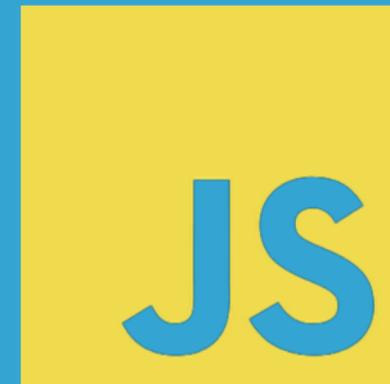
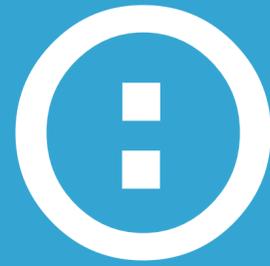


`<simplonBoulogne html5 + js />`

Dessiner avec canvas

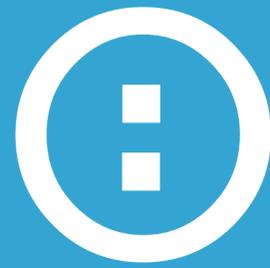




Canvas

Il s'agit d'un espace de pixels initialement transparents, armés de JavaScript pour réaliser un bon nombre de fonctions graphiques.

Permet la création de dessins 2D et 3D directement dans le navigateur, du simple tracé jusqu'à des animations.

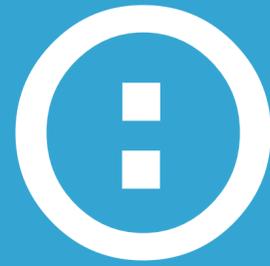


Initialisation HTML

```
<canvas id="mon_canvas" width="350"  
height="350">
```

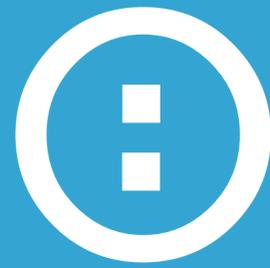
Texte alternatif pour les navigateurs
ne supportant pas Canvas.

```
</canvas>
```



Initialisation JavaScript

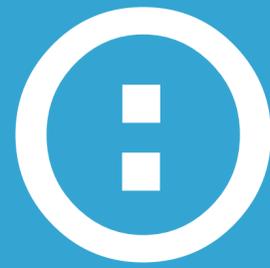
Canvas est défini dans la structure HTML mais tout le reste se passe du côté JavaScript !



Accéder à l'élément `<canvas>`

On utilise donc ce que l'on a vu pour la navigation dans le DOM...

```
getElementById();
```

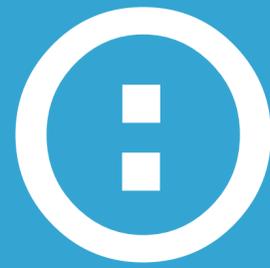


Définir le contexte canvas

Définir le contexte sert à expliciter quel contexte de dessin nous allons utiliser pour l'élément que l'on a sélectionné. Il permet également de préciser à JS quelles fonctions il sera en droit d'utiliser.

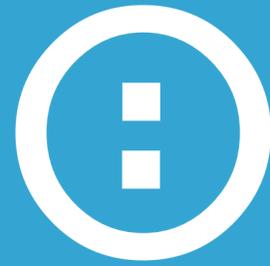
2D

3D

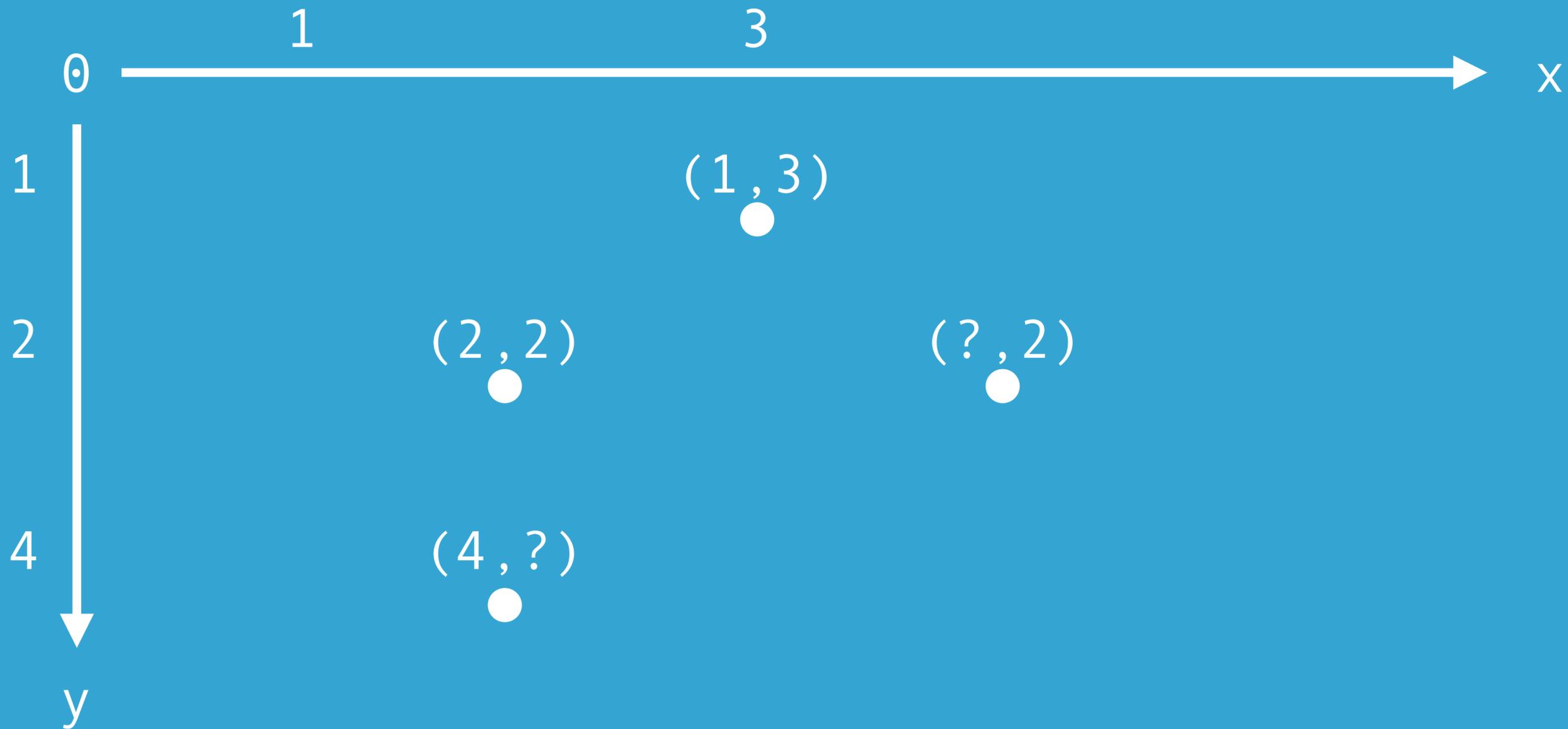


Exemple

```
<script type="text/javascript">  
  var c =  
document.getElementById("mon_canvas");  
  var ctx = c.getContext("2d");  
  // Le reste du script ici...  
</script>
```

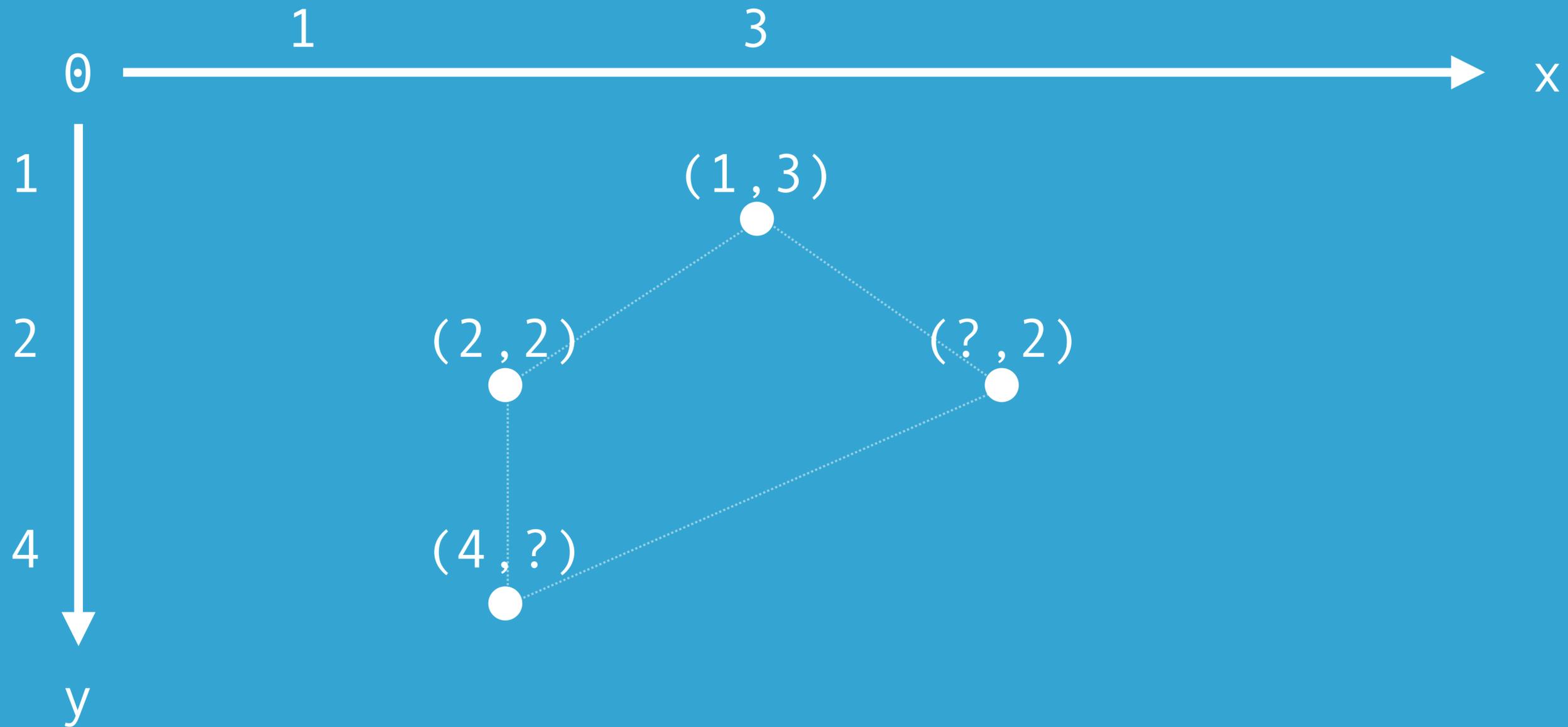


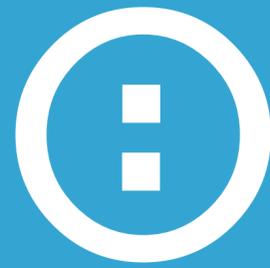
Le système de coordonnées





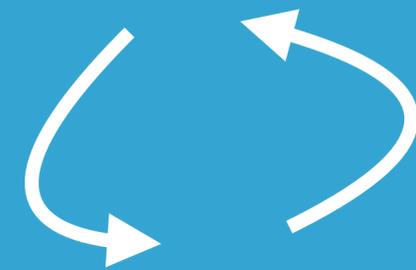
Tracé 2D

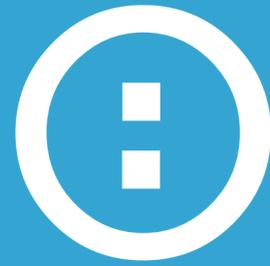




Étapes d'un tracé

- Initialisation - `beginPath();`
- Point de départ - `moveTo(x,y);`
- Point d'arrivée - `lineTo(x,y);`
- Clôture - `closePath();`
- Contour ou remplissage - `stroke();` | `fill();`

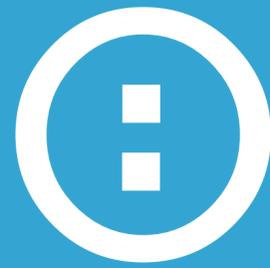




`stroke();` ou `fill();` ?

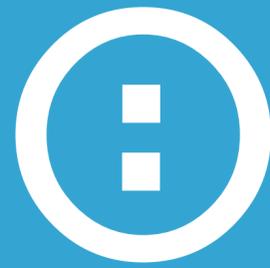
La forme n'apparaît qu'une fois que l'on a appelée l'une des deux méthodes :

- `fill()` pour remplir la forme
- `stroke()` pour le contour de la forme



Exemple de tracé

```
var ctx = c.getContext("2d");  
ctx.beginPath();  
ctx.moveTo(50, 50);  
ctx.lineTo(200, 200);  
ctx.moveTo(200, 50);  
ctx.lineTo(50, 200);  
ctx.closePath();
```

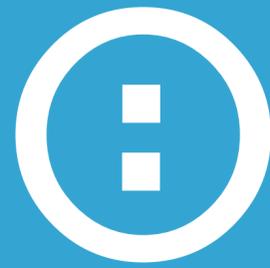


Les propriétés de style du contexte

Couleurs

Tous les codes couleurs utilisés dans un style CSS classique sont reconnus ! Attention pas de parenthèses, ici on applique un style ;)

```
ctx.fillStyle = "red";  
ctx.strokeStyle = "#ecf0f1";
```



Les propriétés de style du contexte

Style de lignes

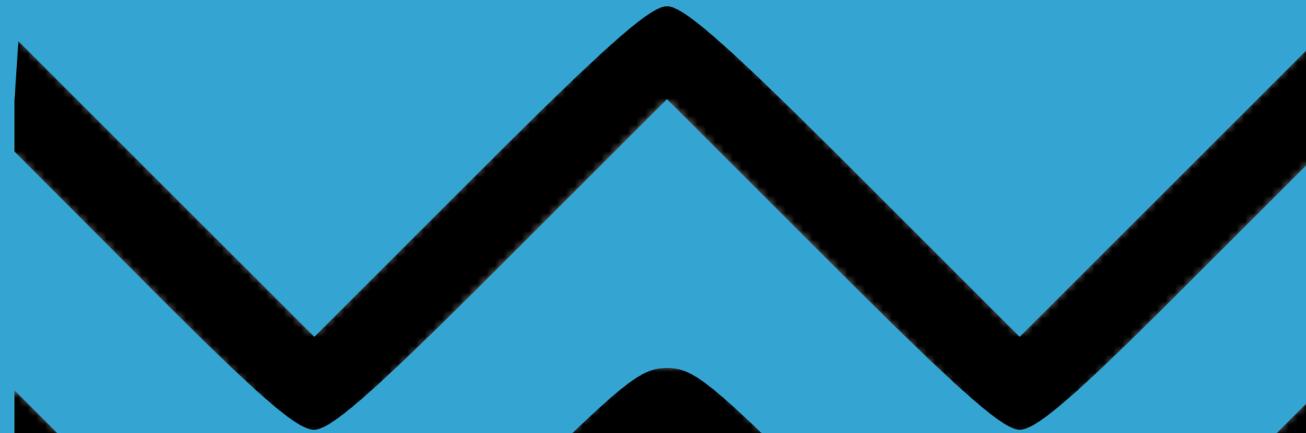
Vous pouvez modifier la largeur des lignes et éditer les « fins de lignes ».

```
ctx.lineJoin = "bevel";  
ctx.lineCap = "round";
```

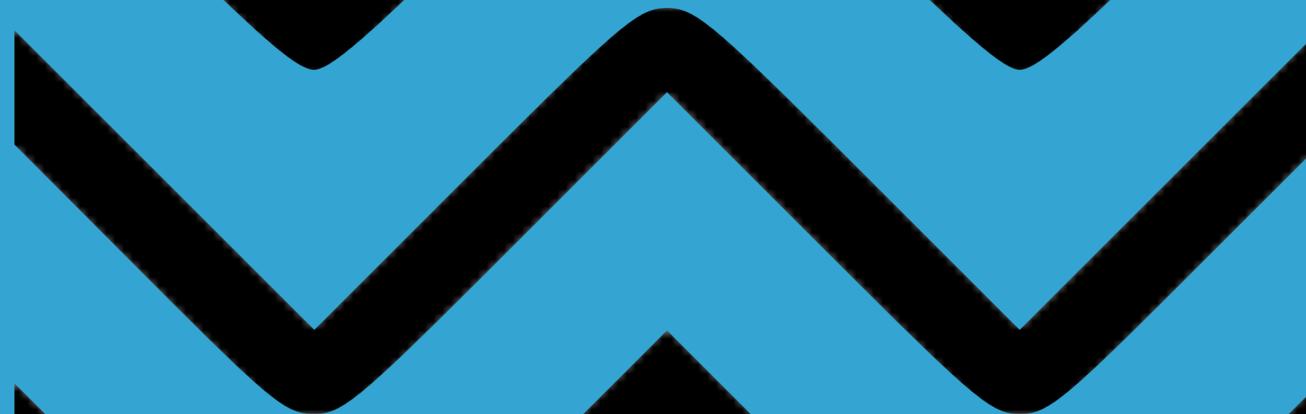


lineJoin

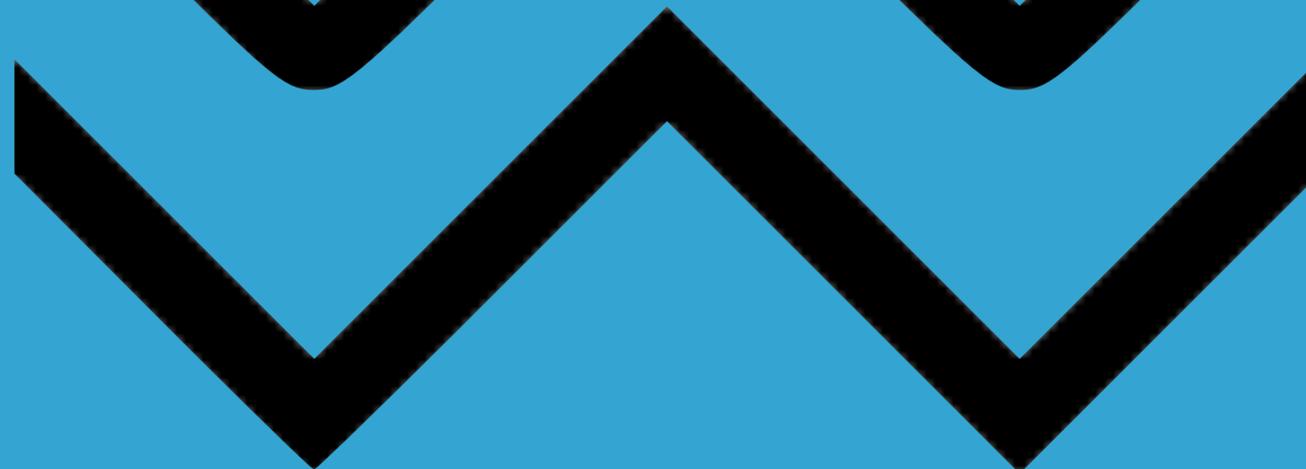
round



bevel



miter



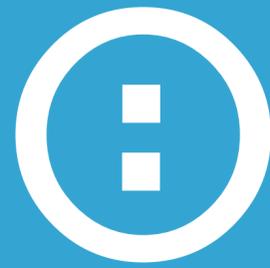
lineCap



butt

round

square

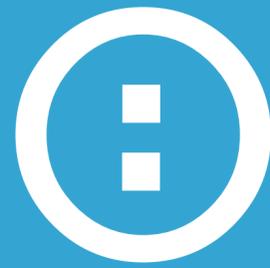


Les formes

Rectangles et carrés

Vous pouvez tracer directement un rectangle en indiquant ses coordonnées de départ et sa longueur/hauteur.

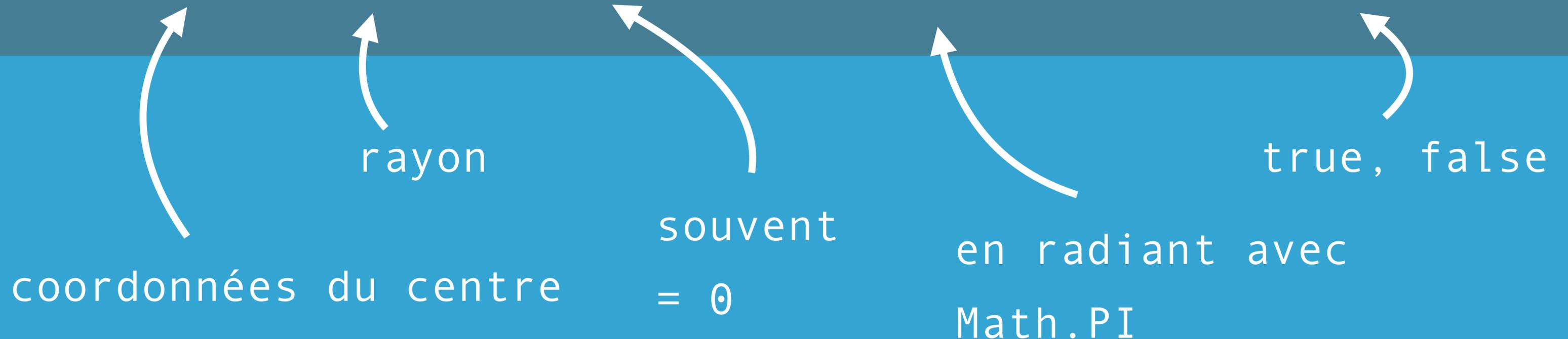
```
fillRect(x, y, height, width);
```

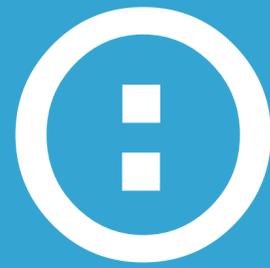


Les formes

Cercles et arc de cercles

```
arc( x, y, rayon, startAngle, endAngle, sensAntiHoraire )
```

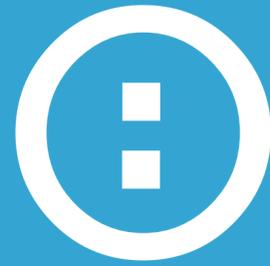




Les formes

Cercles - exemples

```
ctx.arc(150, 150, 60, 0, Math.PI, false);  
ctx.arc(180, 130, 15, 0, Math.PI*2, false);
```



Les formes

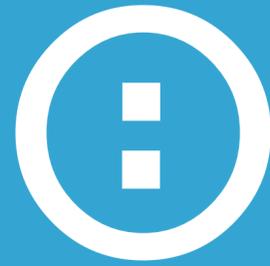
Courbes de Bézier

```
bezierCurveTo( cp1x, cp1y, cp2x, cp2y, destx, desty )
```

coordonnées départ

point de
passage

coordonnées arrivée



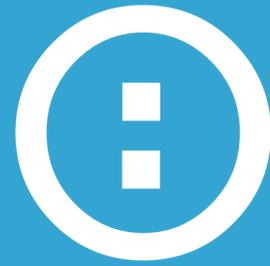
Les formes

Courbes quadratiques

```
quadraticCurveTo( cp1x, cp1y, destx, desty )
```

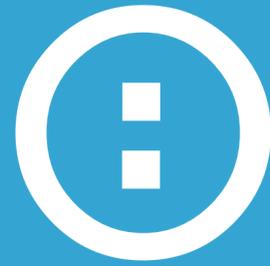
coordonnées départ

coordonnées arrivée



Challenge

Créer le logo de Simplon Boulogne en Canvas :D



Some resources

- <http://caniuse.com/#feat=canvas>
- <https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D>