



Programmation Orientée Objet (POO)

DES OBJETS EN PHP !?



Programmation impérative vs POO

Idée centrale :

- S'appuyer sur un **modèle proche** de celui que nous percevons lorsque nous observons le **monde**
- En **POO**, le raisonnement se fait en termes de **fonctionnalités** et de **propriétés**, alors que la programmation impérative se focalise essentiellement sur les traitements



La notion d'objet : généralités

- **Objet** : représentation abstraite d'une entité
- Structure d'un objet :
 - des propriétés (ou caractéristiques, **attributs**)
 - des fonctionnalités (ou fonctions, **méthodes**)
- **Attributs** : **état** d'un objet
- **Méthodes** : **comportement** d'un objet

EN RÉSUMÉ :

Monde réel	POO
Entité	Objet
État	Attribut(s)
Comportement	Méthode(s)



La notion d'objet : exemple

- Comment décrire un **rectangle** dans une logique « *orientée objet* » ?
- Un **rectangle**, ça peut être :
 - Des **attributs** :
 - une longueur
 - une hauteur
 - Des **méthodes** :
 - calculePerimetre()
 - calculeAire()



La notion d'objet : exemple

- Selon le contexte, un **rectangle** peut également être :
 - Des **attributs** :
 - une couleur de bordure
 - une couleur de fond
 - une position
 - etc.
 - Des **méthodes** :
 - colorerFond()
 - déplacer()
 - etc.



La notion d'objet en PHP : utiliser un objet

- Comment **utiliser** notre **rectangle** dans la **syntaxe PHP** :

```
<?php
include('class.rectangle.php');

/* Instanciation de l'objet $rectangle1 */
$rectangle1 = new Rectangle(12,3);

echo "Aire du rectangle : " . $rectangle1->calculAire();
```



La notion d'objet en PHP : définir une classe

```
<?php
class Rectangle
{
    /* Définition des attributs */
    private $longueur;
    private $hauteur;

    /* Constructeur de la classe */
    public function __construct($l, $h)
    {
        $this->longueur = $l;
        $this->hauteur = $h;
    }

    /* Exemple d'un autre méthode spécifique : un accesseur */
    public function getLongueur()
    {
        return $this->longueur;
    }

    /* Exemple d'un autre méthode spécifique : un modificateur */
    public function setLongueur($l)
    {
        $this->longueur = $l;
    }
}
```

```
/* ----- SUITE et FIN ----- */

/* Méthode de calcul de l'aire */
public function calculAire()
{
    return ($this->longueur * $this->hauteur);
}

/* Destructeur de la classe */
public function __destruct()
{
    /* Instructions visant à libérer la mémoire (ou autres). */
}
}
```



La notion d'objet en PHP : terminologie

- En POO :

- On ne définit pas des **objets**, mais, plus précisément, des **classes d'objets** :

```
class Rectangle { ... }
```

- On ne crée pas des objets, mais on les **instancie** :

```
$rectangle1 = new Rectangle(12,3);
```

- L'objet `$rectangle1` est se nomme une **instance de classe** `Rectangle()`
- L'objet `$rectangle1` est alors de **type** `Rectangle`
- On **accède** à un attribut ou un méthode via : `$rectangle1->calculAire();`



Conclusion

- POO : **représentation** d'objets du **monde** sous une forme **informatique**
- En PHP :
 - **définir** une **classe** d'objets : `class, public, private, __construct(), __destruct(), $this, return`
 - **instancier** un objet, puis l'utiliser : `include, new, ->`
- **À retenir** : objet, classe, instance, type, attribut, méthode

>>> MISSION START ! <<<