# UD 4\_4: EJEMPLOS SASS

### Partiendo de este css genera el scss

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

#### 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;

}

#### 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;

}

#### 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;

}

### Traduce a css

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

|  |  |
| --- | --- |
| SCSS | CSS |
| .blog {  .post & p {  em {  **color**: #fff;  }  }  } | .blog .post .post p em  {  color: #fff;  } |
| .blog {  .post p {  em & {  **color**: #fff;  }  }  } | .blog .post p em p  {  color: #fff;  } |
| .blog {  .post p {  &:hover em {  **color**: #fff;  }  }  } | .blog .post p:hover em  {  color: #fff;  } |
| $btn-bg-color: #ce4dd6;  .menu {  .button {  **background-color**: $btn-bg-color;  &:hover{  **background-color**: lighten($btn-bg-color, 20%);  }  &:active{  **background-color**: darken($btn-bg-color, 20%);  }  &.disabled{  **background-color**: grayscale($btn-bg-color);  }  }  } | .menu .button  {  background-color: #ce4dd6;  }  .menu .button:hover  {  background-color: #e5a0e9;  }  .menu .button:active  {  background-color: #93239a;  }  .menu .button.disabled  {  background-color: #929292;  } |

### Analizar el siguiente sitio con SASS

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



#### 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?

Todo sigue igual

#### 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: #3c641c;

}

.btn-noticia:hover {

background-color: #243c11;

}

.btn-servicio {

background-color: #cc9500;

}

.btn-servicio:hover {

background-color: #997000;

}

#### 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 | @include tablet{  @for $i from 1 through $numero-columnas{  .col-#{$i}{  width:100%/$numero-columnas \*$i;  }  } | Clases llamadas col-n | .col-1 {  width: 8.3333333333%;  }  .col-2 {  width: 16.6666666667%;  }  .col-3 {  width: 25%;  }  .col-4 {  width: 33.3333333333%;  }  … |

#### 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.

Los títulos h2 a partir de 768px tienen display: block

Las listas ul se les da el display flex dirección vertical y se alinea al centro llamando a displayFlex.

Los elementos li dentro de ul tienen el formato: list-style: none; padding: 10px;

#### 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 &?

Al elemento: li