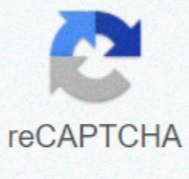I'm not robot

reCAPTCHA

Continue

I'm not robot

reCAPTCHA

Continue

Decode base64 to image javascript

Comments: 3 | Rating: 5/5 Convert Base64 to JPG online using a free decoding tool that allows you to decode Base64 as JPG image and preview it directly in the browser. By and large, the "Base64 to JPG" converter is similar to Base64 to Image, except that it this one forces the MIME type to be "image/jpeg". If you are looking for the reverse process, check JPG to Base64. Important notes about the decoder The "Base64 to JPG" converter will force the decoding result to be displayed as a JPG image, even if it is a different file type. Therefore, if you are not sure that your Base64 string is JPG image, use the Base64 to file converter since it is capable of displaying various file types. In any case, feel free to give it a try, since this converter will inform you if something is wrong. About JPG Name: JPEG Developer: Joint Photographic Experts Group MIME types: image/jpeg File Extensions: .jpg, .jpeg, .jpe, .jif, .jfif, .jfi Uniform Type Identifier: public.jpeg Type of Format: Lossy image Comments: 10 | Rating: 4.3/5 Convert Base64 to PNG online using a free decoding tool that allows you to decode Base64 as PNG image and preview it directly in the browser. By and large, the "Base64 to PNG" converter is similar to Base64 to Image, except that it this one forces the MIME type to be "image/png". If you are looking for the reverse process, check PNG to Base64. Important notes about the decoder The "Base64 to PNG" converter will force the decoding result to be displayed as a PNG image, even if it is a different file type. Therefore, if you are not sure that your Base64 string is PNG image, use the Base64 to file converter since it is capable of displaying various file types. In any case, feel free to give it a try, since this converter will inform you if something is wrong. About PNG Name: Portable Network Graphics Developer: PNG Development Group MIME types: image/png File Extensions: .png Uniform Type Identifier: public.png Type of Format: Lossless bitmap image Base64 Decode is easy to use tool to decode base64 data to plain text. Copy, Paste and Decode. What can you do with Base64 Decode? Base64 Decode is very unique tool to decode base64 data to plain text. This tool saves your time and helps to decode base64 data. This tool allows loading the Base64 data URL, which loads base64 encoded text and decodes to human readable text. Click on the URL button, Enter URL and Submit. Users can also convert base64 data File to Plain text by uploading the file. Base64 Decoder Online works on Windows, MAC, Linux, Chrome, Firefox, Edge, and Safari. What is Base64? Base64 is numeral system of base-64 which uses 64 digit set and can be represented by 6 bits. To learn more about Base64, please visit Base64 Wikipedia page. Example Base64 Encoded Text Try it. R2VpY28= Output: Base64 Decoded text Geico For Advanced Users External URL Load External URL in Browser URL like this Data as Parameter Load Data in Browser URL input like this Base64 (automatically decoded) ASCII Plain Text (automatically encoded) Hex Options Decode as Image This is a simple online base 64 encoder and decoder. This page was designed to be helpful to developers and anyone doing programming work. Base64 is a common format used for the web and email. It allows binary data to be transmitted in plain text format without risk of the data being clobbered by an intermediate server that does not handle certain binary characters. The number system has a base of 64, which means that each character requires 6 bits of storage. This page should be useful to anyone who occasionally comes across a base64 string that they want to decode. This includes things like HTTP basic authentication passwords. This app is helpful to PHP, Ruby on Rails, and other developers. It is also helpful for new programmers who are trying to understand base64 encoding. Please let me know below if you have any ideas on how I can improve the functionality or usability of this page. I would assume that one might want a solution that produces a widely useable base64 URI. Please visit data:text/plain;charset=utf-8;base64,4pi44pi54pi64pi74pi84pi++4pi/ to see a demonstration (copy the data uri, open a new tab, paste the data URI into the address bar, then press enter to go to the page). Despite the fact that this URI is base64-encoded, the browser is still able to recognize the high code points and decode them properly. The minified encoder+decoder is 1058 bytes (+Gzip=589 bytes) !function(e){"use strict";function h(b){var a=b.charCodeAt(0);if(55296=a)if(b=b.charCodeAt(1),b===b&&56320=b){if(a=1024*(a-55296)+b-56320+65536,65535>>18,128|a>>>12&63,128|a>>>6&63,128|a&63)}else return d(239,191,189);return 127>=a?inputString:2047>=a?d(192|a>>>6,128|a&63):d(224|a>>>12,128|a>>>6&63,128|a&63)}function k(b){var a=b.charCodeAt(0)=f){a=a>24+f;for(c=1;c>10)+55296,(a&1023)+56320)}:c=0}for(;c>>0)n|0},d=String.fromCharCode,p=atob,q=btoa;q.btoaUTF8=function(b,a){return q((a?"\u00ef\u00bb\u00bf":"")+b.replace(/[\x80-\u07ff\uDC00-\uFFFF]|[\uD800-\uDBFF][\uDC00-\uDFFF]?/g,h))};e.atobUTF8=function(b,a){a||"\u008[\u00bb\u00bf"!==b.substring(0,3)||(b=b.substring(3));return p(b).replace(/[\xc0-\xff][\x80-\xbf]*/g,k)}}(""+void 0==typeof global?""+void 0==typeof self?this:self:global) Below is the source code used to generate it. var fromCharCode = String.fromCharCode; var btoaUTF8 = (function(btoa, replacer){"use strict"; return function(inputString, BOMit){ return btoa((BOMit ? "\xEF\xBB\xBF" : "") + inputString.replace(/[\x80-\u07ff\uDC00-\uFFFF]|[\uD800-\uDBFF][\uDC00-\uDFFF]?/g, replacer)); } })(btoa, function(nonAsciiChars){"use strict"; // make the UTF string into a binary UTF-8 encoded string var point = nonAsciiChars.charCodeAt(0); if (point >= 0xD800 && point = 0xDC00 && nextcode 0xffff) return fromCharCode((0x1e/*0b11110*/>18), (0x2/*0b10*/>12)&0x3f/*0b00111111*/), (0x2/*0b10*/>6)&0x3f/*0b00111111*/), (0x2/*0b10*/ 0b11111111; 0xc0-0xff => 0b11xxxxxx // 0x80 => 0b10000000; 0xbf => 0b10111111; 0x80-0xbf => 0b10xxxxxx return inputString.replace(/[\xc0-\xff][\x80-\xbf]*/g, replacer); } })(atob, function(encoded){"use strict"; var codePoint = encoded.charCodeAt(0) = leadingOnes) { codePoint = (codePoint>(24+leadingOnes); for (endPos = 1; endPos < leadingOnes; ++endPos) codePoint = (codePoint