

Ejercicios para practicar SASS

Directivas de Control

ENUNCIADO	3
EJERCICIOS	3
@at-root	3
Ejercicio 1	3
Ejercicio 2	3
Ejercicio 3	4
@debug	4
Ejercicio 4	4
Ejercicio 5	4
Ejercicio 6	4
@warn	5
Ejercicio 7	5
Ejercicio 8	5
Ejercicio 9	5
@if	5
Ejercicio 10	5
Ejercicio 11	5
Ejercicio 12	5
@for	6
Ejercicio 13	6
Ejercicio 14	6
Ejercicio 15	6
@each	6
Ejercicio 16	6
Ejercicio 17	6
Ejercicio 18	6
@while	7
Ejercicio 19	7
Ejercicio 20	7
Ejercicio 21	7
SOLUCIONES A LOS EJERCICIOS	8
Solución del Ejercicio 1	8
Solución del Ejercicio 2	8
Solución del Ejercicio 3	8
Solución del Ejercicio 4	9
Solución del Ejercicio 5	9
Solución del Ejercicio 6	9

Solución del Ejercicio 7	9
Solución del Ejercicio 8	9
Solución del Ejercicio 9	10
Solución del Ejercicio 10	10
Solución del Ejercicio 11	10
Solución del Ejercicio 12	11
Solución del Ejercicio 13	11
Solución del Ejercicio 14	11
Solución del Ejercicio 15	12
Solución del Ejercicio 16	12
Solución del Ejercicio 17	12
Solución del Ejercicio 18	12
Solución del Ejercicio 19	13
Solución del Ejercicio 20	13
Solución del Ejercicio 21	13

ENUNCIADO

Convierte los siguientes códigos CSS a SCSS haciendo uso de los operadores SassScript:

EJERCICIOS

@at-root

Las directivas **@at-root** hacen que una o más reglas se generen en la raíz de la hoja de estilos en vez de anidarse en sus selectores. Por defecto la regla **@at-root** simplemente excluye todos los selectores. No obstante, también es posible modificar su comportamiento para que salga o no de cualquier directiva **@media** en la que se encuentre esa regla.

Ejercicio 1

Incluye el segundo body dentro de **@media** usando **@at-root**.

```
@media print {  
    body {  
        padding: left;  
    }  
}  
  
body {  
    padding: 20px;  
}
```

Ejercicio 2

```
h1 {  
    color: #A52A2A;  
    background-color: #DEB887;  
}  
h2 {  
    font-size: 22px;  
    font-style: italic;  
    color: #1E90FF;  
}
```

Ejercicio 3

```
h1 {
    color: #ff8000;
}
h2 {
    color: #A52A2A;
    background-color: #DEB887;
}
p {
    font-size: 25px;
    font-style: italic;
    color: #666600;
}
```

@debug

La regla **@debug** muestra por la consola el valor de la expresión **SassScript** indicada. Se trata de una regla útil para depurar hojas de estilos.

Ejercicio 4

Usa **@debug** para que por consola parezca:

Line 1 DEBUG: 22em

Ejercicio 5

Usa **@debug** para que por consola parezca:

Line 2 DEBUG: 900px

Ejercicio 6

Usa **@debug** para que por consola parezca:

Line 7 DEBUG: <color; #bdc3c7>

Line 8 DEBUG: 30px

@warn

La directiva **@warn** es una herramienta útil en **SCSS** para emitir advertencias durante la compilación. Puede ser útil para alertar al desarrollador sobre posibles problemas en el código, como valores inesperados o errores de sintaxis.

Ejercicio 7

Haz uso de un mixin para que evalúe el color, si este es rojo saca un aviso: "El color es rojo. Ten cuidado." y el mixin se aplica al body para que salga:

```
body {  
    color: red;  
}
```

Ejercicio 8

Haz uso de un mixin para que evalúe el tamaño de la fuente y si es menor que 12px de el siguiente aviso: "El tamaño de fuente es muy pequeño. Debería ser mayor a 12px. Aplica a h1:

```
h1 {  
    font-size: 10px;  
}
```

Ejercicio 9

Haz uso de un mixin para que evalúe el ancho de la fuente (weight); de manera que si es menor de 400px de el aviso: "El peso de la fuente es muy ligero. Debería ser mayor a 400."; Aplica a h2.

```
h2 {  
    font-weight: 300;  
}
```

@if

Ejercicio 10

Crea un mixin que acepte un argumento para el tamaño de la fuente. Si el tamaño es mayor que 36px, el color del texto debe ser rojo. De lo contrario, debe ser verde.

Ejercicio 11

Crea un mixin que acepte un argumento para el ancho de un elemento. Si el ancho es mayor que 960px, el elemento debe tener un margen automático. De lo contrario, no debe tener margen.

Ejercicio 12

Crea un mixin que acepte un argumento para el color de fondo de un botón. Si el color es claro, el color del texto debe ser oscuro. De lo contrario, debe ser claro.

@for

Ejercicio 13

Crea un mixin que genere clases para márgenes con un bucle @for.

Ejercicio 14

Crea un mixin que genere clases para paddings con un bucle @for.

Ejercicio 15

Crea un mixin que genere clases para diferentes tamaños de texto con un bucle @for.

@each

Ejercicio 16

Crea un mixin que genere clases de color de fondo para una lista de colores con un bucle @each.

Ejercicio 17

Crea un mixin que genere clases de color de texto para una lista de colores con un bucle @each.

Ejercicio 18

Crea un mixin que genere clases para una lista de tamaños de fuente con un bucle @each.

@while

Ejercicio 19

Crea un mixin que genere clases para márgenes con un bucle @while.

Ejercicio 20

Crea un mixin que genere clases para paddings con un bucle @while.

Ejercicio 21

Crea un mixin que genere clases para diferentes tamaños de texto con un bucle @while.

SOLUCIONES A LOS EJERCICIOS

Solución del Ejercicio 1

```
@media print {  
  body {  
    padding: left;  
    @at-root (without:media) {  
      padding: 20px;  
    }  
  }  
}
```

Solución del Ejercicio 2

```
h1 {  
  color: #A52A2A;  
  background-color: #DEB887;  
  @at-root {  
    h2 {  
      font-size: 22px;  
      font-style: italic;  
      color: #1E90FF;  
    }  
  }  
}
```

Solución del Ejercicio 3

```
h1 {  
  color: #ff8000;  
  @at-root {  
    h2 {  
      color: #A52A2A;  
      background-color: #DEB887;  
    }  
    p {  
      font-size: 25px;  
      font-style: italic;  
      color: #666600;  
    }  
  }  
}
```


Solución del Ejercicio 4

```
@debug 10em + 12em;
```

Solución del Ejercicio 5

```
$min-width:900px;  
@debug "minimum width:#{ $min-width}";
```

Solución del Ejercicio 6

```
$font-sizes: 10px + 20px;  
$style: (  
  color: #bdc3c7  
);  
  
.container {  
  @debug $style;  
  @debug $font-sizes;  
}
```

Solución del Ejercicio 7

```
$color: red;  
  
@mixin color-warning($color) {  
  @if $color == red {  
    @warn "El color es rojo. Ten cuidado.";  
  }  
  color: $color;  
}  
  
body {  
  @include color-warning($color);  
}
```

Solución del Ejercicio 8

```
@mixin font-size($size) {  
  @if $size < 12px {  
    @warn "El tamaño de fuente es muy pequeño. Debería ser mayor a  
12px.";  
  }  
}
```

```

    font-size: $size;
}

h1 {
    @include font-size(10px);
}

```

Solución del Ejercicio 9

```

@mixin font-weight($weight) {
    @if $weight < 400 {
        @warn "El peso de la fuente es muy ligero. Debería ser mayor a
400.";
    }
    font-weight: $weight;
}

h2 {
    @include font-weight(300);
}

```

Solución del Ejercicio 10

```

@mixin tamaño-fuente($tamaño) {
    font-size: $tamaño;
    @if $tamaño > 36px {
        color: red;
    } @else {
        color: green;
    }
}

p {
    @include tamaño-fuente(40px);
}

```

Solución del Ejercicio 11

```

@mixin ancho-elemento($ancho) {
    width: $ancho;
    @if $ancho > 960px {
        margin: auto;
    } @else {
        margin: 0;
    }
}

```

```
div {
  @include ancho-elemento(1000px);
}
```

Solución del Ejercicio 12

```
@mixin boton($color-fondo) {
  background-color: $color-fondo;
  @if lightness($color-fondo) > 70 {
    color: black;
  } @else {
    color: white;
  }
}

button {
  @include boton(#f5f5f5);
}
```

Solución del Ejercicio 13

```
@mixin margenes {
  @for $i from 1 through 10 {
    .margen-#{ $i } {
      margin: #{ $i * 10 }px;
    }
  }
}

@include margenes;
```

Solución del Ejercicio 14

```
@mixin paddings {
  @for $i from 1 through 10 {
    .padding-#{ $i } {
      padding: #{ $i * 5 }px;
    }
  }
}

@include paddings;
```

Solución del Ejercicio 15

```
@mixin tamaños-texto {
```

```

@for $i from 1 through 10 {
  .texto-#{ $i } {
    font-size: #{ $i + 10 }px;
  }
}

```

```

@include tamaños-texto;

```

Solución del Ejercicio 16

```

@mixin colores-fondo($colores) {
  @each $color in $colores {
    .fondo-#{str-slice(inspect($color), 2)} {
      background-color: $color;
    }
  }
}

```

```

@include colores-fondo(#ff0000, #00ff00, #0000ff);

```

Solución del Ejercicio 17

```

@mixin colores-texto($colores) {
  @each $color in $colores {
    .texto-#{str-slice(inspect($color), 2)} {
      color: $color;
    }
  }
}

```

```

@include colores-texto(#ff0000, #00ff00, #0000ff);

```

Solución del Ejercicio 18

```

@mixin tamaños-fuente($tamaños) {
  @each $tamaño in $tamaños {
    .fuente-#{str-slice(inspect($tamaño), 2)} {
      font-size: $tamaño;
    }
  }
}

```

```

@include tamaños-fuente(12px, 14px, 16px);

```

Solución del Ejercicio 19

```
@mixin margenes {  
  $i: 1;  
  @while $i <= 10 {  
    .margen-#{ $i } {  
      margin: #{ $i * 10 }px;  
    }  
    $i: $i + 1;  
  }  
}  
  
@include margenes;
```

Solución del Ejercicio 20

```
@mixin paddings {  
  $i: 1;  
  @while $i <= 10 {  
    .padding-#{ $i } {  
      padding: #{ $i * 5 }px;  
    }  
    $i: $i + 1;  
  }  
}  
  
@include paddings;
```

Solución del Ejercicio 21

```
@mixin tamaños-texto {  
  $i: 1;  
  @while $i <= 10 {  
    .texto-#{ $i } {  
      font-size: #{ $i * 10 }px;  
    }  
    $i: $i + 1;  
  }  
}  
  
@include tamaños-texto;
```