

MÉMENTO

POO EN PHP

Voici les éléments de syntaxe à connaître au niveau de la POO dans le langage PHP.

```
class NomClasse {  
    ...  
}
```

Une classe peut hériter d'une autre classe :

```
class NomClasse extends ClasseMere {  
    ...  
}
```

Les propriétés se déclarent en début de classe, en précisant leur visibilité mais pas leur type (PHP est un langage non typé) :

```
private $propriétéPrivée;  
protected $propriétéProtégée;  
public $propriétéPublique;
```

Les méthodes sont gérées comme des fonctions classiques PHP, excepté qu'elles ont une visibilité. Pour rappel, "function" désigne aussi bien les fonctions que les procédures : la distinction se fait avec la présence (ou non) du return.

```
public function uneMethode(...){  
    ...  
}
```

Le constructeur porte le nom «__construct» :

```
public function __construct(...){  
    ...  
}
```

Pour utiliser une propriété dans une méthode, il faut faire précéder son nom, non pas de "\$" mais de "\$this->" :

```
public function uneMethode(...){  
    $this->unePropriété = ...;  
}
```

Même remarque pour l'appel d'une méthode dans une autre méthode :

```
public function uneMethode(...){  
    $this->uneAutreMethode(...);  
}
```

D'une manière générale, en PHP, l'accès à une propriété ou une méthode, que ce soit dans la classe ou à partir d'un objet, ne se fait pas avec "." comme dans la plupart des langages, mais avec "->" car le point est déjà utilisé pour la concaténation de chaîne.

Une propriété ou une méthode peuvent être statiques. Il suffit de préciser "**static**" :

```
private static $propriétéStatique;
```

Pour ensuite l'utiliser, il faut la faire précéder du nom de la classe suivi de "::" (cette fois, en gardant \$) :

```
public function uneMethode(...){  
    NomClasse::$propriétéStatique = ...;  
}
```

La création d'un objet en lui affectant une instance de la classe, se fait de façon classique. C'est le constructeur qui est appelé (la méthode `__construct(...)`) qui peut attendre des paramètres.

```
$unObjet = new NomClasse(...);
```

Pour appeler une méthode publique sur un objet, c'est donc "->" et non pas "." qu'il faut utiliser :

```
$unObjet->uneMethode(...);
```

Si une classe a besoin d'accéder à une autre classe, il suffit de l'inclure en début de fichier :

```
include_once("NomAutreClasse.php");
```



Les cours du CNED sont strictement réservés à l'usage privé de leurs destinataires et ne sont pas destinés à une utilisation collective. Les personnes qui s'en serviraient pour d'autres usages, qui en feraient une reproduction intégrale ou partielle, une traduction sans le consentement du CNED, s'exposeraient à des poursuites judiciaires et aux sanctions pénales prévues par le Code de la propriété intellectuelle. Les reproductions par reprographie de livres et de périodiques protégés contenues dans cet ouvrage sont effectuées par le CNED avec l'autorisation du Centre français d'exploitation du droit de copie (20, rue des Grands-Augustins, 75006 Paris).

CNED, BP 60200, 86980 Futuroscope Chasseneuil Cedex, France

© CNED 2021

87D22ANWB1121

