

Funciones de LESS

Módulo 02

Implementación LESS

Implementación

LESS

Generaremos un archivo llamado mixins.less, dentro del mismo haremos lo siguiente cómo si fuese una clase,



```
estilos.css x estilos.less x index.html x mixins.less seccion.less x tipografia.less x

.bordes { border-radius: 10%; border: thin dotted lightblue;}
```



Ahora vamos a ir seccion.less y generamos la siguiente línea,

```
section {
  width: 30%;
  box-shadow: 1px 2px 4px rgba(0,0,0,0.5);
  padding: 1em;
  .bordes; }
```

Implementación

LESS

Siempre nuestro archivo estilos.less, para sumarle este nuevo archivo como veremos en imagen,

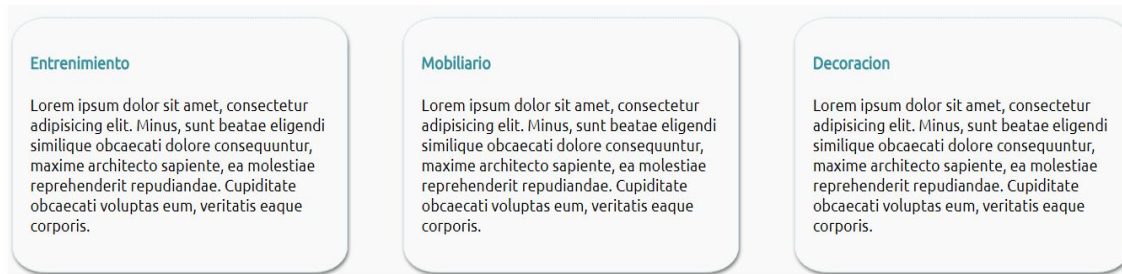
```
*{box-sizing: border-box; padding:2px;}  
@import "variables.less";  
@import "mixins.less";  
@import "tipografia.less";  
@import "seccion.less";
```



Implementación

LESS

El resultado será el siguiente en el navegador,



Pasemos a modificar un poco la línea de mixins.less por ejemplo,

```
.bordes(@grosor) { border-radius: 10%; border: thin dotted lightblue;}
```

Implementación

LESS

Y ahora podremos utilizarlo de la siguiente forma, para indicar donde se hará **dinámico el parámetro**,

```
.bordes(@grosor) { border-radius: 10%; border: @grosor dotted lightblue;}
```



Entonces en **seccion.less** podremos hacer lo siguiente,

```
section {  
  width: 30%;  
  box-shadow: 1px 2px 4px rgba(0,0,0,0.5);  
  padding: 1em;  
  .bordes(medium); }  
}
```

Implementación

LESS

Y utilizarlo cambiando el valor dinámico o parámetro por ejemplo,

```
#metricas { background: @colorFondo; min-height: 300; width: 100%; margin-bottom: 1em;
.bordes(thin);}
```



Inclusive pudiendo modificar varios elementos a la vez, por ejemplo también en este caso el color, en mixins.less haremos los siguiente,

```
.bordes(@grosor, @color) { border-radius: 10%; border: @grosor dotted @color;}
```

Implementación

LESS

Y en nuestro **seccion.less**, agregaremos tanto en section cómo en #metricas las siguientes líneas,

```
section {  
  width: 30%;  
  box-shadow: 1px 2px 4px rgba(0,0,0,0.5);  
  padding: 1em;  
  .bordes(medium, #0ff); }  
  
#metricas { background: @colorFondo; min-height: 300; width: 100%; margin-bottom: 1em;  
  .bordes(thick, #333);}
```

También se puede utilizar variables ya declaradas por ejemplo,

```
section {  
  width: 30%;  
  box-shadow: 1px 2px 4px rgba(0,0,0,0.5);  
  padding: 1em;  
  .bordes(medium, @colorSecundario); }
```



Implementación

LESS

Existen otras funciones interesantes para poder trabajar en cuanto a oscurecer o aclarar colores, por ejemplo,

- darken (oscurecer un color)
- lighten (aclarar un color)



Veamos algunos ejemplos de operaciones incluso matemáticas con estas funciones,

```
@color: blue;
lighten(@color, 10%); // Devuelve un color 10% más claro que @color
darken(@color, 10%); // Devuelve un color 10% más oscuro que @color

saturate(@color, 10%); // Devuelve un color 10% más saturado que @color
desaturate(@color, 10%); // Devuelve un color 10% menos saturado que @color

fadein(@color, 10%); // Devuelve un color 10% menos transparente que @color
fadeout(@color, 10%); // Devuelve un color 10% más transparente que @color
fade(@color, 50%); // Devuelve un color con 50% de transparencia

spin(@color, 10); // Devuelve un color con 10 grados más en el tono de @color
spin(@color, -10); // Devuelve un color con 10 grados menos en el tono de @color

mix(@color1, @color2); // Devuelve una mezcla entre los colores @color1 y @color2
```

Revisión

- Repasar los conceptos vistos de **LESS**.
- Trabajar con el **bonus track** debajo de todo para practicar los **elementos individualmente**.
- Ver todos los videos y materiales necesarios antes de continuar



¡Muchas gracias!

¡Sigamos trabajando!