# shell-quote [![Version Badge][npm-version-svg]][package-url]

[github actions](https://github.com/ljharb/shell-quote/actions) [coverage](https://app.codecov.io/gh/ljharb/shell-quote/) [License](http://license) [Downloads](https://npm-stat.com/charts.html?package=shell-quote)

[npm badge](https://npmjs.org/package/shell-quote)

Parse and quote shell commands.

# example

## quote

var quote = require('shell-quote/quote');

var s = quote([ 'a', 'b c d', '$f', '"g"' ]);

console.log(s);

output

a 'b c d' \$f '"g"'

## parse

var parse = require('shell-quote/parse');

var xs = parse('a "b c" \\$def \'it\\\'s great\'');

console.dir(xs);

output

[ 'a', 'b c', '\\$def', 'it\'s great' ]

## parse with an environment variable

var parse = require('shell-quote/parse');

var xs = parse('beep --boop="$PWD"', { PWD: '/home/robot' });

console.dir(xs);

output

[ 'beep', '--boop=/home/robot' ]

## parse with custom escape character

var parse = require('shell-quote/parse');

var xs = parse('beep ^--boop="$PWD"', { PWD: '/home/robot' }, { escape: '^' });

console.dir(xs);

output

[ 'beep --boop=/home/robot' ]

## parsing shell operators

var parse = require('shell-quote/parse');

var xs = parse('beep || boop > /byte');

console.dir(xs);

output:

[ 'beep', { op: '||' }, 'boop', { op: '>' }, '/byte' ]

## parsing shell comment

var parse = require('shell-quote/parse');

var xs = parse('beep > boop # > kaboom');

console.dir(xs);

output:

[ 'beep', { op: '>' }, 'boop', { comment: '> kaboom' } ]

# methods

var quote = require('shell-quote/quote');

var parse = require('shell-quote/parse');

## quote(args)

Return a quoted string for the array args suitable for using in shell commands.

## parse(cmd, env={})

Return an array of arguments from the quoted string cmd.

Interpolate embedded bash-style $VARNAME and ${VARNAME} variables with the env object which like bash will replace undefined variables with "".

env is usually an object but it can also be a function to perform lookups. When env(key) returns a string, its result will be output just like env[key] would. When env(key) returns an object, it will be inserted into the result array like the operator objects.

When a bash operator is encountered, the element in the array with be an object with an "op" key set to the operator string. For example:

'beep || boop > /byte'

parses as:

[ 'beep', { op: '||' }, 'boop', { op: '>' }, '/byte' ]

# install

With [npm](http://npmjs.org) do:

npm install shell-quote

# license

MIT