

Objectifs de ce TP :

- consolider les notions acquises sur les langages HTML, CSS et javascript
- réaliser des dessins vectoriels et des animations

Conditions de travail :

- travail en binômes (vos deux noms doivent apparaître en clair)
- travail en local (sur votre disque dur)
- utiliser l'éditeur NOTEPAD++
- durée prévue : 2h
- uploader votre travail dans vos travaux et insérer un lien dans votre page perso

1. La balise SVG

Il existe plusieurs méthodes pour tracer des graphismes dans une page HTML. L'une d'entre elles consiste à utiliser la balise svg (Scalable Vector Graphics ou image vectorielle dimensionnable).

La balise svg est une balise non auto-fermante destinée à **contenir** des balises qui correspondent à des éléments graphiques.

Les éléments graphiques sont des balises auto-fermantes (sauf la balise text).

Les éléments graphiques sont affichés dans l'ordre de lecture, ils peuvent se chevaucher.

Comme pour n'importe quelle balise, les couleurs opaques (formats rgb, hexadécimales et hsl) et transparentes (format rgba ou hsla) sont autorisées. On peut également modifier la transparence de toute balise avec la propriété d'opacité.

2. Les dimensions de la balise svg

Comme pour la balise img, les dimensions de la balise svg sont définies par les attributs **width** et **height** dont les valeurs doivent être données en pixel et sans unité. Attention : si votre balise est mal dimensionnée certains tracés qui débordent seront tronqués.

L'attribut **viewBox**, facultatif, permet de définir la zone visible du dessin.

Exemple : dans cet exemple, les éléments graphiques sont affichés dans un espace de dessin de 1000px par 1000px, tout ce qui dépasse est tronqué. La balise svg s'affichera sur la page avec une taille de 300px par 300px.

```
<svg width=300 height=300 viewBox="0 0 1000 1000" >
    éléments graphiques...
</svg>
```

3. Les éléments graphiques de la balise svg

Les éléments graphiques usuels sont :

élément graphique	le cercle	l'ellipse	le rectangle	le trait	le tracé ouvert	le tracé fermé	le tracé complexe	le texte
balise	circle	ellipse	rect	line	polyline	polyline	path	text
attributs	cx cy r	cx cy rx ry	x y width height (rx ry facultatifs pour des coins arrondis)	x1 y1 x2 y2	points	points	d	x y
auto-fermante	oui	oui	oui	oui	oui	oui	oui	non

4. Bordure et couleur de fond

La couleur de fond d'un élément graphique est renseignée par les attributs **fill**, tandis que la bordure l'est par les attributs **stroke** et **stroke-width**.

Un élément graphique svg peut posséder un attribut d'opacité comme ceci par exemple : **opacity=0.5**

Testez cet exemple :

```
<svg width=300 height=300 viewBox="0 0 200 200" style="background-color:rgb(200,200,200)" >
  <circle cx=50 cy=50 r=40 fill=yellow stroke=blue stroke-width=5 />
  <rect x=60 y=60 width=40 height=20 fill=cyan stroke=brown stroke-width=4 opacity=0.5 />
</svg>
```

Pour les exemples suivant seuls les éléments graphiques sont précisés...

5. Les tracés

La valeur de l'attribut **points** est la liste des coordonnées (abscisse , ordonnée) des sommets des segments successifs qui constituent le tracé.

Un tracé peut être ouvert ou fermé.

Essayer ces deux exemples et noter leur différence:

```
<polyline points="100,100 300,100 200,200" fill=red stroke=green stroke-width=5 />
<polygon points="100,100 300,100 200,200" fill=red stroke=green stroke-width=5 />
```

6. Grouper des éléments

Si plusieurs éléments graphiques doivent avoir des caractéristiques similaires, on peut les regrouper dans une balise g.

Tester cet exemple :

```
<g fill=orange stroke=gray stroke-width=5 >
  <circle cx=50 cy=50 r=40 />
  <polygon points="100,100 300,100 200,200" />
</g >
```

7. Les transformations

L'attribut **transform** permet d'effectuer des translations, rotations, redimensionnements ... sur n'importe quelle balise.

Exemples :

```
transform="scale(3)" agrandi d'un facteur 3
transform="rotate(60)" effectue une rotation de 60°
transform="rotate(60 20,30)" effectue une rotation de 60° ayant pour centre de rotation le point de coordonnées (20 , 30)
```

remarque : la rotation se fait dans le sens horaire, l'angle nul correspondant à une aiguille des minutes située à "et quart".

```
transform="translate(24,35)" effectue une translation de vecteur (24px,35px)
```

Tester cet exemple :

```
<rect x=60 y=60 width=40 height=20 fill=cyan stroke=brown stroke-width=4 transform=rotate(10) />
```

8. Interactions avec javascript

On peut modifier n'importe quel attribut avec l'instruction `setAttribute` du langage javascript. Pour cela, il est pratique d'identifier la balise cible de la modification.

Tester cet exemple :

```
<rect x=60 y=60 width=40 height=20 fill=cyan stroke=brown stroke-width=4 id=un_rectangle />
```

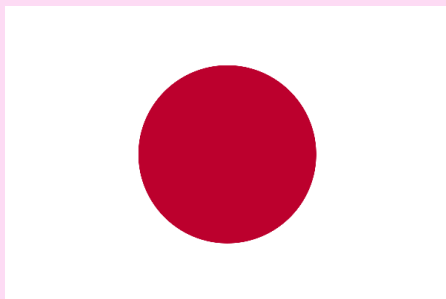
```
code javascript : un_rectangle.setAttribute("transform","rotate(-20)");
```

9. Exercices

*Réalisez chaque exercice dans une page HTML avec un titre.
Pour chaque exercice, montrer votre rendu à l'écran à votre professeur.
Placez des liens vers les exercices dans votre page perso.*

Exercice n°1

Sur un fond gris, dessiner les drapeaux du Japon, de la Belgique et de l'Islande :



Exercice n°2

Sur une feuille de papier, tracer un repère gradué (x,y) , reproduire le polygone représenté ci-dessous et nommez les sommets de ce polygone.



Mesurer les coordonnées de ces points dans le repère donné et inscrivez ces valeurs en mm dans le tableau ci-dessous.

 : remarquer qu'en informatique l'axe des ordonnées est orienté vers le bas !

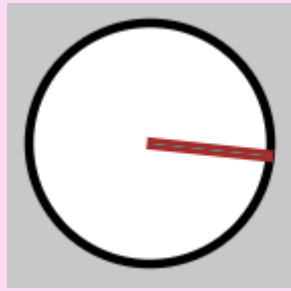
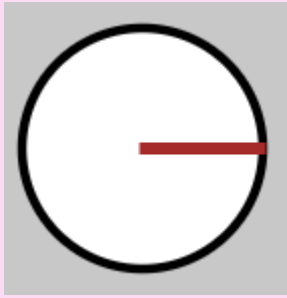
point						
abscisse (mm)						
ordonnée (mm)						
abscisse (px)						
ordonnée (px)						

Construire ce tableau avec un tableur, y inscrire les coordonnées en mm puis calculer automatiquement les coordonnées en pixel (px) en tenant compte de l'échelle 1 mm = 3 px.

Ecrire la balise polygon correspondant à ce dessin.

Exercice n°3

Réaliser un dessin approchant celui-ci (à gauche) avec une balise circle et une balise rect :



Lorsque l'on capture un évènement (par exemple un clic sur la balise svg), effectuer une rotation de 6° dans le sens horaire. Utiliser une variable globale pour mémoriser la valeur de l'angle et pour permettre une rotation supplémentaire de 6° à chaque clic.

Expliquer pourquoi ce choix d'un angle de 6° .