**Javascript : aide API Canvas 2D**

|  |
| --- |
| **API Canvas 2D (ou CanvasRenderingContext2D) :** interface de programmation en JavaScript pour dessiner des graphiques et des animations en 2D sur une page web.  -Elle permet de dessiner des formes, des images, du texte, des gradients et des motifs en utilisant des méthodes comme **fillRect()**, **drawImage()**, **fillText()**, etc.  -Elle offre également la possibilité de modifier des propriétés telles que la couleur, l'épaisseur du trait, le style de ligne, la transparence et bien plus encore. L'API Canvas 2D est largement utilisée pour créer des jeux, des visualisations de données et des applications interactives sur le web. |

|  |
| --- |
| **DOM :** Le **Document Object Model** ou **DOM** (pour modèle objet de document) est une interface de programmation pour les documents HTML, XML et SVG. Il fournit une représentation structurée du document sous forme d'un arbre et définit la façon dont la structure peut être manipulée par les programmes, en termes de style et de contenu. Le DOM représente le document comme un ensemble de nœuds et d'objets possédant des propriétés et des méthodes. Les nœuds peuvent également avoir des gestionnaires d'événements qui se déclenchent lorsqu'un événement se produit. Cela permet de manipuler des pages web grâce à des scripts et/ou des langages de programmation. Les nœuds peuvent être associés à des gestionnaires d'événements. Une fois qu'un événement est déclenché, les gestionnaires d'événements sont exécutés. |

|  |  |
| --- | --- |
| **Canvas :** L'élément <canvas> a seulement deux attributs : [width](https://developer.mozilla.org/fr/docs/Web/HTML/Element/canvas" \l "attr-width) et [height](https://developer.mozilla.org/fr/docs/Web/HTML/Element/canvas" \l "attr-height) (« largeur » et « hauteur »). Ces deux attributs sont optionnels et peuvent aussi être fixés à travers le [DOM](https://developer.mozilla.org/fr/docs/Web/API/Document_Object_Model). Quand les attributs **width** et **height** ne sont pas spécifiés, le canvas sera initialement large de **300 pixels** et haut de **150 pixels**. Les dimensions du canvas peuvent être modifiés par du [CSS](https://developer.mozilla.org/fr/docs/Web/CSS), mais l'image sera dessinée selon les valeurs **width** et **height** du canvas et ensuite étirée pour afficher dans l'espace donné par le CSS.   |  | | --- | | Exemple :  <canvas id="stockGraph" width="150" height="150">  current stock price: $3.15 + 0.15  </canvas>  <canvas id="clock" width="150" height="150">  <img src="images/clock.png" width="150" height="150" alt="une horloge"/>  </canvas> | |

**CanvasRenderingContext2D.beginPath() :**  méthode de l'API Canvas 2D permet de commencer un nouveau chemin en vidant la liste des sous-chemins. Appelez cette méthode quand vous voulez créer un nouveau chemin.

**CanvasRenderingContext2D.arc() :**  méthode  de l'API Canvas 2D permet d'ajouter un arc de cercle au tracé, en le centrant aux positions \_(x, y)\* et avec un rayon r qui démarre à angleDépart et qui finit à angleFin, dans la direction de sensAntiHoraire (par défaut en sens horaire).

**CanvasRenderingContext2D.moveTo() :** méthode de l'API Canvas 2D déplace le point de départ d'un nouveau sous-chemin vers les coordonnées (x, y).

**CanvasRenderingContext2D.stroke() :** méthode de l'API Canvas 2D qui dessine le chemin actuel ou donné avec le style de trait actuel utilisant la règle d'enroulement non nulle.

**CanvasRenderingContext2D.save()** : méthode de l'API Canvas 2D API qui enregistre l'état complet du canvas en plaçant l'état courant dans une stack.

**CanvasRenderingContext2D.fillRect() :** méthode de l'API 2D des Canvas dessine un rectangle plein aux coordonnées (x, y), aux dimensions déterminées par largeur et hauteur et au style déterminé par l'attribut fillStyle.

**CanvasRenderingContext2D.fillStyle**: propriété de l'API Canvas 2D qui spécifie la couleur ou style à utiliser à l'intérieur des formes. La valeur par défaut est #000 (black)

**arc(x,y,r,startangle,endangle) :** créer un arc/une courbe. Pour créer un cercle avec arc() : Définissez l'angle de départ sur 0 et l'angle de fin sur 2\*Math.PI. Les paramètres x et y définissent les coordonnées x et y du centre du cercle. Le paramètre r définit le rayon du cercle.

**createLinearGradient( *x,y,x1,y1* )** - crée un dégradé linéaire

**createRadialGradient( *x,y,r,x1,y1,r1* )** - crée un dégradé radial/circulaire

.

var canvas = document.getElementById("myCanvas"); :

var ctx = canvas.getContext("2d");

|  |
| --- |
| Tracer une ligne  Pour tracer une ligne droite sur un canevas, utilisez les méthodes suivantes :   * moveTo( *x,y* ) - définit le point de départ de la ligne * lineTo( *x,y* ) - définit le point final de la ligne   Pour tracer réellement la ligne, vous devez utiliser l'une des méthodes "ink", comme stroke(). |

|  |
| --- |
|  |

Pour dessiner du texte sur un canevas, les propriétés et méthodes les plus importantes sont :

* font - définit les propriétés de la police pour le texte
* fillText( *text,x,y* ) - dessine du texte "rempli" sur le canevas
* strokeText( *text,x,y* ) - dessine du texte sur le canevas (pas de remplissage)