

Data types

numbers

e.g. 1.2, 13, 10px

strings of text, with and without quotes

e.g. "foo", 'bar', baz

colors

e.g. blue, #04a3f9, rgba(255, 0, 0, 0.5)

booleans

e.g. true, false

null

e.g. null

lists of values, separated by spaces or commas

e.g. 1.5em 1em 0 2em, Helvetica, Arial, sans-serif

maps from one value to another

e.g. (key1: value1, key2: value2)

String::Functions

`unquote($string)`

Removes quotes from a string.

`quote($string)`

Adds quotes to a string.

`str-length($string)`

Returns the number of characters in a string.

`str-insert($string, $insert, $index)`

Inserts `$insert` into `$string` at `$index`.

`str-index($string, $substring)`

Returns the index of the first occurrence of `$substring` in `$string`.

`str-slice($string, $start-at, [$end-at])`

Extracts a substring from `$string`.

`to-upper-case($string)`

Converts a string to upper case.

`to-lower-case($string)`

Converts a string to lower case.

Number :: Functions

`percent-age($number)`

Converts a unitless number to a percentage.

`round($number)`

Rounds a number to the nearest whole number.

`ceil($number)`

Rounds a number up to the next whole number.

`floor($number)`

Rounds a number down to the previous whole number.

`abs($number)`

Returns the absolute value of a number.

`min($numbers...)`

Finds the minimum of several numbers.

`max($numbers...)`

Finds the maximum of several numbers.

`random([$limit])`

Returns a random number.

Maps::functions

`map-get($map, $key)`

Returns the value in a map associated with a given key.

`map-merge($map1, $map2)`

Merges two maps together into a new map.

`map-remove($map, $keys...)`

Returns a new map with keys removed.

`map-keys($map)`

Returns a list of all keys in a map.

`map-values($map)`

Returns a list of all values in a map.

`map-has-key($map, $key)`

Returns whether a map has a value associated with a given key.

Maps::functions (cont)

`keywords($args)`

Returns the keywords passed to a function that takes variable arguments.

Reference : [Sass-lang:Maps Functions](#)

List : Functions

`length($list)`

Returns the length of a list.

`nth($list, $n)`

Returns a specific item in a list.

`join($list1, $list2, [$separator])`

Joins together two lists into one.

`append($list1, $val, [$separator])`

Appends a single value onto the end of a list.

`zip($lists...)`

Combines several lists into a single multidimensional list.

`index($list, $value)`

Returns the position of a value within a list.

`list-separator($list)`

Returns the separator of a list.

Misc :: Functions

`if($condition, $if-true, $if-false)`

Returns one of two values, depending on whether or not `$condition` is true.

`unique-id()`

Returns a unique CSS identifier.

Sources & Docs

[Sass:: Documentation](#)

[SassScript::Functions](#)



Selector Functions

`select or- nes t($ sel ect ors...)`

Nests selector beneath one another like they would be nested in the stylesheet.

`select or- app end ($s ele ctors...)`

Appends selectors to one another without spaces in between.

`select or- ext end ($s ele ctor, $extende e, $extende r)`

Extends \$extende e with \$extender within \$selector.

`select or- rep lac e($ sel ector, $original, $repla cement)`

Replaces \$original with \$replacement within \$selector.

`select or- uni fy($se lec tor1, $selec tor2)`

Unifies two selectors to produce a selector that matches elements matched by both.

`is- sup ers ele cto r($ super, $sub)`

Returns whether \$super matches all the elements \$sub does, and possibly more.

`simple -se lec tor s($ sel ector)`

Returns the simple selectors that comprise a compound selector.

`select or- par se($se lector)`

Parses a selector into the format returned by &.

Introspection :: Functions

`featur e- e xis ts($fe ature)`

Returns whether a feature exists in the current Sass runtime.

`variab le- exi sts($name)`

Returns whether a variable with the given name exists in the current scope.

Introspection :: Functions (cont)

`global -va ria ble -ex ist s($ name)`

Returns whether a variable with the given name exists in the global scope.

`functi on- exi sts($name)`

Returns whether a function with the given name exists.

`mixin- exi sts($name)`

Returns whether a mixin with the given name exists.

`inspec t($ value)`

Returns the string representation of a value as it would be represented in Sass.

`type- o f($ value)`

Returns the type of a value.

`unit($ number)`

Returns the unit(s) associated with a number.

`unitle ss($nu mber)`

Returns whether a number has units.

`compar abl e($ num ber1, $number 2)`

Returns whether two numbers can be added, subtracted, or compared.

`call($ name, $args...)`

Dynamically calls a Sass function.



By **Mist. GraphX** (Mist. GraphX)
cheatography.com/mist-graphx/

www.mister-graphx.com

Published 9th January, 2017.
 Last updated 9th January, 2017.
 Page 2 of 2.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>