

# <TEMA 04>

## HOJAS DE ESTILO WEB

1. Introducción.
2. Elementos y estructura de una hoja de estilos.
3. Diseño de estilos para diferentes dispositivos.
4. Tipos de hojas de estilos: estáticas y dinámicas. Las pseudoclases.
5. Buenas prácticas en el uso de hojas de estilos.

[EMPEZAR >](#)





## 4.1 INTRODUCCIÓN

- La aparición de las hojas de estilo en cascada (*CSS o Cascade Style Sheet*) supuso uno de los avances más importantes en el ámbito del diseño y desarrollo web profesional.

**CSS:** Conjunto de reglas que definen el aspecto de los elementos que forman parte de los documentos web definidos previamente con HTML.

- En este capítulo vamos a estudiar los fundamentos de las hojas de estilo CSS, aprendiendo su sintaxis, diversas maneras de incluirlas en los documentos y algunas propiedades básicas que permitirán modificar el aspecto de los elementos de los documentos web.



## 4.2 ELEMENTOS Y ESTRUCTURA DE UNA HOJA DE ESTILOS

```
button {  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    font-size: 14px;  
    line-height: 20px;  
}  
  
button:focus {  
    outline: none;  
    box-shadow: 0 0 0 0 1px white, 0 0 0 0 3px white;  
}  
  
button:disabled {  
    pointer-events: none;  
    cursor: not-allowed;  
}  
  
@media (min-width: 1024px) {  
    button {  
        font-size: 16px;  
        line-height: 20px;  
    }  
}
```

**En este apartado vamos a estudiar las maneras de definir y aplicar estilos CSS a los documentos web, así como el funcionamiento de las hojas de estilo y como utilizarlas para modificar el formato y la estructura de los documentos web.**





## 4.2.1. Creación de hojas de estilo

Existen tres maneras de definir estilos CSS en un documento web

### MEDIANTE EL ATRIBUTO STYLE

Se utiliza el atributo "style" dentro de la etiqueta a dar estilo, mediante esta **sintaxis**:

```
<etiqueta  
style="propiedad1:valor1;  
       propiedad2:valor2;...>  
</etiqueta>
```

Ejemplo

### MEDIANTE LA DIRECTIVA <style>

Incluyendo los estilos dentro de la etiquetas `<style>` ubicada dentro de la cabecera del documento (`<head>`)

#### Sintaxis:

```
<head>  
  <style type="text/css">  
    selector {  
      propiedad1:valor1;  
      propiedad2:valor2;  
    }  
  </style>  
</head>
```

Ejemplo

### MEDIANTE UN ARCHIVO .CSS EXTERNO

El más utilizado. Los estilos se implementan en un archivo css externo (.css) aparte. El archivo se enlaza al documento web utilizando la directiva `<link>` dentro de `<head>`

#### Sintaxis:

```
<head>  
  <link rel="stylesheet"  
        href="NombreArchivo.css">  
</head>
```

Ejemplo



## Ejemplo con atributo style

Establecemos color rojo y alineación justificada en el párrafo de este documento web (estilo en linea)

```
...
<p style="color:#ff0000; text-align:justify"> Esto es un parrafo</p>
...
```

Modifica este párrafo para que aparezca en azul (#0000ff) y alineado a la derecha (right).



## Ejemplo archivo .css externo

Deseamos vincular una hoja de estilos llamada "estilo.css" que está ubicada en un directorio "css" que cuelga del directorio raiz donde está ubicado el archivo html que realiza el vínculo.

```
<head>
    <link rel="stylesheet" href="css/estilo.css"/>
</head>
<body>
    <p>Esto es un parrafo</p>
</body>
```

**NOTA:** Los archivos .css son de texto plano, como los documentos html



## mediante la directiva <style>

Vamos ha crear el mismo estilo anterior, pero en este caso incluyendo el código CSS dentro de <head> y a su vez dentro de la directiva <style>

```
<head>
    <style type="text/css">
        p {
            color:#ff0000;
            text-align:justify;
        }

    </style>
</head>
<body>
    <p> Esto es un parrafo</p>
    <p> Esto es un parrafo</p>
    <p> Esto es un parrafo</p>
</body>
```



## 4.2.2. Aplicación de estilos. Selectores

- Los selectores se utilizan con estilos CSS en archivos externos o con los incluidos dentro de la directiva <style>
- Informan acerca de que elemento o elementos van a ser los beneficiarios del estilo definido con CSS.
- La mayoría de los sitios web pueden ser definidos con **cuatro tipos** de selectores básicos:

SELECTORES DE ETIQUETAS

+INFO

SELECTORES DESCENDENTES

+INFO

SELECTORES DE CLASE

+INFO

SELECTORES ID

+INFO

h1 { color: green; }

Selector      Declaración  
Propiedad      Valor



# Selectores de etiquetas

Los estilos se aplican a todos los elementos del documento web cuya directiva coincida con el nombre del selector.

## Sintaxis:

```
...
etiqueta {
    propiedad1:valor1;
    propiedad2:valor2;
    ...
}
```

```
...
a {
    color:red;
    text-decoration:none;
}
...
```

## Ejemplo 1:

En este caso el estilo `color:red` y `text-decoration:none` se aplica a todas las etiquetas `<a>` del documento.

```
...
/*todos los elementos p y h1*/
p, h1 {
    color:blue;
    text-align: center;
}
...
```

## Ejemplo 2:

Si se desea aplicar el mismo estilo a dos o mas etiquetas, bastará con separar dichos selectores con **comas**.

En este ejemplo el estilo `color:blue` y `text-align:center` se aplica a todas las etiquetas `<p>` y `<h1>` del documento.



## Selectores id

Se utilizan para dar estilos únicos aplicables a un elemento web en concreto (que tiene un identificador único)

### Sintaxis:

```
...
#identificador {
    propiedad1:valor1;
    propiedad2:valor2;
    ...
}
```

```
...
#autor {
    word-spacing:0.6em;
    font-family:Times;
}
...
```

### Ejemplo 1: css

En este caso el estilo *word-spacing:0.6em* y *font-family:Times* se aplica a la única etiqueta cuyo atributo *id* coincide con 'autor' (sin #)

```
<p id="parrafo1"> Parrafo sin
estilo</p>
<p id="autor"> Nombre del
autor con estilo </p>
```

### Ejemplo 1:html

El *id* de cada elemento ha de ser único, y referencia inequívocamente a ese elemento.

En este ejemplo solo el párrafo con *id="autor"* tomará el estilo definido en el documento CSS.



# Selectores de clase

NOTA

Los estilos CSS definidos en un selector de clase se aplican a aquellos elementos que tengan dicha clase (sin el ".") en su atributo *class*

## Sintaxis:

```
...
.clase {
    propiedad1:valor1;
    propiedad2:valor2;
    ...
}
```

```
<style type="text/css">
    .enlace_con_estilo{
        color:green;
        font-weight:bold;
    }
</style>
```

## Ejemplo 1:

En este caso los estilos solo se aplicarán a las etiquetas que tengan en el valor del atributo *class* (en el documento HTML) el texto "enlace\_con\_estilo"

```
...
article .parrafo_estilo {
    font-size:0.5em;
    text-align: center;
}
```

## Ejemplo 2:

En este otro caso utilizamos una clase combinandola con un selector descendente, de forma que la clase sólo se aplica si en la etiqueta HTML aparece la clase "parrafo\_estilo" y además esa etiqueta esta dentro de otra etiqueta <article>.



## Selectores descendentes

Se utilizan cuando se desean aplicar estilos CSS a elementos que están dentro de otros (y sólo si están dentro de otros)

### Sintaxis:

```
...
selector_1 selector_2
...
selector_n {
  propiedad1:valor1;
  propiedad2:valor2;
  ...
}
```

```
...
p a {
  color:red;
  text-decoration:none;
}
```

### Ejemplo 1:

En este caso el estilo `color:red` y `text-decoration:none` se aplica a todos los enlaces (`<a>`) del documento que se encuentren dentro de un párrafo(`<p>`).

```
...
aside p a {
  color:red;
  text-decoration:none;
}
```

### Ejemplo 2:

Se puede indicar mas de dos selectores descendentes (aparecen seguidos y sin comas)

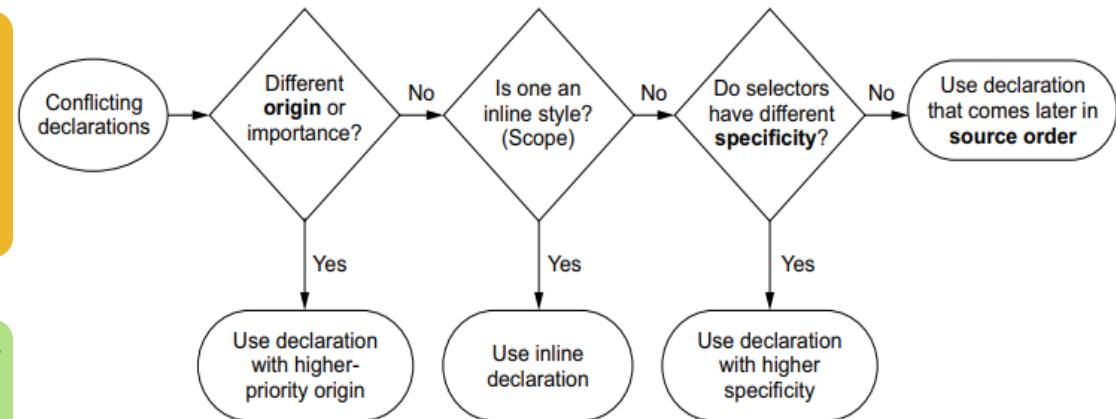
En este ejemplo el estilo solo se aplicará a los enlaces (`<a>`) que estén dentro de párrafos (`<p>`) y a su vez que los párrafos estén dentro de una etiqueta `<aside>`



### 4.2.3. Herencia de estilos y aplicación en cascada

Las hojas de estilo CSS se denominan hojas de estilo en cascada por varias razones:

- Las propiedades CSS pueden estar escritas en un atributo "style", dentro de la etiquetas <style> o en un archivo .css externo, y tendrán distinta **prioridad** dependiendo del sitio donde se encuentren.
- Si un elemento está ubicado dentro de otro, el elemento hijo (el que está contenido en el elemento que lo contiene) **hereda** las propiedades del elemento padre.
- Un mismo elemento puede ser afectado por propiedades CSS definidas en más de un selector.
  - Si no existe conflicto entre las propiedades CSS especificadas, se aplican todas.
  - Si existe conflicto se aplican ciertas reglas para establecer el estilo que va a imperar.





Tenemos un enlace afectado por una clase de nombre "link".

```
<a href="documento1.html" class="link">  
    Ir a Documento1  
</a>
```

Y el siguiente archivo externo .css que está vinculado al documento html:

```
...  
a {  
    text-decoration:none;  
    color:green;  
}  
.link {  
    text-align:center;  
}  
...
```

En este caso el vínculo se visualizará en color **verde** y **sin el subrayado** propio de los enlaces, al estar afectado por las propiedades del *selector de etiquetas* `<a>`.

Del mismo modo, también aparecerá alineado al centro al estar afectado por el *selector de clase* ".link".



## Herencia de estilos css

Tenemos un párrafo `<p>` contenido en un elemento `<aside>` al cual se le establecen propiedades css. Estas propiedades también afectarán al elemento `<p>` que está contenido en él.

```
<aside style="color:red;">
  <p>Parrafo en rojo<p>
</aside>
```

En este caso el párrafo se visualizará en color **rojo**, aunque el elemento `<p>` no tenga asignado ningún estilo, al estar contenido en un elemento padre (`<aside>`) que sí tiene aplicado un estilo, siendo éste heredado.

```
<p style="color:green;">Texto del párrafo</p>
```

X

1. El atributo "style" prima sobre la etiqueta <style>

2. Una etiqueta <style> (dentro del <head>) prima sobre un archivo vinculado externo .css

```
...  
p {  
    color:red;  
}  
...
```

estilos.css

```
<head>  
    <link rel="stylesheet" href="estilos.css">  
    <style>  
        p {color:blue}  
    </style>  
</head>  
<body>  
    <p>Esto es un párrafo azul</p>  
</body>
```

archivo.html

3. Un atributo style prima sobre un archivo vinculado externo .css

```
<head>  
    <link rel="stylesheet" href="estilos.css">  
</head>  
<p style="color:blue;">Esto es un párrafo azul</p>
```



## Prioridad css de mayor a menor

1. CSS en línea (style dentro de un elemento HTML):

```
<p style="color: blue;">Texto del párrafo</p>
```

2. CSS en un <style> interno (dentro del <head>):

```
<head>
  <style>
    p {color: blue;}
  </style>
</head>
<body>
  <p>Texto azul</p>
</body>
```

3. CSS en un archivo externo (.css):

```
<head>
  <link rel="stylesheet" href="estilos.css">
</head>
<body>
  <p>Texto según estilos.css</p>
</body>
```

archivo html

```
p {
  color: green;
}
```

archivo css



#### 4.2.4. Formateado de páginas mediante estilos

Algunas propiedades CSS que se pueden utilizar para formatear el texto de documentos web son:

font-family

Ver

color

Ver

vertical-align

Ver

font-style

Ver

word-spacing

Ver

text-transform

Ver

font-weight

Ver

letter-spacing

Ver

text-align

Ver

font-size

Ver

text-decoration

Ver

text-indent

Ver



line-height

Ver



# line-height

**line-height** en CSS se utiliza para establecer el espacio entre las líneas de un texto. Si el valor es un número, la altura de la linea se establece a partir de la multiplicación del tamaño de la fuente por ese valor. Los valores en porcentaje son relativos al tamaño de la fuente. No permite valores negativos.

```
<!DOCTYPE html>
<html>
<head>
    <style>
        p {
            line-height: 1.8; /* Espaciado entre
líneas 1.8 veces el tamaño de la fuente */
        }
    </style>
</head>
<body>
    <p>Este es un párrafo con mayor espacio entre
líneas. Mejora la legibilidad del texto.</p>
</body>
</html>
```



# letter-spacing

**letter-spacing** es idéntica a **word-spacing** salvo que el espacio se establece entre las *letras* de un texto. Por ejemplo:

```
p { letter-spacing: 0.2em; }
```

```
<!DOCTYPE html>
<html>
<head>
    <style>
        p.espaciado {letter-spacing: 0.2em;}
    </style>
</head>
<body>
    <p class="espaciado">Este es un ejemplo de texto
    con espacio entre letras.</p>
</body>
</html>
```

Ejemplo html



# font-size

El valor de **font-size** (tamaño) en CSS puede tomar varias formas. Aquí mostramos algunas de las más comunes:

- **Valores absolutos:** xx-small, x-small, small, medium (valor por defecto), large, x-large, xx-large. 
- **Valores relativos:** 
  - *smaller* (relativo al tamaño de fuente del elemento padre)
  - *larger* (relativo al tamaño de fuente del elemento padre)
- **Longitudes:** 
  - *px* (píxeles)
  - *em* (relativo al tamaño de la fuente del elemento padre)
  - *rem* (relativo al tamaño de la fuente del elemento raíz)
  - *vw* (porcentaje del ancho de la ventana gráfica)
  - *vh* (porcentaje de la altura de la ventana gráfica)
  - % (porcentaje relativo al tamaño de la fuente del elemento padre)



# word-spacing

**word-spacing** en CSS controla el espacio entre las *palabras* en un texto. Por ejemplo:

**Valor en longitud:** Puedes especificar el espacio en unidades como píxeles (px), em (em), porcentajes (%), etc.

```
p { word-spacing: 10px; }
```

**Valores negativos:** Puedes reducir el espacio entre las palabras usando valores negativos.

```
p { word-spacing: -0.5em; }
```

**Ejemplo html**

```
<!DOCTYPE html>
<html>
<head>
    <style>
        p.espaciado {word-spacing: 20px;}
    </style>
</head>
<body>
    <p class="espaciado">Este es un ejemplo de texto con espacio entre palabras.</p>
</body>
</html>
```



# text-decoration

**text-decoration** permite que el texto sea decorado por alguno de estos efectos:

- **underline**: subrayado.
- **overline**: subrayado superior.
- **line-through**: tachado.
- **blink**: parpadeo.
- **none**: Elimina cualquier efecto.

## Ejemplo html

```
<!DOCTYPE html>
<html>
<head>
    <style>
        a {text-decoration: none; color:blue;}
    </style>
</head>
<body>
    <a href="documento1.pdf">Abrir Documento 1</a>
</body>
</html>
```



# font-family

**font-family** en CSS especifica la **fuente** que se utilizará para el texto de un elemento. Se pueden listar múltiples fuentes como una "familia de fuentes", de manera que si una fuente no está disponible, el navegador intentará usar la siguiente de la lista.

```
p {  
    font-family: "Arial", "Helvetica", sans-serif;  
}
```

## Opciones comunes de fuentes:

- **Serif**: Fuentes con pequeños trazos decorativos (serifa)
  - Ejemplos: Times New Roman, Georgia, Garamond.
- **Sans-serif**: Fuentes sin esos trazos.
  - Ejemplos: Arial, Helvetica, Verdana.
- **Monospace**: Cada carácter ocupa el mismo ancho.
  - Ejemplos: Courier New, Lucida Console.
- **Cursive**: Fuentes que imitan la escritura a mano.
  - Ejemplos: Comic Sans MS, Brush Script MT.
- **Fantasy**: Fuentes decorativas y más inusuales.
  - Ejemplos: Impact, Papyrus.



# text-align

**text-align** en CSS alinea el texto que alberga un contenedor, por lo que solo es aplicable a *elementos contenedores o de bloque* (como `<div>`, `<p>`, etc...) Puede tomar los siguientes valores:

**left:** Texto alineado a la izquierda.

**right:** Texto alineado a la derecha

**center:** Texto alineado al centro.

**justify:** Texto justificado.

## Elementos de bloque

Esto es un párrafo.

Este es otro párrafo.

Los párrafos son los elementos de bloque que se organizan verticalmente.



## Elementos en línea

Los links son elementos en línea que se acomodan uno al lado del otro.



# text-indent

**text-indent** en CSS se utiliza para controlar la sangría de la primera línea de un bloque de texto. Puedes especificar la cantidad de sangría usando varias unidades como píxeles (px), em (em), porcentajes (%), entre otras.

```
<!DOCTYPE html>
<html>
<head>
    <style>
        p {
            text-indent: 30px; /* Sangría de 30
                píxeles */
        }
    </style>
</head>
<body>
    <p>Esta es la primera línea con sangría. El
        resto del texto sigue sin sangría.
    </p>
</body>
</html>
```



# color

El color en CSS se usa para definir el color del texto. El valor del color se puede indicar de varias formas:

**Nombres de colores:** Utiliza los nombres predefinidos.

```
/* Rojo */  
p {color: red;}
```

**Valores HEX:** Especifica colores en formato hexadecimal.

```
/* Rojo */  
p {color: #FF0000;}
```

**Valores RGB:** Utiliza los valores (desde 0 a 255) de Red, Green y Blue.

```
/* Rojo */  
p {color: rgb(255, 0, 0);}
```

**Valores RGBA:** RGB con un canal Alfa para la transparencia.

```
/* Rojo con 50% de opacidad */  
p {color: rgba(255, 0, 0, 0.5);}
```



# vertical-align

**vertical-align** en CSS controla la alineación vertical de un elemento en relación con su línea base de contenedor. Es particularmente útil para elementos en línea y celdas de tabla. Admite porcentajes y las siguientes palabras clave:

**baseline**: Alinea el elemento con la línea base del contenedor (elemento padre)

**sub**: Alinea el texto como un subíndice.

**super**: Alinea el texto como un superíndice.

**top**: Alinea el elemento con la parte superior de la línea más alta del contenedor.

**middle**: Alinea el elemento en el medio de la línea del contenedor.

**bottom**: Alinea el elemento con la parte inferior de la línea del contenedor.

```
img {vertical-align: baseline;}
```

```
sub {vertical-align: sub;}
```

```
sup {vertical-align: super;}
```

```
img {vertical-align: top;}
```

```
img {vertical-align: middle;}
```

```
img {vertical-align: bottom;}
```



# font-weight

**font-weight** en CSS controla el grosor del texto, permitiendo enfatizar ciertas partes del contenido. Estas son las opciones más comunes:

- **Valores numéricos:** De 100 (más delgado) a 900 (más grueso), en incrementos de 100.
- **Palabras clave:**
  - *normal*: Valor por defecto.
  - *bold*: Texto en negrita.
  - *bolder*: Más grueso que el valor del elemento padre.
  - *lighter*: Más delgado que el valor del elemento padre.
- **inherit**: Hereda el valor de font-weight del elemento padre.



# font-style

**font-style** en CSS define la inclinación de la fuente, permitiendo agregar estilos a tu texto. Aquí están las opciones más comunes:

- **normal**: Es el valor por defecto, sin inclinación.

```
p {font-style: "normal";}
```

- **italic**: Aplica un estilo cursivo a la fuente.

```
p {font-style: "italic";}
```

- **oblique**: Similar a italic, pero más inclinado y menos controlado.

```
p {font-style: "oblique";}
```

- **inherit**: Hereda el valor de font-style de su elemento padre.

```
p {font-style: "inherit";}
```



# text-transform

**text-transform** en CSS permite establecer, en el texto, diferentes transformaciones, como son:

**capitalize**: Convierte la primera letra de cada palabra en mayúsculas.

**uppercase**: Convierte todas las letras del texto en mayúsculas.

**lowercase**: Convierte todas las letras del texto en minúsculas.

**none**: No hace ningún cambio al texto.

```
<!DOCTYPE html>
<html>
<head>
    <style>
        .capitalize {
            text-transform: capitalize;
        }
        .uppercase {
            text-transform: uppercase;
        }
        .lowercase {
            text-transform: lowercase;
        }
    </style>
</head>
<body>
    <p class="capitalize">este texto se convierte en Capitalize.</p>
    <p class="uppercase">este texto se convierte en UPPERCASE.</p>
    <p class="lowercase">ESTE TEXTO SE CONVIERTEN EN lowercase.</p>
</body>
</html>
```



## ACTIVIDAD 1



Vamos a desarrollar un sitio web donde, entre etiquetas <style>, aparezcan los siguientes estilos:

1. El encabezado <h1> se visualizará con una fuente Brush Script MT.
2. El encabezado <h2> debe aparecer subrayado.
3. Los párrafos han de llevar asociados la fuente Tahoma.
4. Si un párrafo contiene una clase "azul" el texto se visualizará en azul.
5. Si un párrafo contiene una clase "mayusculas" todos los caracteres del texto han de mostrarse en mayúsculas
6. Una lista desordenada debe mostrarse en itálica, negrita y el primer carácter de cada ítem ha de aparecer en mayúscula.
7. Los vínculos han de mostrarse en rojo, con una separación entre letras de 0.3em y sin el subrayado que, por omisión, lleva incorporado su etiqueta.
8. Sin modificar el código CSS creado hasta ahora, debe aparecer un último párrafo en color azul y en mayúsculas.

Desarrolla el código html que permita mostrar estos estilos.

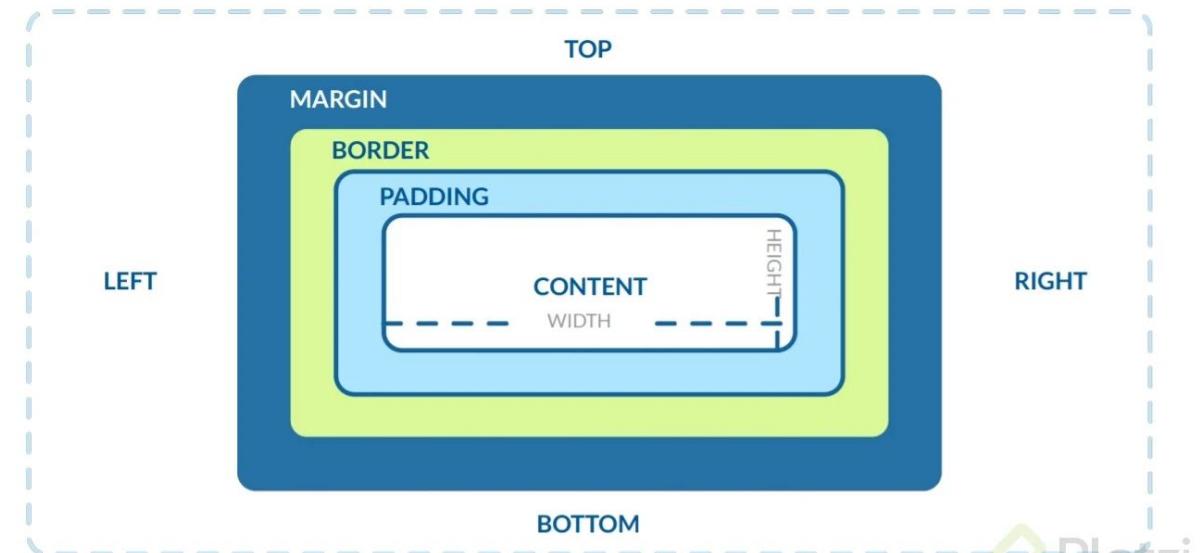
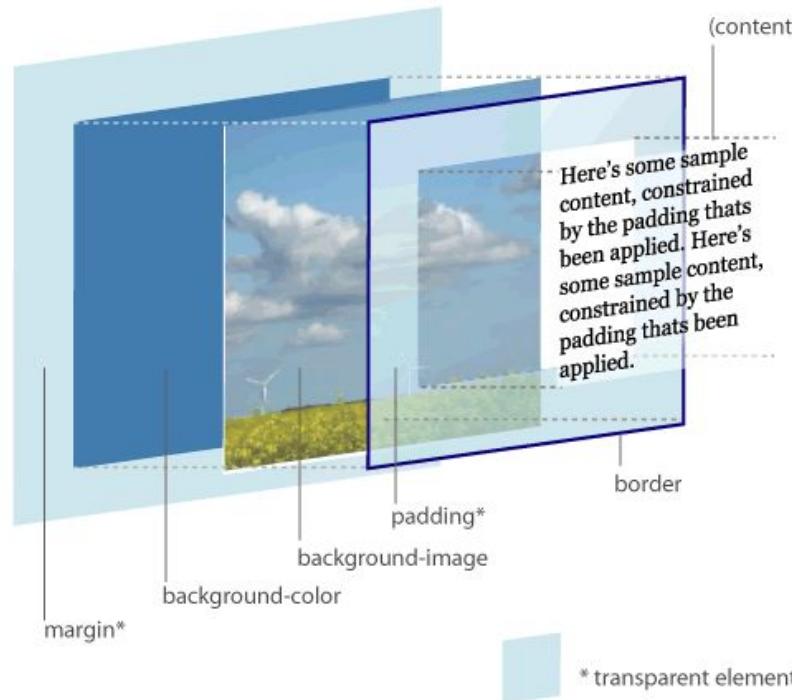
SOLUCIÓN: 15-Actividad1\_CSS.html



## 4.2.5. Estructura de páginas mediante estilos. El modelo de cajas

El modelo de cajas condiciona el diseño de la mayoría de los sitios WWW. Consiste en que hace que todos los elementos incluidos en un documento web presenten una estructura similar a una caja rectangular, estableciendo las siguientes partes:

THE CSS BOX MODEL HIERARCHY



# Partes del modelo de cajas. Estructura

background-color  
(color de fondo)

 Ver

content  
(contenido)

 Ver

padding  
(relleno)

 Ver

border  
(borde)

 Ver

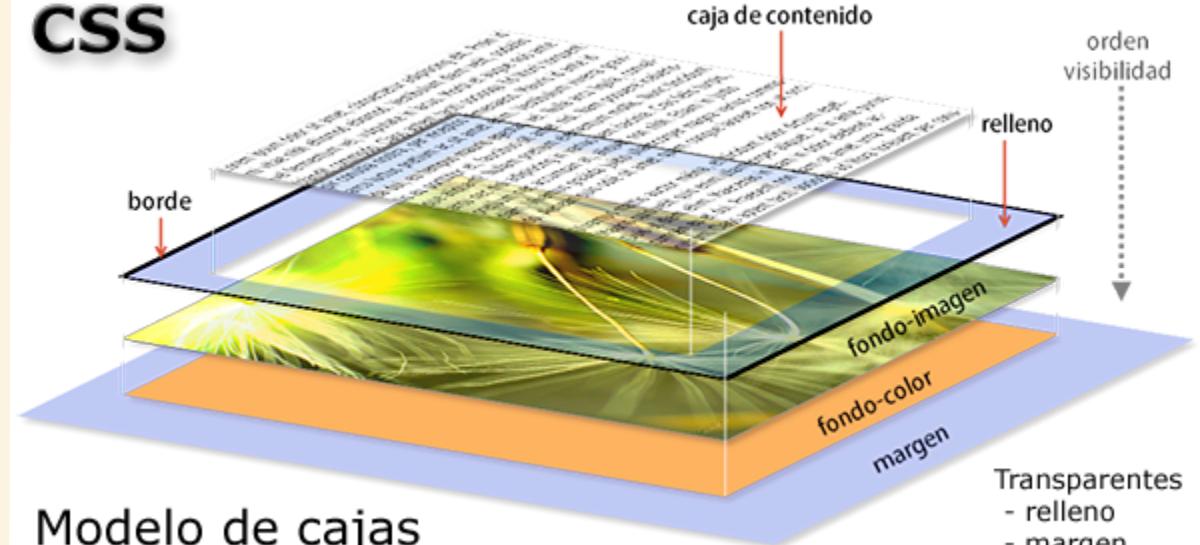
background-image  
(imagen de fondo)

 Ver

margin  
(margen)

 Ver

CSS



Modelo de cajas

Transparentes  
- relleno  
- margen



# margin

**margin** en CSS controla el espacio exterior alrededor de un elemento, creando distancia entre el elemento y sus adyacentes.

Podemos declarar la distancia superior, derecha,inferior e izquierda de forma separada ...

```
.caja {  
    width: 150px;  
    height: 100px;  
    background-color: lightblue;  
    margin-top: 20px; /* Margen superior de 20px */  
    margin-right: 30px; /* Margen derecho de 30px */  
    margin-bottom: 40px; /* Margen inferior de 40px */  
    margin-left: 50px; /* Margen izquierdo de 50px */  
}
```

O bien unificarla en una sola declaración:

```
.caja {  
    ...  
    margin: 20px 30px 40px 50px; /*top right bottom left*/  
    ...  
}
```



# background-image

**background-image** se utiliza para establecer una imagen de fondo para un elemento. Algunas de sus opciones:

## Uso básico:

```
background-image: url('imagen.jpg');
```

## Propiedades adicionales para ajustar imágenes de fondo:

- **background-repeat**: Controla cómo se repite la imagen.
- **background-position**: Define la posición inicial de la imagen.
- **background-size**: Ajusta el tamaño de la imagen de fondo.
- **background-attachment**: Define si la imagen de fondo se desplaza con el contenido de la página

**Estas propiedades tienen una serie de opciones que veremos en la siguiente página.**



# padding

El **padding** (relleno) en el modelo de cajas de CSS es el espacio entre el contenido de un elemento y su borde. Se utiliza para crear espacio interior dentro del borde de un elemento.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .caja {
        width: 150px;
        height: 100px;
        background-color: lightblue;
        padding: 20px; /* Aplica un relleno de 20px en
                         todos los lados */
      }
    </style>
  </head>
  <body>
    <div class="caja">Contenido con relleno.</div>
  </body>
</html>
```

En este ejemplo, el relleno crea espacio adicional dentro del borde del elemento, alrededor del contenido. Esto puede ser útil para asegurar que el contenido no quede pegado al borde del elemento, mejorando la legibilidad y el diseño.



# border

**border** (borde) en el modelo de cajas de CSS define el contorno alrededor del padding y el contenido de un elemento.

## Propiedades:

- **border-width**: Establece el grosor del borde.
- **border-style**: Define el estilo del borde. Algunas opciones son:
  - *solid*: Borde sólido.
  - *dashed*: Borde discontinuo.
  - *double*: Borde doble.
  - *groove*: Borde en relieve (apariencia de ranura).
  - *ridge*: Borde en relieve (apariencia de cresta).
- **border-color**: Establece el color del borde.
- **border**: Propiedad abreviada para establecer el *grosor*, *estilo* y *color* del borde en una sola línea.

```
...
.caja {
  width: 200px;
  height: 100px;
  background-color: lightgray;
  padding: 20px;
  border: 2px solid blue; /* Ancho de 2px, estilo sólido, color azul */
}
...
```



# background-color

Se utiliza para establecer el color de fondo de un elemento.  
Puedes usar varios tipos de valores para definir los colores:

- **Nombres de colores:**

```
background-color: lightblue;
```

- **Valores HEX:**

```
background-color: #FF5733; /* Naranja */
```

- **Valores RGB:**

```
background-color: rgb(255, 87, 51); /* Naranja */
```

- **Valores RGBA:** Permite definir la opacidad.

```
background-color: rgba(255, 87, 51, 0.5);  
/* Naranja con 50% de opacidad */
```

```
<!DOCTYPE html>  
<html>  
  <head>  
    <style>  
      .caja {  
        background-color: lightblue; /*Color de fondo azul claro */  
      }  
    </style>  
  </head>  
  <body>  
    <div class="caja">Contenido con fondo de color.</div>  
  </body>  
</html>
```



# content

El **contenido** es el área central de la caja de un elemento, donde reside el texto, las imágenes o cualquier otro tipo de contenido que el elemento puede albergar. Este área está directamente afectada por propiedades como width, height, min-width, max-width, min-height y max-height.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .caja {
        width: 200px; /* Ancho del contenido */
        height: 100px; /* Altura del contenido */
        background-color: lightblue; /* Color de fondo*/
      }
    </style>
  </head>
  <body>
    <div class="caja">Este es el contenido.</div>
  </body>
</html>
```

Estas propiedades permiten controlar con precisión el tamaño y la apariencia del contenido dentro de un elemento, proporcionando una base sólida para un diseño estructurado y claro.



# Profundizamos en el elemento background-image

Propiedades adicionales para ajustar la imagen de fondo

background-repeat



background-position



background-size



background-attachment



**NOTA:** Existen muchas propiedades CSS que permiten modificar diversos parámetros. Para profundizar en ellas puedes visitar la página <https://www.w3schools.com/css/>



# background-repeat



La propiedad **background-repeat** en CSS controla cómo se repite una imagen de fondo dentro de un elemento. Aquí están las opciones más comunes y cómo puedes usarlas:

- **repeat**: La imagen de fondo se repite tanto horizontal como verticalmente. (por defecto)

```
div {  
    background-image: url(image.jpg);  
    background-repeat:repeat;  
}
```

- **no-repeat**: La imagen de fondo no se repite.

```
div {  
    background-image: url(image.jpg);  
    background-repeat:no-repeat;  
}
```

- **repeat-x**: La imagen de fondo se repite solo horizontalmente.

```
div {  
    background-image: url(image.jpg);  
    background-repeat:repeat-x;  
}
```

- **repeat-y**: La imagen de fondo se repite solo verticalmente.

```
div {  
    background-image: url(image.jpg);  
    background-repeat:repeat-y;  
}
```

- **space**: La imagen de fondo se repite, pero se añade espacio entre cada repetición para que cubra completamente el contenedor sin cortarse

```
div {  
    background-image: url(image.jpg);  
    background-repeat:space;  
}
```

- **round**: La imagen de fondo se repite, pero se escala para asegurarse de que cubre completamente el contenedor sin cortarse.

```
div {  
    background-image: url(image.jpg);  
    background-repeat:round;  
}
```

```
...  
<style>  
    .caja {  
        width: 300px;  
        height: 200px;  
        background-image: url('https://example.com/imagen.jpg');  
        background-repeat: repeat-x; /* Se repite solo horizontalmente */  
    }  
</style>
```



# background-attachment

La propiedad **background-attachment** en CSS controla cómo se comporta la imagen de fondo en relación con el contenido del elemento y la página. Las opciones disponibles son:

- **scroll**: (valor por defecto) La imagen de fondo se desplaza junto con el contenido de la página.

```
div {  
    background-image: url(image.jpg);  
    background-attachment: scroll;  
}
```

- **fixed**: La imagen de fondo se fija en su posición y no se desplaza con el contenido de la página.

```
div {  
    background-image: url(image.jpg);  
    background-attachment: fixed;  
}
```

- **local**: La imagen de fondo se desplaza junto con el contenido del elemento al que se aplica.

```
div {  
    background-image: url(image.jpg);  
    background-attachment: local;  
}
```

## Ejemplo

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
    .caja {  
        width: 300px;  
        height: 200px;  
        background-image: url('https://example.com/imagen.jpg');  
        background-size: cover;  
        background-position: center;  
        background-attachment: fixed; /* La imagen se fija y no se  
                                     desplaza */  
    }  
</style>  
</head>  
<body>  
    <div class="caja">Contenido con imagen de fondo fija.</div>  
</body>  
</html>
```



# background-position

La propiedad **background-position** en CSS se utiliza para definir la posición inicial de una imagen de fondo dentro de un elemento. Puedes utilizar valores específicos o palabras clave para establecer su ubicación. Sus opciones mas comunes son:

- **Palabras clave:**

- left, right: Para posición horizontal
- top, bottom: Para posición vertical
- center: Para posicionar en el centro del elemento

```
div {  
    background-image: url(image.jpg);  
    background-position: top right; /*Esquina superior dcha*/  
}
```

- **Valores de porcentaje:** Especifica la posición en términos de porcentaje del tamaño del contenedor.

```
div {  
    background-image: url(image.jpg);  
    background-position: 50% 50%; /* Centro del elemento*/  
}
```

```
...  
<style>  
    .caja {  
        width: 300px;  
        height: 200px;  
        background-image: url('https://example.com/imagen.jpg');  
        background-position: bottom left; /* Esquina inferior izquierda */  
        background-repeat: no-repeat;  
    }  
</style>
```

- **Valores de longitud:** Usa unidades de longitud como píxeles (px), em (em), etc.

```
div {  
    background-image: url(image.jpg);  
    background-position: 10px 20px;  
    /* 10px desde la izquierda y 20px desde la derecha*/  
}
```

- **initial, inherit:**

- initial: Restaura el valor inicial de la propiedad.
- inherit: Hereda el valor de background-position de su elemento padre.

```
div {  
    background-image: url(image.jpg);  
    background-repeat: space;  
}
```



# background-size

La propiedad background-size en CSS se utiliza para controlar el tamaño de una imagen de fondo. Te permite ajustar cómo se escala la imagen dentro de un elemento. Sus opciones más comunes son:

- **Valores predeterminados:** auto: (valor predeterminado): La imagen conserva su tamaño original. (por defecto)

```
div {  
    background-image: url(image.jpg);  
    background-size: auto;  
}
```

- **Valores de longitud:** Usa unidades como píxeles (px), em (em), etc.
  - Especificar un solo valor cambiará la anchura y ajustará la altura automáticamente para mantener las proporciones.
  - Especificar dos valores cambiará la anchura y altura de la imagen.

```
div {  
    background-image: url(image.jpg);  
    background-size: 100px 50px;  
    /* Ancho de 100px y alto de 50px */  
}
```

- **Porcentajes:** Relativo al tamaño del contenedor.

- Un solo porcentaje ajusta el ancho, manteniendo las proporciones.
- Dos porcentajes ajustan el ancho y el alto.

```
div {  
    background-image: url(image.jpg);  
    background-size: 50% 75%;  
}
```

- **Palabras clave:**

- *cover*: Escala la imagen para cubrir completamente el contenedor, asegurando que no haya espacios vacíos. La imagen puede recortarse.

```
div {  
    background-image: url(image.jpg);  
    background-size: cover;  
}
```

- *contain*: Escala la imagen para que esté completamente contenida dentro del contenedor, asegurando que toda la imagen sea visible. Puede haber espacios vacíos.

```
div {  
    background-image: url(image.jpg);  
    background-size: contain;  
}
```

Ver Ejemplo:





## 4.3 DISEÑO DE ESTILOS PARA DIFERENTES DISPOSITIVOS

Una página alojada en un sitio web es accesible y puede ser visualizada por diferentes dispositivos con distintas resoluciones de pantalla (pc, móvil, tablet, ...) por lo que su diseño debe orientarse hacia su correcta visualización, independientemente del dispositivo con el que se acceda a ella.

Para lograrlo conviene tener en cuenta dos consideraciones relacionadas con el código CSS:

- Utilizar *tamaños relativos* (ej. porcentajes) en cualquier propiedad CSS que establezca alguna medida, de forma que el tamaño de los elementos se adapte a las dimensiones de la ventana de visualización.

Ejemplo: [16-Tamaño relativo para diseño adaptable.html](#)

- Utilizar las denominadas *Media Queries* que incorpora la nueva especificación CSS3. Son una serie de condiciones y requisitos que permiten filtrar estilos CSS dependiendo del rango de resolución de la pantalla.

Ejemplo: [17-Media-queries para diseño adaptable.html](#)



## 4.4 HOJAS DE ESTILO ESTÁTICAS Y DINÁMICAS. LAS PSEUDO-CLASES

**Hojas de estilo estáticas:** El aspecto que presentan los elementos a los que se aplica estilo es siempre el mismo e independiente de cualquier otro factor externo.

**Hojas de estilo dinámicas:** Las **pseudo-clases** aportan dinamismo a los estilos que se pueden aplicar a los elementos web, pudiendo aplicarlos cuando se cumplen ciertas condiciones. Su sintaxis es la siguiente:

Algunas de las pseudo-clases mas utilizadas en CSS son:

- **:first-child** -> Hace referencia al primer elemento de un tipo determinado que está contenido dentro de otro.
- **:link** -> Permite establecer estilos en los vínculos que todavía no han sido visitados.
- **:visited** -> Permite establecer estilos en los vínculos que sí han sido visitados.
- **:hover** -> Permite especificar el estilo de un elemento cuando el puntero del ratón se situa sobre él.
- **:active** -> Permite establecer el aspecto de un elemento cuando se hace click sobre él.
- **:focus** -> Permite establecer el estilo de un elemento cuando éste obtiene el foco.
- **:lang** -> Permite especificar el aspecto de los elementos de un idioma determinado.

```
...  
selector: pseudo-clase {  
    propiedad1: valor1;  
    propiedad1: valor1;  
    propiedad1: valor1;  
    ...  
}
```

+ first-child

+ link

+ visited

+ hover

+ active

+ focus

+ lang



# :focus

## Al obtener el foco

Se utiliza para aplicar estilos a un elemento que tiene el foco del teclado o ha sido seleccionado mediante una interacción (como hacer clic o usar la tecla Tab). Es especialmente útil para mejorar la accesibilidad y la experiencia del usuario al navegar por un sitio web.

```
selector:focus { /* Estilos aplicados al elemento con foco */ }
```

```
<style>
    input {
        padding: 10px;
        font-size: 16px;
        border: 2px solid gray;
        border-radius: 5px;
    }
    input:focus {
        border-color: blue;
        outline: none; /* Elimina el contorno predeterminado del navegador */
        box-shadow: 0 0 5px blue; /* Efecto de sombra */
    }
</style>
</head>
<body>
    <label for="nombre">Nombre:</label>
    <input type="text" id="nombre" placeholder="Escribe tu nombre">
</body>
</html>
```

EJEMPLO



# :hover

## cursor sobre elemento

se utiliza para aplicar estilos a un elemento cuando el usuario coloca el cursor del mouse sobre él. Es ampliamente usada para crear interacciones visuales, especialmente en enlaces, botones o cualquier elemento interactivo.

```
selector:hover { /* Estilos del elemento al pasar el cursor sobre él*/ }
```

```
...
<style>
  a {
    color: blue;
    text-decoration: none;
  }

  a:hover {
    color: red;
    text-decoration: underline;
  }
</style>
</head>
<body>
  <a href="https://www.example.com">Pasa el mouse sobre este enlace</a>
</body>
</html>
```

### EJEMPLO

Ejemplo con transición



# :visited

## enlace visitado

La pseudo-clase :visited en CSS se utiliza para aplicar estilos a los enlaces  que el usuario ya ha visitado (es decir, aquellos cuyos destinos aparecen en el historial de navegación del navegador). Es útil para diferenciar visualmente los enlaces visitados de los no visitados.

```
a:visited /* Estilos aplicados a los enlaces visitados */
```

```
...
<style>
    /* Estilo para enlaces no visitados */
    a:link {
        color: blue;
        text-decoration: none;
    }
    /* Estilo para enlaces visitados */
    a:visited {
        color: purple;
    }
</style>
</head>
<body>
    <a href="https://www.example.com">Enlace a example</a>
    <a href="https://www.google.com">Enlace a google</a>
</body>
</html>
```

### EJEMPLO



# :link

## en vínculos aún no visitados

Se utiliza para aplicar estilos a los enlaces `<a>` que no han sido visitados aún. Es decir, afecta a los enlaces en su estado inicial, antes de que el usuario haya hecho click en ellos y los haya visitado.

```
a:link /* Estilos aplicados a los enlaces no visitados */
```

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Pseudo-clase :link</title>
  <style>
    /* Estilo para enlaces no visitados */
    a:link {
      color: blue;
      text-decoration: none;
    }
  </style>
</head>
<body>
  <a href="https://www.example.com">Enlace no visitado</a>
</body>
</html>
```

### EJEMPLO



# :active

## Al hacer click

Se utiliza para aplicar estilos a un elemento mientras está siendo activado, generalmente durante el momento en que el usuario hace clic sobre él (antes de que se complete la acción). Es comúnmente usada en enlaces, botones o cualquier elemento interactivo.

```
selector:active{ /* Estilos al hacer click sobre el elemento */ }
```

```
<style>
button {
    background-color: lightblue;
    color: black;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}
button:active {
    background-color: darkblue;
    color: white;
    transform: scale(0.95);
}
</style>

<button>Haz clic aquí</button>
```

### EJEMPLO



# :first-child

## Primer elemento hijo

Se utiliza para seleccionar el primer elemento hijo dentro de su contenedor. Esto es útil cuando deseas aplicar estilos específicos solo al primer elemento de un grupo de elementos hermanos.

```
selector:first-child { /* Estilos aplicados al primer hijo */ }
```

**EJEMPLO**

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pseudo-clase :first-child</title>
    <style> /* Selecciona solo el primer párrafo dentro de su contenedor */
      p:first-child { color: red; font-weight: bold; }
    </style>
  </head>
  <body>
    <div>
      <p>Primer párrafo (con :first-child).</p>
      <p>Segundo párrafo (no aparece en rojo ni bold).</p>
      <p>Tercer párrafo (no aparece en rojo ni bold).</p>
    </div>
  </body>
</html>
```



# :lang

## Según idioma definido

Se utiliza para aplicar estilos específicos según el idioma definido en el documento o en un elemento HTML utilizando el atributo lang. Esto es útil para personalizar estilos dependiendo del idioma, especialmente en sitios web multilingües.

```
selector:lang(código-de-idioma) { /* Estilos aplicados a elementos con el  
idioma especificado */ }  
  
<style>  
    p:lang(es) {  
        color: red; /* Texto rojo para español */  
    }  
  
    p:lang(en) {  
        color: blue; /* Texto azul para inglés */  
    }  
  
    p:lang(fr) {  
        color: green; /* Texto verde para francés */  
    }  
</style>  
</head>  
<body>  
    <p lang="es">Este texto está en español.</p>  
    <p lang="en">This text is in English.</p>  
    <p lang="fr">Ce texte est en français.</p>  
</body>  
</html>
```

EJEMPLO



## 4.5 BUENAS PRACTICAS EN EL USO DE HOJAS DE ESTILO

**Algunos consejos recomendables al aplicar estilos css en un sitio web son:**

**Separar el contenido del diseño:** El código CSS se puede escribir en un archivo externo, enlazado en la cabecera del documento. Esto permite separar el contenido (HTML) de la presentación (CSS) 

**Aprovechar los prefijos de proveedor:** Se refiere a utilizar prefijos específicos de navegadores para implementar propiedades y características que aún están en desarrollo, son experimentales o no han sido estandarizadas por completo en todos los navegadores. 

**Utilizar preprocesadores:** Este tipo de aplicaciones permiten escribir código con mayor rapidez y posibilidades, adaptandolo posteriormente a un formato óptimo para todo tipo de navegadores (Sass, Stylus, Less, ...) 

**Depurar el código CSS:** Existen herramientas que permiten depurar CSS, eliminando (entre otras cosas) aquellos fragmentos de código que no se utilizan. 

**Evitar repeticiones:** Debemos evitar las redundancias. Esto permitirá, entre otras cosas, hacer la carga del documento mas liviana. 



## ¿Cómo evitar redundancias?

**1. Usa Selectores Genéricos:** En lugar de repetir estilos para elementos similares, utiliza selectores genéricos o comunes.

Ejemplo

**2. Utiliza Clases en Lugar de Selectores Repetidos:** Reutiliza clases para aplicar los mismos estilos a varios elementos.

Ejemplo

**3. Usa Variables (Preprocesadores o CSS Nativo):** Centraliza valores comunes como colores, tamaños y fuentes.

Ejemplo

**4. Usa Herencia Inteligente:** Centraliza valores comunes como colores, tamaños y fuentes.

Ejemplo

**5. Usa Mixins o Extends (En Preprocesadores):** Agrupa estilos reutilizables en bloques para evitar duplicación.

Ejemplo

**6. Usa Combinadores en Selectores:** Combina selectores para aplicar estilos compartidos.

Ejemplo

**7. Usa Estilos Predeterminados (Reset o Normalize):** Implementa un reset CSS para establecer valores base y evitar definir los mismos estilos repetidamente.

Ejemplo

## Motivos para separar HTML de CSS

X

**1. Claridad y mantenimiento:** Es mas facil de leer, entender y modificar.

**2. Reutilización de estilos:** Permite definir estilos una vez y aplicarlos en múltiples páginas o elementos del sitio web. Esto reduce la duplicación de código, haciendo más eficiente el trabajo y el uso de recursos.

### **3. Escalabilidad:**

- Los proyectos más grandes pueden manejarse mejor cuando la presentación está separada del contenido.
- Las hojas de estilo centralizadas permiten modificar rápidamente el diseño de todo el sitio desde un único archivo CSS.

### **4. Reducción del tamaño del HTML:**

- Separar el diseño del contenido hace que el archivo HTML sea más ligero, ya que no contiene información repetida sobre estilos.
- Esto mejora el tiempo de carga de las páginas y reduce el uso del ancho de banda.



## Herramientas de depuración populares

### 1. Herramientas integradas en navegadores

- Google Chrome DevTools
- Firefox Developer Tools
- Safari Web Inspector
- Microsoft Edge DevTools

### 2. Extensiones y complementos

- VisBug
- CSS Peeper
- Web Developer Toolbar

### 3. Aplicaciones de terceros

- Stylelint
- PostCSS
- Polypane

### 4. Herramientas en linea

- CodePen
- JSFiddle
- CSS Validation Service (W3C)

### 5. Herramientas específicas para depurar diseño avanzado

- Grid Inspector (Firefox/Chrome)
- Flexbox Inspector
- Lighthouse (integrado en Chrome DevTools)



## Prefijos de proveedor

Son un mecanismo que permite a los desarrolladores web probar nuevas propiedades CSS específicas de un navegador antes de que sean oficialmente parte del estándar. Estos prefijos identifican a un navegador o motor de renderizado específico.

### Ejemplos de prefijos comunes:

- **-webkit-**: Utilizado por navegadores basados en WebKit (Google Chrome, Safari, navegadores móviles).
- **-moz-**: Utilizado por Mozilla Firefox.
- **-ms-**: Utilizado por Internet Explorer y versiones tempranas de Microsoft Edge.
- **-o-**: Utilizado por Opera (en motores anteriores a Blink).

### Ejemplo Práctico

Propiedad experimental: border-radius (en sus primeros días de soporte).

```
/* Prefijos de compatibilidad con navegadores antiguos */  
-webkit-border-radius: 10px; /* Chrome, Safari antiguos */  
-moz-border-radius: 10px; /* Firefox antiguo */  
border-radius: 10px; /* Propiedad estandarizada */
```

El prefijo no cambia el significado de la propiedad o valor, pero le dice al navegador que use su implementación específica. Esto asegura que las propiedades experimentales funcionen correctamente mientras se finaliza su definición estándar.



## Ejemplo de SASS (Syntactically Awesome Stylesheets)

Extensiones: .scss o .sass.

### Características:

- **Variables:** Define valores reutilizables (colores, tamaños, etc.).
- **Anidamiento:** Permite estructurar selectores de manera jerárquica.
- **Mixins:** Define bloques de código reutilizable.
- **Funciones y operadores:** Realiza cálculos y transforma valores.
- **Partials e importaciones:** Divide el CSS en múltiples archivos para modularidad.
- **Compatibilidad:** Soporta sintaxis SCSS (similar a CSS) y Sass (minimalista, sin llaves ni punto y coma).

```
$primary-color: #3498db;

body {
    font-family: Arial, sans-serif;
    color: $primary-color;

    h1 {
        font-size: 2rem;
    }
}
```



## 3.19. RESUMEN

- CSS ofrece infinidad de posibilidades de diseño, imposibles de lograr tan solo con HTML.
- Existen 3 maneras de incluir estilos CSS en un documento web:
  - Mediante el atributo *style*.
  - Mediante la etiqueta `<style>`.
  - Mediante un archivo ".css" externo.
- Para añadir estilos con los últimos métodos se utilizan **selectores**, que son indicadores que informan acerca del elemento o elementos que van a ser los beneficiarios de los estilos establecidos.
  - Selectores de *etiquetas*.
  - Selectores *descendentes*.
  - Selectores de *clase*.
  - Selectores *id*.
- También debemos conocer las denominadas **pseudo-clases**, que permiten dotar de cierto dinamismo a los estilos CSS definidos para los elementos de un sitio web.

```
</head>
<body style="margin: 0; background-color: #f0f0f0; font-family: sans-serif; font-size: 14px; color: #333; padding: 10px; position: relative; height: 100%;>
<div id="main_content" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; overflow: auto; scroll-behavior: smooth; margin: 0; padding: 0; border: none; display: flex; justify-content: center; align-items: center; flex-direction: column; gap: 20px;>
<h1 style="font-family: serif; font-size: 2em; margin: 0; padding: 0; border: none; background-color: transparent; color: inherit; text-decoration: none; transition: all 0.3s ease; position: relative; z-index: 1; text-align: center; line-height: 1.2; font-weight: normal; margin-bottom: 10px; margin-top: 0; font-style: italic; letter-spacing: 0.1em; text-transform: lowercase; text-decoration: underline; text-decoration-color: transparent; text-decoration-thickness: 2px; text-decoration-style: solid; text-decoration-position: under;">HTML & CSS
<p style="font-family: serif; font-size: 1.2em; margin: 0; padding: 0; border: none; background-color: transparent; color: inherit; text-decoration: none; transition: all 0.3s ease; position: relative; z-index: 1; text-align: center; line-height: 1.2; font-weight: normal; margin-bottom: 10px; margin-top: 0; font-style: italic; letter-spacing: 0.1em; text-transform: lowercase; text-decoration: underline; text-decoration-color: transparent; text-decoration-thickness: 2px; text-decoration-style: solid; text-decoration-position: under;">HTML & CSS
<img alt="A small decorative icon or logo, possibly a book or a gear, located at the bottom right of the slide." data-bbox="880 880 920 920"/>
```



# Aplicación práctica

..../Ejercicios/Practica3\_CSS/index.html

Tenemos la siguiente estructura (ya dada) de un sitio web:

- En la raíz del sitio aparece un documento web (index.html) y tres carpetas ("view", "images" y "styles")
- Dentro del directorio "view", aparecen dos documentos HTML: "contacto.html" y "sobre\_mi.html".
- Dentro del directorio "images", aparece una carpeta ("gallery") con una serie de imágenes y otros tres archivos de imagen en su raíz.
- El directorio "styles" aparece vacío.

Vamos a cambiar completamente el aspecto del sitio web, **sin realizar modificación alguna** en los archivos HTML

HTML sin estilos CSS en el archivo "style.css"



Lo vamos a transformar  
en ...



HTML con estilos CSS en el archivo "styles.css"



# <TEST>

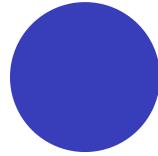
¿Probamos a ver  
que recordamos del  
tema 4?

EMPEZAR >



CSS

```
body {  
    font: x-small  
    background: #  
    color: black;  
    margin: 0;  
    padding: 0;
```



## PREGUNTA 1 DE 10

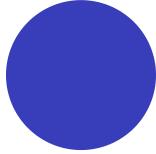
---

**Las clases CSS se aplican en HTML mediante el atributo id.**

Verdadero

Falso

Send



## PREGUNTA 2 DE 10

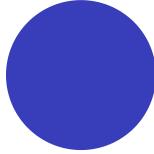
---

**Cuando en CSS se escribe un selector de clase, es necesario escribir, justo delante, el carácter ":"**

Verdadero

Falso

Send



### PREGUNTA 3 DE 10

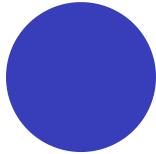
---

**La pseudoclase ":link" permite establecer estilos en los vínculos que aún no han sido visitados**

Verdadero

Falso

Send



## PREGUNTA 4 DE 10

---

**¿que pseudo-clase se utiliza para definir estilos cuando el cursor del ratón se posiciona sobre el elemento?**

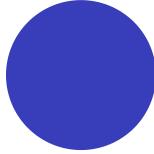
:focus

:link

:hover

:visited

Send



## PREGUNTA 5 DE 10

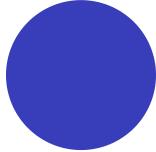
---

**Los selectores id se utilizan para definir estilos únicos aplicables a un elemento web en concreto**

Verdadero

Falso

Send



## PREGUNTA 6 DE 10

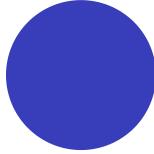
---

**Un selector descendente se utiliza cuando se desean aplicar estilos CSS a elementos que están dentro de ellos (herencia)**

Verdadero

Falso

Send



## PREGUNTA 7 DE 10

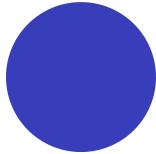
---

**Las propiedades definidas en una etiqueta <style> son prioritarias a las definidas en un atributo style**

Verdadero

Falso

Send



## PREGUNTA 8 DE 10

---

**¿Cómo se pueden insertar estilos CSS en un documento web?**

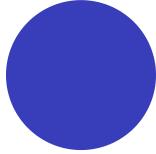
todas son correctas

Mediante el atributo  
style

Mediante un archivo  
externo .css

Mediante la etiqueta  
<style>

Send



## PREGUNTA 9 DE 10

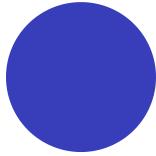
---

**Para que el contenedor <aside> ocupe el 30% del ancho total de la pantalla la sentencia css sería la siguiente: aside { width:30%; }**

Verdadero

Falso

Send



## PREGUNTA 10 DE 10

---

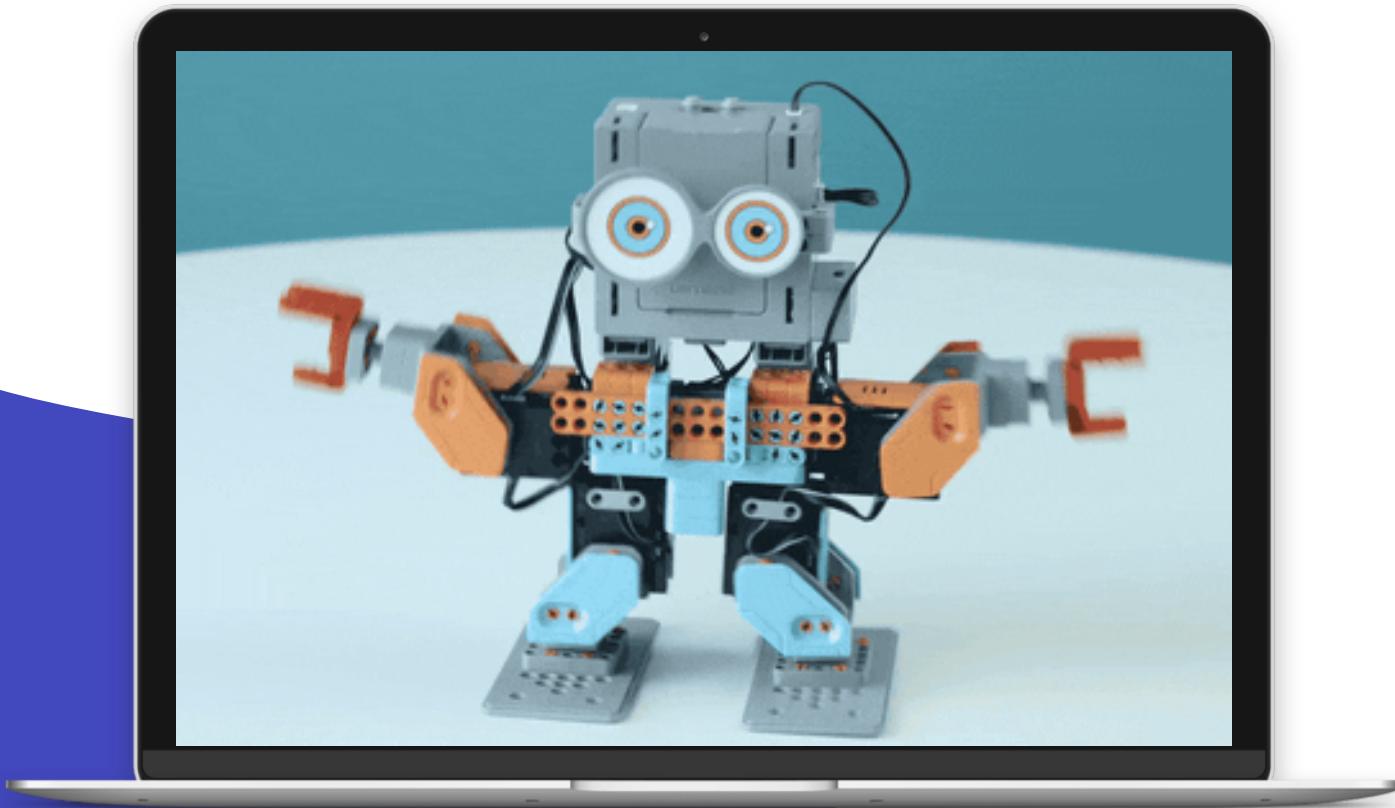
**Para anular los estilos predefinidos usamos el siguiente estilo:**

text-decoration:not

text-decoration:none

Send

# ¡BIEN HECHO!





# Tema 4 finalizado

