

## UD 4\_4: EJEMPLOS SASS

### 1- PARTIENDO DE ESTE CSS GENERA EL SCSS

- Sigue las indicaciones propuestas
- Compílalo en Koala para conseguir en css de nuevo.

---

#### A. GENERA VARIABLES PARA LOS COLORES

```
body {
  padding: 1em 2em;
}

ul, ol {
  list-style: none;
  padding: 0;
}
li {
  text-align: center;
  line-height: 2;
  background: slategrey;
}
div {
  width: 12em;
  float: left;
  margin-right: 2em;
}
hr {
  clear: both;
  padding-top: 1em;
  border: 0;
  border-bottom: 1px solid grey;
}
.rojo {
  color:red;
}
```

---

#### B. AGRUPA REGLAS DE ELEMENTOS

```
.uno li:nth-child(1) {
  background: lightsteelblue;
}

.dos :nth-child(2) :nth-child(3) {
  background: lightsteelblue;
}
```

```
.tres :nth-child(odd) li:nth-child(-n+3) {
  background: lightsteelblue;
}
```

---

### C. GENERA UN MIXIN PARA ESTOS ELEMENTOS

```
.tres:before {
  content:"Soy previo al bloque 3";
  color:blue;}

#i1:after {
  content: "Soy posterior al bloque 2";
  color:green;
  border: 2;
}
```

## 2- TRADUCE A CSS

Traduce el siguiente código a css manual/mental-mente y compruébalo con koala.

SCSS	CSS
<pre>.blog {   .post &amp; p {     em {       color: #fff;     }   } }</pre>	<pre>.post .blog p em{   color:#fff; }</pre>
<pre>.blog {   .post p {     em &amp; {       color: #fff;     }   } }</pre>	<pre>em .blog .post p{   color: #fff; }</pre>
<pre>.blog {   .post p {     &amp;:hover em {       color: #fff;     }   } }</pre>	<pre>.blog .post p:hover em{   color: #fff; }</pre>
<pre>\$btn-bg-color: #ce4dd6; .menu {   .button {     background-color: \$btn-bg-color;     &amp;:hover{       background-color: lighten(\$btn- bg-color, 20%);     }     &amp;:active{</pre>	<pre>.menu .button{   background-color: #ce4dd6; } .menu .button:hover{   background-color: lighten(#ce4dd6, 20%); } .menu .button:active{</pre>

<pre> background-color: darken(\$btn- bg-color, 20%); } &amp; .disabled{ background-color: grayscale(\$btn-bg-color); } } } </pre>	<pre> background-color: darken (#ce4dd6, 20%); } .menu .button .disabled{ background-color: grayscale (#ce4dd6, 20%); } </pre>
--	--

### 3- ANALIZAR EL SIGUIENTE SITIO CON SASS

Analiza el siguiente sitio web (Tienes la solución SASS y Css)



Ejemplo SASS.mp4

#### A. LLAMADAS A UN MISMO MIXIN CON DIFERENTE NÚMERO DE ARGUMENTOS:

Archivo: `_background.scss`

Código1:

```
@include imagenFondo('../img/imagen02.jpg', 600px, top center );
```

Código2:

```
@include imagenFondo('../img/imagen03.jpg', 200px);
```

- Cambia de orden los argumentos del mixin `imagenfondo` y en consecuencia las llamadas que se le hacen desde `_background.scss`
- ¿Qué ocurre? ¿Por qué?

Que se cambian de lugar las imágenes, porque así lo indicas con la clase.

#### B. ANALIZA EL SIGUIENTE CÓDIGO DE `_BOTONES.SCSS`

```

@each $nombre_color, $color in (noticia, $verde), (servicio, $amarillo){
  .btn-#{ $nombre_color }{
    background-color:darken($color,10);
    &:hover{
      background-color: darken($color,20);
    }
  }
}

```

- ¿Cuál es su traducción a css? (Hazlo manual/mental-mente)

```

.btn #noticia {
  background-color:darken($verde,10);
}
.btn #noticia:hover{
  background-color: darken($verde,20);
}
.btn #servicio {

```

```

        background-color:darken($amarillo,10);
    }
    .btn # servicio:hover{
        background-color: darken($amarillo,20);
    }

```

### C. EXPLICA EL SIGNIFICADO (LO QUE HACE Y A QUÉ ELEMENTO AFECTA) EL SIGUIENTE CÓDIGO:

Archivo	Código	A qué elemento del html afecta	Conversion a css
_layout.scss	<pre> @include tablet{ @for \$i from 1 through \$numero- columnas{         .col-#{ \$i}{  width:100%/\$numero- columnas *\$i;         }     } </pre>	<pre> .col .col #1 .col #2 ... </pre>	<pre> .col-#1{ width:8.3%; } .col-#2{ width:16.6%; } ... .col-#12{ width:100%; } </pre>

### D. BUSCA INFORMACIÓN SOBRE EL SIGNIFICADO DEL SIGUIENTE CÓDIGO:

#### - Archivo \_mixins.scss

```

@mixin tablet {
    @media (min-width: #{ $tablet}) {
        @content
    }
}

```

#### - Llamadas desde el archivo \_navegacion.scss

```

h2{
    display:none;
    @include tablet{
        display:block;
    }
}
ul{
    @include displayFlex(column, center,center);
    @include tablet{
        @include displayFlex(row, center,center);
    }
    li{
        list-style: none;
    }
}

```

```
padding: 10px;
}
```

- Explicación del código y las llamadas.

En el texto de color rojo que pone @include tablet hace referencia a un @media de 768px de ancho mínimo:

```
@media (min-width: 768px) {

}
```

@include displayFlex(row, center,center) añade las siguientes líneas de código:

```
display:flex;
flex-direction: $direccion;
justify-content: $disposicion-eje1;
align-items:$disposicion-eje2;
flex-wrap: $wrap;
```

---

#### E. EN EL ARCHIVO \_SERVICIOS.SCSS HAY EL SIGUIENTE CÓDIGO:

```
li{
    margin-top: 20px;
    text-align: center;
    @include tablet{
        padding-right: 10px;
        &:last-child {
            padding-right:0;
        }
    }
}
```

- ¿A qué elemento se refiere el símbolo &?

```
@media (min-width: 768px){
    li:last-child{
        padding-right:0;
    }
}
```