Function	Description & Example	Type
quote(string)	Adds quotes to string, and returns the result. Example: quote(Hello world!) Result: "Hello world!"	string
str-index(string, substring)	Returns the index of the first occurrence of the substring within string. Example: str-index("Hello world!", "H") Result: 1	string
str-insert(string, insert, index)	Returns string with insert inserted at the specified index position. Example: str-insert("Hello world!", " wonderful", 6) Result: "Hello wonderful world!"	string
str-length(string)	Returns the length of string (in characters). Example: str-length("Hello world!") Result: 12	string
str-slice(string, start, end)	Extracts characters from string; start at start and end at end, and returns the sli Example: str-slice("Hello world!", 2, 5) Result: "ello"	string
to-lower-case(string)	Returns a copy of string converted to lower case. Example: to-lower-case("Hello World!") Result: "hello world!"	string

to-upper-case(string)	Returns a copy of string converted to upper case. Example: to-upper-case("Hello World!") Result: "HELLO WORLD!"	string
unique-id()	Returns a unique randomly generated unquoted string (guaranteed to be unique Example: unique-id() Result: tyghefnsv	string
unquote(string)	Removes quotes around string (if any), and returns the result. Example: unquote("Hello world!") Result: Hello world!	string
abs(number)	Returns the absolute value of number. Example: abs(15) Result: 15 abs(-15) Result: 15	number
ceil(number)	Rounds number up to the nearest integer. Example: ceil(15.20) Result: 16	number

comparable(num1, num2)	Returns whether num1 and num2 are comparable. Example: comparable(15px, 10px) Result: true comparable(20mm, 1cm) Result: true comparable(35px, 2em) Result: false	number
floor(number)	Rounds number down to the nearest integer. Example: floor(15.80) Result: 15	number
max(number...)	Returns the highest value of several numbers. Example: max(5, 7, 9, 0, -3, -7) Result: 9	number
min(number...)	Returns the lowest value of several numbers. Example: min(5, 7, 9, 0, -3, -7) Result: -7	number
percentage(number)	Converts number to a percentage (multiplies the number with 100). Example: percentage(1.2) Result: 120	number
random()	Returns a random number between 0 and 1. Example: random() Result: 0.45673	number

random(number)	Returns a random integer between 1 and number. Example: random(6) Result: 4	number
round(number)	Rounds number to the nearest integer. Example: round(15.20) Result: 15 round(15.80) Result: 16	number
append(list, value, [separator])	Adds a single value to the end of the list. separator can be auto, comma, or spac Example: append((a b c), d) Result: a b c d append((a b c), (d), comma) Result: a, b, c, d	list
index(list, value)	Returns the index position for the value in list. Example: index(a b c, b) Result: 2 index(a b c, f) Result: null	list
is-bracketed(list)	Checks whether the list has square brackets. Example: is-bracketed([a b c]) Result: true is-bracketed(a b c) Result: false	list

join(list1, list2, [separator, bracketed])	Appends list2 to the end of list1. separator can be auto, comma, or space. auto i Example: join(a b c, d e f) Result: a b c d e f join((a b c), (d e f), comma) Result: a, b, c, d, e, f join(a b c, d e f, $bracketed: true) Result: [a b c d e f]	list
length(list)	Returns the length of the list. Example: length(a b c) Result: 3	list
list-separator(list)	Returns the list separator used, as a string. Can be either space or comma. Example: list-separator(a b c) Result: "space" list-separator(a, b, c) Result: "comma"	list
nth(list, n)	Returns the nth element in the list. Example: nth(a b c, 3) Result: c	list
set-nth(list, n, value)	Sets the nth list element to the value specified. Example: set-nth(a b c, 2, x) Result: a x c	list

zip(lists)	Combines lists into a single multidimensional list. Example: zip(1px 2px 3px, solid dashed dotted, red green blue) Result: 1px solid red, 2px dashed green, 3px dotted blue	list
map-get(map, key)	Returns the value for the specified key in the map. Example: $font-sizes: ("small": 12px, "normal": 18px, "large": 24px) map-get($font-sizes, "small") Result: 12px	map
map-has-key(map, key)	Checks whether map has the specified key. Returns true or false. Example: $font-sizes: ("small": 12px, "normal": 18px, "large": 24px) map-has-key($font-sizes, "big") Result: false	map
map-keys(map)	Returns a list of all keys in map. Example: $font-sizes: ("small": 12px, "normal": 18px, "large": 24px) map-keys($font-sizes) Result: "small", "normal, "large"	map
map-merge(map1, map2)	Appends map2 to the end of map1. Example: $font-sizes: ("small": 12px, "normal": 18px, "large": 24px) $font-sizes2: ("x-large": 30px, "xx-large": 36px) map-merge($font-sizes, $font-sizes2) Result: "small": 12px, "normal": 18px, "large": 24px, "x-large": 30px, "xx-large'	map

map-remove(map, keys...)	Removes the specified keys from map. Example: $font-sizes: ("small": 12px, "normal": 18px, "large": 24px) map-remove($font-sizes, "small") Result: ("normal": 18px, "large": 24px) map-remove($font-sizes, "small", "large") Result: ("normal": 18px)	map
map-values(map)	Returns a list of all values in map. Example: $font-sizes: ("small": 12px, "normal": 18px, "large": 24px) map-values($font-sizes) Result: 12px, 18px, 24px	map
is-superselector(super, sub)	Checks whether the super selector matches all the elements that sub matches. Example: is-superselector("div", "div.myInput") Result: true is-superselector("div.myInput", "div") Result: false is-superselector("div", "div") Result: true	selector
selector-append(selectors)	Appends the second (and third/fourth etc.) selector to the first selector. Example: selector-append("div", ".myInput") Result: div.myInput selector-append(".warning", "__a") Result: .warning__a	selector
selector-extend(selector, extendee, extender)		selector

selector-nest(selectors)	Returns a new selector containing a nested list of CSS selectors based on the list Example: selector-nest("ul", "li") Result: ul li selector-nest(".warning", "alert", "div") Result: .warning div, alert div	selector
selector-parse(selector)	Returns a list of strings contained in selector using the same format as the paren Example: selector-parse("h1 .myInput .warning") Result: ('h1' '.myInput' '.warning')	selector
selector-replace(selector, original, replace	Returns a new selector with the selectors specified in replacement in place of sel Example: selector-replace("p.warning", "p", "div") Result: div.warning	selector
selector-unify(selector1, selector2)	Returns a new selector that matches only elements matched by both selector1 a Example: selector-unify("myInput", ".disabled") Result: myInput.disabled selector-unify("p", "h1") Result: null	selector
simple-selectors(selectors)	Returns a list of the individual selectors in selectors. Example: simple-selectors("div.myInput") Result: div, .myInput simple-selectors("div.myInput:before") Result: div, .myInput, :before	selector
call(function, arguments...)	Calls a function with arguments, and returns the result.	introspection
content-exists()	Checks whether the current mixin was passed a @content block.	introspection

feature-exists(feature)	Checks whether feature is supported by the current Sass implementation. Example: feature-exists("at-error"); Result: true	introspection
function-exists(functionname)	Checks whether the specified function exists. Example: function-exists("nonsense") Result: false	introspection
get-function(functionname, css: false)	Returns the specified function. If css i true, it returns a plain CSS function instea	introspection
global-variable-exists(variablename)	Checks whether the specified global variable exists. Example: variable-exists(a) Result: true	introspection
inspect(value)	Returns a string representation of value.	introspection
mixin-exists(mixinname)	Checks whether the specified mixin exists. Example: mixin-exists("important-text") Result: true	introspection
type-of(value)	Returns the type of value. Can be number, string, color, list, map, bool, null, fun Example: type-of(15px) Result: number type-of(#ff0000) Result: color	introspection
unit(number)	Returns the unit associated with a number. Example: unit(15px) Result: px	introspection

unitless(number)	Checks whether the specified number has a unit associated with it. Example: unitless(15px) Result: false unitless(15) Result: true	introspection
variable-exists(variablename)	Checks whether the specified variable exists in the current scope. Example: variable-exists(b) Result: true	introspection
rgb(red, green, blue)	Sets a color using the Red-Green-Blue (RGB) model. An RGB color value is specif Example: rgb(0, 0, 255); // rendered as blue because the blue parameter is set to its high	color
rgba(red, green, blue, alpha)	Sets a color using the Red-Green-Blue-Alpha (RGBA) model. RGBA color values a Example: rgba(0, 0, 255, 0.3); // rendered as blue with opacity	color

hsl(hue, saturation, lightness)	Sets a color using the Hue-Saturation-Lightness (HSL) model - and represents a
	
	Example:
	hsl(120, 100%, 50%); // green
	hsl(120, 100%, 75%); // light green
	hsl(120, 100%, 25%); // dark green
	hsl(120, 60%, 70%); // pastel green
		color

hsla(hue, saturation, lightness, alpha)	Sets a color using the Hue-Saturation-Lightness-Alpha (HSLA) model. HSLA color
	
	Example:
	hsl(120, 100%, 50%, 0.3); // green with opacity
	hsl(120, 100%, 75%, 0.3); // light green with opacity
		color

grayscale(color)	Sets a gray color with the same lightness as color. Example: grayscale(#7fffd4); Result: #c6c6c6	color
complement(color)	Sets a color that is the complementary color of color. Example: complement(#7fffd4); Result: #ff7faa	color
invert(color, weight)	Sets a color that is the inverse or negative color of color. The weight parameter i Example: invert(white); Result: black	color

red(color)	Returns the red value of color as a number between 0 and 255.
	Example:
	red(#7fffd4);
	Result: 127
	red(red);
	Result: 255
		color

green(color)	Returns the green value of color as a number between 0 and 255.
	Example:
	green(#7fffd4);
	Result: 255
	green(blue);
	Result: 0
		color

blue(color)	Returns the blue value of color as a number between 0 and 255.
	Example:
	blue(#7fffd4);
	Result: 212
	blue(blue);
	Result: 255
		color

hue(color)	Returns the hue of color as a number between 0deg and 255deg. Example: hue(#7fffd4); Result: 160deg	color
saturation(color)	Returns the HSL saturation of color as a number between 0% and 100%. Example: saturation(#7fffd4); Result: 100%	color
lightness(color)	Returns the HSL lightness of color as a number between 0% and 100%. Example: lightness(#7fffd4); Result: 74.9%	color
alpha(color)	Returns the alpha channel of color as a number between 0 and 1. Example: alpha(#7fffd4); Result: 1	color
opacity(color)	Returns the alpha channel of color as a number between 0 and 1. Example: opacity(rgba(127, 255, 212, 0.5); Result: 0.5	color
mix(color1, color2, weight)	Creates a color that is a mix of color1 and color2. The weight parameter must be	color

adjust-hue(color, degrees)	Adjusts the color's hue with a degree from -360deg to 360deg. Example: adjust-hue(#7fffd4, 80deg); Result: #8080ff	color
adjust-color(color, red, green, blue, hue, s	Adjusts one or more parameters by the specified amount. This function adds or s Example: adjust-color(#7fffd4, blue: 25); Result:	color
change-color(color, red, green, blue, hue,	Sets one or more parameters of a color to new values. Example: change-color(#7fffd4, red: 255); Result: #ffffd4	color
scale-color(color, red, green, blue, satura	Scales one or more parameters of color.	color
rgba(color, alpha)	Creates a new color of color with the given alpha channel. Example: rgba(#7fffd4, 30%); Result: rgba(127, 255, 212, 0.3)	color
lighten(color, amount)	Creates a lighter color of color with an amount between 0% and 100%. The amount parameter increases the HSL lightness by that percent.	color
darken(color, amount)	Creates a darker color of color with an amount between 0% and 100%. The amount parameter decreases the HSL lightness by that percent.	color
saturate(color, amount)	Creates a more saturated color of color with an amount between 0% and 100%. The amount parameter increases the HSL saturation by that percent.	color
desaturate(color, amount)	Creates a less saturated color of color with an amount between 0% and 100%. The amount parameter decreases the HSL saturation by that percent.	color
opacify(color, amount)	Creates a more opaque color of color with an amount between 0 and 1. The amount parameter increases the alpha channel by that amount.	color
fade-in(color, amount)	Creates a more opaque color of color with an amount between 0 and 1. The amount parameter increases the alpha channel by that amount.	color

transparentize(color, amount)	Creates a more transparent color of color with an amount between 0 and 1. The amount parameter decreases the alpha channel by that amount.	color
fade-out(color, amount)	Creates a more transparent color of color with an amount between 0 and 1. The amount parameter decreases the alpha channel by that amount.	color