





**KONVA**

Konva es un framework para canvas que permite añadir interactividad, hacer animaciones con buen desempeño, transiciones, capas, filtrado y más para aplicaciones de escritorio y móviles. A continuación haremos una comparación de los elementos antes vistos ahora utilizando Konva:

|  |  |
| --- | --- |
| **Convencional** | **Konva** |
| Lineas |  |
| context.beginPath(); context.moveTo(0,0); context.lineTo(400,200); context.stroke(); | var line=new Konva.line({ points:[0,0,400,200], stroke: "black", strokeWidth: 1, }); |
| Rectángulos |  |
| context.rect(100, 50, 50, 100); context.stroke(); | ar rect=new Konva.Rect({ x: 100, y: 50, height: 100, stroke: "black", strokeWidth: 5, fill: "blue" }); |
| Arcos |  |
| context.arc(100, 100, 50, Math.PI, Math.PI\*1,5, false); context.stroke(); | var arc=new Konva.Arc({ x: 100, y: 50, innerRadius: 50, outerRadius: 50, angle: 90, stroke: "black", strokeWidth: 1, rotation: 180 }); |
| Circulos |  |
| context.arc(120, 120, 70,0, 2\*Math.PI, false); context.fillStyle="#FF8800"; context.fill(); | var circle=new Konva.Circle({ x: 120, y: 120, radius: 70, fill: "#FF8800" }); |
| Textos |  |
| context.font="bold 30pt Arial, sans"; context.fillText("Hola mundo", 20, 50); | var text=new Konva.Text({ x: 120, y: 120, text: "Hola mundo", fontSize: 30, fontFamily: "Arial", }); |
| Imágenes |  |
| var imageObj=new Image(); imageObj.onload=function(){ context.drawImage(imageObj, 50, 50, 200, 150); }; imageObj.src="images/fruits.png" | var imageObj=new Image(); imageObj.onload=function (){ var fruits=new Konva.Image({ x: 50, y: 50, image: imageObj, width: 200, height: 150 }); layer.add(fruits); stage.add(layer); }; imageObj.src="images/fruits.png" |

**Eventos**

Para añadir manipuladores de eventos a figuras en Konva usamos el método on(). Para equipos de escritorio, Konva soporta los eventos mouseover, mouseout, mouseenter, mouseleave, mousemove, mousedown, mouseup, mousewheel, click, dblclick, dragstart, dragmove, y dragend En dispositivos móbiles, Konva soporta touchstart, touchmove, touchend, tap, dbltap, dragstart, dragmove, y dragend

**Arrastrar y soltar figura**

Para arrastrar y soltar figuras con Konva, podemos utilizar el método draggable() ó configurar en true la propiedad draggable. Esto permite arrastrar y soltar figuras tanto en aplicaciones de escritorio como en móviles.

**Grupos**

Agrupar figuras es muy útil cuando queremos hacerles alguna transformación por ejemplo mover, rotar o escalarlas al mismo tiempo. De hecho se pueden formar agrupaciones más complejas añadiendo grupos a los ya existentes.Para agrupar múltiples figuras con Konva podemos crear un objeto Konva.Group() y luego añadir figuras con el método add()

**Animaciones**

Podemos usar el constructor Konva.Animation para crear animaciones. Este constructor acepta dos parámetros: una función de actualización y opcionalmente una capa o arreglo de capas que serán actualizadas en cada fotograma de la animación.La función de actualización sólo debe contener lógica que actualice propiedades de los nodos como position, rotation, scale, width, height, radius, colors, etc  
Para iniciar la animación utilizamos el método start() y para detenerla el método stop().  
Por ejemplo para hacer rotar una figura podemos hacerlo con:  
  
var anim=new Konva.Animation(function(frame){  
blueRect.rotate(frame.timeDiff\*90/1000);  
}, layer);  
anim.start();  
  
Donde la propiedad timeDiff del objeto frame es el número de milisegundos que han pasado desde el último fotograma.

**Mejorando el desempeño**

Una de las formas que tiene Konva de mejorar el desempeño es el almacenamiento en Caché. Esta técnica te permite dibujar un elemento en un búfer de canvas y luego dibujarlo en el canvas. Para lograrlo se utiliza el método shape.cache()

Para ampliar más la información y ver la lista completa de opciones que Konva ofrece con sus respectivos ejemplos te invitamos a revisar la documentación oficial de Konva: <http://konvajs.github.io/docs/>