

Ejercicio 1 (2 puntos)

Para la web de una empresa de tarjetas de visita, tenemos que generar una clase para cada tarjeta. Esta clase debe

llamarse .card-XXXX (XXXX es el nombre de la tarjeta) y contener el tamaño de fuente y su color, y el color de fondo.

Por ejemplo, para la tarjeta con nombre "fancy2", tamaño de letra 10pt, color azul y fondo blanco, la clase generada debe

ser:

```
.card-fancy2 {  
font-size: 10pt;  
color: blue;  
background-color: white;  
}
```

Elegir la estructura más adecuada para almacenar esta información y luego construir las clases a partir de ellas. Probar

con los siguientes datos:

intrepida: 11pt, red, lightgray

clasica: 12pt, black, white

inversa: 9pt, orange, silver

Opción 1)

```
$list: (  
  ("intrepida", 11pt, red, lightgray),  
  ("clasica", 12pt, black, white),  
  ("inversa", 9pt, orange, silver)  
);  
  
@each $nombre, $letra, $color, $fondo in $list {  
  .card-#{$nombre} {  
    font-size: $letra;  
    color: $color;  
    background-color: $fondo;  
  }  
}
```

Opción 2)

```
$clases: (  
  'intrepida': (11pt, red, lightgray),  
  'clasica': (12pt, black, white),  
  'inversa': (9pt, orange, silver)  
);  
  
@each $key, $value in $clases {  
  .card-#{$key} {  
    font-size: nth($value, 1);  
    color: nth($value, 2);  
    background-color: nth($value, 3);  
  }  
}
```

Opción 3)

```
$tarjetas: ((intrepida, 11pt, red, lightgray),  
  (clasica, 12pt, black, white),  
  (inversa, 9pt, orange, silver)  
);  
  
//for each para recorrer todas las tarjetas  
@each $tarjeta in $tarjetas {  
  .card-#{nth($tarjeta, 1)} {  
    font-size: nth($tarjeta, 2);  
    color: nth($tarjeta, 3);  
    background-color: nth($tarjeta, 4);  
  }  
}
```

Ejercicio 2 (3,5 puntos)

Queremos dar de forma rápida estilos a los encabezados que se encuentran dentro de un elemento que pertenece a una clase determinada. Para ello vamos a crear 2 funciones MIXIN, cada una en un archivo .scss diferente, por lo que usará un 'index.html' diferente para usar el .css resultante que corresponda (Los index.html serán proporcionados en los archivos del examen), a la que se le debe suministrar el nombre de la clase, un color, un tamaño y un "porcentaje de reducción".

Los encabezados del primer Mixin serán:

- h1: color un 10% más claro que el suministrado y el tamaño que se indica.
- h2: color un 20% más claro que el suministrado y el tamaño es más pequeño que el h1 según el "porcentaje de reducción" suministrado.
- h3: color un 30% más claro que el suministrado y el tamaño es más pequeño que el h2 según el "porcentaje de reducción" suministrado.
- h4: color un 40% más claro que el suministrado y el tamaño es más pequeño que el h3 según el "porcentaje de reducción" suministrado.
- ...y así sucesivamente.

Por ejemplo para unos valores de: "miclase", #919b09, 5rem, 70 se obtendría el siguiente CSS (style.css):

```
.miclase h1 {  
  color: #becb0c;  
  font-size: 5rem;  
}
```

```
.miclase h2 {  
  color: #e3f218;  
  font-size: 3.5rem;  
}
```

```
.miclase h3 {  
  color: #e9f449;  
  font-size: 2.45rem;  
}
```

```
.miclase h4 {  
  color: #eff779;  
  font-size: 1.715rem;  
}
```

```
.miclase h5 {  
  color: #f4faa9;  
  font-size: 1.2005rem;  
}
```

```
.miclase h6 {  
  color: #fafdd9;  
  font-size: 0.84035rem;  
}
```

Opción 1)

```
@mixin encabezados($clase, $color, $tam, $reduccion) {  
  @for $i from 1 to 7 {  
    @if $i != 1 {  
      $tam: calc($tam * $reduccion / 100);  
    }  
    .#{$clase} h#{$i} {  
      color: lighten($color, $i*10);  
      font-size: $tam;  
    }  
  }  
}
```

```
@include encabezados("miclase", #919b09, 5rem, 70);
```

Opción 2)

```
@mixin index($clase, $color, $tamanoLetra, $reduccion){  
  $porcentaje: 10;  
  @for $i from 1 to 7{  
    .#{$clase}{  
      h#{$i}{  
        color: lighten($color, $porcentaje);  
        font-size: $tamanoLetra;  
      }  
    }  
  }  
}
```

```
$tamanoLetra: calc(($tamanoLetra*$reduccion)/100);  
$porcentaje: $porcentaje+10;  
}
```

```
@include index("miclase", #919b09, 5rem, 70);
```

Opción 3)

```
@use "sass:color";

@function getFontSize($value, $reduction, $iteration) {
  @if($iteration > 1) {
    @return getFontSize($value, $reduction, $iteration - 1)* calc($reduction/100);
  }
  @else {
    @return $value;
  }
}

@function getBrighterColor($color, $percentage) {
  @return color.adjust($color, $lightness: percentage(calc($percentage/100)));
}

@mixin encabezadoClaro($name, $color, $font-size, $reduction) {
  @for $i from 1 through 6 {
    .#{$name} h#{$i} {
      color: getBrighterColor($color, calc($i*10));
      font-size: getFontSize($font-size, $reduction, $i);
    }
  }
}

@include encabezadoClaro("miclase", #919b09, 5rem, 70);
```

Ejercicio 3 (4,5 puntos)

Queremos dar estilo al interior de las secciones.

Para ello vamos a definir una serie de temas en la que se indica:

- el nombre (que a su vez es la imagen “png” de fondo),
- el color del texto,
- el grado de opacidad del fondo de los artículos
- la fuente que se va a usar.

De forma que para:

('relax', #93a8b7, 60, Verdana)

```
.relax{
background-image:
url("../patrones/relax.png");
padding: 5px;
}

.relax article{
padding:10px;
margin:10px;
color:#93a8b7;

font-family: Verdana;
background-color: rgba(255, 255, 255,
0.6);
text-shadow: 2px 2px 2px black;
}
```

('contrast', #cccccc, -65)

```
.contrast{
background-image:
url("../patrones/contrast.png");
padding: 5px;
}

.contrast article {
padding: 10px;
margin: 10px;
color: #cccccc;

font-family: "Comic Sans MS";
background-color: rgba(0, 0, 0, 0.65);
text-shadow: 2px 2px 2px black;
}
```

Ten en cuenta que:

- Si no se indica explícitamente la fuente, se debe usar por defecto “Comic Sans MS”
- El sombreado del texto será negro si la luminosidad (lightness) del color de texto es mayor del 50%, y si es menor, el sombreado del texto será blanco.
- Si la opacidad es positiva, se debe hacer sobre un fondo blanco, y si es negativa, se aplica sobre fondo negro.

Realiza la prueba usando el archivo patrones.html y con los temas:

('relax', #93a8b7, 60, Verdana),
('contrast', #cccccc, -65),
('love', #f8afb2, 30, Gabriola),
('psico', #1b4d48, 30, "Arial Black"),
('modern', #fec87c, -20, Impact)

Opción 1)

@use "sass:math";

```
@mixin theme($name, $color, $opacity, $font: 'Comic Sans MS') {
.#{$name} {
background-image: url("../patrones/#{ $name }.png");
padding: 5px;
}

.#{$name} article {
padding: 10px;
margin: 10px;
color: $color;
font-family: $font;
@if $opacity > 0 {
background-color: rgba(255, 255, 255, math.div($opacity, 100));
} @else {
background-color: rgba(0, 0, 0, math.div(math.abs($opacity), 100));
}

@if lightness($color) > 50 {
text-shadow: 2px 2px 2px black;
} @else {
text-shadow: 2px 2px 2px white;
}
}
```

```

}

@include theme(relax, #93a8b7, 60, Verdana);
@include theme(contrast, #cccccc, -65);
@include theme(love, #f8afb2, 30, Gabriola);
@include theme(psico, #1b4d48, 30, "Arial Black");
@include theme(modern, #fec87c, -20, Impact);

```

Opción 2)

```

@mixin styleSections($name, $color, $opacityBg, $font: 'Comic Sans MS') {
  $shadowColor: calcShadowColor($color);
  $bgColor: calcBgColor($opacityBg);
  @if ($opacityBg > 0) {
    $bgColor: white;
  } @else {
    $bgColor: black;
  }
  #{ $name } {
    background-image: url("../patrones/#{ $name }.png");
    padding: 5px;
    article {
      padding: 10px;
      margin: 10px;
      color: $color;
      font-family: $font;
      background-color: rgba($bgColor, calc(abs($opacityBg) / 100));
      text-shadow: 2px 2px 2px $shadowColor;
    }
  }
}

```

```

@function calcShadowColor($color) {
  $shadowColor: white;
  @if (lightness($color) > 50%) {
    $shadowColor: black;
  }
  @return $shadowColor;
}

```

```

@function calcBgColor($opacityBg) {
  $bgColor: black;
  @if ($opacityBg > 0) {
    $bgColor: white;
  }
  @return $bgColor;
}

```

```

$listIncludes: (
  ('relax', #93a8b7, 60, Verdana),
  ('contrast', #cccccc, -65),
  ('love', #f8afb2, 30, Gabriola),
  ('psico', #1b4d48, 30, "Arial Black"),
  ('modern', #fec87c, -20, Impact)
);

```

```

@each $include in $listIncludes {
  $name: nth($include, 1);
  $color: nth($include, 2);
  $opacityBg: nth($include, 3);
  @if (length($include) < 4) {
    @include styleSections($name, $color, $opacityBg);
  } @else {
    $font: nth($include, 4);
    @include styleSections($name, $color, $opacityBg, $font);
  }
}

```

Opción 3)

```

@use "sass:color";
@use "sass:math";
@use "sass:list";

```

```

$patrones: (
  ('relax', #93a8b7, 60, Verdana),
  ('contrast', #cccccc, -65),
  ('love', #f8afb2, 30, Gabriola),

```

```
('psico', #1b4d48, 30, "Arial Black"),  
('modern', #fec87c, -20, Impact)  
):
```

```
@function getColorLightness($color) {  
  @return lightness($color);  
}
```

```
@function getTextShadowColor($color) {  
  @if getColorLightness($color) > 50% {  
    @return black;  
  }  
  @else {  
    @return white;  
  }  
}
```

```
@function parseOpacity($opacity) {  
  @return math.abs(math.div($opacity, 100));  
}
```

```
@function getBackgroundColor($opacity) {  
  @if $opacity < 0 {  
    @return color.change(black, $alpha: parseOpacity($opacity));  
  } @else {  
    @return color.change(white, $alpha: parseOpacity($opacity));  
  }  
}
```

```
@mixin getPatrones($name, $color, $opacity, $font: "Comic Sans MS") {  
  .#{$name} {  
    $folderUrl: "../patrones/";  
    background-image: url($folderUrl+$name+".png");  
    padding: 5px;  
  
    & article {  
      padding: 10px;  
      margin: 10px;  
      color: $color;  
      font-family: $font;  
      background-color: getBackgroundColor($opacity);  
      text-shadow: 2px 2px 2px getTextShadowColor($color);  
    }  
  }  
}
```

```
@each $patron in $patrones {  
  $name: list.nth($patron, 1);  
  $color: list.nth($patron, 2);  
  $opacity: list.nth($patron, 3);  
  @if list.length($patron) == 4 {  
    $font: list.nth($patron, 4);  
    @include getPatrones($name, $color, $opacity, $font);  
  } @else {  
    @include getPatrones($name, $color, $opacity);  
  }  
}
```