View on GitHub

Swag

Give your handlebars.js templates some swag son!

Download this project as a .zip file Download this project as a tar.gz file

Swag

Swag is a growing collection of helpers for Handlebars.js. Give your handlebars.js templates some swag son!

Usage

```
<!-- Browser -->
<script src="../path_to/handlebars.js"></script>
<script src="../path_to/swag.js"></script>
<script>Swag.registerHelpers(Handlebars);</script>

// Node
Handlebars = require('handlebars');
Swag = require('swag');

Swag.registerHelpers(Handlebars);
```

Swag.registerHelpers

This method will register all Swag helpers with the instance of Handlebars you pass to it.

```
<script src="../path_to/handlebars.js"></script>
<script src="../path_to/swag.js"></script>
<script>Swag.registerHelpers(window.Handlebars);</script>
```

If you don't pass any instance of Handlebars, Swag will use the Handlebars instace available in the global context.

```
<script src="../path_to/handlebars.js"></script>
<script src="../path_to/swag.js"></script>
<script>
    // Will use window.Handlebars or Ember.Handlebars if you're using Ember.
    Swag.registerHelpers();
</script>
```

This method must be called in order to use Swag helpers in you Handlebars templates.

Strings

lowercase

Turns a string to lowercase.

Parameters: none.

Usage:

```
{{lowercase "BENDER SHOULD NOT BE ALLOWED ON TV"}}
```

```
bender should not be allowed on tv
```

uppercase

Turns a string to uppercase. Opposite of `{{lowercase}}`.

Parameters: none.

Usage:

```
{{uppercase "bender should not be allowed on tv"}}
```

```
BENDER SHOULD NOT BE ALLOWED ON TV
```

capitalizeFirst

Capitalizes the firs word in a string.

Parameters: none.

Usage:

```
{{capitalizeFirst "bender should not be allowed on TV"}}
```

```
Bender should not be allowed on TV
```

capitalizeEach

Capitalizes each word in a string.

Parameters: none.

Usage:

```
{{capitalizeEach "bender should NOT be allowed on TV"}}
```

```
Bender Should NOT Be Allowed On TV
```

titleize

Capitalizes all words within a string. Taken from the templating library Walrus by Jeremy Ruppel.

Parameters: none.

Usage:

```
{{titleize "Bender-should-Not-be-allowed_on_Tv."}}
```

```
Bender Should Not Be Allowed On Tv.
```

sentence

Capitalizes the first word of each sentence in a string and converts the rest of the sentence to lowercase.

Parameters: none.

Usage:

```
{{sentence "bender should NOT be allowed on TV. fry SHOULD be allowed on TV."}}
```

```
Bender should not be allowed on tv. Fry should be allowed on tv.
```

reverse

Reverses a string.

Parameters: none.

Usage:

```
{{reverse "bender should NOT be allowed on TV."}}
```

```
.VT no dewolla eb TON dluohs redneb
```

truncate

Truncates a string given a specified `length`, providing a custom string to denote an `omission`.

Parameters:

```
length [int] - The number of characters to keep (Required)
omission [string] - A string to denote an omission (Optional)
```

Usage:

```
{{truncate "Bender should not be allowed on tv." 31 "..."}}
```

```
Bender should not be allowed...
```

center

Centers a string using non-breaking spaces.

Parameters:

```
spaces [int] - The number of spaces. (Required)
```

Usage:

```
{{center "Bender should not be allowed on tv." 10}}
```

```
          Bender should not be allowed on tv.
```

newLineToBr

Converts new line characters `\n` to line breaks `<br>`.

Parameters: none.

Usage:

```
{{{newLineToBr "Bender \n should \n not \n be allowed on tv."}}}
```

```
Bender <br> should <br> not <br> be allowed on tv.
```

Collections

first

Returns the first item in a collection.

Parameters: none.

Usage:

```
collection = ['Amy Wong', 'Bender', 'Dr. Zoidberg', 'Fry', 'Hermes Conrad', 'Leela', 'Professor Farnsworth', 'Scruffy']
```

```
{{first collection}}
```

```
Amy Wong
```

withFirst

Use the first item in a collection inside a block.

Parameters: none.

Usage:

```
collection = ['Amy Wong', 'Bender', 'Dr. Zoidberg', 'Fry', 'Hermes Conrad', 'Leela', 'Professor Farnsworth', 'Scruffy']
```

```
{{#withFirst collection}}
    <p>{{this}} is smart.</p>
{{/withFirst}}
```

```
<p>Amy Wong is smart.</p>
```

last

Returns the last item in a collection. Opposite of `first`.

Parameters: none.

Usage:

```
collection = ['Amy Wong', 'Bender', 'Dr. Zoidberg', 'Fry', 'Hermes Conrad', 'Leela', 'Professor Farnsworth', 'Scruffy']
```

```
{{last collection}}
```

```
Scruffy
```

withLast

Use the last item in a collection inside a block. Opposite of `withFirst`.

Parameters: none.

Usage:

```
collection = ['Amy Wong', 'Bender', 'Dr. Zoidberg', 'Fry', 'Hermes Conrad', 'Leela', 'Professor Farnsworth', 'Scruffy']
```

```
{{#withLast collection}}
    <p>{{this}} is lazy.</p>
{{/withLast}}
```

```
<p>Scruffy is lazy.</p>
```

after

Returns all of the items in the collection after the specified count.

Parameters:

```
count [int] - How many items to omit from the beginning. (Required)
```

Usage:

```
collection = ['Amy Wong', 'Bender', 'Dr. Zoidberg', 'Fry', 'Hermes Conrad', 'Leela', 'Professor Farnsworth', 'Scruffy']
```

```
{{after collection 5}}
```

```
Leela, Professor Farnsworth, Scruffy
```

withAfter

Use all of the items in the collection after the specified count inside a block.

Parameters:

```
count [int] - How many items to omit from the beginning. (Required)
```

Usage:

```
collection = ['Amy Wong', 'Bender', 'Dr. Zoidberg', 'Fry', 'Hermes Conrad', 'Leela', 'Professor Farnsworth', 'Scruffy']
```

```
{{#withAfter collection 5}}
    {{titleize this}}
{{/withAfter}}
```

```
Leela Professor Farnsworth Scruffy
```

before

Returns all of the items in the collection before the specified count. Opposite of `after`.

Parameters:

```
count [int] - How many items to omit from the end. (Required)
```

Usage:

```
collection = ['Amy Wong', 'Bender', 'Dr. Zoidberg', 'Fry', 'Hermes Conrad', 'Leela', 'Professor Farnsworth', 'Scruffy']
```

```
{{before collection 5}}
```

```
Amy Wong, Bender, Dr. Zoidberg
```

withBefore

Use all of the items in the collection before the specified count inside a block. Opposite of `withAfter`.

Parameters:

```
count [int] - How many items to omit from the end. (Required)
```

Usage:

```
collection = ['Amy Wong', 'Bender', 'Dr. Zoidberg', 'Fry', 'Hermes Conrad', 'Leela', 'Professor Farnsworth', 'Scruffy']
```

```
{{#withBefore collection 5}}
    {{reverse this}}
{{/withBefore}}
```

```
gnoW ymA redneB grebdioZ .rD
```

join

Joins all elements of a collection into a string using a separator if specified.

Parameters:

```
separator [string] - A string to use as a separator between the items. (Optional)
```

Usage:

```
collection = ['Amy Wong', 'Bender', 'Dr. Zoidberg', 'Fry', 'Hermes Conrad', 'Leela', 'Professor Farnsworth', 'Scruffy']
```

```
{{join collection " & "}}
```

```
Amy Wong & Bender & Dr. Zoidberg & Fry & Hermes Conrad & Leela & Professor Farnsworth & Scruffy
```

sort

Returns the collection sorted.

Parameters:

```
none.
```

Usage:

```
collection = ['Dr. Zoidberg', 'Fry', 'Amy Wong', 'Professor Farnsworth', 'Bender', 'Hermes Conrad', 'Leela', 'Scruffy']
```

```
{{sort collection}}
```

```
Amy Wong, Bender, Dr. Zoidberg, Fry, Hermes Conrad, Leela, Professor Farnsworth, Scruffy
```

withSort

Uses the sorted collection inside the block.

Parameters:

```
field [string] - String name of the field or property to sort by. (Optional)
```

Usage:

```
collection = [
        name: 'Leela'
        deliveries: 8021
    ,
        name: 'Bender'
        deliveries: 239
    ,
        name: 'Fry'
        deliveries: -12
]
```

```
{{#withSort collection "deliveries"}}
    {{name}}: {{deliveries}} <br>
{{/withSort}}
```

```
Fry: -12
Bender: 239
Leela: 8021
```

length

Returns the length of the collection.

Parameters: none.

```
collection = ['Dr. Zoidberg', 'Fry', 'Amy Wong', 'Professor Farnsworth', 'Bender', 'Hermes Conrad', 'Leela', 'Scruffy']
```

```
{{length collection}}
```

```
8
```

lengthEqual

Conditionally render a block based on the length of a collection.

Parameters:

```
length [int] - The value to test against. (Required)
```

Usage:

```
collection = [
        name: 'Leela'
        deliveries: 8021
    ,
        name: 'Bender'
        deliveries: 239
    ,
        name: 'Fry'
        deliveries: -12
]


{{#lengthEqual collection 3}}
    There are 3 people in Planet Express.
{{else}}
    This is not Planet Express.
{{/lengthEqual}}
```

```
There are 3 people in Planet Express.
```

empty

Conditionally render a block if the collection is empty.

Parameters: none.

Usage:

```
collection = []
```

```
{{#empty collection}}
    Good news everyone!
{{else}}
    Bad news everyone!
{{/empty}}
```

```
Good news everyone!
```

any

Conditionally render a block if the collection isn't empty. Opposite of `empty`

Parameters: none.

Usage:

```
collection = ['Professor Farnsworth']
```

```
{{#any collection}}
    Good news everyone!
{{else}}
    Bad news everyone!
{{/any}}
```

```
Good news everyone!
```

inArray

Conditionally render a block if a specified value is in the collection.

Parameters:

```
value [string|int] - A value to test against. (Required)
```

Usage:

```
collection = ['Professor Farnsworth', 'Fry', 'Bender']
```

```
{{#inArray collection "Fry"}}
    I'm walking on sunshine!
{{else}}
    I'm walking on darkness.
{{/inArray}}
```

```
I'm walking on sunshine!
```

eachIndex

Renders a block using the array and each item's index.

Parameters: none.

Usage:

```
collection = ['Professor Farnsworth', 'Fry', 'Bender']
```

```
{{#eachIndex collection}}
    {{item}} is {{index}}
{{/eachIndex}}
```

```
Professor Farnsworth is 0, Fry is 1, Bender is 2
```

eachProperty

Uses the key and value of each property in an object to render a block.

Parameters: none.

Usage:

```
collection = fry: 3, bender: 120
```

```
{{#eachProperty collection}}
    {{key}} - {{value}}
{{/eachProperty}}
```

```
fry - 3 bender - 120
```

Math

add

Returns the sum of two numbers.

Parameters:

```
value [int] - The number to add to the expression. (Required)
```

Usage:

```
value = 5
```

```
{{add value 5}}
```

```
10
```

subtract

Returns the difference of two numbers. Opposite of `add`

Parameters:

```
value [int] - The number to subtract from the expression. (Required)
```

Usage:

```
value = 5
```

```
{{subtract value 5}}
```

```
0
```

divide

Returns the division of two numbers.

Parameters:

```
value [int] - The number to divide the expression. (Required)
```

Usage:

```
value = 5
```

```
{{divide value 5}}
```

```
1
```

multiply

Returns the multiplication of two numbers.

Parameters:

```
value [int] - The number to multiply the expression. (Required)
```

Usage:

```
value = 5
```

```
{{multiply value 5}}
```

```
25
```

floor

Returns the value rounded down to the nearest integer.

Parameters: none.

Usage:

```
value = 5.6
```

```
{{floor value}}
```

```
5
```

ceil

Returns the value rounded up to the nearest integer.

Parameters: none.

Usage:

```
value = 5.6
```

```
{{ceil value}}
```

```
6
```

round

Returns the value rounded to the nearest integer.

Parameters: none.

Usage:

```
value = 5.69
```

```
{{round value}}
```

```
6
```

Numbers

toFixed

Returns exactly `digits` after the decimal place. The number is rounded if necessary, and the fractional part is padded with zeros if necessary so that it has the specified length.

Parameters:

```
digits [int] - The number of digits to appear after the decimal point. (Optional)
```

Usage:

```
value = 5.53231
```

```
{{toFixed value 3}}
```

```
5.532
```

toPrecision

Returns the number in fixed-point or exponential notation rounded to `precision` significant digits.

Parameters:

```
precision [int] - The number of digits. If omitted, it returns the entire number (without any formatting). (Optional)
```

Usage:

```
value = 555.322
```

```
{{toPrecision value 4}}
```

```
555.3
```

toExponential

Returns the number in exponential notation with one digit before the decimal point, rounded to `fractions` digits after the decimal point.

Parameters:

```
fractions [int] - An integer specifying the number of digits after the decimal point. (Optional)
```

Usage:

```
value = 5
```

```
{{toExponential value 5}}
```

```
5.00000e+0
```

toInt

Returns an integer.

Parameters: none.

Usage:

```
value = '22.2abc'
```

```
{{toInt value}}
```

```
22
```

toFloat

Returns a floating point number.

Parameters: none.

Usage:

```
value = '22.2abc'
```

```
{{toFloat value}}
```

```
22.2
```

digitGrouping

Adds thousands separator to a number.

Parameters:

```
separator [string] - A string to use as a digit group separator. (Optional)
```

Usage:

```
value = 2222222
```

```
{{digitGrouping value}}
```

```
2,222,222
```

#

Comparisons

is

Conditionally render a block if the condition is true.

Parameters:

```
value [string|int] - the value to test against.
```

Usage:

```
number = 5
```

```
{{#is number 5}}
    Kiss my shiny metal ass!
{{else}}
    Never mind :(
{{/is}}
```

```
Kiss my shiny metal ass!
```

isnt

Conditionally render a block if the condition is false. Opposite of `is`.

Parameters:

```
value [string|int] - the value to test against.
```

Usage:

```
number = 5
```

```
{{#isnt number 5}}
    Kiss my shiny metal ass!
{{else}}
    Never mind :(
{{/isnt}}
```

```
Never mind :(
```

gt

Conditionally render a block if the value is greater than a given number.

Parameters:

```
value [string|int] - the value to test against.
```

Usage:

```
number = 5

{{#gt number 8}}
    Kiss my shiny metal ass!
{{else}}
    Never mind :(
{{/gt}}

Never mind :(
```

gte

Conditionally render a block if the value is greater or equal than a given number.

Parameters:

```
value [string|int] - the value to test against.
```

Usage:

```
number = 5

{{#gte number 5}}
    Kiss my shiny metal ass!
{{else}}
    Never mind :(
{{/gte}}

Kiss my shiny metal ass!
```

lt

Conditionally render a block if the value is less than a given number. Opposite of `gt`.

Parameters:

```
value [string|int] - the value to test against.
```

Usage:

```
number = 5

{{#lt number 3}}
    Kiss my shiny metal ass!
{{else}}
    Never mind :(
{{/lt}}

Never mind :(
```

lte

Conditionally render a block if the value is less or equal than a given number. Opposite of `gte`.

Parameters:

```
value [string|int] - the value to test against.
```

Usage:

```
number = 5

{{#lte number 5}}
    Kiss my shiny metal ass!
{{else}}
    Never mind :(
{{/lte}}

Kiss my shiny metal ass!
```

or

Conditionally render a block if one of the values is truthy.

Parameters:

```
values [string|int] - the values to test against.
```

Usage:

```
great = no
magnificent = true

{{#or great magnificent}}
    Kiss my shiny metal ass!
{{else}}
    Never mind :(
{{/or}}

Kiss my shiny metal ass!
```

and

Conditionally render a block if both values are truthy.

Parameters:

```
values [string|int] - the values to test against.
```

Usage:

```
great = true
magnificent = true

{{#and great magnificent}}
    Kiss my shiny metal ass!
{{else}}
    Never mind :(
{{/and}}

Kiss my shiny metal ass!
```

Dates

formatDate

Formats a date into a string given a format. Accepts any value that can be passed to `new Date()`. This helper is a port of the formatDate-js library by Michael Baldry.

Parameters:

```
format [string] - The format string, according to these tokens: (http://www.ruby-doc.org/core-1.9.3/Time.html#method-i-strftime) (Required)
```

Usage:

```
date = new Date()

{{formatDate date "%m/%d/%Y"}}
{{formatDate date "%I:%M%p"}}
{{formatDate date "%F"}}
{{formatDate date "%Y%m%dT%H%M%S%z"}}

07/26/2012
11:38PM
2012-07-26
20120726T233805-0004
```

now

Returns the current date.

Parameters:

```
format [string] - The format string, according to these tokens: http://www.ruby-doc.org/core-1.9.3/Time.html#method-i-strftime (Optional)
```

Usage:

```
{{now}}
{{now "%m/%d/%Y"}}

Thu Jul 26 2012 23:41:02 GMT-0400 (AST)
07/26/2012
```

timeago

Returns a human-readable time phrase from the given date.

Parameters: none.

Usage:

```
date = 'Thu Jul 22 2012 23:41:02 GMT-0400 (AST)'

{{timeago date}}

4 days ago
```

Inflections

inflect

Returns the plural or singular form of a word based on a count.

Parameters:

```
singular [string] - The singular form of the word. (Required)
plural [string] - The plural form of the word. (Required)
include [boolean] - whether or not to include the count before the word. (Optional)
```

Usage:

```
enemies = 0
friends = 1

{{inflect enemies "enemy" "enemies"}}
{{inflect friends "friend" "friends" true}}

enemies
1 friend
```

ordinalize

Turns a number into an ordinal string. Taken from the templating library Walrus by Jeremy Ruppel.

Parameters: none.

Usage:

```
{{ordinalize 3}}
{{ordinalize 1}}
{{ordinalize 22}}

3rd
1st
22nd
```

HTML

ul

Creates an unordered list.

Parameters:

```
hash [html attributes] - HTML attributes to use on the ul element. (Optional)
```

Usage:

```
collection = [
        name: 'Leela'
        deliveries: 8021
    ,
        name: 'Bender'
        deliveries: 239
    ,
        name: 'Fry'
        deliveries: 1
]

{{#ul collection class="deliveries-list"}}
    {{name}} - {{inflect deliveries "delivery" "deliveries" true}}
{{/ul}}

<ul class="deliveries-list">
    <li>
        Leela - 8021 deliveries
    </li>
    <li>
        Bender - 239 deliveries
    </li>
    <li>
        Fry - 1 delivery
    </li>
</ul>
```

ol

Same as the `ul` helper but creates and ordered list.

br

Returns `<br>` tags based on a count.

Parameters:

```
count [int] - The number of `br` elements to render. (Optional)
```

Usage:

```
{{br 5}}
```

```
<br><br><br><br><br>
```

Logging

log

Simple console.log()

Parameters: none.

Usage:

```
{{log "Hi console :)"}}
```

```
Hi console :)
```

debug

Simple console.debug() that shows the current context.

Parameters: none.

Usage:

```
collection = [
        name: 'Leela'
        deliveries: 8021
    ,
        name: 'Bender'
        deliveries: 239
    ,
        name: 'Fry'
        deliveries: 1
]
```

```
{{#withFirst collection}}
    {{debug name}}
{{/withFirst}}
```

```
Context: { deliveries: 8021, name: "Leela" }
Value: Leela
-----------------------------------------------
```

Miscellaneous

default

Provides a default or fallback value if a value doesn't exist.

Parameters:

```
defaultValue [string|int] - The default value to use.
```

Usage:

```
title = ''
```

```
{{default title "Not title available."}}
```

```
Not title available.
```

partial

Provides an easy way to register and use partials inside your templates. This helper only works if you define your templates as common.js modules, since it uses the common.js `require` function to find and register your templates with `Handlebars.registerPartial`. It was created with brunch in mind (which I use a lot), because brunch automatically wraps your scripts and templates in common.js modules to use in the browser.

Parameters:

```
name [string] - The name or path of the file in which your template is define. You can tell swag where your templates folder is by overriding Swag.Config.partialsPath. (Required)
```

```
data [int|string|collection] - The data you want to use inside the partial. (Optional)
```

Usage:

```
# Path to your templates from where yo override Swag.Config.partialsPath
# The path must finish with a forward slash '/'
Swag.Config.partialsPath = '../views/templates/'

collection = ['Professor Farnsworth', 'Fry', 'Bender']

# Your Partial (planet_express.hbs)
{{sort this}}

# Your template
<p>
    {{partial "planet_express" collection}}
</p>
```

```
<p>Bender, Fry, Professor Farnsworth</p>
```

Swag maintained by elving

Published with GitHub Pages