# didYouMean.js - A simple JavaScript matching engine

[Available on GitHub](https://github.com/dcporter/didyoumean.js).

A super-simple, highly optimized JS library for matching human-quality input to a list of potential matches. You can use it to suggest a misspelled command-line utility option to a user, or to offer links to nearby valid URLs on your 404 page. (The examples below are taken from a personal project, my [HTML5 business card](http://dcporter.aws.af.cm/me), which uses didYouMean.js to suggest correct URLs from misspelled ones, such as [dcporter.aws.af.cm/me/instagarm](http://dcporter.aws.af.cm/me/instagarm).) Uses the [Levenshtein distance algorithm](https://en.wikipedia.org/wiki/Levenshtein_distance).

didYouMean.js works in the browser as well as in node.js. To install it for use in node:

npm install didyoumean

## Examples

Matching against a list of strings:

var input = 'insargrm'

var list = ['facebook', 'twitter', 'instagram', 'linkedin'];

console.log(didYouMean(input, list));

> 'instagram'

// The method matches 'insargrm' to 'instagram'.

input = 'google plus';

console.log(didYouMean(input, list));

> null

// The method was unable to find 'google plus' in the list of options.

Matching against a list of objects:

var input = 'insargrm';

var list = [ { id: 'facebook' }, { id: 'twitter' }, { id: 'instagram' }, { id: 'linkedin' } ];

var key = 'id';

console.log(didYouMean(input, list, key));

> 'instagram'

// The method returns the matching value.

didYouMean.returnWinningObject = true;

console.log(didYouMean(input, list, key));

> { id: 'instagram' }

// The method returns the matching object.

## didYouMean(str, list, [key])

* str: The string input to match.
* list: An array of strings or objects to match against.
* key (OPTIONAL): If your list array contains objects, you must specify the key which contains the string to match against.

Returns: the closest matching string, or null if no strings exceed the threshold.

## Options

Options are set on the didYouMean function object. You may change them at any time.

### threshold

By default, the method will only return strings whose edit distance is less than 40% (0.4x) of their length. For example, if a ten-letter string is five edits away from its nearest match, the method will return null.

You can control this by setting the "threshold" value on the didYouMean function. For example, to set the edit distance threshold to 50% of the input string's length:

didYouMean.threshold = 0.5;

To return the nearest match no matter the threshold, set this value to null.

### thresholdAbsolute

This option behaves the same as threshold, but instead takes an integer number of edit steps. For example, if thresholdAbsolute is set to 20 (the default), then the method will only return strings whose edit distance is less than 20. Both options apply.

### caseSensitive

By default, the method will perform case-insensitive comparisons. If you wish to force case sensitivity, set the "caseSensitive" value to true:

didYouMean.caseSensitive = true;

### nullResultValue

By default, the method will return null if there is no sufficiently close match. You can change this value here.

### returnWinningObject

By default, the method will return the winning string value (if any). If your list contains objects rather than strings, you may set returnWinningObject to true.

didYouMean.returnWinningObject = true;

This option has no effect on lists of strings.

### returnFirstMatch

By default, the method will search all values and return the closest match. If you're simply looking for a "good- enough" match, you can set your thresholds appropriately and set returnFirstMatch to true to substantially speed things up.

## License

didYouMean copyright (c) 2013-2014 Dave Porter.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License [here](http://www.apache.org/licenses/LICENSE-2.0).

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.