**DEVHINTS.IO**

<div style="text-align:right">Edit</div>

# Lodash cheatsheet

## Finding

```
_.filter(list, (n) => n % 2)    // → Array
_.find(list, (n) => n % 2)      // → item
_.findRight(list, ...)          // → item
```

Works for both arrays and objects.

## Iteration

```
_.forEach(list, (item, i) => ...)
_.forEachRight(list, ...)

_.map(list, ...)


_.every(users, (u) => u.active)  // → true|false (aka _.all)
_.any(users, ...)                // → true|false (aka _.some)
```

## Accessing

```
_.at([ abcd
_.at([ abcd
```

## Set/get

```
_.set(object
_.get(object
_.get(object
```

## ♯ Array

### Arrays

```
_.chunk([ abcd ], 2)        // → [ [ab], [cd] ]
_.compact(list)
```

### Filtering

```
_.drop([ abc
_.dropRight(
_.take([ abc
```

```
_.fill(Array(4), 'x')            // → [ 'x', 'x', 'x', 'x' ]        _.takeRight(
_.flatten                                                          _.slice([ ab
_.flattenDeep
                                                                  _.initial([
## Accessing                                                       _.rest([ abc

_.first([ abcdef ])              // → a                                       e(
_.last([ abcdef ])               // → f                                       e(
                                                                             e(
                                                                  _.dropWhile(
## Sets                                                            _.dropRightW

_.uniq()                                                                      [
_.difference([ abc ], [ bc ])       // → [ a    ]
_.intersection([ abc ], [ bcd ])    // → [  bc  ]                            is
_.union([ abc ], [ bcd ])           // → [ abcd ] (unique)

                                                                  ## Indexes
Array#concat()

                                                                  _.findIndex(
                                                                  _.findLastIn

                                                                  _.sortedInde
                                                                  _.sortedLast


                                                                  _.indexOf(li
```

# ♯ Functions

## Currying

```
greet = (greeting, name) => `${greeting}, ${name}!`


fn = _.partial(fn, 'hi')
fn('joe')     // → 'hi, joe!'

fn = _.partial(fn, 'joe')
fn('yo')      // → 'yo, joe!'
```

```
_.curry(greet)('hi')         // → function(name)
_.curryRight(greet)('joe')   // → function(greet)
```

# ♯ Decorating functions

## Throttling

```
_.throttle(fn)
_.debounce(fn)
```

## Limiting

```
_.before(5,
_.after(5, f
_.once(fn)
```

## Etc

```
_.wrap(_.escape, (name) => `hi ${name}`)
// same as doing `name = _.escape(name)`

_.delay(fn, 2000)

_.negate(fn)

_.memoize(fn)
_.memoize(fn, ...)
```

# ♯ Strings

## Capitalization

```
_.capitalize('hello world')   // → 'Hello world'
_.startCase('hello_world')    // → 'Hello World'
_.snakeCase('hello world')    // → 'hello_world'
_.kebabCase('hello world')    // → 'hello-world'
_.camelCase('hello world')    // → 'helloWorld'
```

## Padding

```
_.pad('abc',
_.padLeft('a
_.padLeft('a
_.padRight(.
```

## Trim

## Etc

```
_.repeat('-', 2)                // → '--'                          st
_.deburr('déjà vu')             // → 'deja vu'                     ('
_.trunc('hello world', 5)       // → 'hello...'                    t(


_.startsWith('abc', 'a')   // → true
_.endsWith('abc', 'c')     // → true
```

# ⌇ Objects

## Keys and values

```
_.keys(obj)
_.values(obj)
```

# ⌇ Chaining

## Chain and value

```
_([1, 2, 3])
  .reduce((total, n) => total + n)
  .map((n) => n * n)
  .tap(console.log)
  .thru((n) => n.reverse())
  .value()
```

▶       **0 Comments** for this cheatsheet.  Write yours!