Punycode.js

Punycode.js is a robust Punycode converter that fully complies to RFC 3492 and RFC 5891.

This JavaScript library is the result of comparing, optimizing and documenting different open-source implementations of the Punycode algorithm:

- The C example code from RFC 3492
- `punycode.c` by *Markus W. Scherer* (IBM)
- `punycode.c` by *Ben Noordhuis*
- JavaScript implementation by *some*
- `punycode.js` by *Ben Noordhuis* (note: not fully compliant)

This project was bundled with Node.js from v0.6.2+ until v7 (soft-deprecated).

This project provides a CommonJS module that uses ES2015+ features and JavaScript module, which work in modern Node.js versions and browsers. For the old Punycode.js version that offers the same functionality in a UMD build with support for older pre-ES2015 runtimes, including Rhino, Ringo, and Narwhal, see v1.4.1.

Installation

Via npm:

```
npm install punycode --save
```

In Node.js:

> ⚠️ *Note that userland modules don't hide core modules. For example,* `require('punycode')` *still imports the deprecated core module even if you executed* `npm install punycode`. *Use* `require('punycode/')` *to import userland modules rather than core modules.*

```
const punycode = require('punycode/');
```

API

`punycode.decode(string)`

Converts a Punycode string of ASCII symbols to a string of Unicode symbols.

License

Punycode.js is available under the MIT license.

For maintainers

How to publish a new release

1. On the `main` branch, bump the version number in `package.json`:

   ```
   npm version patch -m 'Release v%s'
   ```

   Instead of `patch`, use `minor` or `major` as needed.
   Note that this produces a Git commit + tag.
2. Push the release commit and tag:

   ```
   git push && git push --tags
   ```

   Our CI then automatically publishes the new release to npm, under both the `punycode` and `punycode.js` names.

Author

Mathias Bynens


```
// decode domain name parts
punycode.decode('maana-pta'); //
        'mañana'
punycode.decode('--dqo34k'); // '💩-⌘'
```

punycode.encode(string)

Converts a string of Unicode symbols to a Punycode string of ASCII symbols.

```
// encode domain name parts
punycode.encode('mañana'); // 'maana-
        pta'
punycode.encode('💩-⌘'); // '--dqo34k'
```

punycode.toUnicode(input)

Converts a Punycode string representing a domain name or an email address to Unicode. Only the Punycoded parts of the input will be converted, i.e. it doesn't matter if you call it on a string that has already been converted to Unicode.

```
// decode domain names
punycode.toUnicode('xn--maana-pta.com');
// → 'mañana.com'
punycode.toUnicode('xn----dqo34k.com');
// → '💩-⌘.com'

// decode email addresses
```

```
punycode.toUnicode('джумла@xn--
        p-8sbkgc5ag7bhce.xn--ba-lmcq');
// → 'джумла@джрумлатест.bpфa'
```

punycode.toASCII(input)

Converts a lowercased Unicode string representing a domain name or an email address to Punycode. Only the non-ASCII parts of the input will be converted, i.e. it doesn't matter if you call it with a domain that's already in ASCII.

```
// encode domain names
punycode.toASCII('mañana.com');
// → 'xn--maana-pta.com'
punycode.toASCII('☃-⌘.com');
// → 'xn----dqo34k.com'

// encode email addresses
punycode.toASCII('джумла@джрумлатест.bpфa');
// → 'джумла@xn--p-8sbkgc5ag7bhce.xn--
        ba-lmcq'
```

punycode.ucs2

punycode.ucs2.decode(string)

Creates an array containing the numeric code point values of each Unicode symbol in the string. While JavaScript uses UCS-2 internally, this function will convert a pair of surrogate halves

(each of which UCS-2 exposes as separate characters) into a single code point, matching UTF-16.

```
punycode.ucs2.decode('abc');
// → [0x61, 0x62, 0x63]
// surrogate pair for U+1D306 TETRAGRAM
        FOR CENTRE:
punycode.ucs2.decode('\uD834\uDF06');
// → [0x1D306]
```

punycode.ucs2.encode(codePoints)

Creates a string based on an array of numeric code point values.

```
punycode.ucs2.encode([0x61, 0x62,
        0x63]);
// → 'abc'
punycode.ucs2.encode([0x1D306]);
// → '\uD834\uDF06'
```

punycode.version

A string representing the current Punycode.js version number.