

EXERCISES QUE TRABAJAREMOS EN LA CUE

- EXERCISE 1: DISEÑOS ADAPTATIVOS: UTILIZANDO MEDIA QUERIES
- EXERCISE 2: PATRONES DE DISEÑO RESPONSIVO.
- EXERCISE 3: CREANDO UN CONVERSOR DE UNIDADES DE MEDIDA.

EXERCISE 1: DISEÑOS ADAPTATIVOS: UTILIZANDO MEDIA QUERIES

En el siguiente ejercicio aprenderemos a utilizar el elemento **media query**.

Haremos uso de este elemento para que el fondo de nuestra página web cambie al variar el tamaño de la pantalla.

Primero, crearemos un documento HTML con su estructura base sin ningún código en su body, y con un documento style.css agregado en su head.

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="UTF-8">
5     <title>Cambio de color del fondo</title>
6     <link rel="stylesheet" href="assets/css/style.css">
7   </head>
8   <body>
9   </body>
10 </html>
```

A partir de este punto, solo trabajaremos sobre el archivo **style.css**. En primer lugar, lo que haremos es determinar los puntos de quiebre o **breakpoints** que utilizaremos para ir cambiando el color del fondo, para esto, nos guiaremos de los **breakpoints** propuestos por W3Schools.

```
1 /* Dispositivos extra pequeños (teléfonos, 600px y menos) */
2
3 @media only screen and (max-width: 600px) {}
4
5
6 /* Dispositivos pequeños (tablets verticales y teléfonos grandes,
7 600px y más) */
8
9 @media only screen and (min-width: 600px) {}
10
11
12 /* Dispositivos medianos (tablets horizontales, 768px y más) */
13
14 @media only screen and (min-width: 768px) {}
15
16
17 /* Dispositivos grandes (computadores portátiles/de escritorio, 992px
18 y más) */
19
20 @media only screen and (min-width: 992px) {}
21
22
23 /* Dispositivos extra grandes (grandes computadores portátiles y de
24 escritorio, 1200px y más) */
25
26 @media only screen and (min-width: 1200px) {}
```

Nótese que se utilizó la filosofía Mobile First para estructurar los **media queries**, por lo que el orden de los tamaños es ascendente.

Con estos **breakpoints** hemos cubierto cualquier tamaño de dispositivo, dándonos la posibilidad de personalizar el estilo de cada rango de medidas definido sin problemas.

Ahora vamos a establecer los colores a mostrarse:

1º En dispositivos extrapequeños, de **599px** y menos, el color de fondo será verde.

```
1 /* Dispositivos extra pequeños (teléfonos, 600px y menos) */
2
3 @media only screen and (max-width: 600px) {
4     body {
5         background-color: green;
6     }
7 }
```

2º Para dispositivos pequeños, entre **600px** y **767px**, el color de fondo será amarillo.

```
1 /* Dispositivos pequeños (tablets verticales y teléfonos grandes,
2 600px y más) */
3
4 @media only screen and (min-width: 600px) {
5     body {
6         background-color: yellow;
7     }
8 }
```

3º Para dispositivos medianos, entre **768px** y **991px**, el color de fondo será rojo.

```
1 /* Dispositivos medianos (tablets horizontales, 768px y más) */
2
3 @media only screen and (min-width: 768px) {
4     body {
5         background-color: red;
6     }
7 }
```

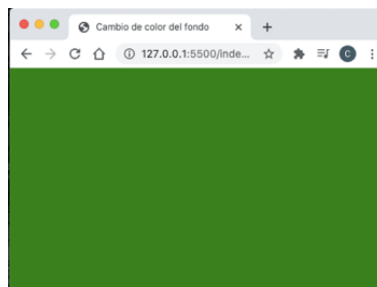
4º Para dispositivos grandes, entre **992px** y **1199px**, el color de fondo será azul.

```
1 /* Dispositivos grandes (computadores portátiles/de escritorio, 992px
2 y más) */
3
4 @media only screen and (min-width: 992px) {
5     body {
6         background-color: blue;
7     }
8 }
```

5º Para dispositivos extragrandes, sobre **1200px** de ancho, el color de fondo será naranja.

```
1 /* Dispositivos extra grandes (grandes computadores portátiles y de
2 escritorio, 1200px y más) */
3
4 @media only screen and (min-width: 1200px) {
5     body {
6         background-color: orange;
7     }
8 }
```

Ahora, probaremos el resultado en el navegador variando el tamaño de la ventana, o utilizando el Inspector de elementos para simular la carga de la página web en distintos dispositivos.



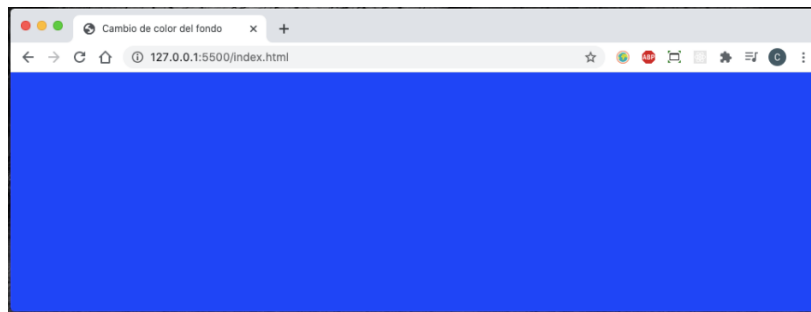
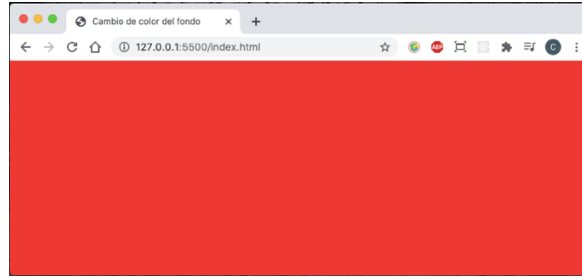


Imagen 1. Página web visualizada en distintos anchos de pantalla.

En este ejercicio hemos aprendido sobre diseños adaptativos y, específicamente, sobre la utilización de media queries. Te invitamos a seguir realizando las actividades correspondientes a este CUE, y a continuar avanzando en el curso.

EXERCISE 2: PATRONES DE DISEÑO RESPONSIVO.

En el siguiente ejercicio, aprenderemos a utilizar los distintos patrones de diseño responsivo. Para esto, realizaremos un ejemplo utilizando cada uno de ellos:

MOSTLY FLUID

Generaremos una página web sencilla, con 6 columnas dentro de un contenedor. Además, agregaremos una hoja de estilos `style.css` al `head`.

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <title>Mostly fluid</title>
5     <link rel="stylesheet" href="assets/css/style.css">
6   </head>
7
8   <body>
9
10    <div class="container">
11      <div class="columna1"></div>
12      <div class="columna2"></div>
13      <div class="columna3"></div>
14      <div class="columna4"></div>
15      <div class="columna5"></div>
16      <div class="columna6"></div>
17    </div>
18
19  </body>
20 </html>
```

Ahora, le agregaremos estilo a las columnas para que se conviertan en rectángulos de distintos colores.

```
1 .columna1,
2 .columna2,
3 .columna3,
4 .columna4,
5 .columna5,
6 .columna6 {
7     height: 200px;
8     width: 100%;
9 }
10
11 .columna1 {
12     background-color: antiquewhite;
13 }
14
15 .columna2 {
16     background-color: aqua;
17 }
18
19 .columna3 {
20     background-color: blueviolet;
21 }
22
23 .columna4 {
24     background-color: lightsalmon;
25 }
26
27 .columna5 {
28     background-color: palevioletred;
29 }
30
31 .columna6 {
32     background-color: cadetblue;
33 }
```

Si revisamos la página resultante, nos encontraremos con 6 rectángulos de distintos colores ordenados verticalmente, uno debajo del otro.



Imagen 1. Resultado sin agregar patrón `mostly fluid`.

Luego, es necesario indicarle al navegador que queremos que siga el patrón de diseño `mostly fluid`, para lo que agregaremos las siguientes líneas de código al archivo CSS.

```
1 /* Mostly Fluid */
2
3 .container {
4     display: -webkit-flex;
5     display: flex;
6     -webkit-flex-flow: row wrap;
7     flex-flow: row wrap;
8 }
9
10 @media (min-width: 600px) {
11     .columna2,
12     .columna3,
13     .columna4,
14     .columna5,
15     .columna6 {
16         width: 50%;
17     }
18 }
19
20 @media (min-width: 800px) {
21     .columna1 {
```



```
22         width: 60%;  
23     }  
24     .columna2 {  
25         width: 40%;  
26     }  
27     .columna3,  
28     .columna4,  
29     .columna5,  
30     .columna6 {  
31         width: 25%;  
32     }  
33     .container {  
34         width: 800px;  
35         margin-left: auto;  
36         margin-right: auto;  
37     }  
38 }
```

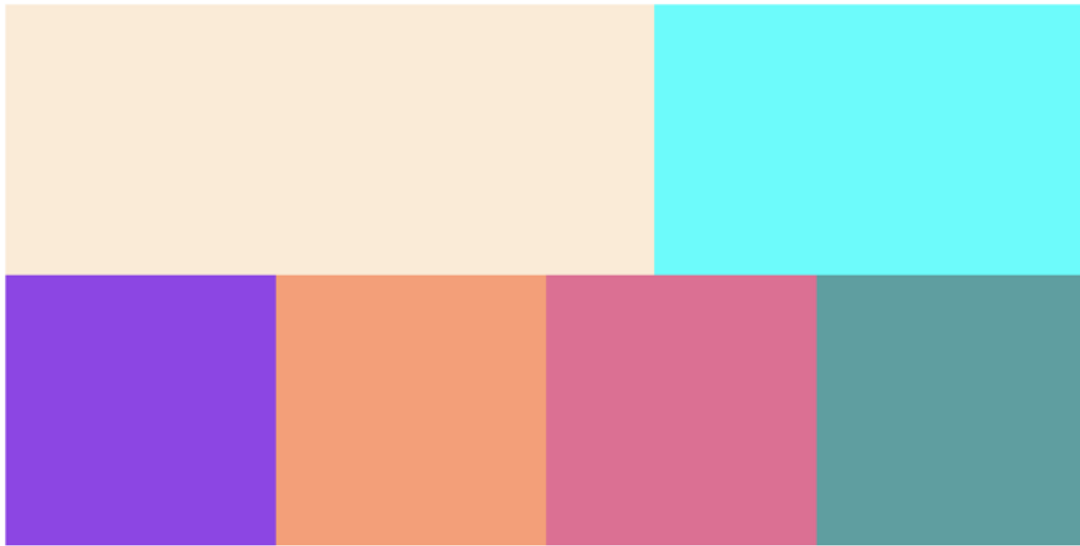


Imagen 2. Resultado con **mostly fluid** en una pantalla de gran tamaño (sobre **800px**).

Al parecer todo funciona correctamente. Ahora, probaremos con un tamaño de pantalla menor para corroborar que los elementos estén como se indicó.

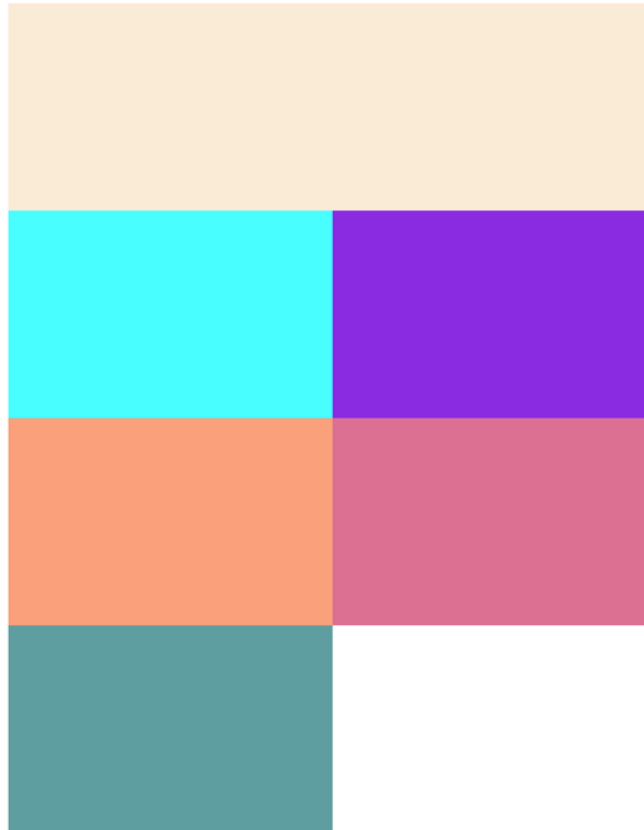


Imagen 3. Resultado con mostly fluid en un dispositivo entre **600px** y **800px**.

COLUMN DROP

Utilizaremos el mismo código HTML para realizar este ejemplo, al igual que la primera parte del código CSS. Solo modificaremos los media queries.

```
1 @media (min-width: 600px) {
2     .columna1 {
3         width: 60%;
4         -webkit-order: 2;
5         order: 2;
6     }
7     .columna2 {
8         width: 40%;
9         -webkit-order: 1;
10        order: 1;
11    }
12    .columna3 {
13        width: 100%;
14        -webkit-order: 3;
15        order: 3;
16    }
17    .columna4 {
18        width: 100%;
19        -webkit-order: 6;
20        order: 6;
21    }
22    .columna5 {
23        width: 100%;
24        -webkit-order: 5;
25        order: 5;
26    }
27    .columna6 {
28        width: 100%;
29        -webkit-order: 4;
30        order: 4;
31    }
32 }
33 @media (min-width: 800px) {
34     .columna2,
35     .columna3,
36     .columna4,
37     .columna5,
38     .columna6 {
39         width: 20%;
40     }
41 }
```

Revisemos cómo se vería nuestra página web en un dispositivo de más de **800px** de ancho.

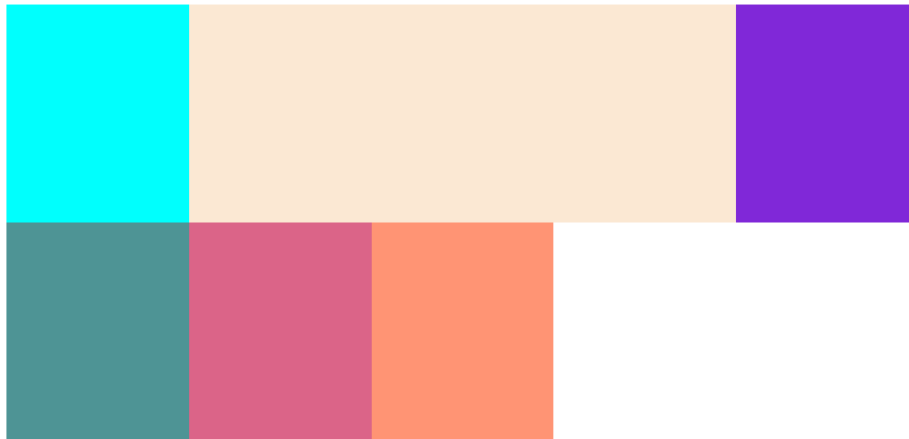


Imagen 4. Resultado con **column drop** en un dispositivo de gran tamaño.

Y así en uno de tamaño entre **600px** y **800px**.

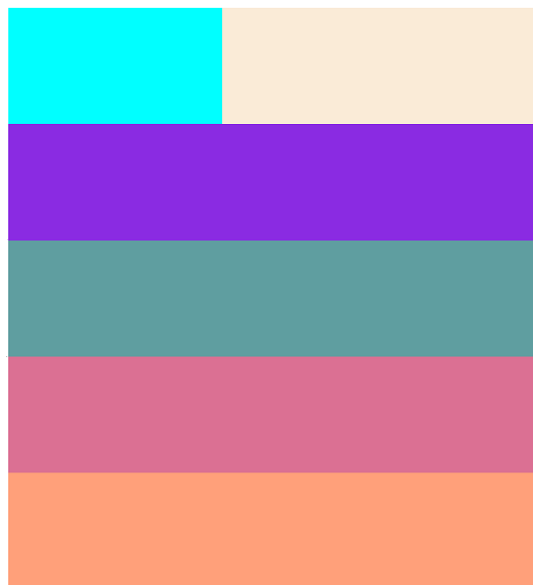


Imagen 5. Resultado con column drop en un dispositivo entre **600px** y **800px**.

LAYOUT SHIFTER

En este caso, realizaremos algunas modificaciones en el código HTML.

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <title>Layout Shifter</title>
5     <link rel="stylesheet" href="assets/css/style.css">
6   </head>
7   <body>
8     <div class="container">
9       <div class="columna1"></div>
10      <div class="columna2"></div>
11      <div class="container-inner">
12        <div class="columna3"></div>
13        <div class="columna4"></div>
14        <div class="columna5"></div>
15      </div>
16    </div>
17  </body>
18 </html>
```

Y también modificaremos la hoja de estilos.

```
1 .columna1,
2 .columna2,
3 .columna3,
4 .columna4,
5 .columna5,
6 .container-inner {
7   width: 100%;
8   height: 100px;
9 }
10
11 .columna1 {
12   background-color: antiquewhite;
13 }
14
15 .columna2 {
16   background-color: aqua;
```

```
17 }
18
19 .columna3 {
20     background-color: blueviolet;
21 }
22
23 .columna4 {
24     background-color: lightsalmon;
25 }
26
27 .columna5 {
28     background-color: palevioletred;
29 }
30
31
32 /* Layout Shifter */
33
34 .container {
35     display: -webkit-flex;
36     display: flex;
37     -webkit-flex-flow: row wrap;
38     flex-flow: row wrap;
39 }
```

Ahora, emplearemos algunas medias queries para aplicar el patrón de diseño **layout shifter**.

```
1 @media (min-width: 600px) {
2     .columna1,
3     .columna2 {
4         width: 20%;
5         height: 300px;
6     }
7     .container-inner {
8         width: 60%;
9     }
10 }
11 @media (min-width: 800px) {
12     .container {
13         width: 800px;
14         margin-left: auto;
15         margin-right: auto;
16     }
17 }
```

Y así sería el resultado en un dispositivo con pantalla de gran tamaño.



Imagen 6. Resultado con **layout shifter** en una pantalla de sobre **800px**.



Imagen 6. Resultado con **layout shifter** en una pantalla de sobre **600px**.

TINY TWEAKS

Para este ejemplo, y teniendo en cuenta que dicho patrón trabaja con una sola columna, modificaremos el documento HTML para que se adapte a él.

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <title>Mostly fluid</title>
5     <link rel="stylesheet" href="assets/css/style.css">
6   </head>
7   <body>
8     <div class="container">
9       <div class="column1"></div>
10    </div>
11  </body>
12 </html>
```

Luego, le daremos estilo a dicha columna:

```
1 .column1 {
2   height: 200px;
3   width: 100%;
4   padding: 10px;
5   background-color: antiquewhite;
6 }
```

Ahora, procederemos a agregar los media queries correspondientes.

```
1 @media (min-width: 600px) {
2   .column1 {
3     padding: 20px;
4     background-color: aquamarine;
5   }
6 }
7
8 @media (min-width: 800px) {
```



```
9      .columnal {  
10          padding: 40px;  
11          background-color: blue;  
12      }  
13 }
```

Para finalizar, revisemos el resultado en distintos tamaños de pantalla.



Imagen 8. Resultado con tiny tweaks en un dispositivo de gran tamaño.



Imagen 9. Resultado con tiny tweaks en un dispositivo entre **600px** y **800px**.

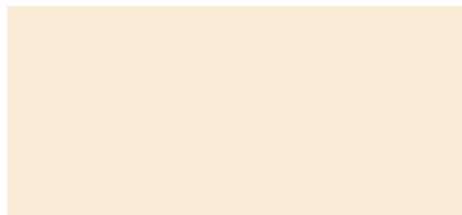


Imagen 10. Resultado con tiny tweaks en un dispositivo de pantalla pequeña.

OFF CANVAS

Para el siguiente ejemplo, crearemos tres columnas de distinto color.

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <title>Mostly fluid</title>
5     <link rel="stylesheet" href="assets/css/style.css">
6   </head>
7   <body>
8     <div class="container">
9       <div class="columna1"></div>
10      <div class="columna2"></div>
11      <div class="columna3"></div>
12    </div>
13  </body>
14 </html>
```

```
1 .columna1,
2 .columna2,
3 .columna3 {
4   height: 200px;
5   width: 100%;
6 }
7
8 .columna1 {
9   background-color: antiquewhite;
10 }
11
12 .columna2 {
13   background-color: aqua;
14 }
15
16 .columna3 {
17   background-color: blueviolet;
18 }
```

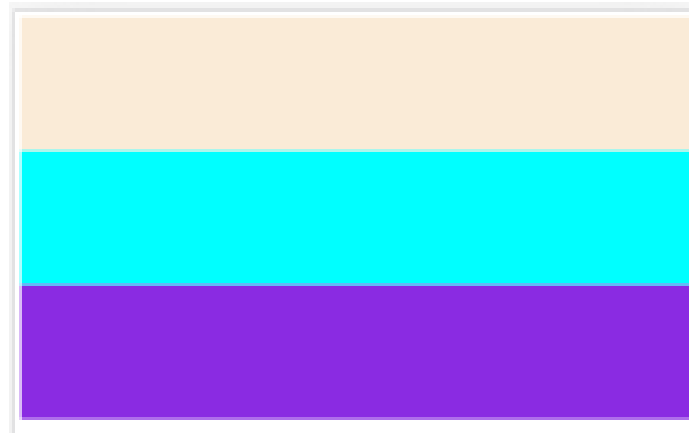


Imagen 11. Resultado sin patrón de diseño.

Ahora, modificaremos el archivo CSS para aplicar este patrón de diseño.

```
1 .columna1,  
2 .columna2,  
3 .columna3 {  
4     height: 200px;  
5     position: absolute;  
6 }  
7  
8 .container {  
9     width: 200px;  
10    margin-left: 200px;  
11 }  
12  
13 .columna1 {  
14     background-color: antiquewhite;  
15     width: 200px;  
16     -webkit-transition: -webkit-transform 0.3s ease-out;  
17     transition: transform 0.3s ease-out;  
18     -webkit-transform: translate(-200px, 0);  
19     transform: translate(-200px, 0);  
20     z-index: 1;  
21 }  
22  
23 .columna1.open {  
24     -webkit-transform: translate(0, 0);  
25     transform: translate(0, 0);
```

```
26 }
27
28 .columna2 {
29     background-color: aqua;
30     width: 200px;
31 }
32
33 .columna3 {
34     background-color: blueviolet;
35     width: 200px;
36     -webkit-transition: -webkit-transform 0.3s ease-out;
37     transition: transform 0.3s ease-out;
38     -webkit-transform: translate(200px, 0);
39     transform: translate(200px, 0);
40     z-index: 2;
41 }
42
43 @media (min-width: 600px) {
44     .container {
45         display: -webkit-flex;
46         display: flex;
47         -webkit-flex-flow: row nowrap;
48         flex-flow: row nowrap;
49         width: 800px;
50     }
51     .columna1,
52     .columna3 {
53         -webkit-transform: translate(0, 0);
54         transform: translate(0, 0);
55     }
56 }
```

Obtendremos el siguiente resultado para pantallas pequeñas:



Imagen 12. Resultado en pantallas bajo los **600px**.

Y éste, en el caso de pantallas grandes:

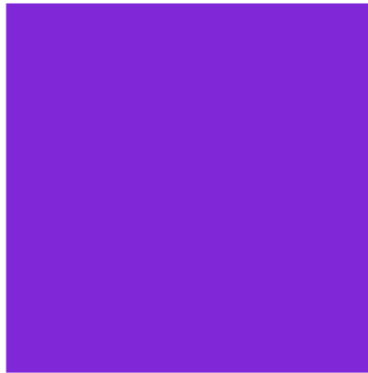


Imagen 13. Resultado con **off canvas** en pantallas grandes.

En este ejercicio hemos aprendido sobre los distintos patrones de diseño. Te invitamos a seguir realizando las actividades correspondientes a este CUE, y a continuar avanzando en el curso.

EXERCISE 3: CREANDO UN CONVERSOR DE UNIDADES DE MEDIDA.

En el siguiente ejercicio crearemos un conversor que nos permitirá automatizar el proceso de transformar unidades absolutas, como los pixeles, a unidades relativas, como rem y porcentaje.

Primero, recordemos qué son los **mixin**. Éstos corresponden a una directiva de Sass, que permite reutilizar reglas CSS dentro de nuestras hojas de estilo.

Ahora que ya lo tenemos presente, utilizaremos dicha directiva para crear nuestro conversor de unidades. Procederemos a ir al archivo **_mixins**, que está contenido en el directorio **abstracts** de nuestro proyecto.

Ya en este archivo, agregaremos una directiva **@mixin** con su nombre al lado. Procuremos utilizar nombres que hagan referencia a la función de éste.

```
1 @mixin conversor-de-unidades
```

Luego, debemos insertar paréntesis **()**, dentro de los cuales agregaremos los valores necesarios para convertir las unidades absolutas a relativas.

```
1 @mixin conversor-de-unidades()
```

Necesitaremos los siguientes valores para que nuestro **mixin** sea lo más flexible posible:

1. La propiedad a afectar con la unidad relativa.
2. La unidad final a la que será transformada la absoluta.
3. El valor que se transformará.
4. El valor contextual de la unidad relativa.

Para agregar dichos valores dentro de los paréntesis, utilizaremos variables de Sass.

```
1 @mixin conversor-de-unidades($propiedad, $unidad, $objetivo, $contexto)
```

Ya con esto listo, procederemos a abrir llaves y, dentro de ellas, definiremos una variable que como valor tendrá el resultado de la división entre la unidad absoluta (**\$objetivo**) y el valor asociado (**\$contexto**) a la unidad relativa.

```
1 @mixin conversor-de-unidades($propiedad, $unidad, $objetivo, $contexto) {  
2     $resultado: ($objetivo / $contexto);  
3 }
```

Posteriormente, debemos declarar la estructura que tendrá el resultado de la conversión, la cual estará dada por el resultado más la unidad en la que se medirá.

```
1 @mixin conversor-de-unidades($propiedad, $unidad, $objetivo, $contexto) {  
2     $resultado: ($objetivo / $contexto);  
3     #{$propiedad}: #{$resultado + $unidad};  
4 }
```

Finalmente, aplicaremos nuestro nuevo **mixin** al proyecto, y para eso debemos dirigirnos a nuestro manifiesto. Una vez ahí, probaremos su funcionamiento transformando el tamaño de un párrafo de pixeles a rem, para lo cual debemos agregar **@include**, y a continuación, el nombre del **mixin** seguido de paréntesis, dentro de los que incluiremos los valores necesarios para hacer la conversión.

```
1 h1 {  
2     @include conversor-de-unidades(font-size, rem, 40, 16);  
3 }
```

Es importante recordar, que el primer valor corresponde a la propiedad que queremos transformar; el segundo, a la unidad relativa a la que queremos transformar; el tercero, al valor de la propiedad en unidades absolutas; y, el cuarto, al valor contextual de la unidad relativa.