fill-range

Fill in a range of numbers or letters, optionally passing an increment or `step` *to use, or create a regex-compatible range with* `options.toRegex`

Please consider following this project's author, Jon Schlinkert, and consider starring the project to show your :heart: and support.

Install

Install with npm:

Usage

Expands numbers and letters, optionally using a `step` as the last argument. *(Numbers may be defined as JavaScript numbers or strings)*.

```
const fill = require('fill-range');
// fill(from, to[, step, options]);

console.log(fill('1',
        '10')); //=> ['1', '2', '3',
        '4', '5', '6', '7', '8', '9',
        '10']
console.log(fill('1', '10', { toRegex:
        true })); //=> [1-9]|10
```

Params

- `from`: **{String|Number}** the number or letter to start with
- `to`: **{String|Number}** the number or letter to end with
- `step`: **{String|Number|Object|Function}** Optionally pass a step to use.
- `options`: **{Object|Function}**: See all available options

Examples

By default, an array of values is returned.

Alphabetical ranges

```
console.log(fill('a', 'e')); //=> ['a',
        'b', 'c', 'd', 'e']
console.log(fill('A',
        'E')); //=> [ 'A', 'B', 'C',
        'D', 'E' ]
```

Numerical ranges

Numbers can be defined as actual numbers or strings.

```
console.log(fill(1, 5));      //=> [ 1,
        2, 3, 4, 5 ]
console.log(fill('1', '5')); //=> [ 1,
        2, 3, 4, 5 ]
```

Negative ranges

Numbers can be defined as actual numbers or strings.

```
console.log(fill('-5', '-1')); //=>
        [ '-5', '-4', '-3', '-2', '-1' ]
console.log(fill('-5', '5')); //=>
        [ '-5', '-4', '-3', '-2', '-1',
        '0', '1', '2', '3', '4', '5' ]
```

Steps (increments)

```
// numerical ranges with increments
console.log(fill('0', '25', 4)); //=>
        [ '0', '4', '8', '12', '16',
        '20', '24' ]
console.log(fill('0', '25', 5)); //=>
        [ '0', '5', '10', '15', '20',
        '25' ]
console.log(fill('0', '25', 6)); //=>
        [ '0', '6', '12', '18', '24' ]

// alphabetical ranges with increments
console.log(fill('a', 'z', 4)); //=>
        [ 'a', 'e', 'i', 'm', 'q', 'u',
        'y' ]
console.log(fill('a', 'z', 5)); //=>
        [ 'a', 'f', 'k', 'p', 'u', 'z' ]
console.log(fill('a', 'z', 6)); //=>
        [ 'a', 'g', 'm', 's', 'y' ]
```

Options

options.step

Type: `number` (formatted as a string or number)

Default: `undefined`

Description: The increment to use for the range. Can be used with letters or numbers.

Example(s)

```
// numbers
console.log(fill('1', '10', 2)); //=>
        [ '1', '3', '5', '7', '9' ]
console.log(fill('1', '10', 3)); //=>
        [ '1', '4', '7', '10' ]
console.log(fill('1', '10', 4)); //=>
        [ '1', '5', '9' ]

// letters
console.log(fill('a', 'z', 5)); //=>
        [ 'a', 'f', 'k', 'p', 'u', 'z' ]
console.log(fill('a', 'z', 7)); //=>
        [ 'a', 'h', 'o', 'v' ]
console.log(fill('a', 'z', 9)); //=>
        [ 'a', 'j', 's' ]
```


Author

Jon Schlinkert

- GitHub Profile
- Twitter Profile
- LinkedIn Profile

Please consider supporting me on Patreon, or start your own Patreon page!

License

This file was generated by verb-generate-readme, v0.8.0, on April 08, 2019.

Running and reviewing unit tests is a great way to get familiarized with a library and its API. You can install dependencies and run tests with the following command:

```
$ npm install && npm test
```

Building docs

(This project's readme.md is generated by verb, please don't edit the readme directly. Any changes to the readme must be made in the .verb.md readme template.)

To generate the readme, run the following command:

```
$ npm install -g verbose/verb#dev verb-
        generate-readme && verb
```

Contributors

Commits	Contributor
116	jonschlinkert
4	paulmillr
2	realityking
2	bluelovers
1	edorivai
1	wtgtybhertgeghgtwtg

options.strictRanges

Type: `boolean`

Default: `false`

Description: By default, `null` is returned when an invalid range is passed. Enable this option to throw a `RangeError` on invalid ranges.

Example(s)

The following are all invalid:

```
fill('1.1', '2');   // decimals not
        supported in ranges
fill('a', '2');     // incompatible
        range values
fill(1, 10, 'foo'); // invalid "step"
        argument
```

options.stringify

Type: `boolean`

Default: `undefined`

Description: Cast all returned values to strings. By default, integers are returned as numbers.

Example(s)

```
console.log(fill(1,
        5));                    //=>
        [ 1, 2, 3, 4, 5 ]
console.log(fill(1, 5, { stringify:
        true })); //=> [ '1', '2', '3',
        '4', '5' ]
```

options.toRegex

Type: `boolean`

Default: `undefined`

Description: Create a regex-compatible source string, instead of expanding values to an array.

Example(s)

```
// alphabetical range
console.log(fill('a', 'e', { toRegex:
        true })); //=> '[a-e]'
// alphabetical with step
console.log(fill('a', 'z', 3, {
        toRegex: true })); //=> 'a|d|g|
        j|m|p|s|v|y'
// numerical range
console.log(fill('1', '100', { toRegex:
        true })); //=> '[1-9]|[1-9]
        [0-9]|100'
// numerical range with zero padding
```



```
console.log(fill('000001', '100000', {
        toRegex: true }));
//=> '0{5}[1-9]|0{4}[1-9][0-9]|0{3}[1-9]
        [0-9]{2}|0{2}[1-9][0-9]{3}|
        0[1-9][0-9]{4}|100000'
```

options.transform

Type: `function`

Default: `undefined`

Description: Customize each value in the returned array (or string). *(you can also pass this function as the last argument to `fill()`).*

Example(s)

```
// add zero padding
console.log(fill(1, 5, value =>
        String(value).padStart(4,
        '0')));
//=> ['0001', '0002', '0003', '0004',
        '0005']
```

About

Contributing

Pull requests and stars are always welcome. For bugs and feature requests, please create an issue.

Running Tests