



INTRODUCTION À CANVAS

JAVASCRIPT - API DE DESSIN

Elisabeth Pecatte

1

LA BASE

```
<canvas id="monCanvas" width="500" height="450">  
  <p>Votre navigateur ne supporte pas les canvas</p>  
</canvas>
```

Et en JavaScript :

Fonctionne par le même principe que paintComponent de Java. On parle cependant de "context", au lieu de "graphics"

methode **getContext** pour définir le contexte de dessin (2D ou 3D) et donc comment le script va pouvoir agir, et de quelles fonctions il pourra disposer.

```
var canvas = document.getElementById(monCanvas');  
ctx = canvas.getContext('2d');
```

DESSINER UN RECTANGLE

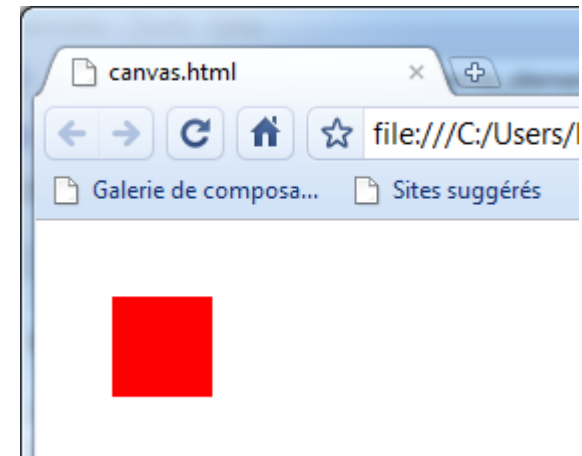
- `fillRect (x, y, largeur, hauteur)` pour un rectangle plein;
- `strokeRect (x, y, largeur, hauteur)` seulement le contour.

x et y : coordonnées.

Coord.[0, 0] = en haut et à gauche.

Exemple :

```
context.fillRect(30, 30, 50, 50);
```



LA COULEUR

- Propriété `fillStyle` pour le fond
- Propriété `strokeStyle` pour le contour.

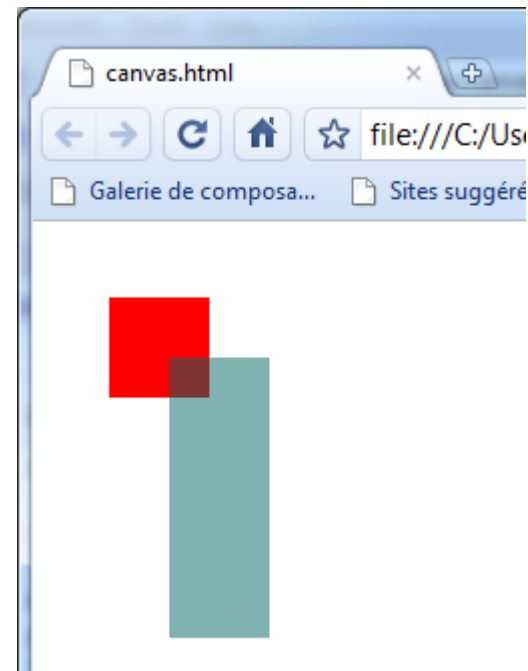
Couleur : nom, hexadecimal, `rgb()`, `rgba()` pour la transparence.

```
context.fillStyle = "rgb(255,0,0)";
```

```
context.fillRect(30, 30, 50, 50);
```

```
context.fillStyle = "rgba(0,100,100, 0.5)";
```

```
context.fillRect(60, 60, 50, 150);
```



DESSINER UNE LIGNE

○ Exemple :

```
ctx.beginPath();
```

```
ctx.lineWidth = 1;
```

```
ctx.strokeStyle = "black";
```

```
ctx.moveTo(50,25);
```

```
ctx.lineTo(300,250);
```

```
ctx.closePath();
```

```
ctx.stroke();
```

Indique au context qu'on effectue un nouveau tracé

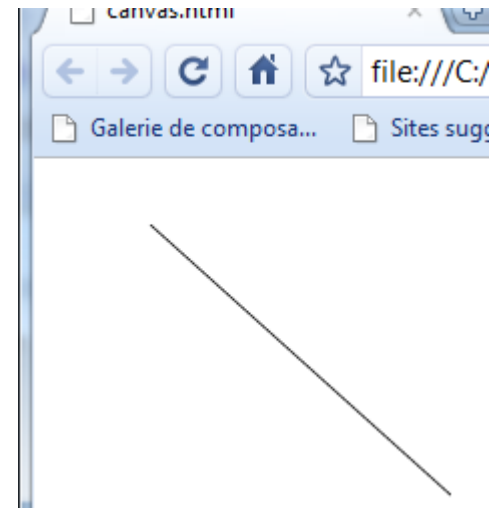
Propriété qui définit l'épaisseur du trait

Position de départ du trait, par défaut 0,0

Trace une ligne jusqu'à ces coordonnées.

Indique que le tracé est terminé

Dessine la ligne



DESSINER UN POLYGONE

○ Exemple :

```
ctx.beginPath();  
ctx.lineWidth = 2;  
ctx.strokeStyle = "rgb(200,0,0)";  
ctx.fillStyle = "rgba(10,10,100, 0.3)";  
ctx.moveTo(25,25);  
ctx.lineTo(150,105);  
ctx.lineTo(105,125);  
ctx.lineTo(25,105);  
ctx.closePath();  
ctx.fill();  
ctx.stroke();
```

Indique au contexte qu'on effectue un nouveau tracé

Propriété qui définit l'épaisseur du trait

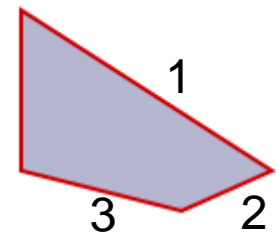
Couleur de trait et couleur de fond(en transparence)

Position de début.

Dessin de 3 lignes du polygone

Ferme le polygone

Dessin du fond puis du contour



DESSINER UN CERCLE

○ `arc(x, y, rayon, angleDebut, angleFin, anti-horaire)`

Angle en radian.

```
ctx.beginPath();  
ctx.arc(50,100,50,0, Math.PI*2, true);
```

Galerie de comp



```
ctx.beginPath();  
ctx.arc(50,30,20,0, Math.PI,true);  
ctx.fill();
```

Galerie de composa



```
ctx.beginPath();  
ctx.arc(50,30,20,0, Math.PI/2,false);  
ctx.fill();
```

Galerie de c



ÉCRIRE DU TEXTE

o `context.fillText('text', x, y);`

// Pour un effet d'ombre

`ctx.shadowOffsetX = 4;`

`ctx.shadowOffsetY = 3;`

`ctx.shadowBlur = 3;`

`ctx.shadowColor = "rgba(0, 0, 0, 0.5)";`

`ctx.textAlign="center";`

`ctx.font = "24px Times New Roman";`

`ctx.fillStyle = "blue";`

`ctx.fillText("Hellow World", 5, 30);`

Décalage horizontal puis
décalage vertical

Épaisseur du flou

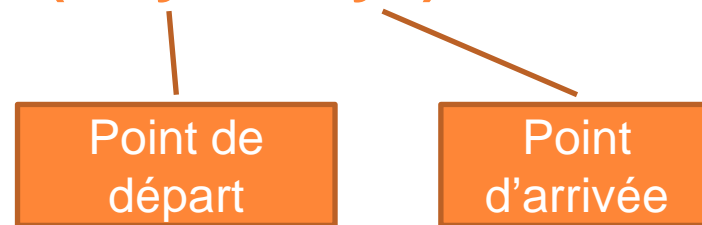
Couleur de l'ombre

Taille et famille de police

Hellow World

UTILISER UN DÉGRADÉ (LINÉAIRE)

o `ctx.createLinearGradient(x1,y1, x2,y2)`



- Le gradient se fait tout au long de cette ligne. Mais pour spécifier de quelle façon il faut faire le dégradé, on utilise “colorStop”

o `addColorStop(%, couleur)`

UTILISER UN DÉGRADÉ – LINÉAIRE EEMPLE

```
var degrade = ctx.createLinearGradient(10,10,150,150);  
degrade.addColorStop(0.2,'rgb(255,100,200)');  
degrade.addColorStop(0.8,'black');
```

```
ctx.fillStyle = degrade;  
ctx.fillRect(10, 10, 150, 150);
```



UTILISER UN DÉGRADÉ (RADIANT)

o **createRadialGradient(x1,y1,r1, x2,y2,r2)**



Coordonnées et rayon
de départ

Coordonnées et rayon
d'arrivée

- Un peu comme le dégradé linéaire, mais permettant de faire un dégradé de style radiant

UTILISER UN DÉGRADÉ – RADIANT (EXEMPLE)

```
var degrade = ctx.createRadialGradient(45,45,10,52,50,30);
```

```
degrade.addColorStop(0, '#A7D30C');
```

```
degrade.addColorStop(0.9, '#019F62');
```

```
degrade.addColorStop(1, 'rgba(1,159,98,0)');
```

Contour transparent

```
ctx.fillStyle = degrade;
```

```
ctx.fillRect(0,0,150,150);
```

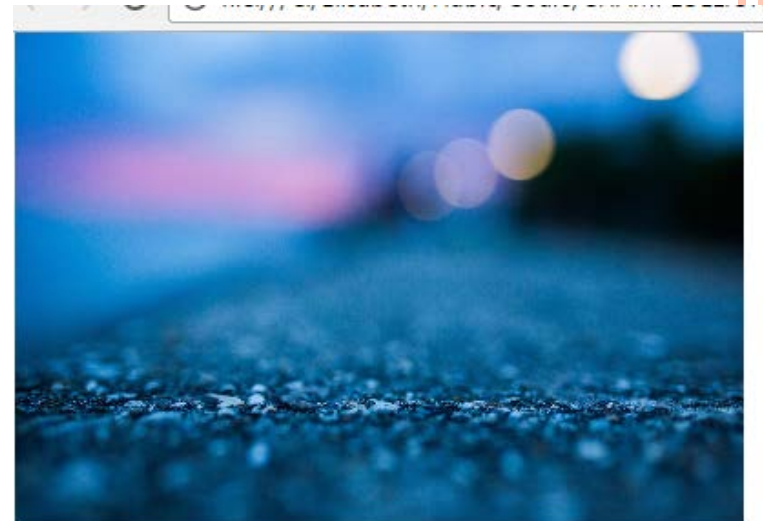
☐ Galerie de compos



INSÉRER UNE IMAGE

- Pour insérer une image, il faut d'abord la charger en mémoire. Lorsqu'elle est chargée, alors on peut l'afficher.
- **ctx.drawImage(imgVar, x1,y1, largeur, hauteur)**

```
var image = new Image();  
image.onload = function ()  
    { afficherImage(); }  
image.src = 'image.jpg';  
function afficherImage() {  
    ctx.drawImage(monImage, 0, 0, 400, 300);  
}
```

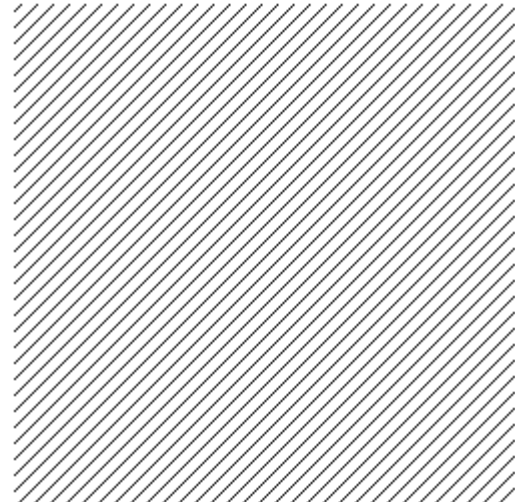


INSÉRER UNE IMAGE EN PATTERN

- o **createPattern(img ,répétition)**

- o Exemple

```
var image = new Image();  
image.src = 'image.jpg';  
image.onload = function(){  
    var motif = ctx.createPattern( this, 'repeat' );  
    ctx.fillStyle = motif;  
    ctx.fillRect( 50, 50, 250, 250 );  
};
```



MANIPULATION DU CONTEXTE

- Comme il est possible de faire des transformations au contexte (rotation, translation), il peut s'avérer utile de sauvegarder son état à certains moments.
 - `ctx.save()`
 - Sauvegarde l'état actuel du contexte
 - `ctx.restore()`
 - Remet le contexte à l'état de la dernière sauvegarde.

LES TRANSFORMATIONS (TRANSLATION)

o `translate(x, y)`

```
ctx.translate(60, 60);
```

```
ctx.fillStyle = "rgba(0,100,100, 0.5)";
```

```
ctx.fillRect(0, 0, 50, 50);
```

Galerie de composa...



LES TRANSFORMATIONS (ROTATION)

o Rotate(angle)

- Pour le paramètre “angle”, utiliser : **$x * \text{Math.PI}/180$**

x est l'angle voulu

```
ctx.translate(60, 60);  
ctx.rotate(45 * Math.PI / 180);
```

```
ctx.fillStyle = "rgba(0,100,100, 0.5)";  
ctx.fillRect(0, 0, 50, 50);
```

Canvas de composition



VIDER/EFFACER UN CANVAS

o `ctx.clearRect(x, y, width, height)`

- Ceci permet d'effacer le canvas (ou une partie du canvas)

ANIMATIONS

- A l'aide de `setInterval(action, temps en millisecondes)`
- Exemple : `var jeu=setInterval(animate, 1000)`

`clearInterval` permet de stopper `setInterval`.

Exemple : `clearInterval(jeu)`

CODE MINIMUM

```
<canvas id="zone" width="250" height="200">
Votre navigateur ne supporte pas la balise canvas
</canvas>
<script type="text/javascript">
window.addEventListener("load", init);

function init(evt)
{
    var canvas = document.getElementById('mon_canvas');
    var context = canvas.getContext('2d');
    if(!context)
    {
        alert("Impossible de récupérer le context du canvas");
    }
}
</script>
```