# Cheatography

## SASS::Script Cheat Sheet
by Mist. GraphX (Mist. GraphX) via cheatography.com/1461/cs/3926/

## 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 occurance 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

`percentage($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{{/link}}

# Cheatography

## SASS::Script Cheat Sheet
by Mist. GraphX (Mist. GraphX) via cheatography.com/1461/cs/3926/

## Selector Functions

`selector-nest($selectors…)`

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

`selector-append($selectors…)`

Appends selectors to one another without spaces in between.

`selector-extend($selector, $extendee, $extender)`

Extends $extendee with $extender within $selector.

`selector-replace($selector, $original, $replacement)`

Replaces $original with $replacement within $selector.

`selector-unify($selector1, $selector2)`

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

`is-superselector($super, $sub)`

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

`simple-selectors($selector)`

Returns the simple selectors that comprise a compound selector.

`selector-parse($selector)`

Parses a selector into the format returned by &.

## Introspection :: Functions

`feature-exists($feature)`

Returns whether a feature exists in the current Sass runtime.

`variable-exists($name)`

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

## Introspection :: Functions (cont)

`global-variable-exists($name)`

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

`function-exists($name)`

Returns whether a function with the given name exists.

`mixin-exists($name)`

Returns whether a mixin with the given name exists.

`inspect($value)`

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

`type-of($value)`

Returns the type of a value.

`unit($number)`

Returns the unit(s) associated with a number.

`unitless($number)`

Returns whether a number has units.

`comparable($number1, $number2)`

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

`call($name, $args…)`

Dynamically calls a Sass function.