

# Base64

Ein kleines JavaScript zur Demonstration von Base64 Encoding und Decoding.

Decoded:



Encoded:

HTML:

[base64.htm \(/code/base64.htm?mode=download\)](#) HTML (534 Bytes) 21.02.2021 00:28

```
<!DOCTYPE html >
<html lang="en">
  <head>
    <meta charset="utf-8" />
  </head>
  <body>
    <form><label for="decoded">Decoded:</label><input type="text"
id="decoded" /><button type="button" id="encode" title="encode
value"></button><button type="button" id="decode" title="decode
value"></button><label for="encoded">Encoded:</label><input type="text"
id="encoded" /></form>
    <script src="base64.js"></script>
  </body>
</html>
```

JavaScript:

[base64.js \(/code/base64.js?mode=download\)](#) JavaScript (4,02 kByte) 20.12.2020 13:55

```
// coding: utf-8
/** Created by: Udo Schmal | https://www.gocher.me/ */
(function () {
  'use strict';
  var Base64 = {
    // private property
    _keyStr :
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/",
    // public method for encoding
    encode : function (input) {
      var output = "";
      var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
      var i = 0;
      input = Base64._utf8_encode(input);
      while (i < input.length) {
        chr1 = input.charCodeAt(i++);
        chr2 = input.charCodeAt(i++);
        chr3 = input.charCodeAt(i++);
        enc1 = chr1 >> 2;
        enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);
        enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);
        enc4 = chr3 & 63;
```

```

        if (isNaN(chr2)) {
            enc3 = enc4 = 64;
        } else if (isNaN(chr3)) {
            enc4 = 64;
        };
        output = output + Base64._keyStr.charAt(enc1) +
Base64._keyStr.charAt(enc2) + Base64._keyStr.charAt(enc3) +
Base64._keyStr.charAt(enc4);
    };
    return output;
},
// public method for decoding
decode : function (input) {
    var output = "";
    var chr1, chr2, chr3;
    var enc1, enc2, enc3, enc4;
    var i = 0;
    input = input.replace(/[^A-Za-z0-9\+\ \/\\=]/g, "");
    while (i < input.length) {
        enc1 = Base64._keyStr.indexOf(input.charAt(i++));
        enc2 = Base64._keyStr.indexOf(input.charAt(i++));
        enc3 = Base64._keyStr.indexOf(input.charAt(i++));
        enc4 = Base64._keyStr.indexOf(input.charAt(i++));
        chr1 = (enc1 << 2) | (enc2 >> 4);
        chr2 = ((enc2 & 15) << 4) | (enc3 >> 2);
        chr3 = ((enc3 & 3) << 6) | enc4;
        output = output + String.fromCharCode(chr1);
        if (enc3 != 64) {
            output = output + String.fromCharCode(chr2);
        };
        if (enc4 != 64) {
            output = output + String.fromCharCode(chr3);
        }
    };
    output = Base64._utf8_decode(output);
    return output;
},
// private method for UTF-8 encoding
_utf8_encode : function (string) {
    string = string.replace(/\r\n/g, "\n");
    var utftext = "";
    for (var n = 0; n < string.length; n++) {
        var c = string.charCodeAt(n);
        if (c < 128) {
            utftext += String.fromCharCode(c);
        } else if ((c > 127) && (c < 2048)) {
            utftext += String.fromCharCode((c >> 6) | 192);
            utftext += String.fromCharCode((c & 63) | 128);
        } else {
            utftext += String.fromCharCode((c >> 12) | 224);

```

```

        utftext += String.fromCharCode((c >> 6) & 63) | 128);
        utftext += String.fromCharCode(c & 63) | 128);
    }
}
return utftext;
},
// private method for UTF-8 decoding
_utf8_decode : function (utftext) {
    var string = "";
    var i = 0;
    var c = 0, c1 = 0, c2 = 0;
    while ( i < utftext.length ) {
        c = utftext.charCodeAt(i);
        if (c < 128) {
            string += String.fromCharCode(c);
            i++;
        } else if ((c > 191) && (c < 224)) {
            c1 = utftext.charCodeAt(i+1);
            string += String.fromCharCode(((c & 31) << 6) | (c1 & 63));
            i += 2;
        } else {
            c1 = utftext.charCodeAt(i+1);
            c2 = utftext.charCodeAt(i+2);
            string += String.fromCharCode(((c & 15) << 12) | ((c1 & 63) << 6) |
(c2 & 63));
            i += 3;
        }
    }
};
return string;
}
}

var encode = document.getElementById('encode');
if (encode) {
    encode.addEventListener('click', function (event) {
        var val_in = document.getElementById('decoded').value;
        var val_out = Base64.encode(val_in);
        document.getElementById('encoded').value = val_out;
    });
}
var decode = document.getElementById('decode');
if (decode) {
    decode.addEventListener('click', function (event) {
        var val_in = document.getElementById('encoded').value;
        var val_out = Base64.decode(val_in);
        document.getElementById('decoded').value = val_out;
    });
}
})();

```

Autor: Udo Schmal ([#udo.schmal](#)), veröffentlicht: 10.10.2015, letzte Änderung: 06.09.2023

© Copyright 2023 Udo Schmal

(/)