

PHP Objet

Les méthodes magiques



Juin 2018

1.

Définitions

Des méthodes prédéfinies

Les méthodes magiques

Se sont des méthodes qui sont appelées automatiquement par PHP lors d'un événement

Par exemple lorsque qu'on crée un instance d'un objet, la méthode magique de la classe **__construct** sera appelée

Si elles ne sont pas définies, elle ont un comportement par défaut

2.

Liste des méthodes

Quelles sont ces méthodes

Constructeur & destructeur

__construct appelée lors de l'instanciation d'un objet

```
public function __construct($param)
{
    echo "Objet créée";
}
```

__destruct appelée lors de la destruction d'un l'objet

```
public function __destruct()
{
    echo "Objet détruit";
}
```

call & staticCall

__call est appelée quand une méthodes inaccessible est appelée (ex. privé ou non définie)

```
public function __call($name, $attr)
{
    echo "Appel de la méthode " .
        $name;
}
```

__staticCall pour les méthodes statiques

```
public function __staticCall($name,
    $attributes)
{
    echo "Appel de la méthode
        statique " . $name;
}
```

get & set

__get est appelée lorsqu'on veut accéder à une propriété inaccessible (private ou non définie)

```
public function __get($name)
{
    echo "L'attribut " . $name . "
        est lu";
}
```

__set est appelée lorsqu'on veut définir une propriété inaccessible

```
public function __set($name, $val)
{
    echo "L'attribut " . $name . "
        est définie";
}
```

isset & unset

__isset est appelée lors de l'appel de la méthode **isset** ou **empty** sur une propriété inaccessible

```
public function __isset($name)
{
    echo "L'attribut " . $name . "
        est testé";
}
```

__unset est appelée lors de l'appel de la méthode **unset** sur une propriété inaccessible

```
public function __unset($name)
{
    echo "L'attribut " . $name . "
        est définie";
}
```


sleep & wakeup

__sleep est appelée lors de la linéarisation d'un objet

```
public function __sleep()  
{  
    echo "Linéarisation de l'objet";  
    return ["prop1", "prop2"];  
}
```

__wakeup est appelée lors de la délinéarisation d'un objet

```
public function __wakeup()  
{  
    echo "Délinéarisation de  
        l'objet";  
}
```

toString & invoke

__toString retourne une chaîne de caractère afin d'afficher un objet

```
public function __toString()  
{  
    return $this->getTitre();  
}
```

__invoke est appelé lorsqu'un objet est utilisé comme une fonction

```
public function __invoke($titre)  
{  
    return $this->setTitre($titre);  
}
```

clone

Lorsque l'on clone un objet, la méthode `__clone` du nouvel objet est appelée

```
public function __clone()  
{  
    self::$counter++;  
}
```

set_state & debugInfo

__set_state est une méthode statique qui est appelée lors de l'appel de la méthode **var_export**

```
public static function __set_state($array)
{
    $obj = new A;
    $obj->var1 = $array['var1'];
    return $obj;
}
```

__debugInfo est appelée par la méthode **var_dump**, elle permet de définir ce qui sera affiché

```
public function __debugInfo()
{
    return array(
        "Titre" => $this->titre,
    );
}
```