

Elementos básicos

Variables

Las variables son elementos típicos de lenguajes de programación en los que guardamos valores para utilizarlos posteriormente.

DEFINICIÓN	USO
<i>\$nombre: expresión;</i>	<i>selector_css {</i>
	<i>...</i>
<i>\$nombre: valor;</i>	<i>regla_css: \$nombre...</i>
	<i>...</i>
	<i>}</i>

Ejemplo

```
//definición
//variables globales
$color-primary: orange;
$color-secondary: gray;
$color-tertiary: royalblue;

$font-lg: 40px;
$font-md: 30px;
$font-sm: 20px;

//uso
.nav {
  background-color: $color-primary;
}
```

Variables locales

```
.nav {
  $fondo-nav: yellow; //variable local
  background-color: $fondo-nav;
}
```

Valores por defecto

En algunos casos, como cuando estemos escribiendo librerías, puede que necesitemos que a la variable se le asigne un valor solo si no estaba definida o su valor era null. Para ello utilizamos el *flag* `!default`

```
$gris: #eee !default; //por defecto, suelen venir de una libreria
```

```
$gris:#ddd;  
$color-letra-pie:$gris; //previamente $gris está definida
```

Tipos de datos básicos

- Números: 48, 48px, 2rem, ...
- Cadenas: 'ledt', "ledt", left
- Colores: rgb(255, 0,0), hsl(0, 100%, 50%), #fff, #f0f0f0
- Booleanos: true, false
- Null
- Listas
- Mapas

Operadores

== y != se usan para comprobar si dos valores son iguales o no

+, -, *, / y % efectúan la correspondiente operación matemática entre los valores

<, <=, > y >= son los operadores de comparación

and, or y not tienen el comportamiento usual de los operadores booleanos

+ también se pueden usar para encadenar *strings*,

Prioridad de los operadores de mayor a menor

1. Los operadores unarios not, +, - y /
2. Los operadores matemáticos *, / y %
3. Los operadores matemáticos + y -
4. Los operadores de comparación <, <=, > y >=
5. Los operadores de igualdad == y !=
6. El operador booleano and
7. El operador booleano or

Referencias

<https://sass-lang.com/documentation>

Listas y mapas

Listas

Colección de valores, se accede por índice. Primer elemento = 1

```
$sizes:(40px, 80px, 160px);
```

```
.banner {  
  h1{  
    font-size: nth($sizes,2);  
  }  
}
```

Mapas

Colección de valores, se accede por clave

```
$breakpoints:(  
  "sm": 576,  
  "md": 768px,  
  "lg": 992px,  
  "xl": 1200px  
);  
  
.banner{  
  background-color: map-get($greyscale, "200" );  
}
```

Interpolación

En selectores

permiten crear variables cuyo contenido puede ser evaluado en tiempo real
para poder referenciar propiedades CSS o selectores de una forma mucho más flexible

```
$button-type: "error";  
$btn-color : #f00;  
button{  
  background-color: $btn-color;  
}
```

```

$button-type: ("error", "warning", "infor");
$btn-color : (red, yellow, blue);
//ahora, voy a crear tres clases para mis botones
.btn-#{nth($button-type,1)} {
    background-color: nth($btn-color,1);
}
.btn-#{nth($button-type,2)} {
    background-color: nth($btn-color,2);
}
.btn-#{nth($button-type,3)} {
    background-color: nth($btn-color,3);
}

```

Interpolación en el uso de funciones

```

$fondo : "../img/image1.jpg";

.container {

    background-image: url("#{ $fondo}");

}

```

Interpolación en el uso de comentarios

```

$autor: "DAW";

/*
    Web desarrollada por #{ $autor}
*/

```

Anidamiento

Anidamiento básico

```

//anidamiento básico
.nav{
    background-color: $color-primary;
    ul li{
        list-style-type: none;
        a{
            text-decoration: none;
        }
    }
}

```

Haciendo referencia al elemento padre: &

```
a {  
  color: $link-normal;  
  text-decoration: none;  
  &:hover {  
    color: $link-hover;  
  }  
  &:visited {  
    color: $link-visited;  
  }  
  &:active {  
    color: $link-active;  
  }  
}
```