

[Download](#)



i2p • Digital3 · koolback · Dviewer · Hacking and sniffing your local area network · Windows * 3D Water Simulation. And no software to install. I have an old laptop 3Planesoft Category:ScreensaversQ: Javascript canvas inverse filter to transparency I'm trying to make a logo that has transparency at the bottom. The canvas element doesn't fully support this. I have a mask that draws the logo onto a canvas, now I want to apply an inverse blur to the canvas so that transparent pixels become black and opaque pixels become white. The problem is that when I apply my inverse filter, even the transparent pixels appear white. How do I fix this? Fiddle: JS: var canvas = document.getElementById("canvas"); var ctx = canvas.getContext("2d"); function drawLogo() { ctx.drawImage(img, 0,0,400,100); } function blur() { ctx.globalCompositeOperation = "difference"; ctx.globalAlpha = 9; ctx.globalCompositeOperation = "source-over"; ctx.fillStyle = "black"; ctx.clearRect(0,0,ctx.canvas.width, ctx.canvas.height); ctx.drawImage(ctx.canvas,0,0); } // Create a canvas element var c = document.createElement("canvas"); c.width = 400; c.height = 50; // Create a two-dimensional context ctx = c.getContext("2d"); drawLogo(); A: You can't have a grayscale source when applying a difference blend mode. But you can have a grayscale image as the source of a mask and then use a blend mode that can combine a grayscale source with a transparent one (add). For example, ctx.globalAlpha = 9; ctx.globalCompositeOperation = "add"; ctx.fillStyle = "black"; ctx.clearRect(0,0,ctx.canvas.width

