Funciones - SASS

Además de la posibilidad de definir nuestras propias funciones, SASS/SCSS viene con gran cantidad de funciones predefinidas a las que podremos llamar utilizando la sintaxis estándar de CSS y, en algunos casos, una sintaxis especial de SASS.

Estas funciones predefinidas están divididas en categorías que se corresponden en la mayoría de los casos con los tipos de variables definidos anteriormente:

- Funciones de Números, normalmente de carácter matemático
- Funciones de Strings, para crearlos, combinarlos, dividirlos y hacer búsquedas
- Funciones de Colores, para generarlos, mezclarlos o modificarlos
- Funciones de Listas, para acceder a ellas o modificarlas
- Funciones de Mapas, para trabajar con ellos
- Funciones de Selectores, para acceder al motor de selectores de Sass
- Funciones de Introspección, para mostrar detalles de la forma interna de trabajar de Sass

Creando una función personalizada

```
@function opacidad($color,$nivel)

    @return transparentize($color,$nivel)
```

Llamando una función

```
border: 1px solid opacidad(black, 0.9)
```

- ### Funciones de Números

- abs($number): Devuelve el valor absoluto del número dado.
- ceil($number): Devuelve el entero superior al número dado.
- comparable($number1, $number2): Devuelve el booleano true si las unidades son compatibles, o false en caso contrario.
- floor($number): Devuelve el entero inferior al número dado.
- max($number...): Devuelve el máximo valor a partir de uno o más números dados.
- min($number...): Devuelve el mínimo valor a partir de uno o más números dados.
- percentage($number): Convierte un número sin unidades dado (Normalmente entre 0 y 1) a porcentaje (es lo mismo que $number * 100)
- random($limit: null): Si $limit es null, devuelve un número aleatorio entre 0 y 1. Si $limit >= 1, devuelve un número aleatorio entre 1 y $limit.
- round($number): Devuelve el entero más cercano (por arriba o por abajo al número dado.
- unit($number): Devuelve un string con las unidades número dado (una cadena vacía si no tuviera unidades). Solo es recomendable su uso para debug.
- unitless($number): Devuelve el booleano true si el número dado no tiene unidades, false en caso contrario.

- ### Funciones de Strings

- quote($string): Devuelve el string dado entre comillas.
- str-index($string, $substring): Devuelve el índice de la primera ocurrencia de $substring en $string o null si no la encuentra.
- str-insert($string, $insert, $index): Devuelve una copia de $string con $insert añadido en $index. Si $index es mayor que la longitud de $string, $insert se añade al final, y si $index es menor que la longitud negativa de $string, $insert se añade al principio.
- str-length($string): Devuelve el número de caracteres en $string.
- str-slice($string, $start-at, $end-at: -1): Devuelve la parte de $string que desde el índice $start-at hasta el índice $end-at (ambos incluidos).
- to-upper-case($string): Devuelve una copia de $string con los caracteres ASCII convertidos a mayúsculas.
- to-lower-case($string): Devuelve una copia de $string con los caracteres ASCII convertidos a minúsculas.
- unique-id(): Devuelve un string aleatorio sin comillas válido como identificador único CSS.
- unquote($string): Devuelve el string dado sin comillas.

- ## Funciones de Colores

- adjust-color($color, $red: null, $green: null, $blue: null, $hue: null, $saturation: null, $lightness: null, $alpha: null): Incrementa o decrementa ciertas propiedades del color dado según los valores de entrada, que deben ser válidos y sin unidades.
- adjust-hue($color, $degrees): Incrementa la propiedad hue del color dado según el valor de $degrees, que debe ser un número entre -360deg y 360deg.
- alpha($color): Devuelve el canal alpha entre 0 y 1 del color dado. Es idéntica a opacity($color).
- blue($color): Devuelve el canal blue entre 0 y 255 del color dado. De igual forma podemos usar red() o green().
- hue($color): Devuelve el canal hue como ángulo entre 0 y 360 del color dado.
- saturation($color): Devuelve el canal saturation como porcentaje entre 0% y 100% del color dado. De igual forma podemos usar lightness($color).
- change-color($color, $red: null, $green: null, $blue: null, $hue: null, $saturation: null, $lightness: null, $alpha: null): Ajusta ciertas propiedades del color dado según los valores de entrada, que deben ser válidos y sin unidades.
- complement($color): Devuelve el color complementario RGB (que si se suman e cancelan entre si) del color dado. Es idéntica a adjust-hue($color, 180deg).
- darken($color, $amount): Hace $color más oscuro decrementando el canal lightness con un $amount entre 0% y 100%.
- desaturate($color, $amount): Hace $color más desaturado decrementando el canal saturation con un $amount entre 0% y 100%.
- grayscale($color): Devuelve un color en escala de grises con el mismo lightness que $color. Es idéntico a change-color($color, $saturation: 0%).
- hsl($hue $saturation $lightness): Devuelve el color con los valores dados. También se puede escribir como hsl($hue $saturation $lightness / $alpha), hsl($hue, $saturation, $lightness, $alpha: 1), hsla($hue $saturation $lightness), hsla($hue $saturation $lightness / $alpha) o hsla($hue, $saturation, $lightness, $alpha: 1).
- invert($color, $weight: 100%): Devuelve el inverso o negativo de $color.
- lighten($color, $amount): Hace $color más claro incrementando el canal lightness con un $amount entre 0% y 100%.
- mix($color1, $color2, $weight: 50%): Devuelve un color mezcla de los dos dados, siendo $weight opcional y el porcentaje de mezcla entre ambos (100% indica que solo se usa $color1).
- opacify($color, $amount): Hace $color más opaco decrementando el canal alpha con un $amount entre 0 y 1. Es idéntico a fade-in($color, $amount).
- rgb($red $green $blue): Devuelve el color con los valores dados. También se puede escribir como rgb($red $green $blue / $alpha), rgb($red, $green, $blue, $alpha: 1), rgb($color, $alpha), rgba($red $green $blue) rgba($red $green $blue / $alpha), rgba($red, $green, $blue, $alpha: 1) o rgba($color, $alpha).
- saturate($color, $amount): Hace $color más saturado incrementando el canal saturation con un $amount entre 0% y 100%.
- scale-color($color, $red: null, $green: null, $blue: null, $hue: null, $saturation: null, $lightness: null, $alpha: null): Escala ciertas propiedades del color dado según los valores de entrada, que deben ser porcentajes entre -100% y 100%.
- transparentize($color, $amount): Hace $color más transparente incrementando el canal alpha con un $amount entre 0 y 1. Es idéntico a fade-out($color, $amount).

- ## Funciones de Listas

- append($list, $val, $separator: auto): Devuelve una copia de $list con $val añadido al final de la misma.
- index($list, $value): Devuelve el índice de $value si está en la lista o null en caso contrario.
- is-bracketed($list): devuelve true si la lista tiene corchetes o false en caso contrario.
- join($list1, $list2, $separator: auto, $bracketed: auto): devuelve una nueva lista con los valores de $list1 seguido de los valores de $list2, con el $separator indicado y entre corchetes si $bracketed es true.
- length($list): devuelve la longitud de la lista.
- list-separator($list): devuelve el separador utilizado (espacio por defecto o coma).
- nth($list, $n): devuelve el elemento con índice $n de $list, contando desde el final si $n es negativo y devolviendo un error si no existe el índice.
- set-nth($list, $n, $value): Devuelve una copia de $list con el elemento de índice $n reemplazado por $value, , contando desde el final si $n es negativo y devolviendo un error si no existe el índice.
- zip($lists...): Combina cada lista en $lists en una única lista de sub-listas de longitud la más corta de las sub-listas y separada por comas.

- ## Funciones de Mapas/Objetos

- keywords($args): Devuelve un mapa con las keywords pasadas a un mixin o función con argumentos opcionales (estando estos en forma de lista de argumentos)
- map-get($map, $key): Devuelve el valor en $map asociado a $key, o null si no lo encuentra
- map-has-key($map, $key): Devuelve el booleano true si $map tiene un valor asociado a $key, false en caso contrario.
- map-keys($map): Devuelve una lista separada por comas con todas las claves en $map.
- map-merge($map1, $map2): Devuelve un nuevo mapa con todas las claves y valores de $map1 y $map2. También se puede usar para añadir sobre-escribir valores en $map1. Si los dos mapas tienen la misma clave, prevalece la de $map2.
- map-remove($map, $keys...): Devuelve una copia de $map sin los valores asociados de $keys (si alguna de las claves de $keys no existe, será ignorada).
- map-values($map): Devuelve una lista separada por comas con todos los valores en $map.

- ## Funciones de Selectores

- is-superselector($super, $sub): Devuelve el booleano true si $super encaja con al menos todos los elementos con los que selector $sub encaja.
- selector-append($selectors...): Combina $selectors sin usar combinadores descendentes, esto es, sin espacios entre ellos.
- selector-extend($selector, $extendee, $extender): Extiende $selector usando la regla @extend.
- selector-nest($selectors...): Combina $selectors como si estuvieran anidados.
- selector-replace($selector, $original, $replacement): Devuelve una copia de $selector con todas las instancias de $original reemplazadas por $replacement.
- selector-unify($selector1, $selector2): Devuelve un selector que encaja solo con elementos que encajen en $selector1 y en $selector2. Devuelve null si no encaja ninguno.
- simple-selectors($selector): Devuelve una lista de selectores simples en $selector.

- ## Funciones de Introspección

- call($function, $args...): Invoca $function con sus $args y devuelve el resultado.
- content-exists(): Devuelve el booleano true si un mixin ha pasado un bloque @content, o false en caso contrario (o si es llamada fuera del mixin).
- feature-exists($feature): Devuelve el booleano true si una cierta implementación de Sass soporta una $feature, o false en caso contrario.
- function-exists($name): Devuelve el booleano true si existe una función llamada $name, o false en caso contrario.
- get-function($name, $css: false): Devuelve la función llamada $name.
- global-variable-exists($name): Devuelve el booleano true si existe una variable global llamada $name, o false en caso contrario.
- inspect($value): Devuelve $value en formato string. Pensado para debug.
- mixin-exists($name): Devuelve el booleano true si existe un mixin llamado $name, o false en caso contrario.
- type-of($value): Devuelve el tipo de $value.
- variable-exists($name): Devuelve el booleano true si existe una variable llamada $name en el ámbito actual, o false en caso contrario.