minimist v1.2.8

parse argument options

This module is the guts of optimist’s argument parser without all the fanciful decoration.

example

```
var argv = require('minimist')
        (process.argv.slice(2));
console.log(argv);
```

```
$ node example/parse.js -a beep -b boop
{ _: [], a: 'beep', b: 'boop' }
```

```
$ node example/parse.js -x 3 -y 4 -n5
-abc --beep=boop foo bar baz
{
    _: ['foo', 'bar', 'baz'],
    x: 3,
    y: 4,
    n: 5,
    a: true,
    b: true,
    c: true,
    beep: 'boop'
}
```

license

MIT

install

With npm do:

```
npm install minimist
```

security

Previous versions had a prototype pollution bug that could cause privilege escalation in some circumstances when handling untrusted user input.

Please use version 1.2.6 or later:

- https://security.snyk.io/vuln/SNYK-JS-MINIMIST-2429795 (version <=1.2.5)
- https://snyk.io/vuln/SNYK-JS-MINIMIST-559764 (version <=1.2.3)

methods

```
var parseArgs = require('minimist')
```

var argv = parseArgs(args, opts={})

Return an argument object `argv` populated with the array arguments from `args`.

`argv._` contains all the arguments that didn't have an option associated with them.

Numeric-looking arguments will be returned as numbers unless `opts.string` or `opts.boolean` is set for that argument name.

Any arguments after `'--'` will not be parsed and will end up in `argv._`.

options can be:

- `opts.string` - a string or array of strings argument names to always treat as strings
- `opts.boolean` - a boolean, string or array of strings to always treat as booleans. if `true` will treat all double hyphenated arguments without equal signs as boolean (e.g. affects `--foo`, not `-f` or `--foo=bar`)
- `opts.alias` - an object mapping string names to strings or arrays of string argument names to use as aliases

- `opts.default` - an object mapping string argument names to default values
- `opts.stopEarly` - when true, populate `argv._` with everything after the first non-option
- `opts['--']` - when true, populate `argv._` with everything before the `--` and `argv['--']` with everything after the `--`. Here's an example:

```
  > require('./')('one two three -- four
five --six'.split(' '), { '--': true })
{
  _: ['one', 'two', 'three'],
  '--': ['four', 'five', '--six']
}
```

Note that with `opts['--']` set, parsing for arguments still stops after the `--`.

- `opts.unknown` - a function which is invoked with a command line parameter not defined in the `opts` configuration object. If the function returns `false`, the unknown option is not added to `argv`.