



## DÉFI RAPIDE !

*Peux-tu modifier l'exemple d'animation pour que le carré rebondisse contre les bords du canvas ?*

Pour que le carré rebondisse contre les bords du canvas, tu devras ajouter une logique pour inverser la direction du mouvement lorsque le carré atteint un bord. Voici comment tu peux modifier le code :

### EXEMPLE DE CODE VALIDE POUR LE DÉFI

```
let canvas = document.querySelector("#monCanvas");
let ctx = canvas.getContext("2d");

let x = 0; // Position x de départ du carré
let y = 50; // Position y de départ du carré
let dx = 10; // Vitesse de déplacement en x
let dy = 10; // Vitesse de déplacement en y
let squareSize = 50; // Taille du carré

function draw() {
  ctx.clearRect(0, 0, canvas.width, canvas.height); // On efface le canvas
  ctx.fillStyle = "blue";
  ctx.fillRect(x, y, squareSize, squareSize); // On dessine le carré
  // On met à jour la position du carré
  x += dx;
  y += dy;
  // On vérifie si le carré atteint les bords du canvas
  if (x + squareSize > canvas.width || x < 0) {
    dx = -dx; // Inverse la direction horizontale
  }
  if (y + squareSize > canvas.height || y < 0) {
    dy = -dy; // Inverse la direction verticale
  }
  requestAnimationFrame(draw); // On appelle draw au prochain rafraîchissement
}

draw(); // On appelle draw pour commencer
```

### EXPLICATION DU CODE

**Initialisation** : La position initiale du carré est définie par **x** et **y**, et les vitesses de déplacement par **dx** et **dy**.

**Animation** : La fonction **draw** est appelée de manière répétée grâce à **requestAnimationFrame**.

**Mise à jour de la position** : À chaque appel de **draw**, le carré est effacé et redessiné à sa nouvelle position.

**Collision avec les bords** : Lorsque le carré atteint un bord (droite, gauche, haut ou bas), la direction de son mouvement est inversée en changeant le signe de **dx** ou **dy**.

### SCÉNARIOS DE TEST

1. Le carré bleu de 50x50 pixels commence à se déplacer à partir de la position (0, 50).
  2. Le carré se déplace de manière continue et rebondit lorsqu'il atteint n'importe quel bord du canvas.
  3. Le mouvement se poursuit indéfiniment, changeant de direction à chaque collision avec un bord.
- Ce défi te permet de pratiquer la manipulation du canvas, les animations et la logique de collision en JavaScript, des compétences clés pour créer des jeux et des animations interactives.