

Hojas de estilo. CSS.

Recursos.

[W3C CSS](#)

[MDN Web Docs](#)

[Materialize](#)

[w3schools](#)

Introducción.

CSS (Cascading Style Sheets) es un lenguaje que describe el estilo de un documento HTML. CSS describe cómo se mostrarán los elementos HTML en la pantalla, en papel o en otros medios.

HTML fue diseñado para describir el contenido de las páginas web, utilizarlo para maquetar y describir estilos supone un proceso largo y costoso con grandes inconvenientes en el proceso de desarrollo. Para resolver estos inconvenientes, el World Wide Web Consortium (W3C) definió las especificaciones del estándar CSS.

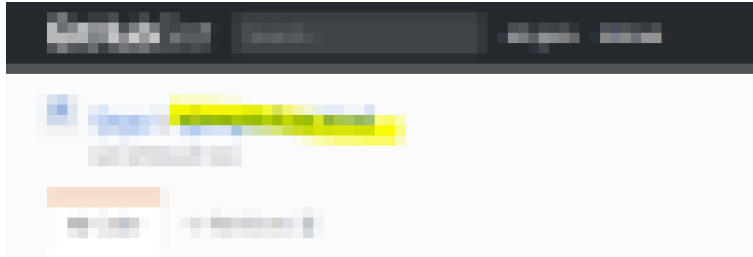
CSS permitió eliminar de los documentos HTML la descripción de los estilos. Las definiciones de estilo normalmente se guardan en archivos .css externos de manera que modificando los archivos css es posible cambiar el aspecto completo de un sitio web.

Actividad.

Referencia:

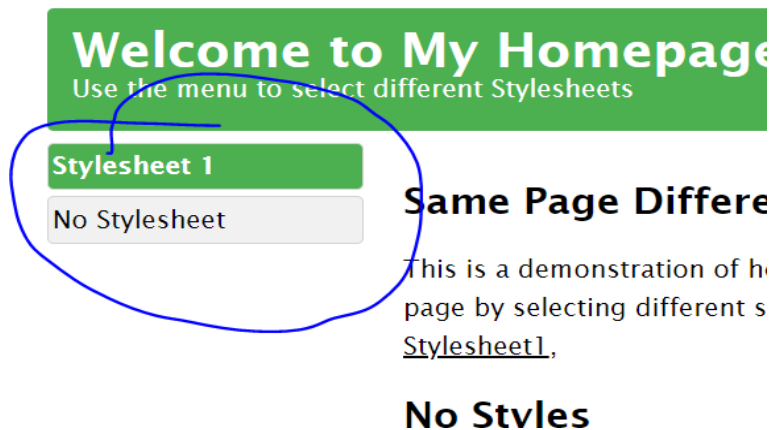
[w3schools](https://www.w3schools.com)

Hacemos una copia local de ejemplo1css.html localizado GitHubGist.



Analizamos el contenido identificando: estilos, html, scripts.

Visualizamos y probamos el documento en el navegador.



Añadimos los enlaces necesarios para incorporar el resto de estilos.

...

```
<li class="menuitem" onclick="reStyle(1)">Stylesheet 2</li>
<li class="menuitem" onclick="reStyle(2)">Stylesheet 3</li>
<li class="menuitem" onclick="reStyle(3)">Stylesheet 4</li>
```

...

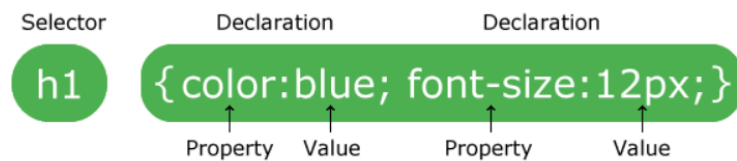
```
<a href="#" onclick="reStyle(1);return false">Stylesheet2</a>,
<a href="#" onclick="reStyle(2);return false">Stylesheet3</a>,
<a href="#" onclick="reStyle(3);return false">Stylesheet4</a>.
```

Probamos la página.

Hojas de estilo. Reglas.

Una **regla** es una declaración sobre un aspecto estilístico de uno o más elementos HTML. Una **hoja de estilo** es un conjunto de reglas que se aplican a un documento HTML.

Cada **regla** está formada por un **selector** y un **bloque de declaración**.



Selector

Referencia los elementos HTML a los que se aplicarán los estilos declarados en la regla.

Bloque de declaración.

Contiene una o más declaraciones agrupadas entre `{ }` y separadas por `;`

Cada declaración está formada por el par **propiedad, valor** separados por `:` y terminada en `;`

Como buena práctica se recomienda escribir cada par propiedad:valor en una línea diferente, cuando el bloque de declaración contenga varias declaraciones.

```
p {  
  font-size: 12px;  
  line-height: 15px;  
  color: #223344;  
}
```

Se pueden incluir comentarios en las hojas de estilo utilizando como delimitadores los caracteres `/*`, `*/`

```
/* Esto es un comentario CSS */
```

Actividad.

Poema de Jorge Drexler.

Esta actividad la vamos a desarrollar utilizando el entorno web online [codepen](#).

Añadimos el siguiente contenido al documento HTML.

```
<body>
<h1>Milonga del moro judío.</h1>
<address rel="author">
  <a href="https://es.wikipedia.org/wiki/Jorge_Drexler" target="_blank">
    Jorge Drexler
  </a>
</address>
<pre>
  Por cada muro un lamento
  En Jerusalén la dorada
  Y mil vidas malgastadas
  Por cada mandamiento.
  Yo soy polvo de tu viento
  Y aunque sangro de tu herida,
  Y cada piedra querida
  Guarda mi amor más profundo,
  No hay una piedra en el mundo
  Que valga lo que una vida.
  Yo soy un moro judío
  Que vive con los cristianos,
  No sé qué dios es el mío
  Ni cuáles son mis hermanos.
  No hay muerto que no me duela,
  No hay un bando ganador,
  No hay nada más que dolor
  Y otra vida que se vuela.
  La guerra es muy mala escuela
  No importa el disfraz que viste,
  Perdonen que no me aliste
  Bajo ninguna bandera,
  Vale más cualquier quimera
  Que un trozo de tela triste.
  Y a nadie le dí permiso
  Para matar en mi nombre,
  Un hombre no es más que un hombre
  Y si hay dios, así lo quiso.
  El mismo suelo que piso
  Seguirá, yo me habré ido;
  Rumbo también del olvido
  No hay doctrina que no vaya,
  Y no hay pueblo que no se haya
  Creído el pueblo elegido.
</pre>
</body>
```

Lenguaje Marcas

Añadimos en la ventana CSS algunos estilos para mejorar la presentación.

- Para body y html cambia fuente y tamaño.
- Pon color a la cabecera.
- Centra el título.
- Cambio de fuente y tamaño más pequeño para address.

Sacamos una copia del ejercicio en local y lprobamos el resultado.

Lenguaje Marcas

Hay tres formas de aplicar estilos a un documento HTML.

Estilos en línea.

Se puede incluir el estilo utilizando el **atributo style** del elemento al que queremos aplicar el estilo. El valor de este atributo puede ser cualquier regla CSS.

```
<h1 style="color:blue;margin-left:30px;">Encabezado</h1>
```

No es recomendable aplicar los estilos de este modo, ya que se pierden las grandes ventajas que aporta la utilización de CSS.

Hoja de estilos interna.

En este caso, las reglas CSS se definen como contenido del **elemento style** incluido en la sección **head** del documento.

```
<head>
  <style>
    body {
      background-color: linen;
    }

    h1 {
      color: maroon;
      margin-left: 40px;
    }
  </style>
</head>
```

Hojas de estilo externas.

Es la forma más habitual de trabajar con estilos. Utilizamos ficheros de texto plano con extensión **.css** que incluyen las hojas de estilo. En los documentos HTML son referenciados en el elemento **link**, dentro de la sección **head**.

```
<head>
  ...
  <link rel="stylesheet" type="text/css" href="css/estilos.css">
  ...
</head>
```

css/estilos.css

```
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
```

Lenguaje Marcas

Los estilos se asignan en cascada (de ahí el nombre hojas de estilo en cascada o Cascading Style Sheets en inglés). Los elementos en los niveles bajos de la jerarquía heredan los estilos asignados a los elementos en los niveles más altos.

Los estilos heredados de elementos en niveles superiores se pueden reemplazar por nuevos estilos definidos para los elementos en niveles inferiores de la jerarquía.

Selectores.

El selector es el elemento CSS que permite referenciar o seleccionar al conjunto de elementos HTML al que se le deben aplicar los estilos.

Los selectores pueden ser:

Nombres de elementos.

Las reglas que utilizan como selector el nombre de un elemento se aplica a todos los elementos de este tipo encontrados en el documento.

Por ejemplo, la regla siguiente se aplicaría a todos los elementos **p** del documento.

```
p {  
    font-size: 12px;  
}
```

Valor de atributo id.

Otra forma de referenciar elementos es utilizando su atributo **id**. Si queremos aplicar una regla concreta a un atributo deberemos identificarlo con el atributo **id** y utilizar como selector el **valor del atributo** precedido del **carácter #**.

```
#p1 {  
    font-size: 24px;  
}
```

```
</head>  
<p id="p1">Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Nulla quis sem libero. Integer lobortis tincidunt dui, ac fringilla  
Vestibulum rutrum orci eu lectus consequat, et consequat augue plac  
Mauris augue eros, porta interdum vestibulum nec, maximus ac metus.  
urna mi consectetur metus, in bibendum velit nibh non dolor. Aliqua  
fermentum pretium libero. Donec tempor est at metus sodales, ac auc  
nec mollis lorem. Vivamus aliquam ultrices elit non aliquam. Quisqu  
vel luctus felis fermentum. Nunc dapibus sapien metus, eget commodo  
Suspendisse potenti. Fusce luctus scelerisque ligula.  
<p>Maecenas id erat pharetra, mollis urna ut, luctus nunc.
```

Valor de atributo class. .

Este atributo class se suele utilizar para agrupar elementos y poder aplicar estilos a un determinado grupo. Para referenciar un elemento usando su atributo **class**, el selector debe incluir el **valor del atributo** precedido por un **punto**.

```
.puntointeres {  
    font-size:16px;  
}
```

```
<p class ="puntointeres">Proi  
Fusce quis tristique dui. Ali  
libero sem hendrerit ex, sed
```

Las reglas asignadas a través del atributo **class** se denominan **clases**. A un mismo elemento se le pueden asignar varias clases, declarando los nombres de las clases separados por un espacio. Por ejemplo, **class="texto1 color1"**.

Las clases también se pueden declarar como exclusivas para un tipo específico de elementos declarando el nombre del elemento antes del punto.

```
p.puntointeres {  
    font-size:16px;  
}
```

Otros atributos. []

CSS permite referenciar elementos utilizando cualquier atributo. Este tipo de selector se define incluyendo el nombre del elemento seguido del nombre de atributo entre corchetes.

```
p[name] {  
    font-size:16px;  
}
```

La regla se aplica a todos los párrafos que incluyan el atributo **name**. Es posible incluir también el valor del atributo como criterio para la selección y utilizar distintos modificadores de igualdad para conseguir una selección más precisa.

- = Elementos cuyo valor de atributo sea igual al especificado.
- ~= Elementos cuyo valor de atributo incluya la palabra especificada.
- ^= Elementos cuyo valor de atributo empiece por la cadena especificada.
- \$= Elementos cuyo valor de atributo termina por la cadena especificada.
- *= Elementos cuyo valor de atributo contiene la cadena especificada.

Actividad guiada.

Prueba el siguiente ejemplo.

```
p[name] {  
    background: yellow;  
}  
p[name="p1"] {  
    background: red;  
}  
p[name~="p2"] {  
    background: green;  
}  
p[name^="p3"] {  
    background: grey;  
}  
p[name$="p4"] {  
    background: purple;  
}  
p[name*="p5"] {  
    background: orange;  
}  
</style>  
</head>  
<body>  
<p>Selección.</p>  
<p name="p">p.</p>  
<p name="p1">p1.</p>  
<p name="** p2 **">p2.</p>  
<p name="p3*****">p3.</p>  
<p name="****p4">p4.</p>  
<p name="*****p5*****">p5.</p>
```

Selección.

p.

p1.

p2.

p3.

p4.

p5.

Lenguaje Marcas

Es posible incluir una lista de selectores en una regla, en este caso hay que tener en cuenta que:

Si queremos asignar los mismos estilos a elementos con nombres diferentes, podemos declarar los nombres separados por una coma.

```
p, span {  
    font-size: 12px;  
}
```

Podemos referenciar solo elementos que se encuentran dentro de otro elemento utilizando como selector una lista de selectores separados por un espacio. Este tipo de selectores reciben el nombre de **selectores de descendencia**.

```
p span {  
    font-size: 12px;  
}
```

elemento_padre > elemento_hijo

Referencia un elemento que es hijo **directo** de otro elemento.

```
section > p {  
    font-size: 20px;  
}
```

elemento2 + elemento1

Este selector referencia un elemento que está inmediatamente después de otro (no dentro). Ambos deben compartir el mismo elemento padre. La siguiente regla se aplica a los elementos **p** que están inmediatamente después de un elemento **h1**.

```
h1 + p {  
    font-size: 20px;  
}
```

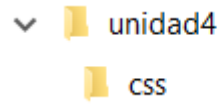
elemento1 ~ elemento2

Este selector hace referencia a todos los elementos que se ubican a continuación de otro elemento.

```
p ~ p {  
    font-size: 20px;  
}
```

Actividad.

Creamos una estructura de directorios para el ejercicio.



Creamos en el directorio css una hoja de estilos, style.css

```
p,span {  
    font-size: 16px;  
}  
p span {  
    font-size: 14px;  
}  
header p {  
    font-size: 22px;  
}
```

Creamos un documento HTML con una estructura básica y que incluya algunos párrafos para probar el estilo. Guárdalo en el mismo directorio donde has creado la carpeta css

Seudoclases

Las pseudoclases son selectores especiales de CSS que nos permiten referenciar elementos HTML por medio de sus características, como sus posiciones en el código o sus condiciones actuales.

```
selector:pseudo-class {  
  property:value;  
}
```

:nth-child(valor). Esta pseudoclase selecciona un elemento de una lista de elementos hermanos que se encuentra en la posición especificada por el valor entre paréntesis. La palabra clave **odd** afecta a los elementos que son hijos de otro elemento y tienen un índice impar, y la palabra clave **even** afecta a aquellos que tienen un índice par.

:first-child. Esta pseudoclase selecciona el primer elemento de una lista de elementos hermanos.

:last-child. Esta pseudoclase selecciona el último elemento de una lista de elementos hermanos.

:only-child. Esta pseudoclase selecciona un elemento cuando es el único hijo de otro Elemento

:first-of-type. Esta pseudoclase selecciona el primer elemento de una lista de elementos del mismo tipo.

:not(selector). Esta pseudoclase selecciona los elementos que no coinciden con el selector entre paréntesis.

Seudoclases de anclaje.

:link. Enlace no visitado.

:visited. Enlace visitado.

:hover. Selecciona el elemento sobre el que se encuentra el cursor del ratón.

:active. Enlace activo.

Tienen que ir en este orden para que funcionen correctamente.

Se pueden combinar con clases.

```
a.highlight:hover {  
  color: #ff0000;  
}
```

¿A qué elementos se aplica la regla anterior? Crea un documento para comprobarlo.

Seudoelementos

Se utilizan para dar estilo a partes específicas de un elemento. Por ejemplo, se pueden utilizar para dar estilo a la primera letra, o línea, de un elemento y para insertar contenido antes o después el contenido de un elemento.

```
selector::seudoelemento {  
    property:value;  
}
```

::after. Inserta contenido después.

::before. Inserta contenido antes. content of each <p> element

::first-letter. Selecciona primera letra

::first-line. Selecciona primera línea.

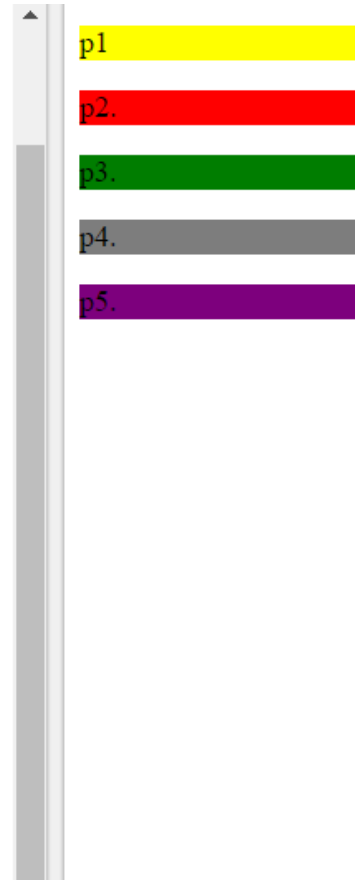
::selection. Selecciona la parte del elemento seleccionada por el usuario.

Actividad.

Probamos los siguientes ejemplos.

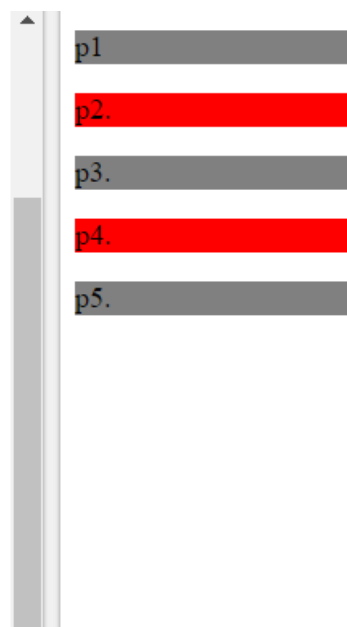
Ejemplo 1.

```
p:nth-child(1) {  
    background: yellow;  
}  
p:nth-child(2){  
    background: red;  
}  
p:nth-child(3) {  
    background: green;  
}  
p:nth-child(4) {  
    background: grey;  
}  
p:nth-child(5) {  
    background: purple;  
}  
  
</style>  
</head>  
<body>  
    <p>p1</p>  
    <p>p2.</p>  
    <p>p3.</p>  
    <p>p4.</p>  
    <p>p5.</p>  
</body>  
</html>
```



Ejemplo 2.

```
p:nth-child(odd) {  
    background: grey;  
}  
p:nth-child(even){  
    background: red;  
}  
  
</style>  
</head>  
<body>  
    <p>p1</p>  
    <p>p2.</p>  
    <p>p3.</p>  
    <p>p4.</p>  
    <p>p5.</p>  
</body>  
</html>
```



Propiedades CSS.

Recursos:

[Lista completa de propiedades CSS. W3C](#)

[Referencia MDN](#)

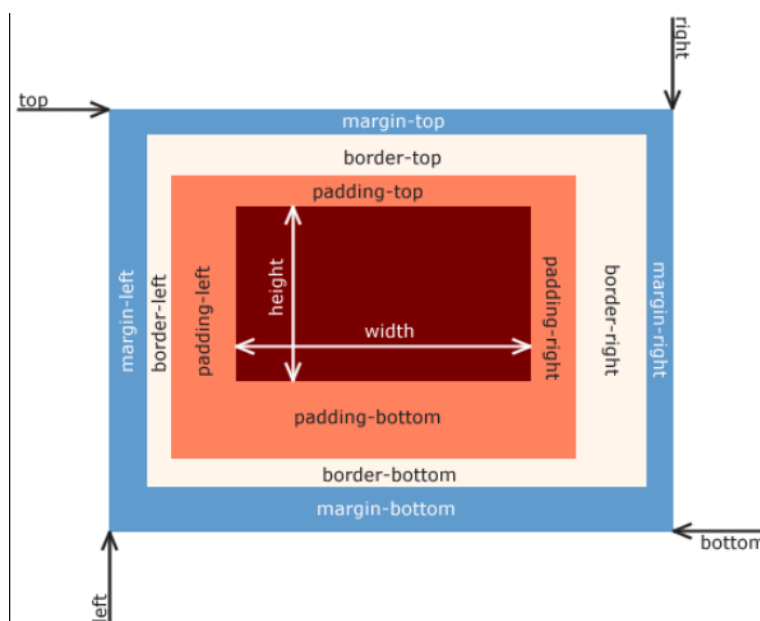
Las propiedades son la parte fundamental de las reglas de las hojas de estilo. Según su función podemos clasificarlas en propiedades de formato y propiedades de diseño.

Algunas propiedades y funciones CSS todavía se consideran experimentales. Por esta razón, se deben declarar con un prefijo que representa el motor web utilizado. Por ejemplo, si queremos que la función **linear-gradient()** funcione en Google Chrome, tenemos que declararla como **-webkit-linear-gradient()**. Para conseguir un funcionamiento correcto en todos los navegadores deberemos repetir estas propiedades para cada navegador con su correspondiente prefijo.

- moz- para Mozilla Firefox.
- webkit- para Safari y Google Chrome.
- o- para Opera.
- ms- para Internet Explorer.

Modelo de cajas.

Cualquier elemento contenido en un documento HTML tiene la estructura de una caja rectangular dentro del formato del documento. El modelo de cajas es la base del diseño web, cada elemento se representa como una caja rectangular, con su contenido, padding (espacio interior), borde y margen. El navegador, para interpretar el diseño de una página, procesa los estilos que se aplicarán a cada caja.



El tamaño del elemento no queda solo determinado por el ancho y la altura de su caja, sino también por el relleno y los márgenes, es decir el tamaño final de un elemento se calcula con la fórmula: tamaño + márgenes + relleno + bordes.

Margen, borde, padding y otras muchas propiedades se asignan mediante unidades de medida. La longitud se establece con un número seguido de una unidad de longitud: 10px, 2em, etc. Un espacio

Lenguaje Marcas

en blanco no puede aparecer entre el número y la unidad. Sin embargo, si el valor es 0, la unidad puede omitirse. Para algunas propiedades CSS, se permiten longitudes negativas.

Las unidades de longitud pueden ser:

Absolutas:

cm	centímetros.
mm	milímetros
in	pulgadas
px	pixels
pt	points
pc	picas

Relativas:

em	Relativa al tamaño de fuente del elemento (2em significa 2 veces el tamaño de la fuente actual)
ex	Relativo a la altura x de la fuente actual. No suele utilizarse
ch	Relativo al ancho del "0" (cero)
rem	Relativo al tamaño de fuente del elemento raíz
vw	Relativo al 1% del ancho de la ventana gráfica
vh	Relativo al 1% de la altura de la ventana gráfica.
vmin	Relativo al 1% de la dimensión más pequeña de la ventana gráfica.
vmax	Relativo al 1% de la dimensión más grande de la ventana gráfica.
%	Relativo al elemento padre.

Por defecto, el tamaño de la mayoría de los elementos se determina según el espacio disponible en el contenedor. El ancho de un elemento se define como 100 %, lo cual significa que será tan ancho como su contenedor, y tendrá una altura determinada por su contenido.

Los elementos se clasifican en dos tipos principales: **Block** (bloque) e **Inline** (en línea). Los elementos Block pueden tener un tamaño personalizado, pero los elementos Inline solo pueden ocupar el espacio determinado por sus contenidos.

CSS define las siguientes propiedades para declarar un tamaño personalizado:

width

Esta propiedad declara el ancho de un elemento. El valor se puede especificar en píxeles, porcentaje, o con la palabra clave auto (por defecto). Cuando el valor se especifica en porcentaje, el ancho se calcula según el navegador a partir del ancho del contenedor, y cuando se declara con el valor **auto**, el elemento se expande hasta ocupar todo el espacio horizontal disponible dentro del contenedor.

height

Esta propiedad declara la altura de un elemento. El valor se puede especificar en píxeles, porcentaje, o con la palabra clave auto (por defecto). Cuando el valor se especifica en porcentaje, el navegador

calcula la altura a partir de la altura del contenedor, y cuando se declara con el valor auto, el elemento adopta la altura de su contenedor.

margin

Esta propiedad declara el margen de un elemento. El margen es el espacio que hay alrededor de la caja. Puede recibir cuatro valores que representan el margen **superior, derecho, inferior, e izquierdo**, en ese orden y separados por un espacio. Si solo se declaran uno, dos o tres valores, los otros toman los mismos valores (por ejemplo, `margin: 10px 30px` asigna 10 píxeles al margen superior e inferior y 30 píxeles al margen izquierdo y derecho). Los valores se pueden declarar independientemente usando las propiedades asociadas ***margin-top*, *margin-right*, *margin-bottom* y *margin-left***. La propiedad también acepta el valor auto para obligar al navegador a calcular el margen.

border

Esta propiedad permite declarar todos los atributos del borde al mismo tiempo. También podemos usar las propiedades ***border-top*, *border-bottom*, *border-left*, y *border-right*** para definir los valores de cada borde de forma independiente. Definiremos el estilo del borde con la propiedad con la propiedad ***border-style***. Una vez que se define el estilo, el navegador usa los valores por defecto para generar el borde. Si no queremos dejar que el navegador determine estos valores, podemos usar el resto de las propiedades para configurar todos los atributos del borde.

padding

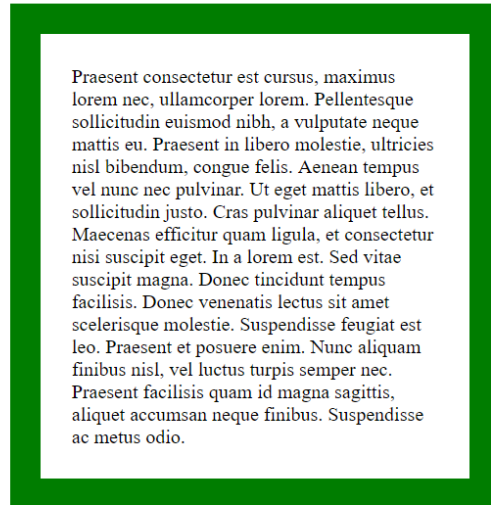
Esta propiedad declara el relleno (espacio interior) de un elemento. Los valores se fijan de la misma forma que se hace en la propiedad ***margin***. Para declarar los valores de forma independiente utilizamos: ***padding-top*, *padding-right*, *padding-bottom* y *padding-left***.

Actividad.

Escribimos un documento html y css para conseguir un resultado similar al siguiente:^o

Modelo de cajas

Neque porro quisquam est qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit...".



Solución:

```
section {  
    background-color: lightgrey;  
    width: 300px;  
    border: 25px solid green;  
    padding: 25px;  
    margin: 25px;  
}
```

overflow

Esta propiedad especifica al navegador como actuar con el contenido excedente de un elemento de bloque. Cuando los elementos tienen un tamaño definido puede ocurrir que éste no pueda verse en su totalidad dentro de la caja. Por defecto, los navegadores mostrarán el resto del contenido fuera del área de la caja pudiendo posicionarse sobre el contenido de otros elementos. Con ***overflow*** configuramos el comportamiento del navegador en estos casos. Los valores disponibles son:

visible. Se muestra el contenido, es el valor por defecto.

hidden. Esconde el contenido que no entra dentro de la caja.

scroll. Muestra barras de desplazamiento),

auto. Es el navegador el que decide qué hacer con el contenido.

overflow-x

Esta propiedad especifica cómo se mostrará el contenido que desborda el elemento horizontalmente. Acepta los mismos valores que la propiedad ***overflow***.

overflow-y

Esta propiedad especifica cómo se mostrará el contenido que desborda el elemento verticalmente. Acepta los mismos valores que la propiedad ***overflow***.

overflow-wrap

Esta propiedad indica si una palabra debería ser dividida en un punto arbitrario cuando no hay suficiente espacio para mostrarla en la línea. Los valores disponibles son:

normal. La línea será dividida naturalmente.

break-word. Las palabras se dividirán en puntos arbitrarios para acomodar la línea de texto en el espacio disponible.

Actividad.

Probamos el siguiente ejemplo.

```
section {  
  background-color: #eee;  
  width: 200px;  
  height: 50px;  
  border: 1px dotted black;  
  overflow: visible  
};
```

1º DAW. LM

"Neque porro quisquam est qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit..."

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Curabitur iaculis vulputate
placerat. Integer pulvinar
aliquet sem id ultrices. .

Modificamos el valor de **overflow** y observamos el resultado.

Preguntas.

¿Qué es una seudoclase? Pon un ejemplo.

¿Qué selector utilizarías para referenciar el siguiente elemento?

```
<p class="principal"> ...</p>
```

¿Qué selector utilizarías para referenciar el siguiente elemento?

```
<p id="p1"> ...</p>
```

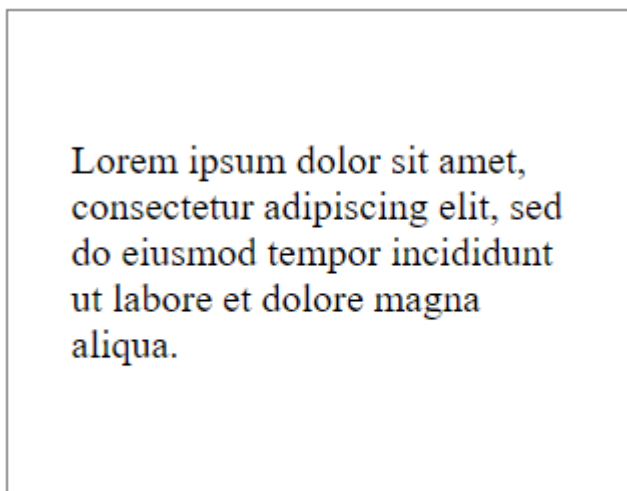
Actividades propuestas.

Bloque 1.

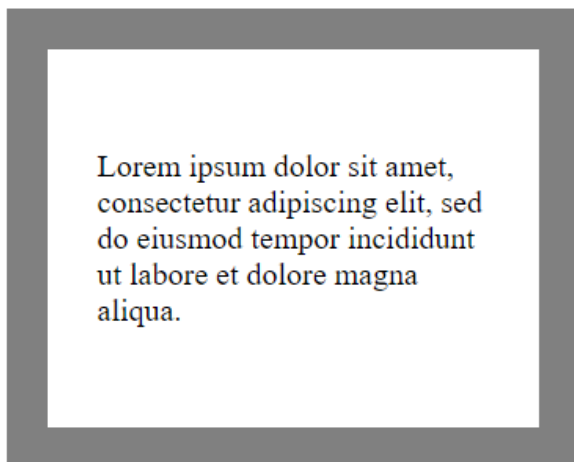
- 1 Escribe un documento html y css para conseguir un resultado similar al siguiente.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

- 2 Escribe documento html y css para conseguir un resultado similar al siguiente. Se trata de un elemento con borde y el relleno de arriba y de abajo es el doble que el relleno lateral.



- 3 Escribe documento html y css para conseguir un resultado similar al siguiente. Tiene que aumentar el ancho de borde del ejercicio anterior.



Texto.

Las propiedades de fuente CSS definen la familia de fuentes, negrita, tamaño y estilo de un texto.

En CSS hay dos tipos familia de fuentes:

Familia genérica: un grupo de familias de fuentes con un aspecto similar (como "Serif" o "Monospace")

Familia de fuentes: una familia de fuentes específica (como "Times New Roman" o "Arial")

Familia genérica	Familia de fuentes	Descripción
Serif	Times New Roma Gorgia	Pequeño remate de las líneas de los caracteres. Facilita la lectura
Sans-serif	Arial Verdana	No presentan serif
Monospace	Courier New Lucida Console	Misma anchura para todos los caracteres.

font-family

Esta propiedad declara el tipo de letra que se usa para mostrar el texto. Se pueden declarar múltiples valores separados por coma para ofrecer al navegador varias alternativas en caso de que algunos tipos de letra no se encuentren disponibles en el ordenador del usuario. Las fuentes que suelen estar en todos los ordenadores se suelen llamar fuentes seguras y garantizan la carga correcta desde cualquier equipo. Algunos de los valores estándar son Georgia, Times New Roman, Arial, Helvetica, "Arial Black", Gadget, Tahoma, Geneva, Helvetica, Verdana, Geneva, Impact, y sans-serif .

Esta propiedad determina el tamaño de la letra. El valor puede ser declarado en píxeles (px), porcentaje (%), o usando cualquiera de las unidades disponibles en CSS.

font-weight

Esta propiedad determina si el texto se mostrará en negrita o no. Los valores disponibles son: ***normal, bold***

Algunos tipos de letra permiten utilizar un valor numérico para especificar el grosor.

font-style

Esta propiedad determina el estilo de la letra. Los valores disponibles son: ***normal, italic, oblique***.

font

Esta propiedad permite establecer un conjunto de valores simultáneamente. Los valores deben declararse separados por un espacio y en orden: Estilo, grosor, tamaño, tipo de letra.

Actividad.

Supongamos la siguiente regla.

```
p.b{  
    font: italic bold 12px/30px Georgia, serif;  
}
```

¿Qué etiqueta de apertura utilizamos para aplicar al párrafo la regla declarada?

```
<????>This is a paragraph. The font is set to italic and bold, the  
font size is set to 12 pixels, the line height is set to 30 pixels,  
and the font family is Georgia.</p>
```

¿Qué significa 12px/30px?

@font-face

Se trata de una regla reservada para incluir un archivo con la tipografía que vamos a utilizar. Los navegadores no descargan las fuentes desde el servidor web, las cargan desde el ordenador del usuario. Si no utilizamos fuentes seguras, nuestros sitios web pueden verse de manera diferente en distintos ordenadores y lamentablemente el número de fuentes seguras es escaso. La regla **@font-face** necesita al menos dos propiedades para declarar la fuente y cargar el archivo. La propiedad **font-family** con el nombre que queremos usar para referenciar este tipo de letra y la propiedad **src** con la URL del archivo con las especificaciones de la fuente (requiere el uso de la función `url()`).

```
@font-face {  
    font-family: fuentePersonal;  
    src: url(sansation_light.woff);  
}
```

text-align

Esta propiedad alinea el texto dentro de un elemento. Los valores disponibles son: **left, right, center, justify**.

text-align-last

Esta propiedad alinea la última línea de un párrafo. Los valores disponibles son: **left, right, center, justify**.

text-indent

Esta propiedad define el tamaño de la sangría de un párrafo.

text-indentation.

Se utiliza para especificar la sangría en la primera línea de un texto.

text-decoration

Esta propiedad resalta el texto con una línea. Los valores disponibles son ***underline***, ***overline***, ***line-through***, y ***none***. La propiedad ***text-decoration*** es particularmente útil con enlaces. Por defecto, los navegadores muestran los enlaces subrayados. Si queremos eliminar la línea, podemos declarar esta propiedad con el valor ***none***.

text-shadow

Se utiliza para poner sombra al texto. Permite utilizar color.

text-transform

Se utiliza para especificar letras mayúsculas y minúsculas en un texto: ***uppercase***, ***lowercase***, ***capitaliza***.

letter-spacing

Esta propiedad define el espacio entre letras.

word-spacing

Esta propiedad define el ancho del espacio entre palabras.

vertical-align

Esta propiedad alinea elementos verticalmente. Se usa frecuentemente para alinear texto con imágenes (la propiedad se aplica a la imagen). Los valores disponibles son: ***baseline***, ***sub***, ***super***, ***text-top***, ***text-bottom***, ***middle***, ***top***, y ***bottom***. Por defecto, los navegadores alinean el texto a la izquierda, pero podemos cambiar la alineación con la propiedad ***text-align***.

line-height.

Se utiliza para especificar el espacio entre líneas.

direction.

Se utiliza para cambiar la dirección del texto de un elemento. De derecha a izquierda lo establecemos con el valor: **rtl**

Actividades propuestas.

Bloque 2

- 1 Corrige, completa y explica el significado del siguiente código.

```
<!DOCTYPE html>
<title>Ej Fuentes Google</title>

<link href="https://fonts.googleapis.com/css?family=Fredericka+the+Great" rel="stylesheet">

.example {
  font-family: 'Fredericka the Great', cursive;
  font-size: 7vw;
}
```

- 2 Centra el encabezado del siguiente documento.

Encabezado del documento

Primer párrafo.

Segundo párrafo.

- 3 Suprime el subrayado en el enlace del siguiente documento.

Encabezado del documento

Lista de enlaces.

[Puente Romano](#)

- 4 Utiliza transformaciones de texto para que se muestren en mayúsculas el encabezado y las iniciales del primer párrafo.

Encabezado del documento

Primer párrafo.

Segundo párrafo.

- 5 Establece en una sangría de 20 píxeles en los párrafos del siguiente documento.

Encabezado del documento

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

- 6 Escribe un documento HTML/CSS para conseguir un resultado similar al siguiente. [font-family en W3C](#)

Example: font-family

Examples

This is an example of a serif font.

This is an example of a sans-serif font.

This is an example of a monospace font.

This is an example of a cursive font.

This is an example of a fantasy font.

- 7 Escribe un documento HTML/CSS para conseguir un resultado similar al de la imagen. El segundo párrafo del documento está en verdana, cursiva, y un 20% mayor al tamaño de la fuente 1.2 em.

Encabezado del documento

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

- 8 Escribe un documento HTML/CSS para conseguir un resultado similar al siguiente. La primera línea del segundo párrafo es roja.

Encabezado del documento

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

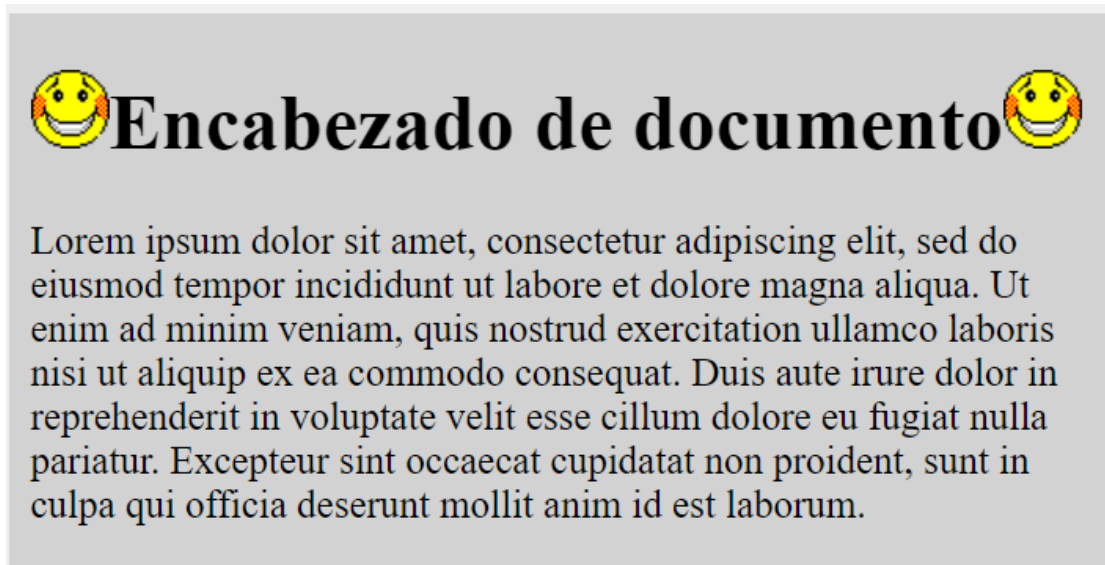
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

- 9 Escribe un documento HTML/CSS para conseguir un resultado similar al siguiente. La primera letra de la primera línea es roja y xx-large

Encabezado del documento

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

- 10 Escribe un documento HTML/CSS para conseguir un resultado similar al de la imagen. smile.gif antes y después del encabezado.



- 11 Escribe un documento HTML/CSS para conseguir un resultado similar al siguiente.

"Neque porro quisquam est qui dolorem..."

- 12 Escribe un documento HTML/CSS con una lista horizontal sin decoración.

Lista en horizontal

HTML CSS JavaScript

Colores

color

Establece color de texto. Puede especificarse utilizando los siguientes formatos:

rgb(rojo, verde, azul). Esta función define un color por medio de los valores especificados por los atributos (desde 0 a 255). El primer valor representa el nivel de rojo, el segundo valor representa el nivel de verde y el último valor el nivel de azul.

rgba(rojo, verde, azul, alfa). Esta función es similar a la función **rgb()**, pero incluye un componente adicional para definir la opacidad (alfa). El valor se puede declarar entre 0 y 1, con 0 como totalmente transparente y 1 como totalmente opaco.

hsl(matiz, saturación, luminosidad). Esta función define un color desde los valores especificados por los atributos. Los valores se declaran en números decimales y porcentajes.

hsla(matiz, saturación, luminosidad, alfa). Esta función es similar a la función **hsl()**, pero incluye un componente adicional para definir la opacidad (alfa). El valor se puede declarar entre 0 y 1, con 0 como totalmente transparente y 1 como totalmente opaco. nombre, valor hexadecimal o valor RGB

Número en hexadecimal #

[Nombre](#)

Recursos:

[Materialize.](#)

[Calculadora de colores](#)

[Enciclopedia de colores](#)

opacity

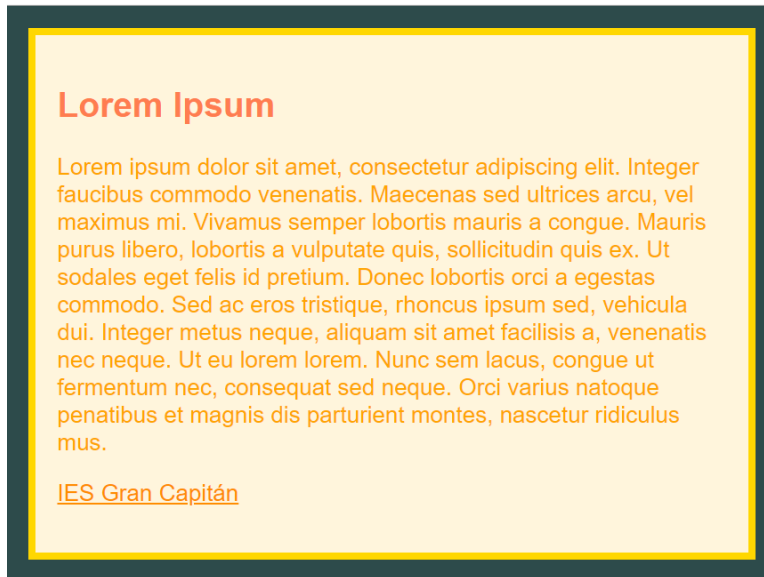
Permite declarar la opacidad de un elemento. Presenta el inconveniente de que su valor es heredado por los hijos. Se resuelve con las funciones **rgba()** y **hlsa()** .

Actividades propuestas.

Bloque 3.

- 1 Escribe un documento HTML/CSS para conseguir un resultado similar al siguiente. Cambia los estilos del enlace en función de su estado.

ej1_colores.html



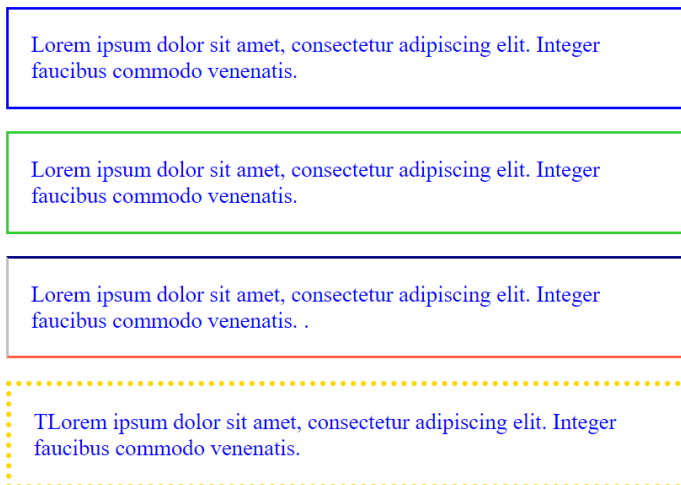
- 2 Prueba la siguiente regla **css** y explica el significado del valor marcado.

```
.box {  
    background-color: gold;  
    font-size: 5vw;  
    padding: 10vw;  
    margin: 5vw;  
    border: solid;  
}
```

Lenguaje Marcas

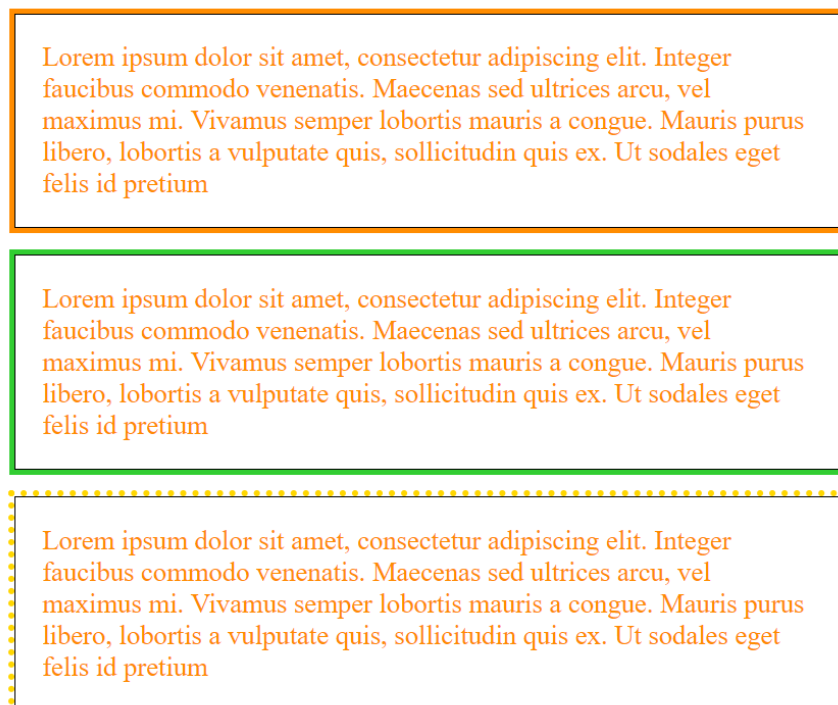
- 3 Escribe un documento HTML/CSS para conseguir un resultado similar al siguiente. El fondo del cuadro cambia a gris al pasar el ratón por encima.

ej2_colores.html



- 4 Escribe un documento HTML/CSS para conseguir un resultado similar al siguiente. Consulta la propiedad **outline**.

ej4_colores.html



- 5 Prueba la siguiente regla y explica el significado de la propiedad marcada.

```
input {  
  caret-color: orange;  
  color: white;  
  background: black;  
  font-size: 7vw;  
  padding: 3vw;  
}
```

- 6 Escribe un documento HTML/CSS para conseguir un resultado similar al siguiente.

ej5_colores.html



Fondo

Los elementos pueden incluir un fondo que se muestra detrás del contenido del elemento y a través del área ocupada por el contenido y el relleno. El fondo puede estar compuesto por colores e imágenes.

background-color

Esta propiedad asigna un fondo de color a un elemento.

background-image

Esta propiedad asigna una o varias imágenes al fondo de un elemento. La URL del archivo se declara con la función `url()` . Si se requiere más de una imagen, los valores se deben separar por una coma.

background-position

Esta propiedad declara la posición de comienzo de una imagen de fondo. Los valores se pueden especificar en función de la posición: **center**, **left**, **right**, **top**, y **bottom**.

background-size

Esta propiedad declara el tamaño de la imagen de fondo. Los valores se pueden especificar en porcentaje, píxeles, o usando las palabras **cover** y **contain**. La palabra clave **cover** expande la imagen hasta que su ancho o su altura cubren el área del elemento, mientras que **contain** estira la imagen para ocupar toda el área del elemento.

background-repeat

Esta propiedad determina cómo se distribuye la imagen de fondo: **repeat**, **repeat-x**, **repeat-y** y **no-repeat**.

background-origin

Esta propiedad determina si la imagen de fondo se posicionará considerando el borde, el relleno o el contenido del área del elemento. Los valores disponibles son:

border-box, ***padding-box***, y ***content-box***.

background-clip

Esta propiedad declara el área a cubrir por el fondo usando los valores ***border-box***, ***padding-box***, y ***content-box***. El primer valor corta la imagen al borde de la caja del elemento, el segundo corta la imagen en el relleno de la caja y el tercero corta la imagen alrededor del contenido de la caja.

background-attachment

Esta propiedad determina si la imagen es estática o se desplaza con el resto de los elementos usando dos valores: ***scroll*** (por defecto) y ***fixed***.

background

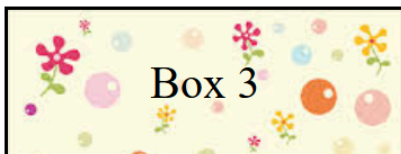
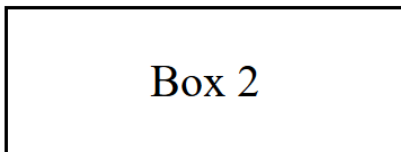
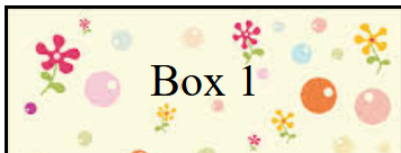
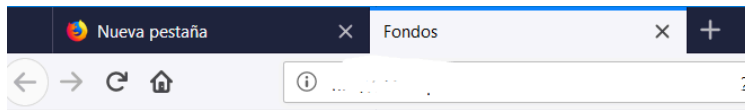
Esta propiedad nos permite declarar todos los atributos del fondo al mismo tiempo.

Actividades propuestas.

Bloque 4.

- 1 Escribe un documento HTML/CSS para conseguir un resultado similar al siguiente.

ej1_fondos.html



- 2 Escribe un documento html y css para conseguir un resultado similar al siguiente.

ej2_fondos.html

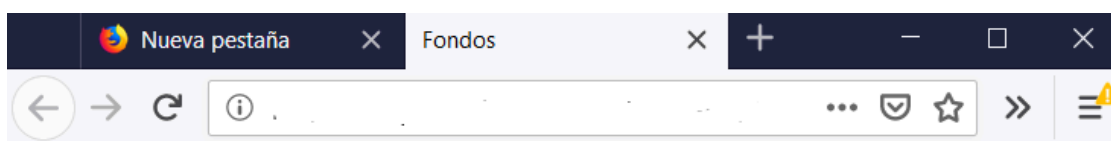
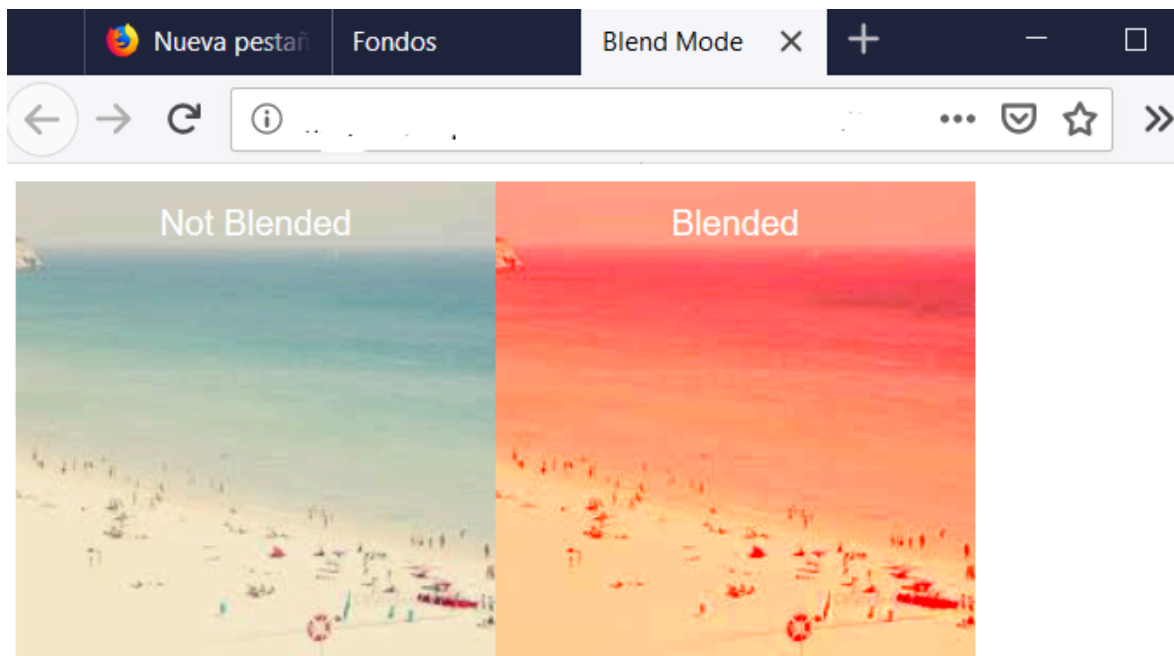


Imagen de fondo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer faucibus commodo venenatis. Maecenas sed ultrices arcu, vel maximus mi

- 3 Escribe un documento HTML/CSS para conseguir un resultado similar al siguiente. Consulta ***background-blend-mode***.

e3_fondos.htm



- 4 Prueba en un documento las siguientes reglas css. Indica para que se utiliza la propiedad ***mix-blend-mode***

```
div {  
  background-image: url(2.jpg);  
  background-size: cover;  
  background-repeat: no-repeat;  
  font-family: georgia, serif;  
  font-size: 7vw;  
  text-align: center;  
  border: solid;  
}  
.blender {  
  mix-blend-mode: difference;  
  color: yellow;  
}
```


Bordes

Los elementos pueden incluir un borde en los límites de la caja del elemento. Por defecto, los navegadores no muestran ningún borde, pero podemos usar las siguientes propiedades para definirlo.

border-width

Esta propiedad define el ancho del borde. Acepta hasta cuatro valores separados por un espacio para especificar el ancho de cada lado del borde (superior, derecho, inferior, e izquierdo). También podemos declarar el ancho para cada lado de forma independiente con las propiedades ***border-top-width***, ***borderbottom-width***, ***border-left-width***, y ***border-right-width***.

border-style

Esta propiedad define el estilo del borde. Acepta hasta cuatro valores separados por un espacio para especificar los estilos de cada lado del borde (superior, derecho, inferior, e izquierdo, en ese orden). Los valores disponibles son ***none***, ***hidden***, ***dotted***, ***dashed***, ***solid***, ***double***, ***groove***, ***ridge***, ***inset***, y ***outset***. El valor por defecto es ***none***, lo que significa que el borde no se mostrará a menos que asignemos un valor diferente a esta propiedad. También podemos declarar los estilos de forma independiente con las propiedades ***border-top-style***, ***border-bottom-style***, ***border-leftstyle***, y ***border-right-style***.

border-color

Esta propiedad define el color del borde. Acepta hasta cuatro valores separados por un espacio para especificar el color de cada lado del borde (superior,derecho, inferior, e izquierdo). También podemos declarar los colores de forma independiente con las propiedades ***border-top-color***, ***border-bottom-color***, ***border-left-color***, y ***border-right-color***

border-radius

Esta propiedad define el radio del círculo virtual que el navegador utilizará para dibujar las esquinas redondeadas. Acepta hasta cuatro valores para definir los radios de cada esquina (superior izquierdo, superior derecho, inferior derecho e inferior izquierdo). También podemos usar las **propiedades *border-top-left-radius*, *border-top-right-radius*, *border-bottom-right-radius*, y *border-bottom-left-radius*** para definir el radio de cada esquina de forma independiente.

outline-style

Esta propiedad define el estilo de un segundo borde. Los valores disponibles son ***none*, *auto*, *dotted*, *dashed*, *solid*, *double*, *groove*, *ridge*, *inset*, y *outset***.

outline-color

Esta propiedad define el color del borde.

outline-offset

Esta propiedad define el desplazamiento (a qué distancia de los límites de la caja se dibujará el segundo borde). Acepta valores en cualquiera de las unidades disponibles en CSS (px, %, em, etc.).

outline

Esta propiedad nos permite especificar el ancho, estilo y color del borde al mismo tiempo (el desplazamiento aún se debe definir con la propiedad ***outline-offset***). Por defecto, el desplazamiento se declara con el valor 0, por lo que el segundo borde se dibujará a continuación del borde de la caja. Si queremos separar los dos bordes, tenemos que definir la propiedad ***outline-offset***.

border-image-source

Se utiliza para crear bordes personalizados. La propiedad determina la imagen que se usará para crear el borde. La URL del archivo se declara con la función url().

border-image-width

Esta propiedad define el ancho del borde. Acepta valores en cualquiera de las

unidades disponibles en CSS (px, %, em, etc.).

border-image-repeat

Esta propiedad define cómo se usa la imagen para generar el borde. Los valores disponibles son **repeat**, **round**, **stretch**, y **space**.

border-image-slice

Esta propiedad define cómo se va a cortar la imagen para representar las esquinas del borde. Debemos asignar cuatro valores para especificar los cuatro trozos de la imagen que se utilizarán (si solo se declara un valor, se usa para todos los lados). El valor se puede especificar como un entero o en porcentaje.

border-image-outset

Esta propiedad define el desplazamiento del borde (la distancia a la que se encuentra de la caja del elemento). Acepta valores en cualquiera de las unidades disponibles en CSS (px, %, em, etc.).

border-image

Esta propiedad establece una imagen como el borde alrededor de un elemento.

border

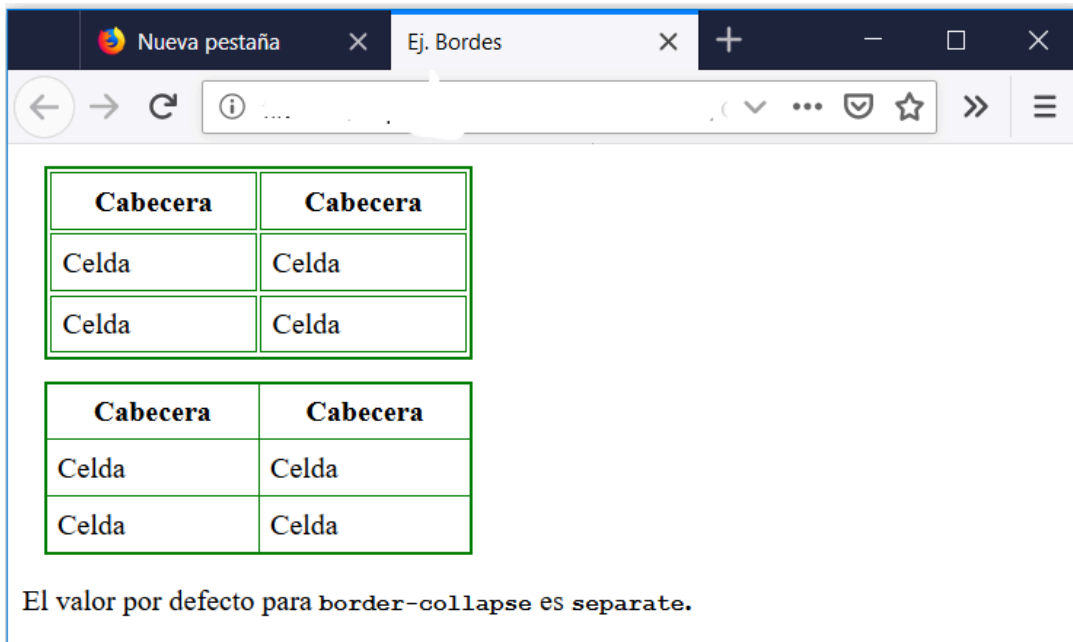
Esta propiedad nos permite especificar todos los atributos del borde al mismo tiempo.

Actividades propuestas.

Bloque 5.

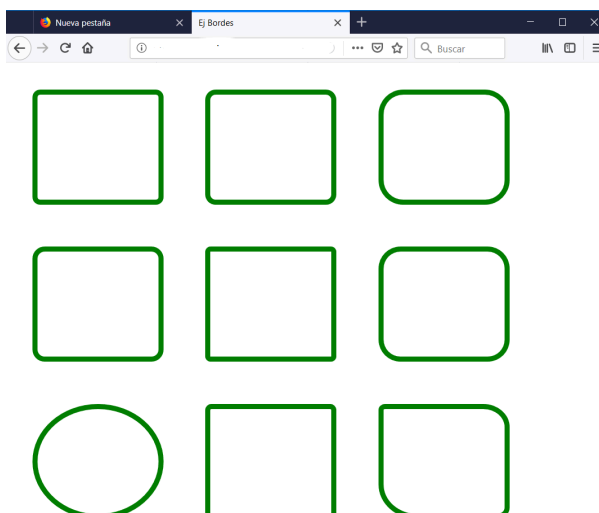
- 1 Crea un estilo de borde y aplícalo a cuatro elementos ***div***.
- 2 Escribe un documento HTML/CSS para conseguir un resultado similar al siguiente.

ej1_bordes.html



- 3 Aplica estilos al horario semanal creado en las actividades de la unidad anterior.
- 4 Escribe un documento HTML/CSS para conseguir un resultado similar al siguiente. Consulta la propiedad ***float***.

ej2_bordes.html



Sombras

Otro efecto interesante que podemos aplicar a un elemento html es la sombra. CSS incluye propiedades para generar sombras para la caja de un elemento y también para formas irregulares como texto.

box-shadow

Esta propiedad genera una sombra desde la caja del elemento. Acepta hasta seis valores. Podemos declarar el desplazamiento horizontal y vertical de la sombra, el radio de difuminado, el valor de propagación, el color de la sombra, y también podemos incluir el valor ***inset*** para indicar que la sombra deberá proyectarse dentro de la caja.

text-shadow

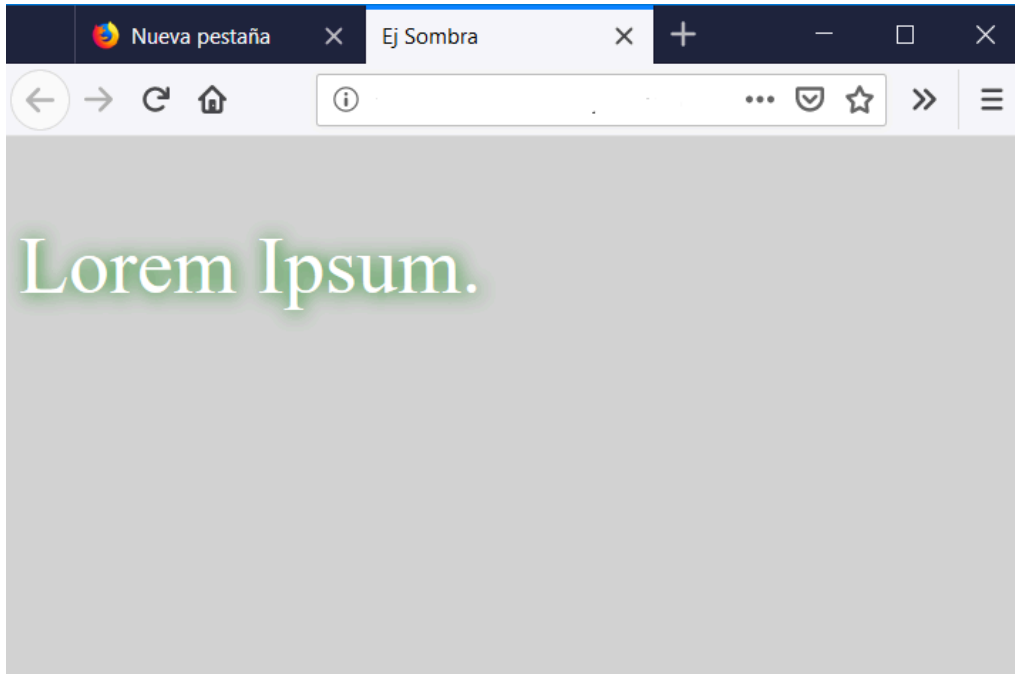
Esta propiedad genera una sombra desde un texto. Acepta hasta cuatro valores. Podemos declarar el desplazamiento horizontal y vertical, el radio de difuminado y el color de la sombra.

Actividades propuestas.

Bloque 6.

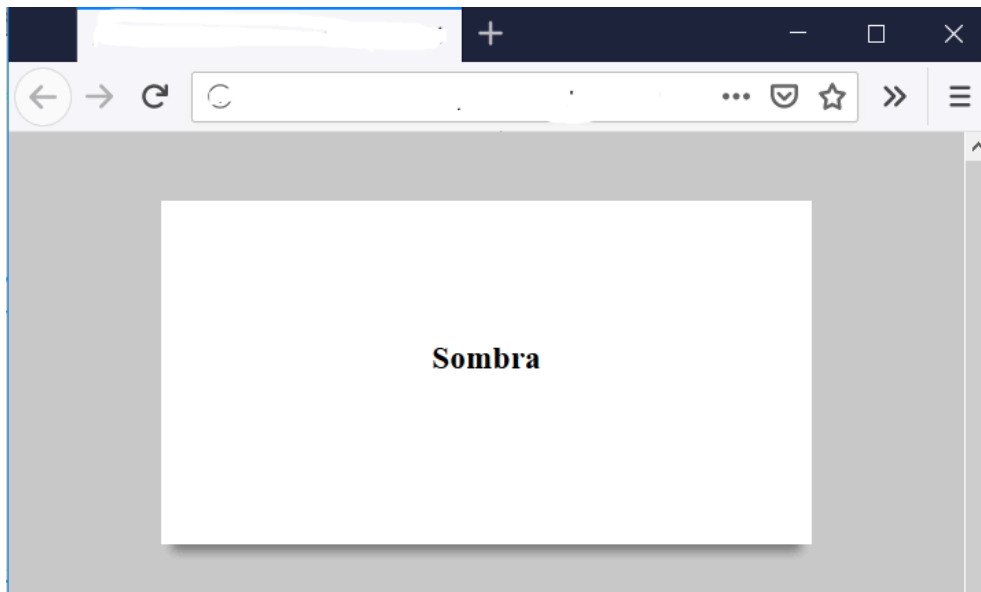
- 1 Escribe un documento HTML/CSS para conseguir un resultado similar al siguiente.

ej1_sombra.html



- 2 Escribe un documento HTML/CSS para conseguir un resultado similar al siguiente.

ej2_sombra.html



Gradientes

Un gradiente se forma mediante una serie de colores que varían continuamente con una transición suave de un color a otro. Los gradientes se crean como imágenes y se agregan al fondo del elemento con las propiedades ***background-image*** o ***background***.

linear-gradient(posición, ángulo, colores)

Esta función crea un gradiente lineal. El atributo posición determina el lado o la esquina desde la cual comienza el gradiente y se declara con los valores ***top***, ***bottom***, ***left*** y ***right***; el atributo ángulo define la dirección del gradiente y se puede declarar en las unidades ***deg*** (grados), ***grad*** (gradianes), ***rad*** (radianes), o ***turn*** (espiras), y el atributo colores es la lista de colores que participan en el gradiente separados por coma. Los valores para el atributo colores pueden incluir un segundo valor en porcentaje separado por un espacio para indicar la posición donde finaliza el color.

radial-gradient(posición, forma, colores, extensión)




Esta función crea un gradiente radial. El atributo posición indica el origen del gradiente y se puede declarar en píxeles, en porcentaje, o por medio de la combinación de los valores ***center***, ***top***, ***bottom***, ***left*** y ***right***; el atributo forma determina la forma del gradiente y se declara con los valores ***circle*** y ***ellipse***; el atributo colores es la lista de los colores que participan en el gradiente separados por coma; el atributo extensión determina la forma que el gradiente va a adquirir con los valores ***closest-side***, ***closest-corner***, ***farthest-side***, y ***farthest-corner***. Los valores para el atributo colores pueden incluir un segundo valor en porcentaje separado por un espacio para indicar la posición donde finaliza el color.

La sintaxis para gradientes radiales es similar a la utilizada con los gradientes lineales. La única diferencia es que tenemos que usar la función ***radial-gradient()*** en lugar de la función ***linear-gradient()*** e incluir un parámetro que determina la forma del gradiente con los valores ***circle*** o ***ellipse***.

La posición del gradiente se puede personalizar y podemos usar varios colores con un segundo valor para determinar los límites de cada uno de ellos.

Actividad. Gradiente.

Crea una copia de la plantilla de trabajo para realizar el ejercicio.

	css	26/11/2019 8:35	Carpeta de archivos	
	img	26/11/2019 8:28	Carpeta de archivos	
	index	26/11/2019 8:47	Chrome HTML Do...	1 KB

Edita y modifica el fichero index.html añadiendo.

```
<div class="myclass">
  Ejemplo de gradiente radial.
</div>
```

Añade título y favicon al documento.

...

```
<title>Ejemplo. Gradiente</title>
```

...

```
<link rel="shortcut icon" href="img/favicon.ico" />
```

Edita y modifica el fichero style.css



```
.myclass {
  text-align: center;
  font-size: 5vw;
  padding: 10vw;
  border: solid;
  background: repeating-radial-gradient(grey 20%, white 40%);
}
```

Visualiza el documento en el navegador.

Modifica los ficheros para añadir un gradiente lineal.



```
<div class="myclass">
  Ejemplo de gradiente lineal.
</div>
```


Actividades propuestas.

Bloque 7

Utiliza y describe la siguiente regla css.

```
.box {  
  font-size: 5vw;  
  padding: 10vw;  
  border: solid;  
  background: repeating-radial-gradient(circle, yellow 20%, purple 40%);  
}
```

Utiliza y describe las siguientes reglas css.

```
body {  
  background-image: url('1.png');  
}  
.box {  
  font-size: 5vw;  
  padding: 10vw;  
  border: solid;  
  background: linear-gradient(blue, transparent, purple);  
}
```

Filtros

Los filtros agregan efectos a un elemento y su contenido.

filter

Se utiliza para asignar un filtro a un elemento. Se utilizan las siguientes funciones para crearlo:

blur(valor). Esta función produce un efecto de difuminado. Acepta valores en píxeles desde 1px a 10px.

grayscale(value). Esta función convierte los colores de la imagen en una escala de grises. Acepta números decimales entre 0.1 y 1.

drop-shadow(x, y, tamaño, color). Esta función genera una sombra. Los atributos x e y determinan la distancia entre la sombra y la imagen, el atributo tamaño especifica el tamaño de la sombra, y el atributo color declara su color.

sepia(valor). Esta función le otorga un tono sepia (ocre) a los colores de la imagen. Acepta números decimales entre 0.1 y 1.

brightness(valor). Esta función cambia el brillo de la imagen. Acepta números decimales entre 0.1 y 10.

contrast(valor). Esta función cambia el contraste de la imagen. Acepta números decimales entre 0.1 y 10.

hue-rotate(valor). Esta función aplica una rotación a los matices de la imagen. Acepta un valor en grados desde 1deg a 360deg.




invert(valor). Esta función invierte los colores de la imagen y produce un negativo. Acepta números decimales entre 0.1 y 1.

saturate(valor). Esta función satura los colores de la imagen. Acepta números decimales entre 0.1 y 10.

opacity(valor). Esta función cambia la opacidad de la imagen. Acepta números decimales entre 0 y 1 (0 es totalmente transparente y 1 totalmente opaco).

Actividad. Filtros.

Crea una copia de la plantilla de trabajo para realizar el ejercicio.

	css	26/11/2019 8:35	Carpeta de archivos	
	img	26/11/2019 8:28	Carpeta de archivos	
	index	26/11/2019 8:47	Chrome HTML Do...	1 KB


Descarga la foto de una guitarra acústica. No olvides respetar las licencias de uso de los recursos utilizados.

Edita y modifica el fichero index.html añadiendo.

```

```

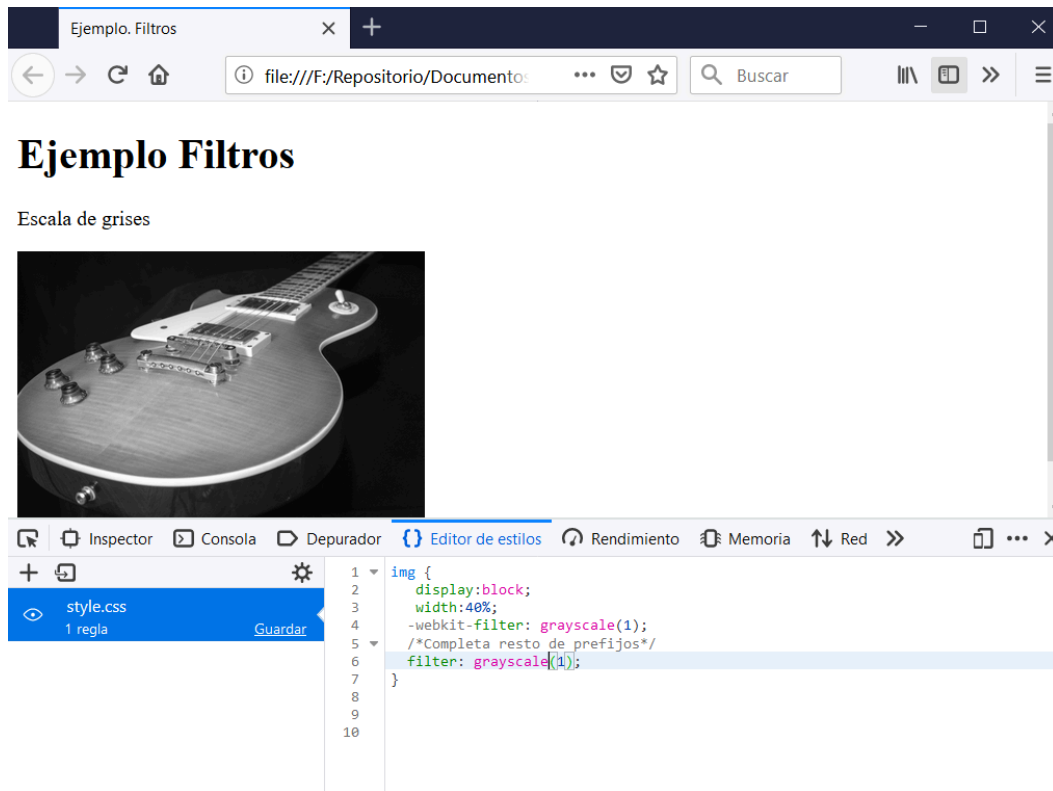
Edita y modifica el fichero style.css

```
img {  
    display: block;   
    width: 40%;  
    -webkit-filter: grayscale(1);  
    /*Completa resto de prefijos*/  
    filter: grayscale(1);  
}
```

Visualiza el documento en el navegador.

Lenguaje Marcas

Abre la ventana de depuración del navegador (F12) y accede al editor de estilos.



Prueba utilizando el editor de estilos las propiedades: sepia, saturate, invert, etc

Actividades propuestas.

Bloque 8

- 1 Crea un documento css con las siguientes reglas e incorpóralo a una página de prueba.

```
div {  
  color: red;  
  background-color: gold;  
  font-size: 4vw;  
  padding: 2vw;  
  margin: 2vw;  
  border: solid;  
}  
.contrast {  
  filter: contrast(100);  
}  
.brightness {  
  filter: brightness(50%);  
}  
.inverted {  
  filter: invert(100%);  
}  
.rotated {  
  filter: hue-rotate(200deg);  
}  
.grayscale {  
  filter: grayscale();  
}
```

Transformaciones

Una vez que se crean los elementos HTML, estos permanecen inmóviles, pero podemos modificar su posición, tamaño y apariencia por medio de transformaciones.

transform

Esta propiedad realiza cuatro transformaciones básicas a un elemento: escalado, rotación, inclinación y traslación. Las siguientes son las funciones definidas para este propósito.

scale(x, y). Esta función modifica la escala del elemento. Existen otras dos funciones relacionadas llamadas ***scaleX()*** y ***scaleY()*** para especificar los valores horizontales y verticales independientemente.

rotate(ángulo). Esta función rota el elemento. El atributo representa los grados de rotación y se puede declarar en ***deg*** (grados), ***grad*** (gradianes), ***rad*** (***radianes***) o ***turn*** (***espiras***).

skew(ángulo). Esta función inclina el elemento. El atributo representa los grados de desplazamiento. La función puede incluir dos valores para representar el ángulo horizontal y vertical. Los valores se pueden declarar en ***deg*** (grados), ***grad*** (gradianes), ***rad*** (radianes) o ***turn*** (espiras).

translate(x, y). Esta función desplaza al elemento a la posición determinada por los atributos x e y. La pantalla de un dispositivo se divide en filas y columnas de píxeles (la mínima unidad visual de la pantalla). Con el propósito de identificar la posición de cada píxel, los ordenadores usan un sistema de coordenadas, donde las filas y columnas de píxeles se cuentan desde la esquina superior izquierda a la esquina inferior derecha de la cuadrícula, comenzando por el

valor 0 (los valores se incrementan de izquierda a derecha y de arriba bajo)

skew(). Esta función cambia la simetría del elemento en grados y en una o ambas dimensiones. Esta función recibe dos valores, pero a diferencia de otras funciones, cada parámetro solo afecta a una dimensión (los parámetros son independientes). Existen las funciones adicionales ***skewX()*** y ***skewY()*** para lograr el mismo efecto.

Se pueden aplicar varias transformaciones simultáneamente, para ello tenemos que declarar las funciones separadas por un espacio. Cuando combinamos múltiples funciones, debemos considerar que el orden de combinación es importante. Las funciones pueden modificar el punto de origen y el centro del elemento y, por lo tanto, cambian los parámetros sobre los que el resto de las funciones trabajan.

De la misma manera que podemos generar transformaciones en dos dimensiones sobre elementos HTML, también podemos hacerlo en tres dimensiones. Estos tipos de transformaciones se realizan considerando un tercer eje que representa profundidad y se identifica con la letra z.

scale3d(x, y, z)

Esta función asigna una nueva escala al elemento en un espacio 3D.

Acepta tres valores en números decimales para establecer la escala en los ejes x, y y z. Al igual que con transformaciones 2D, el valor 1 preserva la escala original.

rotate3d(x, y, z, ángulo)

Esta función rota el elemento en un ángulo y sobre un eje específicos. Los valores para los ejes se deben especificar en números decimales y el ángulo se puede expresar en deg (grados), grad (gradianes), rad (radianes), o turn (espiras). Los valores asignados a los ejes determinan un vector de rotación, por lo que los valores no son importantes, pero sí lo es la relación entre los mismos.

translate3d(x, y, z)

Esta función mueve el elemento a una nueva posición en el espacio 3D. Acepta tres valores en píxeles para los ejes x, y y z.

perspective(valor)

Esta función agrega un efecto de profundidad a la escena incrementando el tamaño del lado del elemento cercano al espectador.

Algunas transformaciones 3D se pueden aplicar directamente al elemento, como hemos hecho con las transformaciones 2D, pero otras requieren que primero declaremos la perspectiva.

CSS también incluye algunas propiedades que podemos usar para lograr un efecto más realista.

perspective

Esta propiedad trabaja de forma similar a la función `perspective()`, pero opera en el elemento padre. La propiedad crea un contenedor que aplica el efecto de perspectiva a los elementos en su interior.

perspective-origin

Esta propiedad cambia las coordenadas x e y del espectador. Acepta dos valores en porcentaje, píxeles, o las palabras clave ***center, left, right, top y bottom***. Los valores por defecto son 50% 50%.

backface-visibility

Transformaciones dinámicas.

Podemos crear transformaciones de forma dinámica utilizando la pseudo clase ***:hover***

Actividades propuestas.

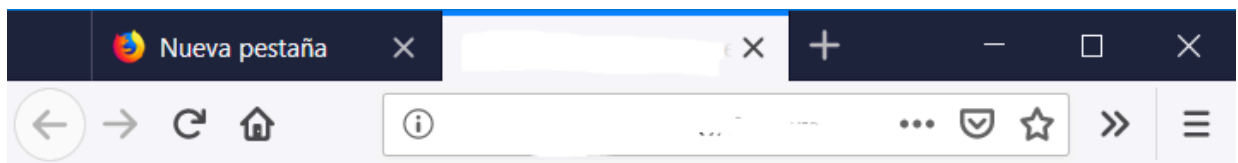
Bloque 9.

- 1 Añade una transformación a la siguiente regla que genere un desplazamiento de 100px en el eje X y 200px en el eje Y.

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: green;  
  border: 1px solid black;  
}
```

- 2 Aplica una transformación que rote el elemento 45°. Se trata de un elemento div con fondo verde.

Ej1_tranformaciones.html



Transiciones

transition-property

Esta propiedad especifica las propiedades que participan en la transición. Además de los nombres de las propiedades, podemos asignar el valor **all** para indicar que todas las propiedades participarán de la transición.

transition-duration

Esta propiedad especifica la duración de la transición en segundos (s).

transition-timing-function

Esta propiedad determina la función que se usa para calcular los valores para la transición. Los valores disponibles son **ease**, **ease-in**, **easeout**, **ease-in-out**, **linear**, **step-start**, y **step-end**.

transition-delay

Esta propiedad especifica el tiempo que el navegador esperará antes de iniciar la animación.

transition

Esta propiedad nos permite declarar todos los valores de la transición al mismo tiempo. La propiedad **transition** puede recibir hasta cuatro parámetros separados por un espacio. El primer valor es la propiedad que se considerará para crear la transición, el segundo parámetro determina la duración de la animación, el tercer parámetro es un valor que determina cómo se llevará a cabo la transición por medio de una curva Bézier, y el último parámetro determina cuántos segundos tarda la animación en comenzar.

Si la transición tiene que considerar los valores de más de una propiedad, tenemos que declarar los nombres de las propiedades separadas por coma. Cuando se tienen que considerar todas las propiedades que se modifican para crear la animación, podemos usar el valor **all** en su lugar.

Actividades propuestas.

Bloque 10

- 1 Crea un documento HTML que aplique las siguientes reglas.

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition: width 2s;  
  transition-timing-function: ease-in-out;  
}  
div:hover {  
  width: 300px;  
}
```

- 2 Prueba las siguientes reglas y analiza su funcionamiento.

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition: background 2s, transform 2s;  
}  
div:hover {  
  background: blue;  
  transform: rotate(180deg);  
}
```

3 Prueba las siguientes reglas y analiza su funcionamiento.

ej2_transiciones.html

```
#transicion{
  margin:30px auto;
  width:200px;
  height:200px;
  background-color:#16a085;
  color:#16a085;
  line-height:300px;
  font-size:25px;
  text-align:center;
}
#transicion:hover{
  width:400px;
  height:400px;
  border-radius:60%;
  background-color:#2c3e50;
  cursor:pointer;
  padding:30px;
  box-shadow:0px 2px 20px #000;
  color:#fff;
  transition: width 1s ease, height 1s ease, background-color 2s ease-out,
              border-radius 2s ease-in-out, padding 1s ease 2s, box-shadow 1s ease 2s,
              color 3s linear;
  /*Repetir transition para -moz- -o- -ms- -webkit-*/
}
```

- 4 Utilizando como referencia los ejemplos anteriores, diseña e implementa una transición con el logo creado para el proyecto “Córdoba, Ciudad Patrimonio”.
- 5 Crea un formulario para introducir nombre y e-mail con transiciones. [Ejemplo](#)
- 6 Crea un menú de navegación con transiciones. [Ejemplo](#)

Animaciones

La propiedad `transition` crea una animación básica, pero solo se involucran dos estados en el proceso: el estado inicial determinado por los valores actuales de las propiedades y el estado final, determinado por los nuevos valores. Para crear una animación real, necesitamos declarar más de dos estados, como los fotogramas de una película. CSS incluye las siguientes propiedades para componer animaciones más complejas.

`animation-name`

Esta propiedad especifica el nombre usado para identificar los pasos de la animación. Se puede usar para configurar varias animaciones al mismo tiempo declarando los nombres separados por coma.

`animation-duration`

Esta propiedad determina la duración de cada ciclo de la animación. El valor se debe especificar en segundos (por ejemplo, 1s).

`animation-timing-function`

Esta propiedad determina cómo se llevará a cabo el proceso de animación por medio de una curva Bézier declarada con los valores `ease`, `linear`, `ease-in`, `ease-out` y `ease-in-out`.

`animation-delay`

Esta propiedad especifica el tiempo que el navegador esperará antes de iniciar la animación.

`animation-iteration-count`

Esta propiedad declara la cantidad de veces que se ejecutará la animación. Acepta un número entero o el valor `infinite`, el cual hace que la animación se ejecute por tiempo indefinido. El valor por defecto es 1.

`animation-direction`

Esta propiedad declara la dirección de la animación. Acepta cuatro valores: `normal` (por defecto), `reverse`, `alternate`, y `alternate-reverse`. El valor `reverse` invierte la dirección de la animación, mostrando los pasos en la dirección opuesta en la que se han declarado. El valor `alternate` mezcla los ciclos de la animación, reproduciendo los

que tienen un índice impar en dirección normal y el resto en dirección invertida. Finalmente, el valor `alternate-reverse` hace lo mismo que `alternate`, pero en sentido inverso.

animation-fill-mode




Esta propiedad define cómo afecta la animación a los estilos del elemento. Acepta los valores ***none*** (por defecto), ***forwards***, ***backwards***, y ***both***. El valor ***forwards*** mantiene al elemento con los estilos definidos por las propiedades aplicadas en el último paso de la animación, mientras que ***backwards*** aplica los estilos del primer paso tan pronto como se define la animación (antes de ser ejecutada). Finalmente, el valor ***both*** produce ambos efectos.

animation

Esta propiedad nos permite definir todos los valores de la animación al mismo tiempo. Estas propiedades configuran la animación, pero los pasos se declaran por medio de la regla `@keyframes`. Esta regla se debe identificar con el nombre usado para configurar la animación, y debe incluir la lista de propiedades que queremos modificar en cada paso. La posición de cada paso de la animación se determina con un valor en porcentaje, donde 0 % corresponde al primer fotograma o al comienzo de la animación, y 100 % corresponde al final.

Actividad. Animaciones

Crea una copia de la plantilla de trabajo para realizar el ejercicio.

 css	26/11/2019 8:35	Carpeta de archivos	
 img	26/11/2019 8:28	Carpeta de archivos	
 index	26/11/2019 8:47	Chrome HTML Do...	1 KB

Descarga la foto de una guitarra acústica. No olvides respetar las licencias de uso de los recursos utilizados.

Edita y modifica el fichero index.html añadiendo.

```

```

Edita y modifica el fichero style.css

```
img {  
  width:200px;  
  animation: myAnimation 2s ease 1s 3 alternate forwards;  
}  
  
@keyframes myAnimation {  
  100% {  
    transform: rotate(360deg);  
  }  
}
```

Visualiza el documento en el navegador.

Actividades propuestas.

Bloque 11.

1 Prueba y analiza las siguientes reglas.

```
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: example;
  animation-duration: 2s;
  animation-delay: 1s;
}
@keyframes example {
  0%   {background-color: red; left:0px;}
  50%  {background-color: yellow; left:200px;}
  100% {background-color: red; left:0px;}
}
```

2 Prueba y analiza las siguientes reglas.

```
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: infinite;
  animation-direction: alternate;
}
@keyframes example {
  0%   {background-color: red; left:0px; top:0px;}
  25%  {background-color: blue; left:0px; top:200px;}
  50%  {background-color: green; left:200px; top:200px;}
  75%  {background-color: yellow; left:200px; top:0px;}
  100% {background-color: red; left:0px; top:0px;}
}
```