

Objectif : Manipuler et comprendre le format SVG sur plusieurs exemples.

Introduction : le document minimal

Un fichier svg doit contenir au minimum les éléments suivants :

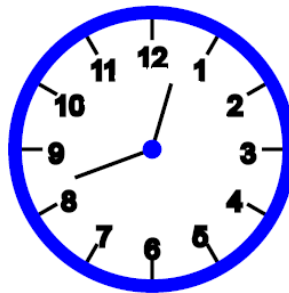
```
<?xml version="1.0" standalone="no"?>
<!--définition du type de document pour la conformité aux spécifications -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
"http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<!--Référencement de l'espace de nommage => les noms des balises utilisables-->
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" >
<!-- ici on met le code svg -->
</svg>
```

Remarque : <!-- est un commentaire pas obligatoire-->

Exercice 1 : Horloge

Réalisez le codage SVG de l'image suivante :

En faisant tourner les aiguilles à vitesse réelle.
(i.e. faites une véritable horloge)



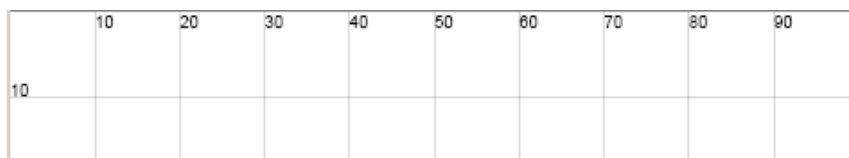
Exercice 2 : Manipuler les tracés et les animations le long des tracés

Préambule construire le fichier grille.svg :

Objectif : Définir son système de coordonnées

- o Dans la **balise « svg »**
 1. Définissez une « viewport » de 10cm par 10cm.
 2. A l'aide de l'attribut « viewBox » redéfinissez l'espace de coordonnées de 0 à 100 unités dans les 2 directions x et y.
code => viewBox="0 0 100 100"
Cela impose que la zone définie par la viewBox rentre dans la viewport de 10cm x 10cm. Ce qui revient à redéfinir l'unité comme étant le millimètre.

- o A l'aide de la **primitive <path>** tracez une grille de 10 x 10 unités
- o **Numérotez** les lignes et les colonnes comme suit :



...

- o Construisez cette grille dans un **groupe** et donnez l'**identifiant « grille »** à ce groupe pour pouvoir le réutiliser par la suite

Construire le fichier **Ref_grille.svg** :

Objectif : Réutiliser des éléments définis dans des fichiers SVG à l'aide de la balise <use> et des référencements xlink.

Dans le **fichier ref_grille.svg** :

Pour utiliser les référencements suivant la **norme xlink**, insérez le référencement à son espace de nommage comme suit :

```
<svg xmlns=http://www.w3.org/2000/svg xmlns:xlink=http://www.w3.org/1999/xlink >  
(Rq il faut conserver celui de SVG)
```

Complétez les **attributs de <svg>** pour définir le même système de coordonnées que pour le fichier précédant.

A l'aide de la **primitive <image>**, réutilisez la grille définie dans le **fichier grille.svg** par :

```
< image xlink:href="grille.svg" x="0" y="0" height="100" width="100" />
```

Afficher un **cercle rouge** de centre (20, 30) et de rayon 5

Construire le fichier **Path_line.svg** :

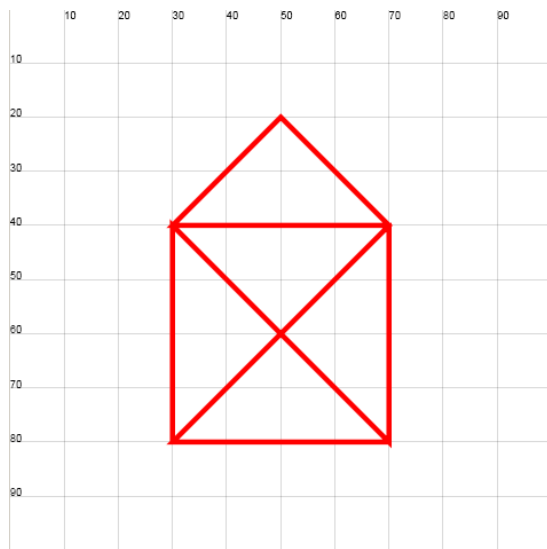
Objectif : Tracer un chemin composé de lignes et utiliser les chemins pour animer un élément.

Dans le **fichier Path_line.svg** :

Définissez toujours le même système de coordonnées.

Réutilisez la grille

Tracez la figure suivante en rouge (sans lever le crayon)



A l'aide de l'élément **animateMotion**, déplacez un cercle bleu de rayon 3 le long du tracé (utilisez le **sous élément <mpath>** pour référencer le chemin déjà défini ci-dessus).

1. Durée de l'animation 10s
2. Répétez l'animation (indéfiniment)

Construire le fichier **Path_arc.svg** :

Objectif : Tracer un chemin composé d'arcs elliptiques

Un arc d'ellipse se dessine toujours à l'aide de la **balise <path>** en utilisant dans la description du chemin la **lettre 'a' (resp. 'A')** pour la description en **relatif (en absolu)**.

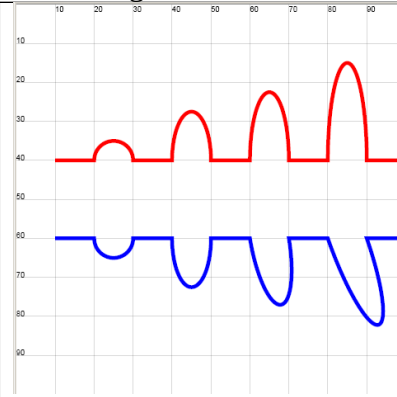
'a' ou 'A' est suivi de :

- * **rx, ry** : pour les rayons en x et en y de l'arc elliptique
- * **rotation-axe-x** : pour la valeur de rotation de l'axe x de l'ellipse
- * **drapeau-arc-large, drapeau-balayage** : pour déterminer de quel « côté » est tracé l'arc
- * **x, y** : pour le point final de l'arc (le point de départ étant le point courant du tracé => on commencera par un « **moveto** » pour le spécifier s'il n'y a pas de point courant)

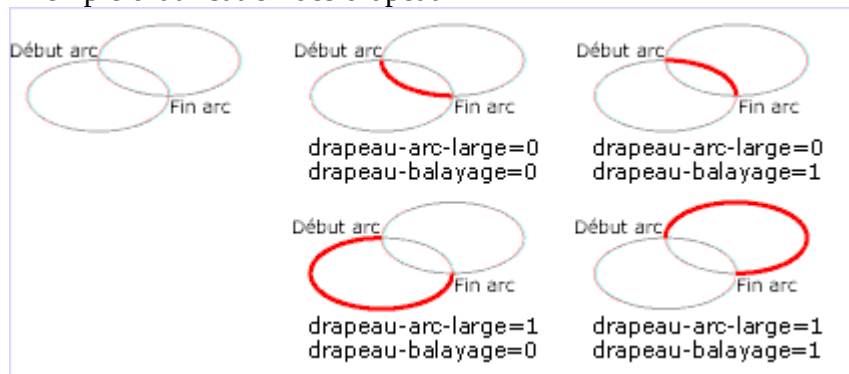
Exemple : Rq on peut combiner comme ici les tracés de lignes et d'arcs

```
<path d="M 10,40 | 10,0
a2,2 0 0,1 10,0 | 10,0
a2,5 0 0,1 10,0 | 10,0
a2,7 0 0,1 10,0 | 10,0
a2,10 0 0,1 10,0 | 10,0"
fill="none" stroke="red" stroke-width="1" /
>

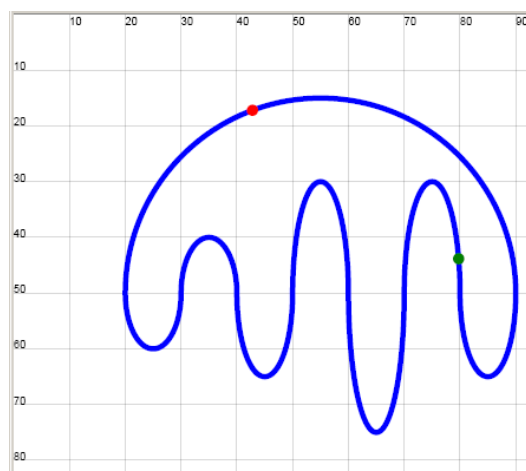
<path d="M 10,60 | 10,0
a2,2 0 0,0 10,0 | 10,0
a2,5 0 0,0 10,0 | 10,0
a2,7 -10 0,0 10,0 | 10,0
a2,10 -20 0,0 10,0 | 10,0"
fill="none" stroke="blue" stroke-
width="1" />
```



Exemple d'utilisation des drapeaux



Dans le **fichier Path_arc.svg**, définissez-le tracé suivant et faites le parcourir par un **disque rouge** (en 6s) et un **disque vert** (en 9s) :



Exercice 3 : Construire le fichier *Path_arc_interact.svg*

Objectif : Rendre une animation interactive

Les paramètres des animations (begin, end, ...) peuvent être liés à des événements de type focus, souris, animation,...

Exemple :

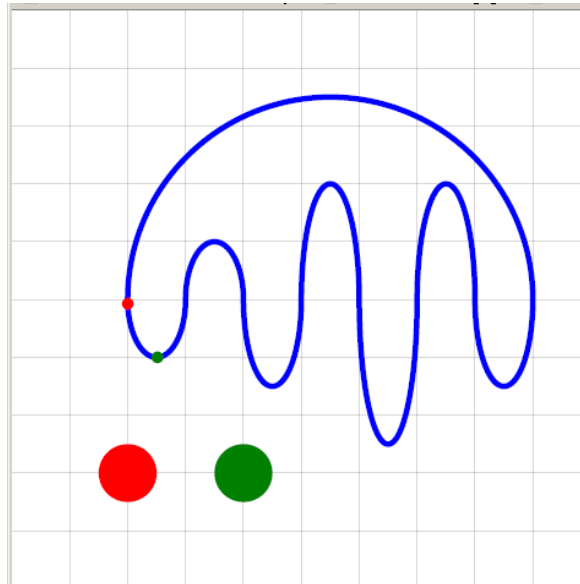
begin= « truc.click » spécifie que l'animation commence si on clique sur l'élément nommé par « truc » (id= « truc »)

end= « truc.mouseout » spécifie que l'animation s'arrête si le pointeur de la souris n'est plus sur l'élément « truc »

Les événements sont répertoriés dans la documentation.

Application :

A partir du fichier *Path_arc*, ajoutez 2 cercles de rayon 5 (un rouge nommé « rouge » et un vert nommé vert) comme suit :



- o Déclenchez le départ du point rouge le long du tracé si le pointeur de souris est au dessus de « rouge » (événement mouseover)
- o Déclenchez l'arrêt du point rouge le long du tracé si le pointeur de souris est au dessus de « rouge » (événement mouseout)
- o Déclenchez le départ du point vert le long du tracé si le pointeur de souris est au dessus de « vert »
- o Déclenchez l'arrêt du point vert le long du tracé si le pointeur de souris est au dessus de « vert »

Exercice 4 : Marche autour de la Terre

Construire le fichier *marcheur.svg*

Construire un marcheur ayant l'allure suivante et marchant sur place en oscillant des jambes (évidemment) et des bras.



Indication : utilisez `animateTransform`

Construire le fichier marche.svg

Utilisez le marcheur précédent et faites-le avancer (le faire marcher quoi !)

Indication : remettez une couche d'animateTransform

Construire le fichier marche_en_rond.svg

Construire un chemin représentant un cercle et faites marcher le marcheur autour de ce cercle.

Indication : Pour le chemin, utilisez la définition des chemins à l'aide des arcs de cercles.

Construire le fichier around_the_world.svg

Importez le fichier terre.png (définition de 300x300)

Faites marcher le marcheur autour de la terre.

Mettez une couleur de fond bleu nuit

Agrémentez le tout de quelques étoiles de différentes tailles

Et vous obtiendrez :



L'image de la Terre se trouve dans le même répertoire que ce sujet (dans Plubel).

Exercice 5 : Mille pattes qui avance

On se propose de réaliser en SVG l'animation d'un mille pattes dont une copie d'écran est donnée ci-dessous.



Le mille pattes est composé de **13 éléments**. Chacun de ces éléments est composé d'un corps (cercle de rayon 20 unités) et d'une patte (de longueur 30 unités) fixée au centre du corps et tournant autour de ce même centre.

Deux éléments consécutifs sont distants de 30 unités et les pattes ont un déphasage de 30 degrés pour permettre au mille pattes de se mouvoir.

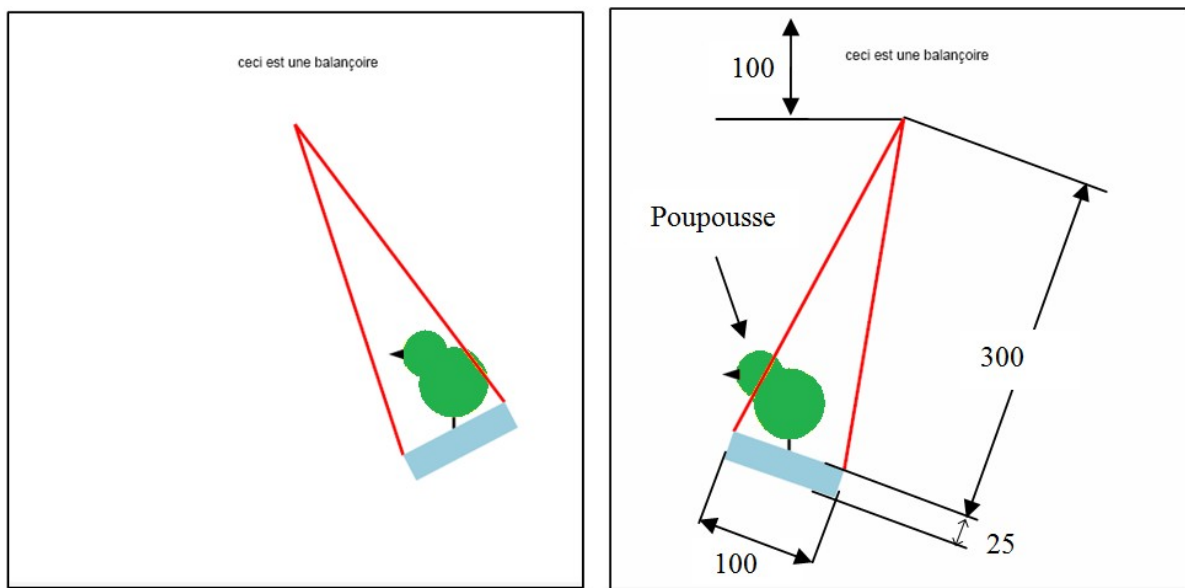
Dans une région de 500 (en y) par 1000 (en x) unités, le mille pattes doit se déplacer de droite à gauche en 18 secondes, ce qui correspond à une vitesse de rotation des pattes de 360° en 4s.

Réalisez le code SVG le plus compact possible de cette animation (réutilisation de définitions, mise en facteur des transformations,...). Expliquez votre construction.

Exercice 6 : Poussin qui se balance

On se propose de réaliser une animation représentant « Poussin » le poussin faisant de la balançoire. Deux étapes sont représentées ci-dessous dont l'une précise un certain nombre de caractéristiques sur les dimensions des composants.

La balançoire doit **osciller de +/- 30°**. Poussin est situé sur la balançoire et suit le mouvement de celle-ci. Mais comme il n'apprécie pas les mouvements d'inclinaison, il les contre de manière à avoir toujours une orientation verticale (voir ci-dessous).



L'animation doit se réaliser dans une **fenêtre de 500 par 500 pixels**.

Le balancement doit démarrer **au clic sur le texte « ceci est une balançoire »**.

L'aller retour doit se faire en **2s**.

L'animation doit se dérouler **indéfiniment**. Pour cela, utilisez le fait qu'il est possible de mettre des identifiants dans les balises d'animation et d'utiliser les événements associés.

Hiérarchisez votre construction au maximum, de manière à utiliser le moins possible de transformations.

Utiliser le code SVG suivant pour dessiner Poussin :

```
<g id="poussin" transform="translate(0,-40)"> <!-- le bas de sa patte est en (0,0) -->
  <circle cx="-25" cy="-25" r="20" fill="yellow" /> <!-- tête -->
  <circle r="30" fill="yellow" /> <!-- corps -->
  <line x1="0" y1="30" x2="0" y2="40" stroke="black" stroke-width="3" /> <!-- patte -->
  <path transform="translate(-42,-20)" d="M 0 0 v -10 l -15 5 z" fill="black" /> <!-- bec -->
</g>
```