

```
/**
 * [js-md2]{@link https://github.com/emn178/js-md2}
 *
 * @namespace md2
 * @version 0.2.2
 * @author Chen, Yi-Cyuan [emn178@gmail.com]
 * @copyright Chen, Yi-Cyuan 2014-2017
 * @license MIT
 */
/*jslint bitwise: true */
(function () {
  'use strict';

  var root = typeof window === 'object' ? window : {};
  var NODE_JS = !root.JS_MD2_NO_NODE_JS && typeof process === 'object' && process.versions &&
process.versions.node;
  if (NODE_JS) {
    root = global;
  }
  var COMMON_JS = !root.JS_MD2_NO_COMMON_JS && typeof module === 'object' && module.exports;
  var AMD = typeof define === 'function' && define.amd;
  var HEX_CHARS = '0123456789abcdef'.split('');

  var S = [
    0x29, 0x2E, 0x43, 0xC9, 0xA2, 0xD8, 0x7C, 0x01, 0x3D, 0x36, 0x54, 0xA1, 0xEC, 0xF0, 0x06, 0x13,
    0x62, 0xA7, 0x05, 0xF3, 0xC0, 0xC7, 0x73, 0x8C, 0x98, 0x93, 0x2B, 0xD9, 0xBC, 0x4C, 0x82, 0xCA,
    0x1E, 0x9B, 0x57, 0x3C, 0xFD, 0xD4, 0xE0, 0x16, 0x67, 0x42, 0x6F, 0x18, 0x8A, 0x17, 0xE5, 0x12,
    0xBE, 0x4E, 0xC4, 0xD6, 0xDA, 0x9E, 0xDE, 0x49, 0xA0, 0xFB, 0xF5, 0x8E, 0xBB, 0x2F, 0xEE, 0x7A,
    0xA9, 0x68, 0x79, 0x91, 0x15, 0xB2, 0x07, 0x3F, 0x94, 0xC2, 0x10, 0x89, 0x0B, 0x22, 0x5F, 0x21,
    0x80, 0x7F, 0x5D, 0x9A, 0x5A, 0x90, 0x32, 0x27, 0x35, 0x3E, 0xCC, 0xE7, 0xBF, 0xF7, 0x97, 0x03,
    0xFF, 0x19, 0x30, 0xB3, 0x48, 0xA5, 0xB5, 0xD1, 0xD7, 0x5E, 0x92, 0x2A, 0xAC, 0x56, 0xAA, 0xC6,
    0x4F, 0xB8, 0x38, 0xD2, 0x96, 0xA4, 0x7D, 0xB6, 0x76, 0xFC, 0x6B, 0xE2, 0x9C, 0x74, 0x04, 0xF1,
    0x45, 0x9D, 0x70, 0x59, 0x64, 0x71, 0x87, 0x20, 0x86, 0x5B, 0xCF, 0x65, 0xE6, 0x2D, 0xA8, 0x02,
    0x1B, 0x60, 0x25, 0xAD, 0xAE, 0xB0, 0xB9, 0xF6, 0x1C, 0x46, 0x61, 0x69, 0x34, 0x40, 0x7E, 0x0F,
```

```

    0x55, 0x47, 0xA3, 0x23, 0xDD, 0x51, 0xAF, 0x3A, 0xC3, 0x5C, 0xF9, 0xCE, 0xBA, 0xC5, 0xEA, 0x26,
    0x2C, 0x53, 0x0D, 0x6E, 0x85, 0x28, 0x84, 0x09, 0xD3, 0xDF, 0xCD, 0xF4, 0x41, 0x81, 0x4D, 0x52,
    0x6A, 0xDC, 0x37, 0xC8, 0x6C, 0xC1, 0xAB, 0xFA, 0x24, 0xE1, 0x7B, 0x08, 0x0C, 0xBD, 0xB1, 0x4A,
    0x78, 0x88, 0x95, 0x8B, 0xE3, 0x63, 0xE8, 0x6D, 0xE9, 0xCB, 0xD5, 0xFE, 0x3B, 0x00, 0x1D, 0x39,
    0xF2, 0xEF, 0xB7, 0x0E, 0x66, 0x58, 0xD0, 0xE4, 0xA6, 0x77, 0x72, 0xF8, 0xEB, 0x75, 0x4B, 0x0A,
    0x31, 0x44, 0x50, 0xB4, 0x8F, 0xED, 0x1F, 0x1A, 0xDB, 0x99, 0x8D, 0x33, 0x9F, 0x11, 0x83, 0x14
];

```

```

var M = [], X = [], C = [];

```

```

var md2 = function (message) {
    var code, i, j, k, t, L = 0, loop = 1, B,
        index = 0, start = 0, bytes = 0, length = message.length;

```

```

    for (i = 0; i < 16; ++i) {
        X[i] = C[i] = 0;
    }

```

```

M[16] = M[17] = M[18] = 0;

```

```

do {
    M[0] = M[16];
    M[1] = M[17];
    M[2] = M[18];
    M[16] = M[17] = M[18] = M[3] =
    M[4] = M[5] = M[6] = M[7] =
    M[8] = M[9] = M[10] = M[11] =
    M[12] = M[13] = M[14] = M[15] = 0;
    for (i = start; index < length && i < 16; ++index) {
        code = message.charCodeAt(index);
        if (code < 0x80) {
            M[i++] = code;
        } else if (code < 0x800) {
            M[i++] = 0xc0 | (code >> 6);
            M[i++] = 0x80 | (code & 0x3f);

```

```

    } else if (code < 0xd800 || code >= 0xe000) {
        M[i++] = 0xe0 | (code >> 12);
        M[i++] = 0x80 | ((code >> 6) & 0x3f);
        M[i++] = 0x80 | (code & 0x3f);
    } else {
        code = 0x10000 + (((code & 0x3ff) << 10) | (message.charCodeAt(++index) & 0x3ff));
        M[i++] = 0xf0 | (code >> 18);
        M[i++] = 0x80 | ((code >> 12) & 0x3f);
        M[i++] = 0x80 | ((code >> 6) & 0x3f);
        M[i++] = 0x80 | (code & 0x3f);
    }
}
bytes += i - start;
start = i - 16;

if (index === length && i < 16) {
    loop = 2;
    t = 16 - (bytes & 15);
    for (; i < 16; ++i) {
        M[i] = t;
    }
}

for (i = 0; i < 16; ++i) {
    C[i] ^= S[M[i] ^ L];
    L = C[i];
}

for (i = 0; i < loop; ++i) {
    B = i === 0 ? M : C;

    X[16] = B[0];
    X[32] = X[16] ^ X[0];
    X[17] = B[1];

```

```
X[33] = X[17] ^ X[1];
X[18] = B[2];
X[34] = X[18] ^ X[2];
X[19] = B[3];
X[35] = X[19] ^ X[3];
X[20] = B[4];
X[36] = X[20] ^ X[4];
X[21] = B[5];
X[37] = X[21] ^ X[5];
X[22] = B[6];
X[38] = X[22] ^ X[6];
X[23] = B[7];
X[39] = X[23] ^ X[7];
X[24] = B[8];
X[40] = X[24] ^ X[8];
X[25] = B[9];
X[41] = X[25] ^ X[9];
X[26] = B[10];
X[42] = X[26] ^ X[10];
X[27] = B[11];
X[43] = X[27] ^ X[11];
X[28] = B[12];
X[44] = X[28] ^ X[12];
X[29] = B[13];
X[45] = X[29] ^ X[13];
X[30] = B[14];
X[46] = X[30] ^ X[14];
X[31] = B[15];
X[47] = X[31] ^ X[15];

t = 0;
for (j = 0; j < 18; ++j) {
    for (k = 0; k < 48; ++k) {
        X[k] = t = X[k] ^ S[t];
```

```
        }
        t = (t + j) & 0xFF;
    }
}
} while (loop === 1);

var hex = '';
for (i = 0; i < 16; ++i) {
    hex += HEX_CHARS[(X[i] >> 4) & 0x0F] + HEX_CHARS[X[i] & 0x0F];
}
return hex;
};

if (COMMON_JS) {
    module.exports = md2;
} else {
    root.md2 = md2;
    if (AMD) {
        define(function() {
            return md2;
        });
    }
}
}()};
```