

PHP3 - Painter

Liens sur les notions utiles :

Comprendre : [Introduction au SVG](#)

Les Formes :

- Documentation officielle : [Formes de base](#)
- OpenClassRoom : [Cours OpenClassRoom](#)
- Exemples d'usages : [Les bases de svg rectangles et carres](#)

PHP - Objet - Héritage - Abstraction - Polymorphisme:

- Objet Php : <http://php.net/manual/fr/language.oop5.basic.php>
- Visibilité des variables : <http://php.net/manual/fr/language.oop5.visibility.php>
- Abstraction : <http://php.net/manual/fr/language.oop5.abstract.php>
- Constructeur : <http://php.net/manual/fr/language.oop5.decon.php>

Détails des actions à mener dans chaque fichiers

index.phtml : **Template d'affichage**

- Code html qui executera le programme

Index.php : **Program starter**

- Lister des classes dans l'ordre des dépendances.

```
/****** CODE PRINCIPAL *****/  
// Création d'une instance de notre programme  
// Création d'une instance du moteur SVG.  
// Démarrage du program  
// Inclusion du template.
```

Point.class.php : **Manage positions/moves**

- Propriétés : x, y
- Functions :
 - __construct() -> init to 0
 - getter/setter

Shape.class.php : **Manage shapes**

- Propriétés : color, location (qui deviendra dans le construct une instance de **Point()**), opacity
- Functions :
 - __construct() -> init to 0
 - getter/setter

- Méthode abstraite à redéfinir dans chaque class child : draw(SvgRenderer \$renderer)

Rectangle.class.php : **Classe pour faire des rectangles**

- Propriétés : width, height
- Fonctions :
 - __construct() -> son constructeur hérite de son parent (Shape) et init properties to 0
 - getter/setter
 - draw(SVGRenderer \$renderer) : *la méthode abstraite à redéfinir obligatoirement*
 - Utilisation de l'objet renderer pour dessiner une ellipse avec ses propriétés
 - Elle appelle la méthode correspondant à l'objet dans lequel nous sommes (ici -> dans \$renderer->drawRectangle())
 - setSize(\$width, \$height) *init des dimensions du rectangle*

Program.class.php : **Construction des formes à réaliser dans le run et envoi via le \$renderer**

- Fonctions :
 - run(SvgRenderer \$renderer) : *Exécution du programme*
 - Création et initialisation d'un rectangle.
 - Création et initialisation d'une ellipse.
 - Création et initialisation d'un carré.
 - Création et initialisation d'un cercle.
 - Création et initialisation d'un triangle.
 - Ajout des différents objets géométriques au moteur graphique.
 - Exécution du moteur graphique.

SvgRenderer.class.php : **Manage rendered shapes**

Les fonctions utiles : **array_push** et **implode**

- Propriétés :
 - \$results (Tableau d'objets géométriques de la classe Shape ou de ses dérivés (= ses enfants))
 - \$shapes (Tableau de chaînes de caractères de balises SVG)
- Fonctions :
 - __construct() -> init to []
 - addShape(Shape \$shape)
 - Ajout d'un objet géométrique au tableau d'objets
 - drawCircle(Point \$center, \$color, \$opacity, \$radius)
 - Ajout d'une balise SVG
 - drawEllipse(Point \$center, \$color, \$opacity, \$xRadius, \$yRadius)
 - Ajout d'une balise SVG
 - drawRectangle(Point \$origin, \$color, \$opacity, \$width, \$height)
 - Ajout d'une balise SVG
 - drawTriangle(array \$points, \$color, \$opacity)
 - Ajout d'une balise SVG
 - getResult()
 - Ajout d'un conteneur SVG et concaténation de toutes les chaînes de balises SVG.
 - run()
 - Boucle sur le tableau d'objets géométriques.