

# PHP

PHP: Hypertext Preprocessor

## OBJET

### INSTANCIATION

```
$water = new Drink();  
$beer = new Booze("Chimay rouge", 33, 9);  
$whisky = new Booze("Caol ila", 4, 43);
```

### UTILISATION

Affichage d'un objet  
avec `__toString` :  
`echo $beer;`

Appel de méthode *static* :  
`echo Booze::howMany() . " dranked!";`

Utilisation de constante de classe :  
`echo Booze::MAX_TO_DRIVE;`

Ajout d'attribut à la volée (bizarre) :  
`$beer->color = "Brune";`

### AUTOLOAD

Utile pour remplacer les includes

```
function loadMyClass($className) {  
    include_once $className.".php";  
}  
spl_autoload_register('loadMyClass');
```

### METHODES MAGIQUES

Méthodes exécutées automatiquement lors  
d'accès à une instance

<code>__get</code>	Lecture d'un attribut
<code>__set</code>	Ecriture d'un attribut
<code>__call</code>	Appel d'une méthode
<code>__callstatic</code>	Appel d'une méthode <i>static</i>
<code>__isset</code>	isset sur un attribut
<code>__unset</code>	unset sur un attribut

<https://www.php.net>

### EXEMPLE

```
class Drink {  
    private $_name = "Unknown";  
    private $_volume = 0;  
  
    public function __construct($name="Water",  
                                $volume=12.5) {  
        $this->_name = $name;  
        $this->_volume = $volume;  
    }  
  
    public function getName() {  
        return $this->_name;  
    }  
    public function setName($name) {  
        $this->_name = $name;  
    }  
  
    public function getVolume() {  
        return $this->_volume;  
    }  
    public function setVolume($volume) {  
        $this->_name = $volume;  
    }  
}  
  
class Booze extends Drink {  
    const MAX_TO_DRIVE = 0.5;  
    private $_degree;  
    public static $_nb = 0;  
  
    public function __construct($name, $volume,  
                                $degree) {  
        parent::__construct($name, $volume);  
        $this->_degree = $degree;  
        self::$_nb++;  
    }  
  
    public function __toString() {  
        return $this->getName().", ".  
            $this->getVolume()."cl, ".  
            $this->getDegree()."%";  
    }  
  
    public function getDegree() {  
        return $this->_degree;  
    }  
    public function setDegree($degree) {  
        $this->_degree = $degree;  
    }  
  
    public static function howMany() {  
        return self::$_nb;  
    }  
}
```