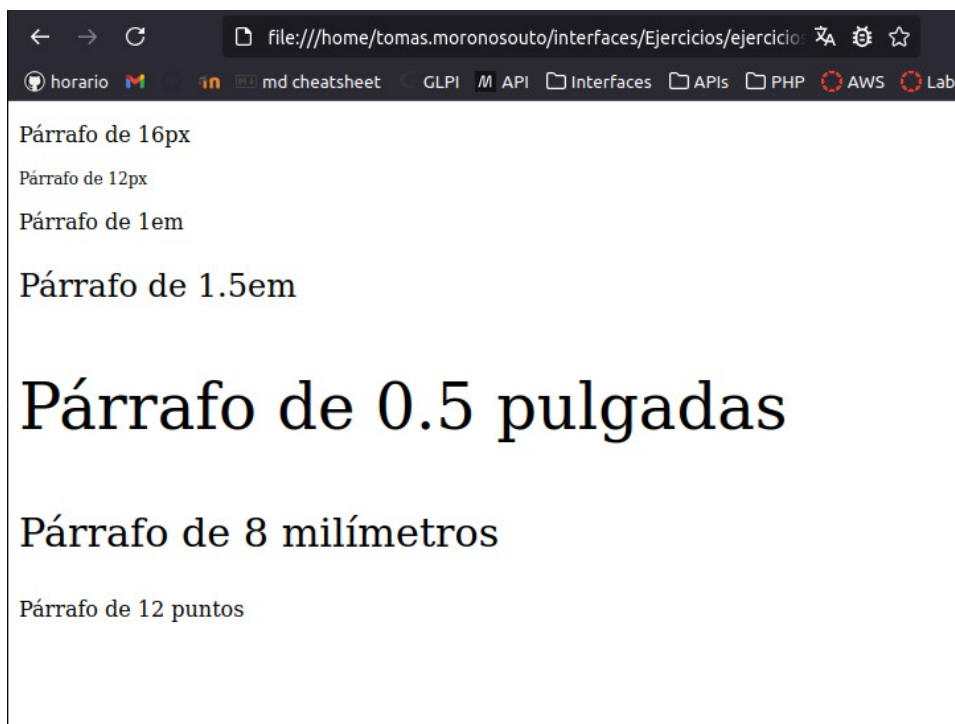


EJERCICIOS UD3.3

1. Si el tamaño de fuente por defecto en un navegador es de 12px. ¿Cuántos em son? Puedes ver la conversión de píxel a ems en el conversor online de: w3schools.com/tags/ref_pxtoemconversion.asp

0.75em

2. Observa el resultado del siguiente código en un navegador y estudia las diferencias que hay entre los distintos tamaños definidos. Adjunta el resultado de ejecutar este código.



3. Observa el resultado de los siguientes estilos y estudia las diferencias que hay entre los distintos valores definidos.

```
.a { background-color: red; }  
.b { background-color: #FF0000; }  
.c { background-color: RGB(255,0,0); }  
.d { background-color: HSL(0,100%,50%); }  
.e { background-color: red; opacity: 0.5; }  
.f { background-color: rgba(255, 0, 0, 0.5); }
```

Los 4 primeros son del mismo color a diferencia de los 2 últimos. 'e' y 'f' son del mismo tono ya que ambos tienen un 50% de opacidad, pero por como se ha aplicado esta, en 'e' la opacidad se aplica a todo el contenido y en 'f' solamente al color de fondo.

4. Agrega los siguientes estilos en una página web y explica las diferencias que hay entre ellos:

```
body { background-image: url("https://bit.ly/2slA7jo");}
```

- Se ve una imagen en su tamaño real **repetida** suficientes veces a lo largo y ancho del DOM como para cubrirlo.

```
body { background-image: url("https://bit.ly/2slA7jo");  
background-repeat: repeat-x;}
```

- Se ve una imagen en su tamaño real **repetida** suficientes veces a lo ancho del DOM como para cubrirlo **horizontalmente**.

```
body { background-image: url("https://bit.ly/2slA7jo");  
background-repeat: repeat-y;}
```

- Se ve una imagen en su tamaño real **repetida** suficientes veces a lo largo del DOM como para cubrirlo **verticalmente**.

```
body { background-image: url("https://bit.ly/2slA7jo");  
background-repeat: no-repeat;}
```

- Se ve **una** imagen en su tamaño real.

```
body { background-image: url("https://bit.ly/2slA7jo");  
background-repeat: repeat;}
```

- Se ve una imagen en su tamaño real **repetida** suficientes veces a lo largo y ancho del DOM como para cubrirlo. Este es el valor por defecto para esta propiedad CSS.

```
body { background-image: url("https://bit.ly/2slA7jo");  
background-size: 100px 100px;}
```

- Se ve una imagen **de tamaño un dado** (100x100 píxels) repetida suficientes veces a lo largo y ancho del DOM como para cubrirlo.

5. La propiedad background-attachment indica si una imagen se mantiene fija o se desplaza con el resto de la página. Explica las **diferencias** que encuentras al establecer las siguientes propiedades:

```
body { background-image: url("https://bit.ly/2slA7jo");  
background-repeat: no-repeat; background-attachment: fixed;}
```

- Se ve una imagen en su tamaño real que **se mueve al hacer scroll**.

```
body { background-image: url("https://bit.ly/2slA7jo");  
background-repeat: no-repeat; background-attachment: scroll;}
```

- Se ve una imagen en su tamaño real que **no se mueve al hacer scroll**.

6. Crea un **efecto parallax** con fondo fijo y letra. Para ello fíjate en los ejemplos 1 y 2 del apartado 6.2 de la teoría.

Adjunto

7. Escribe un documento en HTML con estilos con un **efecto parallax** sencillo de la siguiente manera: una imagen de background fija, la cual desaparecerá cuando el usuario "le de" al scroll vertical, dando paso a un background de color (a tu elección) con opacity del 30%. Puedes ver un ejemplo en:

https://www.w3schools.com/howto/tryit.asp?filename=tryhow_css_parallax

8. Realiza los siguientes degradados:

Adjunto

9. Escribe un documento en HTML y su hoja de estilo correspondiente, en CSS, donde se recojan los siguientes criterios de diseño:

Adjunto

10. Escribe los documentos HTML y CSS necesarios para que se muestre por pantalla la siguiente salida de texto. (Entre paréntesis se indica la familia o nombre de fuente).

Adjunto

11. Comprueba cuál de los siguientes códigos es correcto y determina qué ocurre si algunos valores están bien situados y otros no.

```
p { font: oblique bold 120% cursive; }  
p { font: small-caps 100%/200% serif; }  
p { font: monospace 150%; }  
p { font: bold 90%; } → Este no funciona apropiadamente por el desorden.  
p { font: oblique 120% sans-serif bold; } → Este tampoco.  
p { font: 120% sans-serif small-caps; } → Este tampoco.
```

12. Conéctate a la página web de **Google Fonts**, selecciona 3 tipos de fuentes distintas, y realiza las siguientes tareas:

Adjunto

13. Propiedades **hyphens** y **line-clamp**. Copia los ejemplos 9.4.1 y 9.4.2, en donde se crean cards con ambas propiedades, y observa el resultado. ¿Cuál es el propósito de hyphens? ¿y el de line-clamp?

Adjunto

El propósito de **hyphens** se encarga de dividir palabras al final de una línea e **line-clamp** limita la cantidad de texto visualizado contado en número de líneas.

14. ¿Cuál es la diferencia entre **display:none** y **visibility:hidden** ?

La primera opción elimina el elemento seleccionado del DOM y la segunda simplemente lo oculta visualmente, pese a que el elemento sigue ahí y podrá ser estilizado para verse si cierto evento ocurre (i.e.: un hover).

15. Realiza las siguientes tareas:

a) Indica con tus propias palabras, cuál es la **diferencia** entre un elemento con **position absolute** y **position relative**.

Position: relative sirve para que las propiedades de posicionamiento indicadas se apliquen tomando como referencia a su padre directo y position: absolute toma como referencia el ancestro mas cercano que tenga una posición distinta de static.

b) [Adjunto](#)

16. Realiza y responde a las siguientes cuestiones/tareas:

a) [Adjunto](#)

b) Cuando reduces el tamaño de pantalla, ¿se reorganizan los bloques? ¿cómo harías para que se reorganizaran dichos bloques según el tamaño de la pantalla? Indica la propiedad que utilizarías.

Si. Usaría media-queries con la etiqueta **@media** y **min-width** o **max-width**.

17. Uno de los usos más comunes de la propiedad float es la creación de botones de avance y retroceso en un sitio web. Ejecuta el primer bloque de código en tu navegador y observa lo que ocurre:

[Adjunto](#)

18. Crea un breadcrumb (migas de pan) navegacional utilizando HTML y CSS. El breadcrumb debe contener enlaces que permitan navegar hacia páginas anteriores y siguientes, y se deben aplicar estilos específicos.

[Adjunto](#)

19. Diseña y crea un menú de navegación horizontal utilizando HTML y CSS. El menú debe contener cuatro enlaces y aplicar estilos específicos.

20. Crea un documento HTML que incluya una caja de contenido con un texto largo. Tu tarea es utilizar la propiedad overflow en CSS para controlar cómo se muestra el contenido de la caja. Sigue estos pasos:

21. Escribe un código HTML con estilos que contenga en el que utilices las propiedades z-index, visibility y overflow.