# Ajouter du texte ou une image dans un canevas en JavaScript

L’API Canvas nous offre certaines propriétés et méthodes nous permettant d’insérer du texte ou des images directement dans un canevas

## Dessiner du texte dans un canevas

Deux méthode s’offre pour dessiner du textes dans un canvas, soit nous voulons un texte creux, alors nous utiliserons la méthode strokeText(), sinon un texte plein, nous utiliserons dans ce cas ci fillText() de l’interface CanvasRenderingContext2D.

Ces méthodes prennent 3 arguments :

Un texte à insérer

Une paire de coordonnées indiquant la position où le texte dans être insérer dans le canvas.

Cette paire de coordonnées représente l’écart entre du début du texte par rapport aux bords gauche et supérieur du canvas.

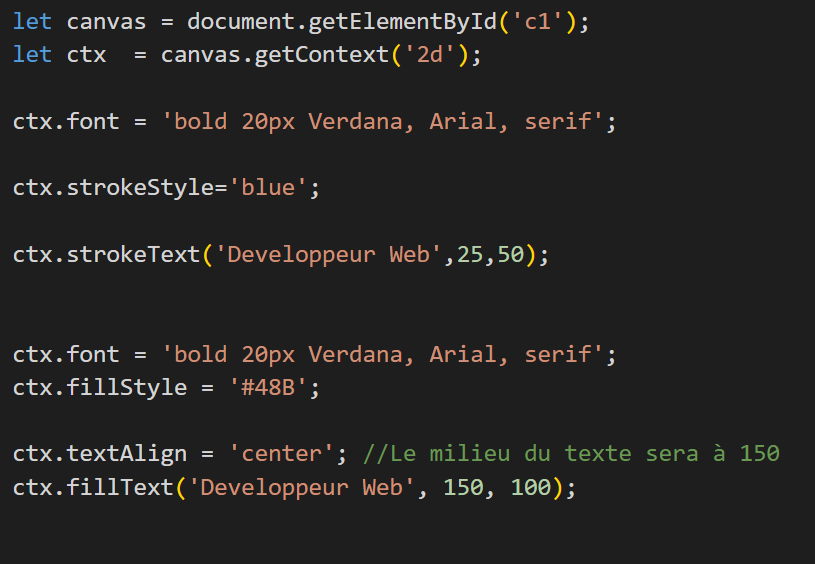
Pour styliser notre texte et son affichage, nous allons pouvoir utiliser les propriétés font, textAlign, textBaseline et direction.

La propriété font est la plus utilisée. Elle utilise la même syntaxe que la propriété raccourcie CSS font, ce qui signifie qu’on va pouvoir lui passer la taille, police, épaisseur, etc. de notre texte en valeur.

La propriété textAlign gère l’alignement du texte par rapport au point de départ. Les valeurs possibles sont start, end, left, right et center.

La propriété textBaseline permet de définir l’alignement de la ligne de base du texte. Les valeurs possibles sont top, hanging, middle, alphabetic, ideographic et bottom.

La propriété direction permet de définir la direction du texte. Les valeurs possibles sont ltr (gauche à droite), rtl (droite à gauche) et inherit.





## Insérer une image dans un canevas

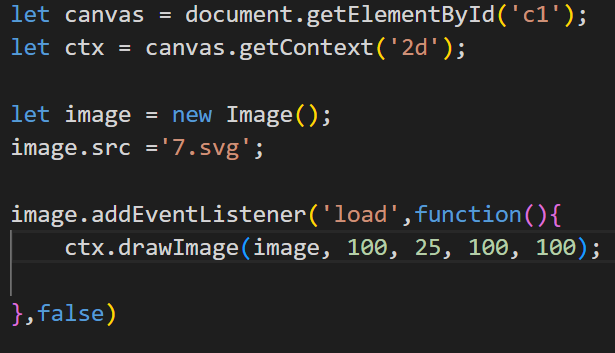
La manière pour insérer une image dans un canvas est relativement similaire a celle pour inserer un motif.

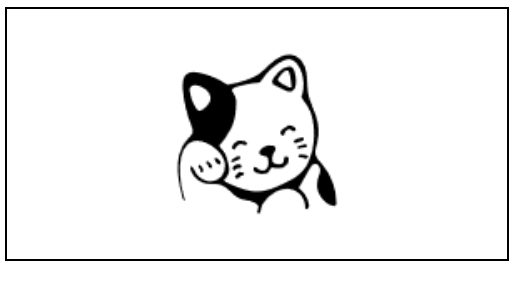
Pour commercer nous créons un nouvelle objet image (HTMLImageElement) grâce au constructeur new image(), puis nous lui donnerons le chemin vers l’image grâce a la propriété src.

Il faut ensuite utiliser la méthode drawImage(), qui prend 5 arguments (3 obligatoire et 2 optionnelles) :

1. La référence à l’image ;
2. Une paire de coordonnées indiquant où l’image doit être insérée par rapport aux bords gauche et supérieur du canevas ;
3. La largeur et la hauteur de l’image qui doit être insérée (facultatifs).

Notez que la méthode drawImage() a besoin que l’image ait été complètement chargée pour fonctionner. Pour s’assurer que c’est bien le cas, on l’utilisera généralement avec un évènement de type load. Le principe ici va être d’attendre la fin du chargement de l’image pour exécuter la méthode drawImage() .





drawImage() peut également être utilisée avec 9 arguments. Dans ce cas, la méthode va nous servir à découper une partie d’une image de base puis à coller cette partie dans le canevas. Les arguments possèdent un sens différent de précédemment :

1. Le premier argument est toujours une référence à l’image ;
2. Les 2è et 3è arguments suivants servent à indiquer un point où commencer la découpe dans l’image de base ;
3. Les 4è et 5è arguments servent à indiquer une largeur et une hauteur de l’image de base qui doit être découpée ;
4. Les 6è et 7è arguments servent à indiquer le point de départ où coller la partie de l’image découpée dans le canevas ;
5. Les 8è et 9è arguments servent à indiquer la largeur et la hauteur que doit prendre l’image dans le canevas.