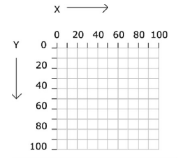


L'essentiel de Canvas

Canvas est un terme utilisé dans de nombreux langages. Il désigne l'objet d'affichage, l'écran en général. En HTML, les balises Canvas permettent de réaliser des dessins ou des animations par l'intermédiaire de script.

Dessiner avec canvas se fait par le biais de coordonnées où l'origine du repère est située au coin supérieur gauche de la fenêtre.



Exemple de code à insérer entre les balises body

Voici un exemple de code à insérer entre les balises body

```
<canvas id="mon_dessin" width="150" height="200">
  <p>Désolé, votre navigateur ne supporte pas Canvas.</p>
</canvas>

<script>

  window.onload = function() {
    var canvas = document.getElementById("mon_dessin");
    var ctx = canvas.getContext('2d');
    ctx.beginPath(); // La méthode beginPath() permet d'initialiser le tracé.
    ctx.moveTo(50,100); // Le tracé part du point de coordonnées 50;100.
    ctx.lineTo(200,200); // On trace un segment jusqu'au point de coordonnées 200,200.
    ctx.stroke(); // affiche les contours des divers tracés.
    ctx.closePath(); // permet de fermer le dessin.

  };
</script>
```

Les attributs width et height permettent de définir les dimensions du dessin que l'on souhaite réaliser.

La méthode getElementById permet de récupérer l'élément canvas qui est nommé par son id à savoir "mon_dessin".

La méthode getContext permet de définir dans quel contexte de dessin nous allons travailler.

Quelques éléments de canvas

- Figures géométriques
 - `ctx.rect(x,y,largeur,hauteur)`; représente un rectangle à partir du point de coordonnées x, y dont les dimensions sont précisées par les paramètres largeur et hauteur.
 - `arc(x, y, rayon, startAngle, endAngle, sensAntiHoraire)` permet de construire un arc de cercle. x, y et rayon désignent respectivement les coordonnées du centre et le rayon de cet arc. startAngle et endAngle sont des angles qui sont mesurés en radian. Enfin, sensAntiHoraire est un boolean (true ou false) qui permet de réaliser cet arc en tournant dans le sens des aiguilles d'une montre ou non.
- Couleur et épaisseur
 - `ctx.fillStyle="green"`; permet de définir la couleur de remplissage.
 - `ctx.strokeStyle="green"`; permet de définir la couleur de contour.
 - `ctx.lineWidth=5`; permet de définir l'épaisseur du trait du contour.
- Texte
 - `ctx.fillText("Bonjour",x,y)`; affiche le mot "Bonjour" au point de coordonnées x,y.
 - `ctx.strokeText("Bonjour",x,y)`; affiche également le mot "Bonjour" au point de coordonnées x, y mais par contre seul le contour des lettres est représenté.
 - `ctx.font="30pt verdana"`; permet de définir la taille et le type de la police.
- Transformations
 - `ctx.scale(x,y)`; permet de changer l'échelle du contexte.
Une valeur de x inférieure à 1 permet de réduire horizontalement alors qu'une valeur supérieure à 1 permet d'agrandir.
Il en est de même pour la valeur de y mais verticalement.
Il est possible d'indiquer des valeurs négatives, dans ce cas il y a un effet miroir en plus d'un éventuel agrandissement ou d'une réduction.
 - `ctx.rotate(angle)`; permet de faire une rotation d'un angle donné en radian. Le centre de rotation est l'origine du dessin.
 - `ctx.translate(x,y)`; permet de déplacer la figure de x pixels horizontalement et de y pixels verticalement.