

☐

I'm not robot


reCAPTCHA

Continue

Download canvas as image jquery

to create an Instagram WebDev Offline, we all spent half of the last day hacking on our future Mozilla Marketplace app. One mobile app that recently got a lot of attention was Instagram, which sold to Facebook for the bat shit crazy price of one billion dollars. Since I wouldn't mind having a bill in my back account, I decided to last week's Instagram-style app (which I'll share with you in the future). This post details how you can convert an image to canvas and convert a canvas back to an image. Convert an Image to Canvas with JavaScript To create an image to canvas, you use a canvas element's context's drawImage method: // Converts image to canvas; returns new canvas element function convertImageToCanvas(image) { var canvas = document.createElement("canvas"); canvas.width = image.width; canvas.height = image.height; canvas.getContext("2d").drawImage(image, 0, 0); return canvas; } The 0, 0 arguments map to coordinates on the canvas where the image data should be placed. Convert Canvas to an Image with JavaScript Assuming modifications to the image have been made, you can easily convert the canvas data to image data with the following snippet: // Converts canvas to an image function convertCanvasToImage(canvas) { var image = new Image(); image.src = canvas.toDataURL("image/png"); return image; } The code above magically converts the canvas to a PNG data URI! Alas, converting an image to canvas and canvas to an image is probably much easier than you think. In future posts, I'll detail how you can apply different image filters to your canvased image. In the mean time, start buying fancy cars and houses with the future billion you'll have! For maximum performance, we all know we must put our assets on CDN (another domain). Along with those assets are custom web fonts. Unfortunately custom web fonts via CDN (or any cross-domain font request) don't work in Firefox or Internet Explorer (correctly so, by spec) though... As much as developers now loathe Flash, we're still playing a bit of catch up to natively duplicate the animation capabilities that Adobe's old technology provided us. Of course we have canvas, an awesome technology, one which I highlighted 9 mind-blowing demos. Another technology available...A while back David Walsh published a list of 7 MooTools Plugins You Should Use on Every Website which included 'AutoGrow' a text area expander plugin. 'AutoGrow' is very similar in results to the class I wrote for Education.com, Flexi. I decided to release this...I released my first MooTools class over a year ago. It was a really minimalistic approach to zebra tables and a great first class to write. I took some time to update and improve the class. The XHTML you may have as many tables as... We are excited to announce that the ASP.NET Forums are moving to the new Microsoft to draw an SVG to canvas: Find the width and height of an SVGNode class blobObjectCreate a canvas object from the SVGCreate a URL for the blobLoad the URL into an image elementCreate a canvas with width and height of the SVGDraw the image to the canvas To find the width and height of the SVG, we can use the SVGElement.getBBox(). This method will return an object with left, top, width, height value. In other words, the bounding box of the SVG element. To clone a node we can use cloneNode method. Now we need to create a blob object from the cloned node: To create a URL from the blob Object: We have a URL for the blob, now we need to load the blobURL to the image element. Also, we need to add onload event to the image element. This will be triggered once the image is loaded. Once the image is loaded we can draw the image to canvas. Now we have drawn the image to the canvas, let's download the canvasDownload canvas as image (png | webp | jpeg) We have a canvas, and we need to convert it into png or jpg. For that we will convert the canvas into DataURL. To download the image, we will write a function that will create a link element with href pointing to the DataURL created from the image. Selecting the quality of the image Consider we have a canvas. We can also set the quality of the image for jpeg and webp images. The quality value is a number between 0 and 1, indicating the image quality to use for image formats that use lossy compression such as image/jpeg and image/webp. The default value is 0.92. Downloading multiple files as zip Consider we have multiple files. We can zip multiple files into a single file and download it. For creating a zip, we can use JSZip library. Download jszip.js and include in your html. Once included, we need to: Create a JSZip() objectCreate a folderAdd the file to the folderDownload the zipCreating a JSZip objectCreate a folderAdd an image to the folderWe have two images: These two images are in the DataURL format, and we need to pass the base64 string only for hat we need to split the base64 string from the dataURL. Generally a dataURL looks like this: We need to split the data:image/png;base64, from the data URL. Then we need to add the baseString to the folder as file, so the final code looks like: Generate the zip fileThis will generate a zip and start the download. If you found any errors, please let me know. Because Code without bug is like a book without title. Please support me at paypal. Follow JavaScript Jeepe for more JavaScript Projects. I'm trying to download a canvas image to the desktop using the following code: var canvas; \$(document).ready(function() { if (\$('#designins-wrapper').length) { \$('#designins').each(function() { var design = \$('#this').attr('data-design'); var canvas = \$('#this').find('canvas'); var ctx = canvas.getContext('2d'); var img = new Image; img.onload = function() { ctx.drawImage(this, 0, 0); }; img.src = design; }); \$('#canvas').click(function() { this.href = canvas.toDataURL(); this.download = 'design.png'; }); }); Sadly I'm getting the following error: Uncaught TypeError: Cannot read property 'toDataURL' of undefined Does anyone have an idea how to fix this? Live preview: Introductions: 1) Click a image (this, 2) See Dev console EDIT: After updating the code to the following: Define canvas in global-scope Remove var from var canvas = \$('#this').find('canvas')[0]; The following error pops up: Uncaught SecurityError: Failed to execute 'toDataURL' on 'HTMLCanvasElement': Tainted canvas: Cannot be exported. How to save an HTML 5 Canvas as an image on the server? Saving HTML canvas as an image is pretty easy, it can be done by just right-clicking on the canvas and save it as an image. But saving canvas as an image on the server is quite different. This article will show you how to achieve that. Sometimes it is required to save canvas image after doing some server processing and this article will help in sending canvas image over the server for processing. Here you will see, how to convert an HTML content into an image using JQuery. We know that there is a plugins html2canvas, by using that plugin we can easily convert the HTML content into image content after that we can save that file by right-clicking and choosing the save image option. After that, we will convert that CANVAS image into the URL format and that to the server by using ajax after that the main part will be done by the PHP code. The PHP code will save that image on your server. Below steps will illustrate the approach clearly: Step 1: HTML code to convert canvas into image. Step 2: Display the output to check that the canvas is successfully converted into an image. Before clicking the Button: After clicking the Button: Step 3: Converting the image into URL format using canvas.toDataURL() Step 4: Sending the converted URL format into your server via Ajax.\$ajax({ type: "POST", url: "script.php", data: { imgBase64: dataURL } }) Step 5: This php code will save the image into the server.

[illegible]