# which

Like the unix which utility.

Finds the first instance of a specified executable in the PATH environment variable. Does not cache the results, so hash -r is not needed when the PATH changes.

## USAGE

var which = require('which')

// async usage

which('node', function (er, resolvedPath) {

// er is returned if no "node" is found on the PATH

// if it is found, then the absolute path to the exec is returned

})

// or promise

which('node').then(resolvedPath => { ... }).catch(er => { ... not found ... })

// sync usage

// throws if not found

var resolved = which.sync('node')

// if nothrow option is used, returns null if not found

resolved = which.sync('node', {nothrow: true})

// Pass options to override the PATH and PATHEXT environment vars.

which('node', { path: someOtherPath }, function (er, resolved) {

if (er)

throw er

console.log('found at %j', resolved)

})

## CLI USAGE

Same as the BSD which(1) binary.

usage: which [-as] program ...

## OPTIONS

You may pass an options object as the second argument.

* path: Use instead of the PATH environment variable.
* pathExt: Use instead of the PATHEXT environment variable.
* all: Return all matches, instead of just the first one. Note that this means the function returns an array of strings instead of a single string.