

# PROJET : GÉNÉRER DES FICHIERS SVG

# OBJECTIFS PROGRESSIFS

# OBJECTIFS

1. Créer différentes classes pour les primitives de base (rectangle, cercle/ellipse) permettant de générer un SVG avec des attributs simples (position, couleur, éventuellement bordure)
2. Générer un SVG avec plusieurs primitives graphiques
3. Avoir la possibilité de réaliser des opérations complexes (mettre en noir et blanc, inverser les couleurs, optimiser la zone d'affichage, ...)
4. Utiliser des primitives plus complexes (groupes, ...)

# Y ALLER ÉTAPE PAR ÉTAPE !

- Commencer par un rectangle ou un cercle uniquement
- Créer un fichier de classe, y mettre les attributs strictement nécessaires
- Définir **toXML()** qui permet de renvoyer la chaîne de caractères XML
- Quand ça marche, complexifier un peu la classe

# DES MÉTHODES UTILES À ENVISAGER :

- Une méthode pour définir la position et une pour traduire
- Une méthode pour gérer la rotation
- Méthodes pour changer certains paramètres (couleur, bordures...)

# RESSOURCES (FICHIERS **ICI**)

- Un squelette de fonction **main** à personnaliser (*App.java*)
- Une classe **File** pour l'écriture dans les fichiers
- Un script **run**
- Une librairie **XML** pour ceux qui ont un **niveau avancé** (*XML.java* et *Element.java*)

# SCRIPT RUN

Si le script ne s'exécute pas en raison d'un problème de droits d'exécution

```
sudo chmod u+x run
```

# CONSEILS



# ENTRAÎNEZ-VOUS À MANIPULER LE SVG

- Avant d'implémenter une nouvelle fonction dans votre classe, testez là “manuellement”
- Créez un fichier “.xml”, mettez-y le code de votre figure, renommez en SVG pour visualiser le résultat
- Inspirez-vous du [tutoriel Mozilla](#)

# ÉCRIVEZ DU XML “SIMPLE”

- Ne pas se préoccuper des indentations
- En cas de doute, inspectez votre sortie avec un navigateur Web

# ATTRIBUTS VIEWBOX, WIDTH ET HEIGHT DE SVG

Si vos dessins ont des problèmes de mise à l'échelle automatique dans Repl.it :

- Vérifier en ouvrant le fichier dans le navigateur
- Ajuster ViewBox, width et height (attributs du noeud svg) pour contrôler finement les échelles et unités (voir [cette page du tutoriel](#))