

# Programmation Web – Avancé

JavaScript & Node.js

Animations via l'API Canvas



Attribution –  
Partage dans les  
Mêmes Conditions  
4.0 International  
(CC BY-SA 4.0)

*Presentation template  
by [SlidesCarnival](#)*



# Interaction avec l'API Canvas pour créer une animation

Apprendre à animer en JS pur grâce au browser...



## Interaction avec l'API Canvas pour créer une animation

---

- **<canvas>** : élément HTML utilisé pour dessiner (2D ou 3D) via des scripts
- Création d'animation en 2D sans librairie externe via l'API Canvas
- Existence aussi de l'API WebGL, de SVG DOM Interface...



## Création d'une animation avec l'API Canvas : style via Bootstrap

- Définition du style du **canvas** et éventuellement du **body** (si pas de marge) :

```
<link rel="stylesheet" href="style.css" />
```



- Ajout de l'élément **<canvas>**

```
<canvas id="canvas2D" ></canvas>
```





## Création d'une animation avec l'API Canvas : style via CSS

```
canvas {  
  display: block;  
}  
body {  
  margin: 0;  
  height: 100vh;  
}
```



- Définition du style du **canvas** et éventuellement du **body** (si pas de marge) :

```
<link rel="stylesheet" href="style.css" />
```



- Ajout de l'élément **<canvas>**

```
<canvas id="canvas2D" ></canvas>
```





## Création d'une animation avec l'API Canvas

- Accès à l'objet **canvas** : **querySelector()** ou autre
- Accès au contexte de rendu du **canvas** pour pouvoir dessiner dessus en 2D

```
let myCanvas = document.querySelector("canvas");
let myContext = myCanvas.getContext("2d");
function drawOneFrame() {
  myContext.clearRect(0, 0, pageWidth, pageHeight);
  myContext.fillStyle = "blue";
  myContext.fillRect(Math.floor(Math.random() * pageWidth),
                     Math.floor(Math.random() * pageHeight), 20, 20);
  // ...
}
```

JS



## Création d'une animation avec l'API Canvas

- Affichage successif de « frames » :
  - Appel d'une callback quand le browser est prêt (~60 callbacks / s) :  
`window.requestAnimationFrame(callback)` [\[41.\]](#)
  - Autre option : `setInterval()`





## Création d'une animation avec l'API Canvas

JS

```
requestAnimationFrame(drawOneFrame);  
function drawOneFrame() {  
  // 1. clear de canvas  
  // 2. draw the current frame  
  // 3. refresh automatically the animation via a recursive call  
  requestAnimationFrame(drawOneFrame);  
}
```

🕒 Tutoriel sur l'élément `<canvas>` [\[40.\]](#)





## Interaction avec l'API Canvas pour créer une animation

### 🕒 DEMO : Design d'une animation avec l'API Canvas

Générer une animation 2D de manière aléatoire reproduisant 101 fois un rectangle à des endroits différents du browser.

L'animation est appelée suite au chargement de la page.