

Base du vocabulaire technique du développeur (extrait)

```
<?php
namespace jeu;

class Personnage {

    private $_force;
    private $_localisation;
    private $_experience;
    private $_degats;

    // Déclarations des constantes en rapport avec la force.
    const FORCE_PETITE = 20;
    const FORCE_MOYENNE = 50;
    const FORCE_GROUPE = 80;

    public function __construct($forceInitiale)
    {
        // N'oubliez pas d'assigner la valeur d'un attribut
        // uniquement depuis son setter !
        $this->setForce($forceInitiale);
    }

    public function setForce($force) {
        if (in_array($force, [self::FORCE_PETITE, self::FORCE_MOYENNE]) {
            $this->_force = $force;
        }
    }

    public function getForce() : int {
        return $this->_force;
    }
}
```

nom du package

nom de la classe

Déclaration des **attributs**

Par convention, les **CONSTANTES** sont notées en majuscule.

paramètre (conteneur)

argument (valeur)

Déclaration et définition d'une méthode d'instance

Type (de retour) de la méthode

```
$p = new Personnage(42);
```

Instanciation de la classe Personnage.
Création d'un objet de type Personnage.
remarque : 42 est un argument passé au constructeur de Personnage.

```
echo "Start personnage with : " . $p->getForce() . " force";
```

Appel de la méthode *getForce* de l'objet référencé par \$p

L'instruction `$p->getForce()`, `getForce` fait référence à la méthode de même nom dans la classe `Personnage`, et `$p` à : **this dans le corps de la méthode**

Parmi les types de variable, on dénombre : **variable locale, variable globale, attribut d'instance, attributs de classe (static), paramètre.**

Attention : contrairement à ce que l'on pourrait penser, un constructeur n'est pas une méthode. Il ne s'emploie qu'avec *new* (appelé *opérateur de création d'instance*).