

Développer en Orienté Objet avec le langage PHP

L'héritage

L'héritage établit une relation de généralisation-spécialisation.

Possibilité d'une classe fille d'hériter des attributs et méthodes d'une classe mère. C'est une spécialisation de la classe fille par rapport à la classe mère. Une classe B hérite d'une classe A. La classe A est donc considérée comme la classe mère et la classe B est considérée comme la classe fille.

L'héritage

Pour procéder à un héritage (c'est-à-dire faire en sorte qu'une classe hérite des attributs et méthodes d'une autre classe), il suffit d'utiliser le mot-clé **extends**

```
class ClasseFille extends ClasseMere{  
    //instructions  
  
    .....  
  
    .....  
  
}
```

L'héritage

```
class ClassA{  
    // déclaration des attribut de la classe  
    private $attribut_1;  
    // déclaration de méthodes de la classe  
    public function nomMethode(){  
        // instructions  
    }  
}  
  
class ClassB extends ClassA{  
}
```

Le mot clé protected

Comme public et private le mot protected permet de définir la visibilité d'un attribut ou d'une méthode.

Quand un attribut ou une méthode est défini avec le mot clé protected il sera accessible dans toute classes filles de la classe qui la défini.

Classe abstraite

Une classe abstraite est une classe non instanciable. L'objectif est de représenter une factorisation, une généralisation. Dans de nombreux cas il est souhaitable de mettre certaines méthodes Par exemple les getters et les setters dans une classe abstraite.

syntaxe

```
abstract class ClassAbstraite {  
    ...  
}
```

Méthode statique, propriété statique

Le fait de déclarer des propriétés ou des méthodes comme statiques vous permet d'y accéder sans avoir besoin d'instancier la classe.

pour déclarer une méthode ou une propriété statique on fait précéder le nom de la propriété ou méthode du mot clé **static** juste après le mot désignant sa portée.

Méthode statique, propriété statique

Les éléments statiques (static) sont des éléments des éléments de classe et non pas des éléments d'instances.

La syntaxe de déclaration d'une propriété statique est la suivante :

```
public static $nomDePropriete = "valeur_Initiale" ;
```

La syntaxe de déclaration d'une méthode statique est la suivante :

```
public static function methode(){  
    ...  
}
```


Les constantes de classe

Comme pour les méthodes et propriété statiques les constantes de classe sont utilisables dans le script sans instantiation d'objet.

pour définir une constante de classe on utilise le mot clé const.

```
const NOM_DE_CONSTANTE = "valeur";
```

L'opérateur ::

L'opérateur :: , appelé opérateur de résolution de portée, permet d'accéder à des éléments statiques et aux constantes d'une classe.

```
NomClasse::NOMDECONSTANTE;
```

```
NomClasse::$nomDeVariableStatique;
```

```
NomDeClasse::nomDeMethodeStatique();
```

Le mot clé self

Le mot clé **self** et l'opérateur de résolution de portée permettent d'accéder aux éléments statiques et constantes de la classe depuis l'intérieur de celle-ci.

```
self::NOMDECONSTANTE;
```

```
self::$nomDeVariableStatique;
```

```
self::nomDeMethodeStatique();
```

Le mot clé parent

Le mot clé **parent** et l'opérateur de résolution de portée permettent d'accéder aux attributs et méthodes statiques publique ou protégé d'une classe à partir de sa classe fille.

```
parent::NOMDECONSTANTE;
```

```
parent::$nomDeVariable;
```

```
parent::nomDeMethode();
```