PHP_EOL;
echo "taille = $sparrow->taille" . PHP_EOL;
echo "poids = $sparrow->poids" . PHP_EOL;
```

L'appel au constructeur se fait toujours en utilisant le nom de la classe. Une fois qu'un constructeur personnalisé a été défini, le constructeur par défaut n'existe plus. L'appel à un constructeur vide déclencherait une erreur puisque le constructeur personnalisé attend plusieurs arguments.

```php
<?php

// Humain.php
class Humain {

    public $age;
    public $taille;
    public $poids;

    public function __construct($page, $ptaille, $ppoids) {
        $this->age = $page;
        $this->taille = $ptaille;
        $this->poids = $ppoids;
    }

}

$sparrow = new Humain(); // erreur !
echo "age = $sparrow->age" . PHP_EOL;
echo "taille = $sparrow->taille" . PHP_EOL;
echo "poids = $sparrow->poids" . PHP_EOL;
```

Le mot clé this peut aussi être utilisé dans les méthodes pour accéder à un attribut ou à une autre méthode, par exemple si nous ajoutons une méthode afficherPoids() qui affichera la valeur de l'attribut poids de la personne :

```php
<?php

// Humain.php
class Humain {

    public $age;
    public $taille;
    public $poids;

    public function __construct($page, $ptaille, $ppoids) {
        $this->age = $page;
        $this->taille = $ptaille;
        $this->poids = $ppoids;
    }

    public function afficherPoids() {
        echo "Je pèse {$this->poids}kg.";
    }

}

$sparrow = new Humain(42, 1.84, 75);
$sparrow->afficherPoids();
```

Autre exemple avec une fonction afficher() qui appelle la méthode afficherPoids() :

```php
<?php

// Humain.php
class Humain {

    public $age;
    public $taille;
    public $poids;

    public function __construct($page, $ptaille, $ppoids) {
        $this->age = $page;
        $this->taille = $ptaille;
        $this->poids = $ppoids;
    }
```

```php

    public function afficherPoids() {
        echo "Je pèse {$this->poids}kg.";
    }

    public function afficher() {
        $this->afficherPoids();
    }

}

$sparrow = new Humain(42, 1.84, 75);
$sparrow->afficher();
```

Fin

NAVIGATION

Accueil
- Ma page
  - Pages du site
  - Mon profil
  - Cours actuel
    - POO PHP
      - Participants
      - Généralités
      - La programmation orientée objet : premiers pas
      - Introduction
      - Classes et objets
      - T.P. première classe
      - Les classes en PHP
      - T.P. Personne
      - **Le constructeur et this**
      - T.P. Personne v2
      - Accesseurs et mutateurs
      - Les méthodes magiques
      - T.P. Personne v3
      - Les constantes de classe
      - Le modificateur static
      - T.P. Personne v4
      - Une Personne et une Adresse : la relation has-a
      - Histoire de références
      - T.P. formation
      - T.P. formation v2
      - Une Personne et plusieurs Adresse(s)
      - T.P. formation v3
      - Bricolage et dépendance
      - L'héritage
      - Les interfaces
      - Le typage
      - Les namespaces
      - Les exceptions
      - Les bases de données avec PDO
      - Les tests avec PHPUnit
      - Petite application version 2
      - Petitie application version 3
  - Mes cours

ADMINISTRATION

Administration du cours

Réglages de mon profil

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