

Satyanarayan Pooja Katha In Kannada

[Download](#)



Category:Public holidays in India Category:Hindu holy days Category:Religious festivals in India Category:Hinduism in India Category:Festivals in KarnatakaQ: How to display an image on a canvas using asynchronous image loading I have an app that has a rather significant number of images to load in order to be displayed in a canvas (1920x1080). I was wondering if it's possible to use asynchronous image loading to solve this problem. As I understand, in a requestAnimationFrame loop I'm able to update the canvas with the image, but in order to do this I have to re-position the canvas as it's drawn overtop of the image. I'm having trouble finding a way to do this without having to re-set the width and height of the canvas element. A: Working with the built-in browser capabilities, this should be pretty straightforward. Just start an animation and use requestAnimationFrame to callback when the image finishes loading. Using a deferred object If you have an existing deferred object from elsewhere in your code (perhaps a request/XHR or something), you can directly use that. Your function would look like: function loadImage(image, canvas) { var deferred = \$.Deferred(); // do whatever to load the image and resolve the deferred // with the image loaded. // or... // if you already have the image, or just want to get // the image in an inline fashion, use this: canvas.width = canvas.width || image.width; canvas.height = canvas.height || image.height; deferred.resolve(canvas); return deferred.promise(); } Using a simple XHR object, such as the XHR object in jQuery: Of course, if you already have a script that uses the XHR object, you can reuse that code; or you can use the deferred approach, but then just need to wrap the promise around the XHR object and resolve it. Using your own promise Use the jQuery promise object to wrap any promise you want: function loadImage(image, canvas) { var deferred = \$.Deferred(); // do whatever to

