

# **Yargs**

**Yargs be a node.js library fer hearties tryin' ter parse optstrings**

ci [NPM version](https://www.npmjs.com/package/yargs) [js-standard-style](http://standardjs.com/) [Coverage](https://github.com/yargs/yargs/blob/main/.nycrc) [Conventional Commits](https://conventionalcommits.org/) [Slack](http://devtoolscommunity.herokuapp.com)

## Description

Yargs helps you build interactive command line tools, by parsing arguments and generating an elegant user interface.

It gives you:

* commands and (grouped) options (my-program.js serve --port=5000).
* a dynamically generated help menu based on your arguments:

mocha [spec..]

Run tests with Mocha

Commands

mocha inspect [spec..] Run tests with Mocha [default]

mocha init <path> create a client-side Mocha setup at <path>

Rules & Behavior

--allow-uncaught Allow uncaught errors to propagate [boolean]

--async-only, -A Require all tests to use a callback (async) or

return a Promise [boolean]

* bash-completion shortcuts for commands and options.
* and [tons more](about:blank).

## Installation

Stable version:

npm i yargs

Bleeding edge version with the most recent features:

npm i yargs@next

## Usage

### Simple Example

#!/usr/bin/env node

const yargs = require('yargs/yargs')

const { hideBin } = require('yargs/helpers')

const argv = yargs(hideBin(process.argv)).argv

if (argv.ships > 3 && argv.distance < 53.5) {

console.log('Plunder more riffiwobbles!')

} else {

console.log('Retreat from the xupptumblers!')

}

$ ./plunder.js --ships=4 --distance=22

Plunder more riffiwobbles!

$ ./plunder.js --ships 12 --distance 98.7

Retreat from the xupptumblers!

Note: hideBin is a shorthand for [process.argv.slice(2)](https://nodejs.org/en/knowledge/command-line/how-to-parse-command-line-arguments/). It has the benefit that it takes into account variations in some environments, e.g., [Electron](https://github.com/electron/electron/issues/4690).

### Complex Example

#!/usr/bin/env node

const yargs = require('yargs/yargs')

const { hideBin } = require('yargs/helpers')

yargs(hideBin(process.argv))

.command('serve [port]', 'start the server', (yargs) => {

return yargs

.positional('port', {

describe: 'port to bind on',

default: 5000

})

}, (argv) => {

if (argv.verbose) console.info(`start server on :${argv.port}`)

serve(argv.port)

})

.option('verbose', {

alias: 'v',

type: 'boolean',

description: 'Run with verbose logging'

})

.parse()

Run the example above with --help to see the help for the application.

## Supported Platforms

### TypeScript

yargs has type definitions at [@types/yargs](https://github.com/DefinitelyTyped/DefinitelyTyped/tree/master/types/yargs).

npm i @types/yargs --save-dev

See usage examples in [docs](about:blank).

### Deno

As of v16, yargs supports [Deno](https://github.com/denoland/deno):

import yargs from 'https://deno.land/x/yargs/deno.ts'

import { Arguments } from 'https://deno.land/x/yargs/deno-types.ts'

yargs(Deno.args)

.command('download <files...>', 'download a list of files', (yargs: any) => {

return yargs.positional('files', {

describe: 'a list of files to do something with'

})

}, (argv: Arguments) => {

console.info(argv)

})

.strictCommands()

.demandCommand(1)

.parse()

### ESM

As of v16,yargs supports ESM imports:

import yargs from 'yargs'

import { hideBin } from 'yargs/helpers'

yargs(hideBin(process.argv))

.command('curl <url>', 'fetch the contents of the URL', () => {}, (argv) => {

console.info(argv)

})

.demandCommand(1)

.parse()

### Usage in Browser

See examples of using yargs in the browser in [docs](about:blank).

## Community

Having problems? want to contribute? join our [community slack](http://devtoolscommunity.herokuapp.com).

## Documentation

### Table of Contents

* [Yargs' API](about:blank)
* [Examples](about:blank)
* [Parsing Tricks](about:blank)
  + [Stop the Parser](about:blank)
  + [Negating Boolean Arguments](about:blank)
  + [Numbers](about:blank)
  + [Arrays](about:blank)
  + [Objects](about:blank)
  + [Quotes](about:blank)
* [Advanced Topics](about:blank)
  + [Composing Your App Using Commands](about:blank)
  + [Building Configurable CLI Apps](about:blank)
  + [Customizing Yargs' Parser](about:blank)
  + [Bundling yargs](about:blank)
* [Contributing](about:blank)

## Supported Node.js Versions

Libraries in this ecosystem make a best effort to track [Node.js' release schedule](https://nodejs.org/en/about/releases/). Here's [a post on why we think this is important](https://medium.com/the-node-js-collection/maintainers-should-consider-following-node-js-release-schedule-ab08ed4de71a).