

# PHP Objet Sérialisation & clonage



Juin 2018

# 1.

## Sérialisation

Transformer un objet en chaîne de caractères

## Sérialisation

Il est parfois utile de sauvegarder les attributs d'objet dans une **session** ou une **base de donnée** pour le récupérer ensuite.

La méthode **serialize** permet de transformer les valeurs des attributs d'un objet en chaîne (linéarisation)

**unserialize** transforme une chaîne valide en objet

## Sérialisation

```
class Article
{
    public function
    __construct($title){
        $this->title = $title;
    }
    private $date;
    private $title;
    private $content;
}
```

```
$article = new Article("Titre");
$articleStr = serialize($article);
echo $articleStr;
$article2 = unserialize($articleStr);
```

# 2.

## **Le clonage**

Faire une copie proprement

## Le clonage

Lorsqu'un objet est créé, PHP crée un identifiant de cet objet et la stock dans la variable de l'objet déclarée

Si nous faisons une simple affectation (*\$copie = \$obj*) de variable, les propriétés modifiées sur un objet le seront également sur l'objet copié

La méthode **clone** permet de régler ce problème

## Clonage

```
class Article
{
    public function
    __construct($title){
        $this->title = $title;
    }
    private $date;
    private $title;
    private $content;
}
```

```
$a = new Article("Article");
$aCopie = $a;
$aCopie->setTitle("New");
echo $a->getTitle(); // New
$aClone = clone($a);
$aClone->getTitle("Clone");
echo $a->getTitle(); // New
echo $Clone->getTitle(); // Clone
```



## Le clonage

Lors d'un clonage, la méthode magique `__clone` de l'objet cloné est appelée.