



EJERCICIOS UD3 – 3.3 – PARTE 1

1. Si el tamaño de fuente por defecto en un navegador es de 12px. ¿Cuántos em son 16px? Puedes ver la conversión de píxel a ems en el conversor online de w3schools: [w3schools.com/tags/ref_pxtoemconversion.asp](https://www.w3schools.com/tags/ref_pxtoemconversion.asp)
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.**

```
<p style="font-size: 16px;">Párrafo de 16px</p>
<p style="font-size: 12px;">Párrafo de 12px</p>
<p style="font-size: 1em;">Párrafo de 1em</p>
<p style="font-size: 1.5em;">Párrafo de 1.5em</p>
<p style="font-size: 0.5in;">Párrafo de 0.5 pulgadas</p>
<p style="font-size: 8mm;">Párrafo de 8 milímetros</p>
<p style="font-size: 12pt;">Párrafo de 12 puntos</p>
```

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); }
```



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"); }

body { background-image: url("https://bit.ly/2sLA7jo");
background-repeat: repeat-x; }

body { background-image: url("https://bit.ly/2sLA7jo");
background-repeat: repeat-y; }

body { background-image: url("https://bit.ly/2sLA7jo");
background-repeat: no-repeat; }

body { background-image: url("https://bit.ly/2sLA7jo");
background-repeat: repeat; }

body { background-image: url("https://bit.ly/2sLA7jo");
background-size: 100px 100px; }
```

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; }

body { background-image: url("https://bit.ly/2sLA7jo");
background-repeat: no-repeat; background-attachment: 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.
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

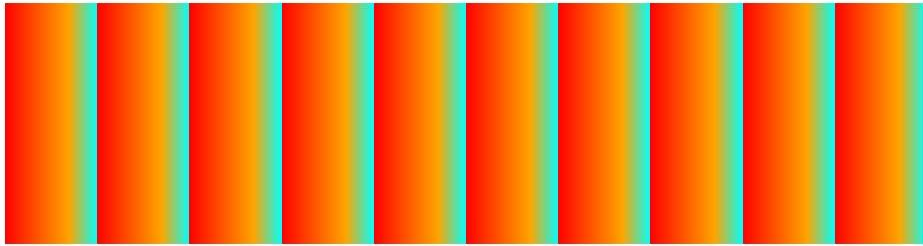


8. Realiza los siguientes degradados:

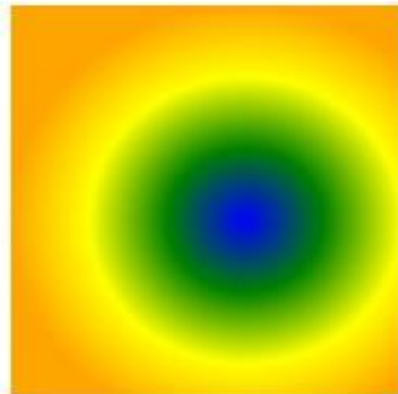
a)



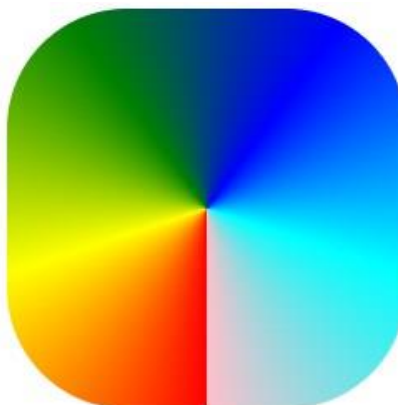
b)



c)



d)





9. Escribe un documento en HTML y su hoja de estilo correspondiente, en CSS, donde se recojan los siguientes criterios de diseño:
- La fuente base de la página será de color negro, con tipografía Courier, un tamaño de 12 puntos e interlineado a 1.2.
 - Debe haber, **al menos, tres bloques** compuestos por elementos de tipo título `<h1>` y elementos de párrafo `<p>`. El resto de los elementos se deja a criterio del diseñador.
 - Los títulos serán de un color seguro, y su tamaño debe ser el doble del texto base del resto de la página.
 - Los títulos, además, tendrán una sombra de color azul, la primera letra de cada palabra estará en mayúscula, y tendrá un espaciado entre palabras de 5px.
 - Los elementos que forman parte del listado deben aparecer centrados y subrayados.
 - Finalmente, se debe incorporar un enlace directo a cualquier buscador de Internet (por ejemplo, Google). Si no se ha visitado este sitio web, el enlace aparecerá en cursiva y de color gris. Cuando se haya visitado, su aspecto cambiará y será de color rojo y en negrita, subrayado por dos líneas.
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).

ACTIVIDAD EN CSS (Times New Roman)

- Pongamos a prueba las fuentes en CSS (Helvetica)
- También **el tamaño** (Arial)
- Y el *estilo* (Times New Roman)

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%;  
}  
p {  
    font: oblique 120% sans-serif bold;  
}  
p {  
    font: 120% sans-serif small-caps;  
}
```

12. Conéctate a la página web de **Google Fonts**, selecciona 3 tipos de fuentes distintas, y realiza las siguientes tareas:
- Importa la primera fuente elegida de forma interna y aplícala a un párrafo de texto.
 - Importa la segunda fuente elegida de forma externa utilizando la etiqueta `<link>`.
 - Importa la tercera fuente elegida de forma externa utilizando `@import` en tu código CSS.
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`?
14. ¿Cuál es la diferencia entre **display:none** y **visibility:hidden** ?
15. Realiza las siguientes tareas:
- Indica con tus propias palabras, cuál es la **diferencia** entre un elemento con **position absolute** y **position relative**.
 - Escribe un código HTML con estilos que contenga **un contenedor padre**, y **5 elementos hijo** posicionados de la siguiente manera: el primero en posición estática, el segundo con posición relativa, el tercero con posición fija, el cuarto con posición absoluta, y el quinto con posición "sticky".



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

- a) Escribe un código HTML con estilos con el código correspondiente para que dé el siguiente resultado. Utiliza las propiedades de **float** y **clear**.



- b) Cuándo 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.

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:

```
<div>
  <a class="anterior" href="#">Anterior</a>
  <a class="siguiente" href="#">Siguiente</a>
</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit...</p>
<p>Vestibulum ante ipsum primis in faucibus orci luctus et
...</p>

a.anterior {
  float: left;
}

a.siguiente {
  float: right;
}
```

Como puede observarse, la caja que contiene los enlaces anterior y siguiente no los recoge por completo; esto suele ocurrir cuando un elemento tiene en su interior otros



de tipo flotante. Para evitar esto se indica en el código CSS del elemento contenedor la propiedad `'overflow: auto'`, de esta forma consigue adaptarse al tamaño de los enlaces flotantes.

Incorpora al código anterior la propiedad indicada y comprueba el resultado.

```
div {  
  overflow: auto;  
}
```

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.

« Anterior

Siguiente »

Las instrucciones son las siguientes:

- Dentro del archivo HTML, crea una estructura para el breadcrumb utilizando un elemento `<div>` con la clase `breadcrumb`. El breadcrumb debe contener dos enlaces con las clases `left` y `right`. El primer enlace debe mostrar «« Anterior» y el segundo enlace debe mostrar «Siguiente »».
- Dentro del archivo CSS, asigna la fuente `Arial` y establece un relleno de `0.5em` alrededor del breadcrumb.
- Establece los estilos para los enlaces dentro del breadcrumb. Asigna el color `#0095eb`, elimina la decoración de texto, y ponlo en negrita.
- Define los estilos de cambio de color al hacer `hover` en los enlaces. Cuando el usuario pase el cursor sobre un enlace, cambia el color del texto a `#006196` y subraya el texto.
- Define los estilos para el enlace con la clase `right`. Utiliza `float: right`; para alinear este enlace a la derecha del breadcrumb.
- Define los estilos para el enlace con la clase `left`. Utiliza `float: left`; para alinear este enlace a la izquierda del breadcrumb.



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



Las **instrucciones** son las siguientes:

- Dentro del archivo HTML, crea una estructura para el menú de navegación utilizando elementos `<nav>`, ``, ``, y `<a>`. El menú debe contener cuatro enlaces con los siguientes nombres: "Inicio", "Servicios", "Nosotros" y "Contacto".
- Dentro del archivo CSS, define el estilo global para todos los elementos de la página utilizando el selector universal. Asigna a la fuente la familia **Arial** y ponla en negrita.
- Establece los estilos para la etiqueta `<nav>`. Asegúrate de que tenga un margen y un relleno de `0` y que su lista de estilos sea **none**.
- Define los estilos para los enlaces dentro del menú de navegación. Establece el ancho en **100px**, elimina la decoración de texto, alinea el texto al centro, establece el color de texto en blanco y el color de fondo en negro. Agrega un relleno superior e inferior de **10px** y un relleno izquierdo y derecho de **5px**. Agrega un borde derecho de **1px** sólido con el color **#336699**. Asegúrate de que los enlaces sean elementos de bloque.
- Define los estilos para los elementos de lista dentro del menú de navegación. Utiliza **float: left;** para que los elementos se coloquen horizontalmente.
- Agrega estilos de cambio de color al hacer **hover** en los enlaces. Cuando el usuario pase el cursor sobre un enlace, cambia el color de fondo a **#336699**.
- Utiliza la **pseudo-clase :last-child** para seleccionar el último elemento de la lista y, a través de su enlace (`<a>`), elimina el borde derecho para evitar que aparezca en el último enlace.

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**:

- Crea un archivo **HTML** con la siguiente estructura:

```
<div class="content-box">
```




```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum in tincidunt libero. Nullam sit amet risus sed elit bibendum tincidunt. Cras consequat ligula id metus hendrerit, id accumsan odio fermentum. Integer non feugiat orci. Donec eu tristique lectus. Fusce in viverra arcu. Proin condimentum, odio nec auctor tempor, justo elit vestibulum nunc, vel laoreet nunc est vel dolor. Aenean tempus, erat at vulputate consequat, velit lectus sollicitudin dui, ac dignissim est eros at erat. Sed congue posuere diam, vel vehicula metus viverra in. Vestibulum laoreet ex quis euismod. Aenean eu ex eget turpis congue interdum eu in erat.</p></div>
```

b) En un archivo CSS (llamado **styles.css** en este caso), agrega estilos a la caja de contenido (**content-box**) de la siguiente manera:

- Establece un ancho y un alto fijo para la caja (**width y height**).
- Utiliza la propiedad correspondiente para incluir una **barra de scroll en el eje vertical**.

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