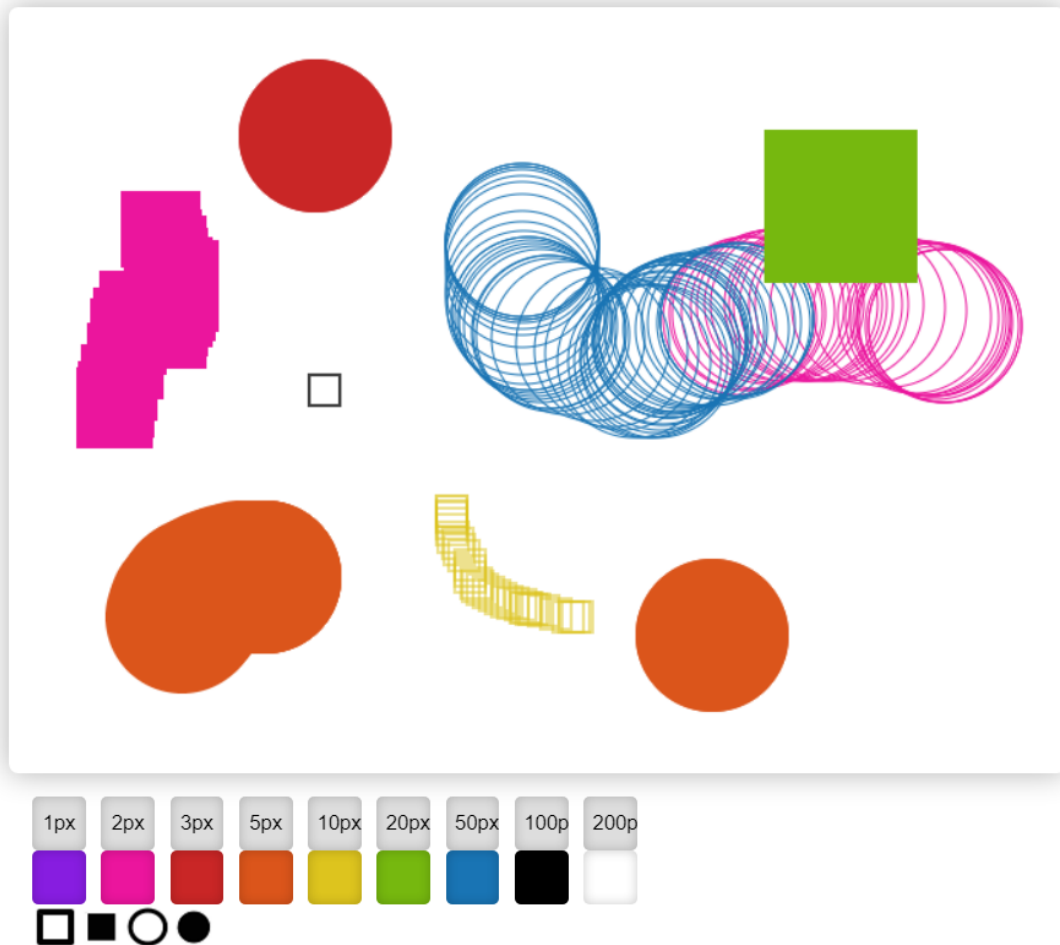


## Utilisation du javascript pour créer une ardoise magique

Aspect de l'ardoise:

### Ardoise magique



Enregistrer le dossier ardoise\_magique sur votre clé usb.

Ce dossier contient:

la page html : ardoise.html

le javascript : ardoise.js

les images : carre.png, carre\_plein.png, cercle.png et disque.png.

Une partie de l'ardoise est déjà réalisée. Pour ajouter les fonctionnalités, il vous faudra modifier les fichiers ardoise.html, ardoise.js et ardoise.css.

#### 1- Fichier ardoise.html

##### **Travail:**

*Ajouter les boutons de sélection de la taille, la couleur\* et la forme du pinceau.*

*\*Pour la couleur, il faudra aussi modifier le fichier ardoise.css.*

#### **La zone de dessin**

```
<canvas width="800" height="500" id="ardoise">
```

Ici se placera une zone de dessin.

```
</canvas>
```

**id="ardoise"** servira pour le javascript pour désigner la zone de dessin.

## Affichage d'un bouton dans une page html:

`<input type="submit" value="20px" onclick="changeSize(20);" />` bouton carré avec texte "20px"

`onclick="changeSize(20);"` action javascript à réaliser quand on clique sur le bouton.

`<input type="submit" onclick="changeColor('#871de0');" class="purple" value="" />` bouton carré de couleur

La couleur est définie dans ardoise.css par la ligne:

```
.purple { background: #871de0;}
```

`#871de0` est le code HTML de la couleur purple.

`<input type="image" onclick="changeForm('1');" class="carre" value="" src="carre.png"/>` bouton carré avec une image (Le carré correspond à la forme 1.)

## 2- Fichier ardoise.js

Ce fichier contiendra l'ensemble des fonctions javascript.

### Travail:

Créer les fonctions carré plein, cercle et disque.  
Ajouter ces fonctions à la fonction draw. Tester.

Pour aller plus loin:  
d'autres formes de pinceau  
bouton reset,  
choix d'un background (couleur, image...),  
sauvegarde dans un fichier...

### Quelques commandes javascript pour dessiner

```
nom_canvas_javascript=document.getElementById("nom du canvas HTML");  
nom_context=nom_canvas_javascript.getContext("2d");  
Les commandes suivantes s'appliquent au contexte de dessin.  
nom_context.beginPath();
```

<code>beginPath();</code>	Début d'une nouvelle figure
<code>closePath();</code>	Ferme la figure
<code>lineWidth =5;</code>	Épaisseur du trait en pixel
<code>strokeStyle = '#ff0000'; ="red"; ="rgb(255,0,0)"; ="rgba(255,0,0,1)";</code>	Couleur du trait
<code>moveTo(x,y);</code>	Déplacement du pinceau sans tracer de trait au point (x;y)
<code>lineTo(x,y);</code>	Trace un trait de la position actuelle au point (x;y)
<code>arc(x,y,50,startAngle,endAngle,booléen);</code>	Dessine un arc de centre (x;y) de rayon 50 avec les angles de départ et d'arrivée en radians. Si le booléen est true, l'arc est dans le sens direct.
<code>rect(x,y,width,height);</code>	Dessine un rectangle de sommet (x;y), de longueur width pixels et de largeur height pixels.
<code>strokeRect(x,y,width,height);</code>	Dessine un rectangle "creux" de sommet (x;y), de longueur width pixels et de largeur height pixels.
<code>quadraticCurveTo(x1,y1,x,y);</code>	Trace une courbe quadratique du point actuel au point (x;y) avec le point de contrôle (x1;y1).
<code>bezierCurveTo(x1,y1,x2,y2,x,y);</code>	Trace une courbe de Bézier du point actuel au point (x;y) avec les points de contrôle (x1;y1) et (x2;y2)
<code>stroke();</code>	Affiche la figure (cas des figures fil de fer)

<b>translate(x,y);</b>	Déplacement au point (x;y)
<b>scale(x,y);</b>	Mise à l'échelle
<b>rotate(radian);</b>	Rotation en radian dans le sens indirect (centre de la rotation (0;0))
<b>fillRect(x,y,width,height);</b>	Dessine un rectangle plein de sommet (x;y), de longueur width pixels et de largeur height pixels.
<b>clearRect(x,y,width,height);</b>	Efface le contenu du rectangle
<b>fillStyle = '#ff0000'; = "red"; = "rgb(255,0,0)"; = "rgba(255,0,0,1)";</b>	Couleur de la figure
<b>fill();</b>	Affiche la figure (cas des figures pleines)
<b>drawImage(img,x,y);</b>	Dessine l'image img à partir du point de coordonnées (x;y) cf exemple en fin de tableau
<b>createImageData(width,height);</b>	Crée un tableau de pixels en rgba de dimension width x height
<b>createImageData(image);</b>	Crée un tableau de pixels en rgba de image
<b>getImageData(x,y,width,height);</b>	Crée un tableau de pixels en rgba de dimension width x height à partir du point (x;y) du canvas
<b>putImageData(image,x,y);</b>	Place image dans le canvas
<b>nom_image.data;</b>	Tableau unidimensionnel en rgba des pixels de nim_image
<b>save();</b>	Sauvegarde d'un état du dessin (on empile)
<b>restore();</b>	Le dessin revient à l'état sauvegardé précédemment (on dépile)

```
function draw(context){
    var img = new Image();
    img.src = "nom.png";
    img.onload=function(){
        context.drawImage(img,150,15);
    }
}
```

références:

<https://www.alsacreations.com/tuto/lire/1484-introduction.html>

<http://www.html5canvastutorials.com/>

[https://developer.mozilla.org/fr/docs/Tutoriel\\_canvas/Formes\\_g%C3%A9om%C3%A9triques](https://developer.mozilla.org/fr/docs/Tutoriel_canvas/Formes_g%C3%A9om%C3%A9triques)