Contenido

[CSS son las siglas de Cascade Style Sheet 3](#_Toc100772544)

[Definición de estilos a nivel de elemento HTML 4](#_Toc100772545)

[Ejercicio 1 4](#_Toc100772546)

[Definición de estilos a nivel de página. 6](#_Toc100772547)

[Ejercicio 2 6](#_Toc100772548)

[Propiedades relacionadas a fuentes. 8](#_Toc100772549)

[Ejercicio 3 y 4 8](#_Toc100772550)

[Font-family 8](#_Toc100772551)

[Font-size 9](#_Toc100772552)

[Font-style 9](#_Toc100772553)

[Font-weight 9](#_Toc100772554)

[Font-variant 10](#_Toc100772555)

[Color 10](#_Toc100772556)

[Text-align 10](#_Toc100772557)

[Text-decoration 11](#_Toc100772558)

[Letter-spacing y word-spacing 12](#_Toc100772559)

[Text-transform 12](#_Toc100772560)

[Definición de hojas de estilo en un archivo externo 14](#_Toc100772561)

[Ejercicio 5 14](#_Toc100772562)

[Definición de estilos por medio de clases. 16](#_Toc100772563)

[Ejercicio 5 16](#_Toc100772564)

[Definición de estilos por medio de id. 18](#_Toc100772565)

[Ejercicio 6 18](#_Toc100772566)

[Propiedades relacionadas al borde de un elemento HTML (border-width, border-style, border-color) 20](#_Toc100772567)

[Posicionamiento de un elemento 24](#_Toc100772568)

[Posicionamiento relativo (position: relative) 24](#_Toc100772569)

[Ejercicio 7 24](#_Toc100772570)

[Posicionamiento absoluto (position: absolute) 26](#_Toc100772571)

[Ejercicio 8 26](#_Toc100772572)

[Disposición del contenido en columnas 28](#_Toc100772573)

[Ejercicio 8 28](#_Toc100772574)

CSS son las siglas de Cascade Style Sheet   
(que traducido significa hojas de estilo en cascada).

Las hojas de estilo es una tecnología que nos permite controlar la apariencia de una página web.   
  
En un principio, los sitios web se concentraban más en su contenido que en su presentación.

Con CSS podemos especificar estilos como el tamaño, fuentes, color, espaciado entre textos y recuadros así como el lugar donde disponer texto e imágenes en la página.

Veremos que podemos asociar las reglas de estilo a las marcas HTML de tres maneras: directamente a la marca, en el head de la página o agrupar las reglas de estilo en un archivo independiente con extensión \*.css

## Definición de estilos a nivel de elemento HTML

### Ejercicio 1

Es la forma más fácil pero menos recomendada para aplicación de un estilo a un elemento HTML. Se define en la propiedad style los estilos a definir para dicho elemento.

Es común a veces definir estilos directamente en los elementos HTML cuando estamos probando diseños de elementos particulares de la página y posteriormente trasladar el estilo creado a la zona de definición de estilos.

La sintaxis para definir un estilo a un elemento HTML es la siguiente:

<!DOCTYPE html>

<html>

<head>

<title>Título de la página</title>

<meta charset="UTF-8">

</head>

<body>

<h1 style="color:#ff0000;background-color:#ffff00">

Este mensaje es de color rojo sobre fondo amarillo.

</h1>

</body>

</html>

El resultado en el navegador al cargar esta página es:



Por defecto, todo navegador tiene un estilo definido para cada elemento HTML, lo que hacemos con la propiedad style es redefinir el estilo por defecto. En este problema definimos que el elemento h1 defina como color de texto el rojo y como color de fondo el amarillo:

<h1 style="color:#ff0000;background-color:#ffff00">

Este mensaje es de color rojo sobre fondo amarillo.

</h1>

Veremos más adelante que hay muchas propiedades en CSS. En este primer ejemplo inicializamos las propiedades color (define el color del texto) y background-color (define el color de fondo del texto).

Cada vez que inicializamos una propiedad debemos separarla de la siguiente por punto y coma.

Cuando definimos estilos directamente en el elemento HTML, tenemos que tener en cuenta que el estilo se aplica únicamente a dicho elemento donde inicializamos la propiedad style, es decir, si tenemos dos secciones h1, deberemos definir la propiedad style para cada elemento:

<h1 style="color:#ff0000;background-color:#ffff00">

Primer título

</h1>

<h1 style="color:#ff0000;background-color:#ffff00">

Segundo título

</h1>

Como podemos observar, más allá que los dos estilos son exactamente iguales, estamos obligados a definirlos para cada elemento HTML.

## Definición de estilos a nivel de página.

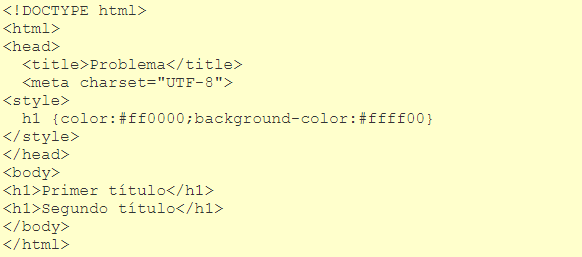
Ejercicio 2  
También podemos hacer la definición de estilos para los distintos elementos HTML de la página en una sección especial de la cabecera que la encerramos entre las marcas HTML (en su interior definimos los estilos para los elementos HTML que necesitemos):

<style>

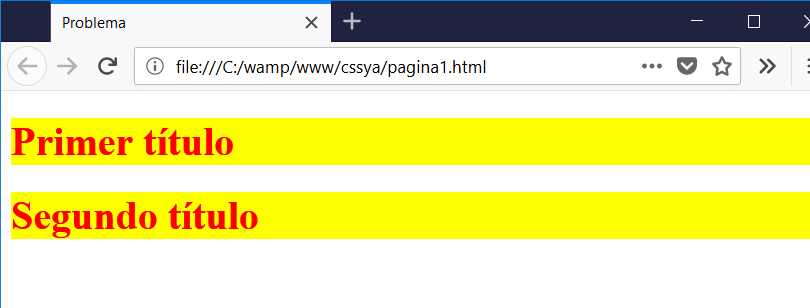
</style>

El problema del concepto anterior donde debíamos crear un estilo similar para el elemento h1 se puede resolver en forma más adecuada empleando la definición de estilos a nivel de página.

Problema: Mostrar dos títulos con texto de color rojo sobre fondo amarillo.



El resultado en el navegador al cargar esta página es:



Podemos observar que en la cabecera de la página definimos la sección:

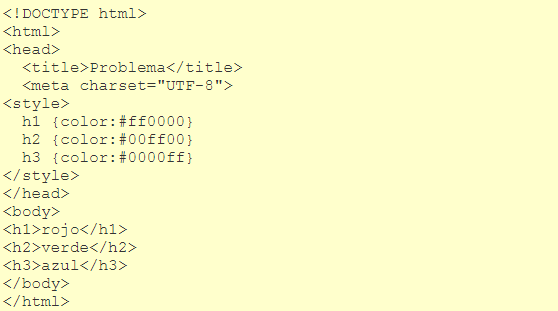
<style>

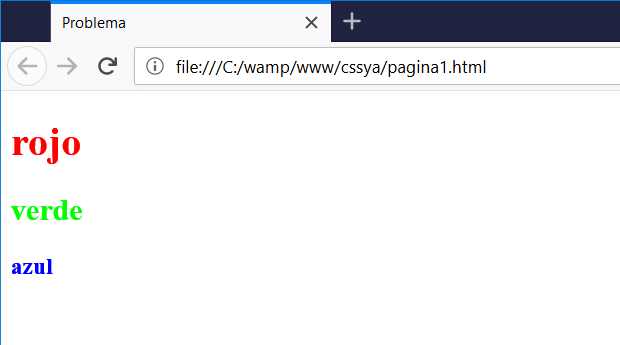
h1 {color:#ff0000;background-color:#ffff00}

</style>

Debe estar encerrada por el elemento style. En este ejemplo indicamos al navegador que en todos los lugares de esta página donde se utilice el elemento h1 debe aplicar como estilo de color de texto el rojo y fondo el amarillo. Podemos observar que es mucho más eficiente que definir los estilos directamente sobre los elementos HTML dentro del cuerpo de la página.

Podemos definir estilos para muchos elementos en la sección style:





## Propiedades relacionadas a fuentes.

Ejercicio 3 y 4  
Contamos con una serie de propiedades relacionadas a fuentes:

font-family

font-size

font-style

font-weight

font-variant

color

text-align

text-decoration

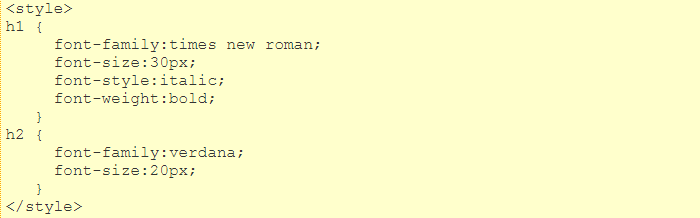
letter-spacing

word-spacing

text-indent

text-transform

Podemos inicializar algunas o todas las propiedades relacionadas a fuentes, un ejemplo:



### Font-family

La primera propiedad inicializada es font-family, algunas de las fuentes más comunes que puede acceder el navegador son:

Arial

Arial Black

Arial Narrow

Courier New

Georgia

Impact

Tahoma

Times New Roman

Verdana

En caso que la fuente no esté disponible el navegador selecciona el estilo por defecto para ese elemento HTML.

Podemos definir varias fuentes por si acaso alguna no se encuentra disponible para el navegador, esto lo hacemos separando por coma todas las fuentes (se eligen de izquierda a derecha):

font-family: verdana, arial, georgia;

### Font-size

La segunda propiedad inicializada es font-size, en nuestro caso indicamos en píxeles:

font-size:30px;

### Font-style

La tercera propiedad es font-style, que puede tener los siguientes valores :

normal

*italic*

oblique

### Font-weight

La última propiedad es font-weight (peso de la fuente), pudiendo tomar los siguientes valores:

normal

bold

bolder

lighter

100

200

300

400

500

600

700

800

900

Esta propiedad indica el peso de la fuente (mientras tenga un valor mayor los caracteres serán más rellenos)

Las propiedades que no se inicializan quedan como responsabilidad al navegador, quien elegirá los valores correspondientes.

### Font-variant

Existe una última propiedad que es font-variant que puede asumir estos dos valores:

small-caps

normal

Define si la fuente se muestra en mayúsculas tipo normal o pequeñas.

### Color

Lo podemos indicar por medio de tres valores hexadecimales que indican la mezcla de rojo, verde y azul. Por ejemplo si queremos rojo puro debemos indicar:

color:#ff0000;

Si queremos verde puro:

color:#00ff00

Si queremos azul puro:

color:#0000ff

### Text-align

Otra propiedad relacionada al texto es text-align, que puede tomar alguno de estos cuatro valores:

left

right

center

justify

Si especificamos:

text-align:center;

### Text-decoration

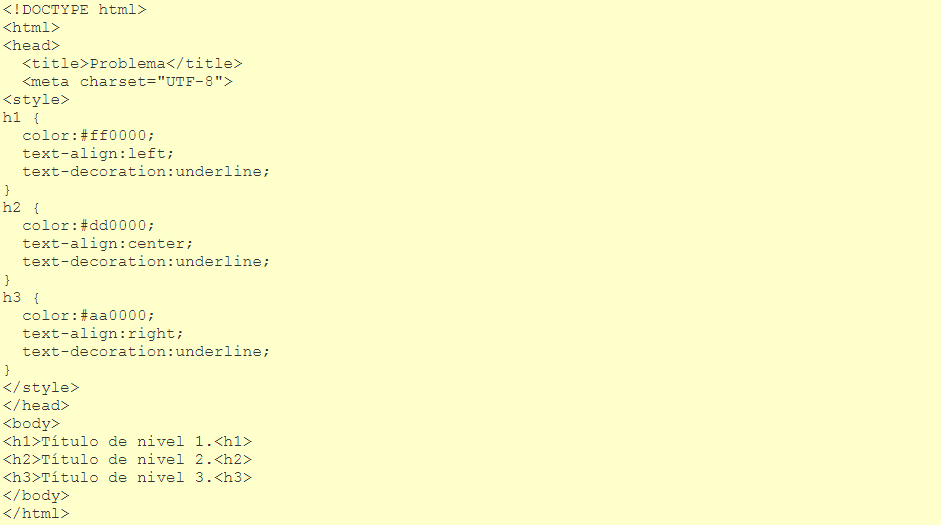
Podemos emplear text-decoration que nos permite entre otras cosas que aparezca subrayado el texto, tachado o una línea en la parte superior, los valores posibles de esta propiedad son:

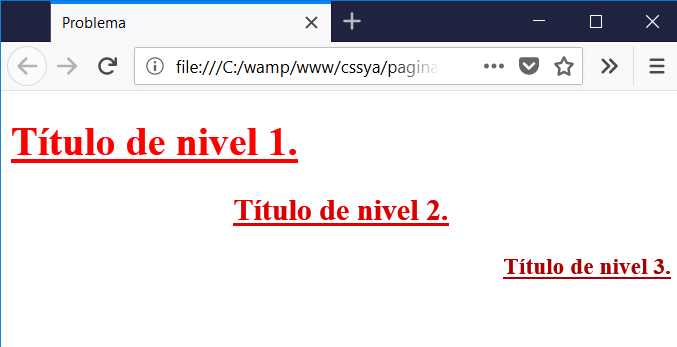
none

underline

overline

line-through





### Letter-spacing y word-spacing

La propiedad letter-spacing y word-spacing permiten indicar el espacio que debe haber entre los caracteres y entre las palabras.

La propiedad text-indent, indenta la primera linea de un texto. A partir de la segunda línea, el texto aparece sin indentación. Podemos indicar un valor negativo con lo que la indentación es hacia la izquierda.

### Text-transform

Por último, la propiedad text-transform puede inicializarse con alguno de los siguientes valores:

none - capitalize - lowercase - uppercase

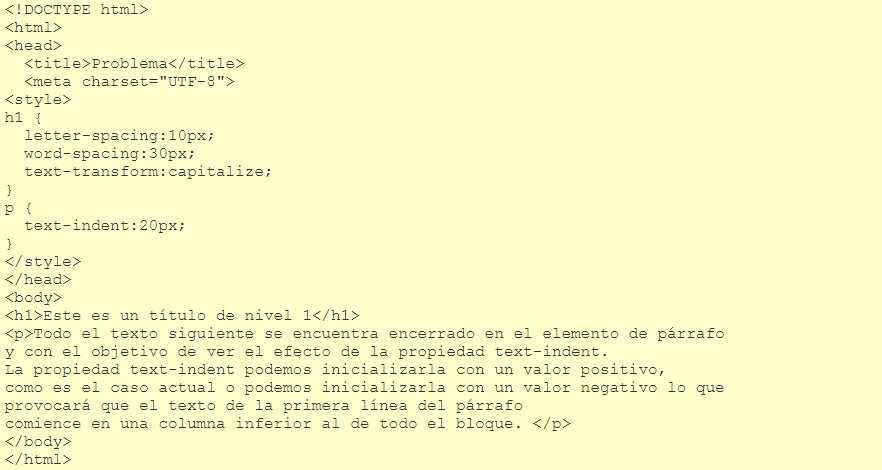
Cada uno de estos valores transforman el texto como sigue:

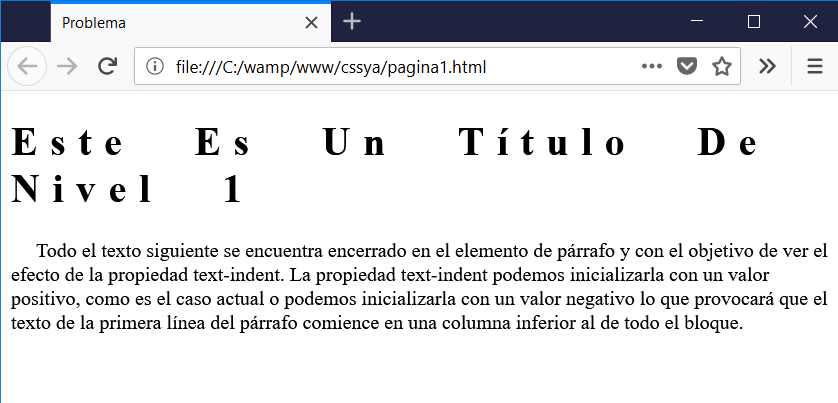
capitalize: Dispone en mayúsculas el primer caracter de cada palabra.

lowercase: Convierte a minúsculas todas las letras del texto.

uppercase: Convierte a mayúsculas todas las letras del texto.

none: No provoca cambios en el texto.





## Definición de hojas de estilo en un archivo externo

Ejercicio 5  
Hasta ahora hemos visto la definición de estilos a nivel de elemento HTML y la definición de estilos a nivel de página. Dijimos que la primera forma es muy poco recomendada y la segunda es utilizada cuando queremos definir estilos que serán empleados sólo por esa página.

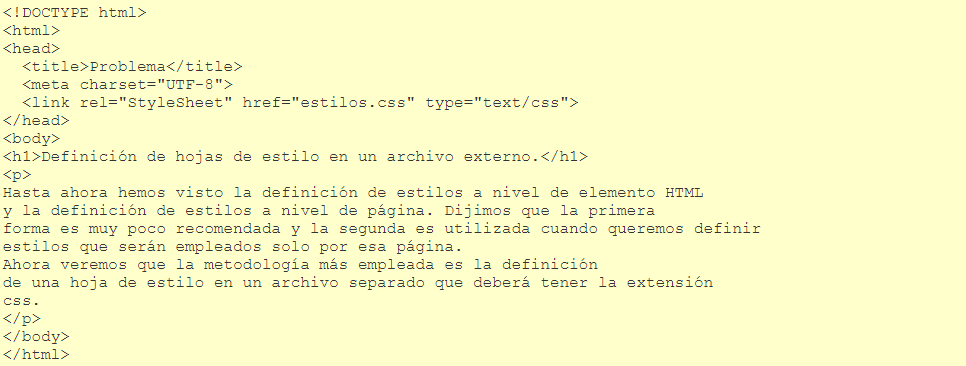
Ahora veremos que la metodología más empleada es la definición de una hoja de estilo en un archivo separado que deberá tener la extensión css.  
Este archivo contendrá las reglas de estilo (igual como las hemos visto hasta ahora) pero estarán separadas del archivo HTML.

La ventaja fundamental es que con esto podemos aplicar las mismas reglas de estilo a parte o a todas las páginas del sitio web. Veremos que esto será muy provechoso cuando necesitemos hacer cambios de estilo (cambiando las reglas de estilo de este archivo estaremos cambiando la apariencia de múltiples páginas del sitio).

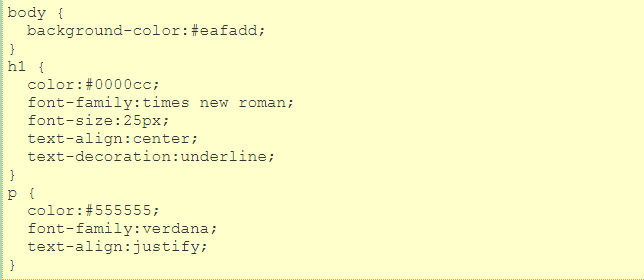
También tiene como ventaja que al programador le resulta más ordenado tener lo referente a HTML en un archivo y las reglas de estilo en un archivo aparte.

Otra ventaja es que cuando un navegador solicita una página, se le envía el archivo HTML y el archivo CSS, quedando guardado este último archivo en la caché de la computadora, con lo cual, en las sucesivas páginas que requieran el mismo archivo de estilos, ese mismo archivo se rescata de la caché y no requiere que el servidor web se lo reenvíe (ahorrando tiempo de transferencia)

Veamos la primera página HTML que tiene asociada una hoja de estilo en un archivo externo. El archivo HTML es (pagina.html):



La hoja de estilos es:



Así quedaría:



Como podemos observar, para indicar el archivo de estilos externo, debemos agregar en la cabecera (head) del archivo HTML la siguiente marca:

<link rel="StyleSheet" href="estilos.css" type="text/css">

La propiedad href hace referencia al archivo externo que afectará la presentación de esta página. En la propiedad type, indica al navegador cual es el formato de archivo. El atributo rel se usa para definir la relación entre el archivo enlazado y el documento HTML.  
Sólo queda probar esta página funcionando.

## Definición de estilos por medio de clases.

Ejercicio 5  
En muchas situaciones una regla de estilo puede ser igual para un conjunto de elementos HTML, en esos casos conviene plantear una regla de estilo con un nombre genérico que posteriormente se puede aplicar a varios elementos de HTML.

Para el planteo de una regla de estilo por medio de una clase creamos un nombre de clase y le antecedemos el caracter punto:

.resaltado{

color:#000000;

background-color:#ffff00;

font-style:italic;

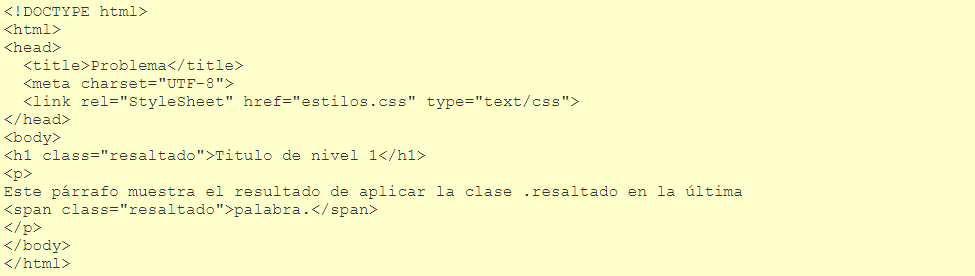
}

Luego para asignar dicha regla a un elemento HTML definimos la propiedad class al elemento que necesitamos fijarle este estilo:

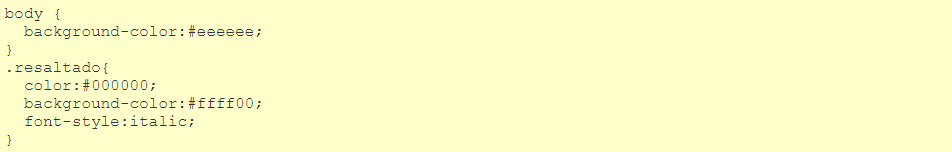
<h1 class="resaltado">Este elemento h1 aparece con la clase resaltado</h1>

Podemos especificar la clase "resaltado" a todos los elementos HTML que necesitemos aplicarle dichas reglas de estilo.

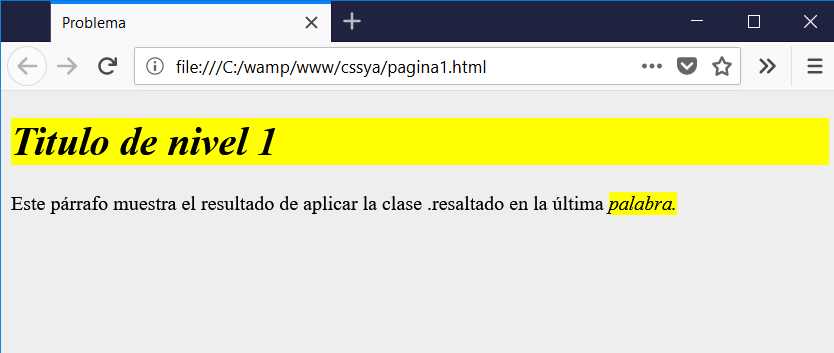
Veamos un ejemplo, la pagina.html es:



La hoja de estilos es:



Así quedaría:



La sintaxis para definir una clase aplicable a cualquier elemento HTML es:

.resaltado{

color:#000000;

background-color:#ffff00;

font-style:italic;

}

Es decir, la inicializamos con el caracter punto y seguidamente un nombre de clase. Dentro definimos las reglas de estilo como hemos venido trabajando normalmente.

El nombre de la clase no puede comenzar con un número.

Luego, para indicar que un elemento sea afectada por esta regla:

<h1 class="resaltado">Titulo de nivel 1</h1>

Es decir, agregamos la propiedad class y le asignamos el nombre de la clase (sin el punto).

## Definición de estilos por medio de id.

Ejercicio 6  
La diferencia fundamental en la definición de un estilo por medio de id con respecto a las clases, es que sólo podremos aplicar dicho estilo a una solo elemento HTML dentro de la página, ya que todos los id que se definen en una página HTML deben tener nombres distintos.

La sintaxis para definir un estilo por medio de id es:

#cabecera {

font-family:Times New Roman;

font-size:30px;

text-align:center;

color:#0000ff;

background-color:#bbbbbb;

}

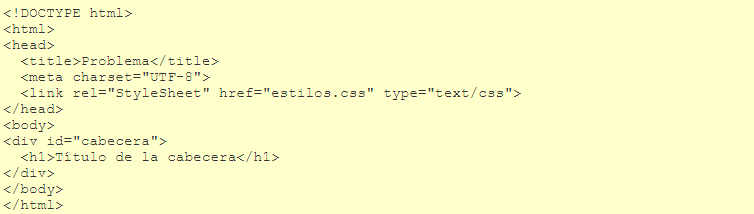
Es decir, utilizamos el carácter numeral (#) para indicar que se trata de un estilo de tipo id.

Luego, sólo un elemento HTML dentro de una página puede definir un estilo de este tipo:

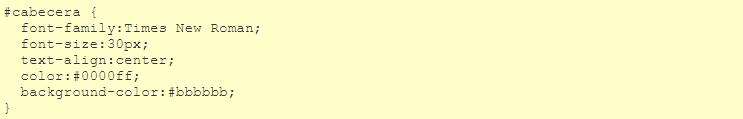
<div id="cabecera">

Hay que subrayar que sólo un elemento HTML puede definir la propiedad id con el valor de cabecera.

Los dos archivos completos del ejemplo entonces quedan (pagina.html):



La hoja de estilo:

En el navegador lo veríamos:



## Propiedades relacionadas al borde de un elemento HTML (border-width, border-style, border-color)

Debemos ahora hacernos la idea que todo elemento que se crea dentro de una página HTML genera una caja. Imaginemos los controles que hemos creado h1, h2, h3, p, em, etc. si fijamos la propiedad background-color veremos que el contenido se encuentra dentro de un rectángulo.

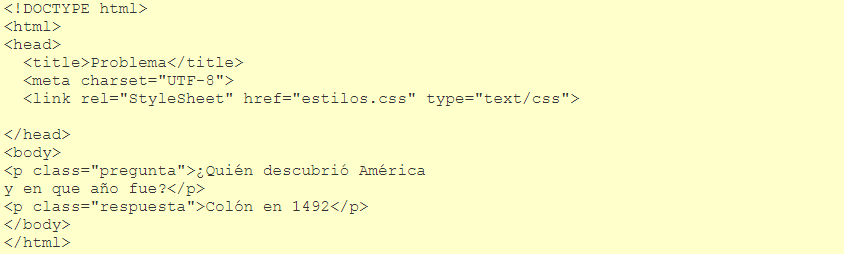
Podemos acceder a las propiedades del borde de ese rectángulo mediante las hojas de estilo CSS; las propiedades más importantes a las que tenemos acceso son:

border-width

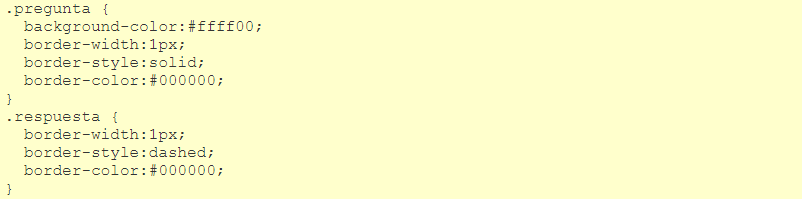
border-style

border-color

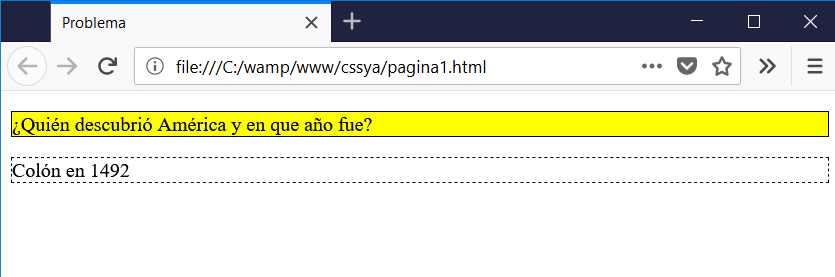
Veamos un ejemplo que inicialice estas propiedades:



La hoja de estilo:



Quedaría así:



Como podemos ver, hemos definido dos clases ".pregunta" que inicializa el color de fondo en amarillo y luego define el ancho del borde en un pixel, el estilo es sólido y el color de la línea de borde es negro.

Luego recordar que para indicar que un elemento tenga este estilo debemos inicializar la propiedad class del elemento HTML respectivo:

<p class="pregunta">Quién descubrió América

y en que año fue?</p>

Al segundo estilo definido lo hemos hecho con la clase ".respuesta"

.respuesta {

border-width:1px;

border-style:dashed;

border-color:#000000;

}

En ésta hemos cambiado el estilo de borde por el valor dashed.

Disponemos de los siguientes estilos de borde:

none

hidden

dotted

dashed

solid

double

groove

ridge

inset

outset

Podemos manejar cada borde de manera independiente con las siguientes propiedades:

border-top-width

border-right-width

border-bottom-width

border-left-width

border-top-style

border-right-style

border-bottom-style

border-left-style

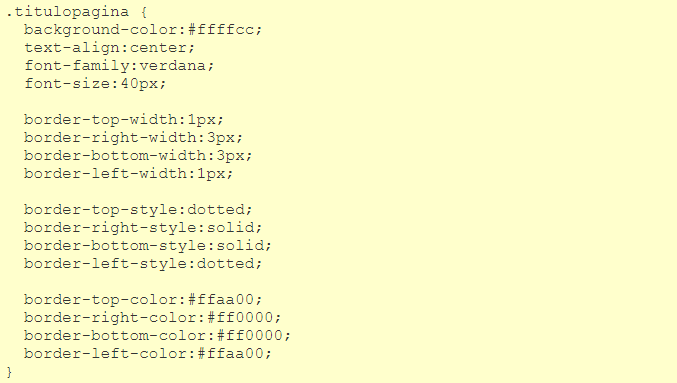
border-top-color

border-right-color

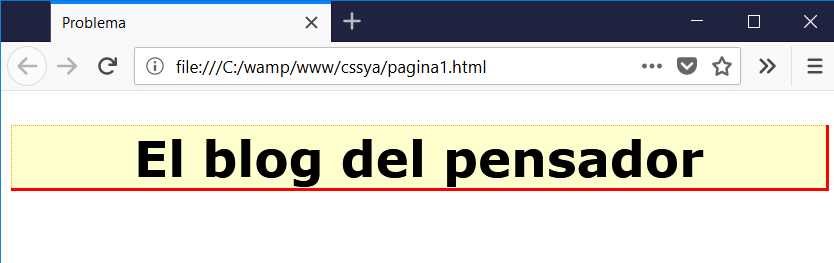
border-bottom-color

border-left-color

Por ejemplo:



Daría como resultado:



## Posicionamiento de un elemento

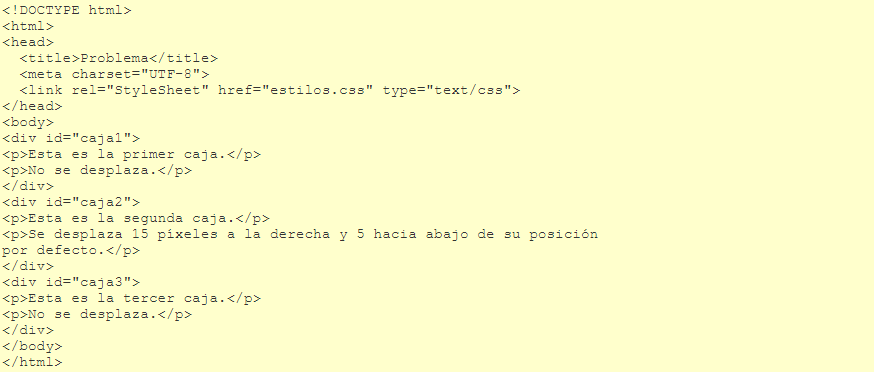
## Posicionamiento relativo (position: relative)

Ejercicio 7  
La propiedad position determina el punto de referencia donde se debe localizar cada elemento HTML. Por defecto esta propiedad se inicializa con el valor static.  
Con el valor por defecto static, cada elemento HTML se localiza de izquierda a derecha y de arriba hacia abajo.

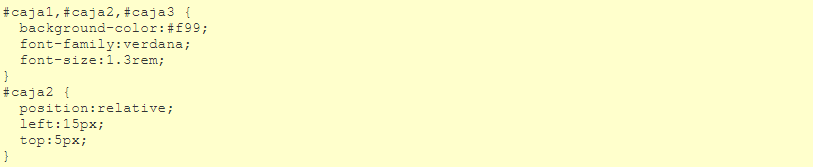
El segundo valor posible para esta propiedad es relative. En caso de fijar la propiedad position con el valor relative, podemos modificar la posición por defecto del elemento HTML modificando los valores left y top (con valores positivos o inclusive negativos)

El posicionamiento relativo mueve un elemento (div, h1, p etc.) en relación a donde se dispondría en el flujo normal. Es decir deja espacio libre.

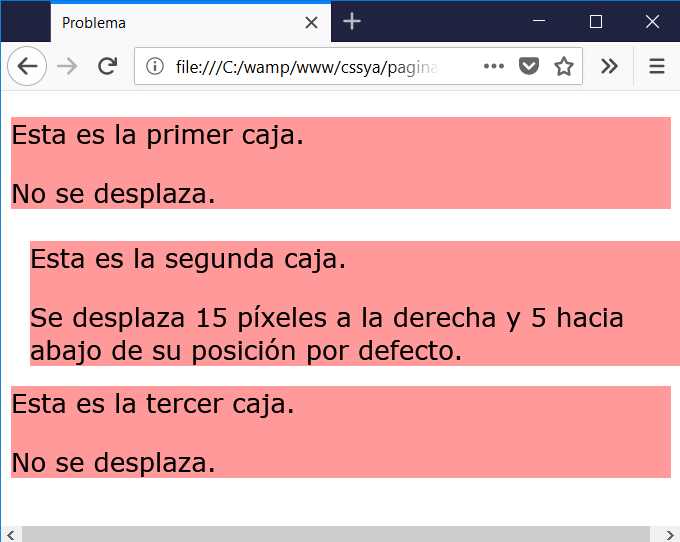
Veamos un ejemplo con tres div, de los cuales el segundo lo desplazamos 20 pixeles a nivel de columna y 5 pixeles a nivel de fila:



La hoja de estilos:



En el navegador lo veríamos:



## Posicionamiento absoluto (position: absolute)

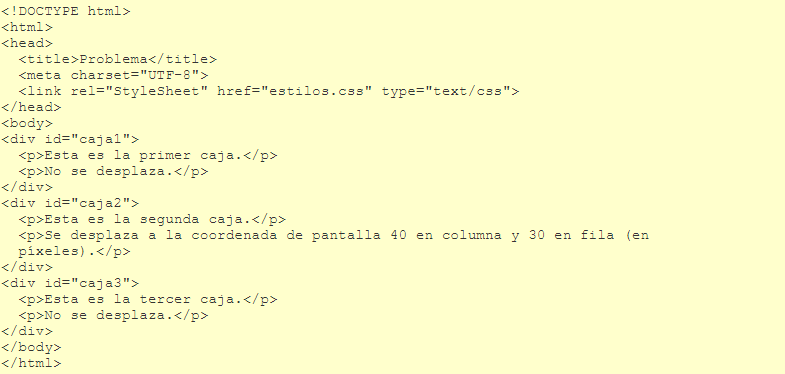
### Ejercicio 8

El posicionamiento absoluto dispone un elemento HTML completamente fuera del flujo de la página. El valor que debemos asignar a la propiedad position es absolute.  
Hay que tener en cuenta que no se reserva espacio en el flujo del documento como pasaba con el posicionamiento relativo (recordemos que con este posicionamiento podemos desplazar el elemento a cualquier parte de la página, pero el espacio por defecto para dicho elemento queda vacío).

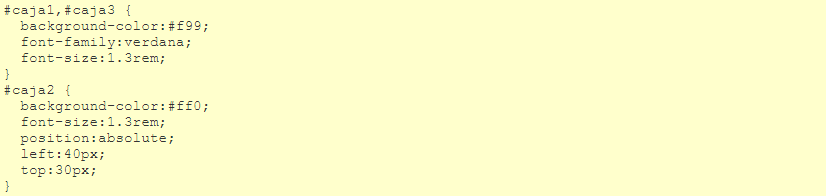
El posicionamiento es siempre con respecto a la página.

Normalmente cuando utilizamos el posicionamiento absoluto inicializamos las propiedades top y left.

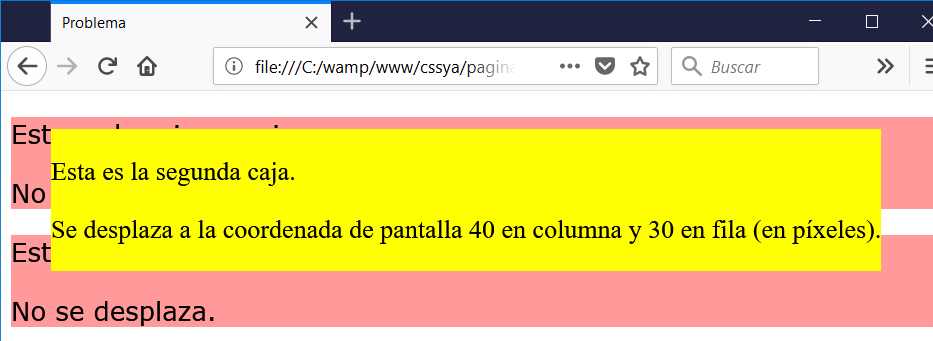
Veamos un ejemplo para ver el funcionamiento del posicionamiento absoluto:



La hoja de estilo quedaría:



En el navegador lo veríamos:



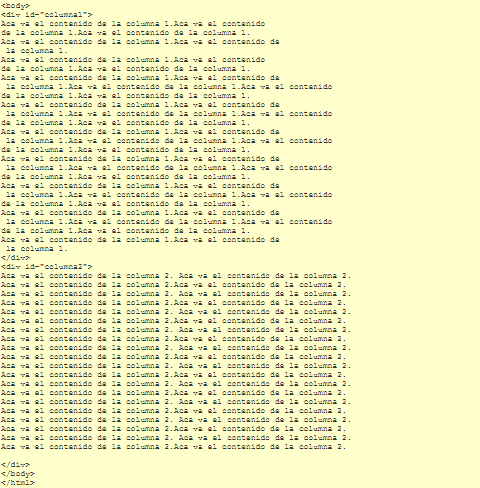
## Disposición del contenido en columnas

### Ejercicio 8

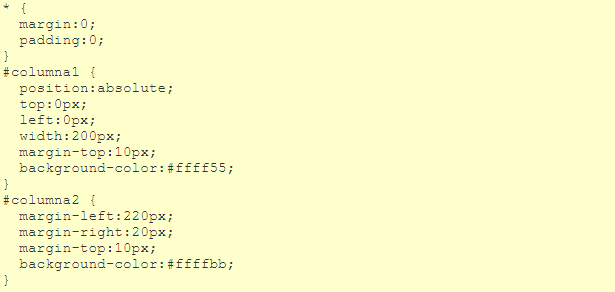
Empezaremos a ver como componer una página sin utilizar las tablas HTML (un medio muy utilizado hasta hace muy poco, ya que es ampliamente soportado por navegadores antiguos).

Una solución para crear una página con dos columnas es utilizar el posicionamiento absoluto:

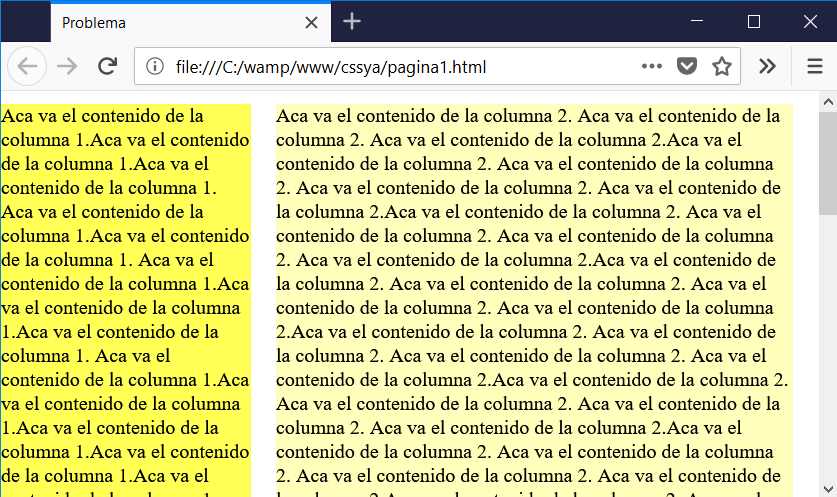
El body del html quedaría así:



La hoja de estilo quedaría así para dos columnas:



La página quedaría así:



La regla definida para la primer columna es:

#columna1 {

position:absolute;

top:0px;

left:0px;

width:200px;

margin-top:10px;

background-color:#ffff55;

}

Es decir, inicializamos la propiedad position con el valor absolute, con lo que debemos indicar la posición del div en la página por medio de las propiedades top y left, en esta caso lo posicionamos en la columna 0 y fila 0 y además inicializamos la propiedad width, con lo cual le estamos indicando que esta columna tendrá siempre 200 píxeles de ancho.

Además inicializamos la propiedad margin-top con 10 píxeles, recordemos que todos los elementos tienen margin y padding cero.

Ahora veamos cómo inicializamos la segunda columna:

#columna2 {

margin-left:220px;

margin-right:20px;

margin-top:10px;

background-color:#ffffbb;

}

Esta regla no inicializa la propiedad position, por lo que el div ocupa la posición que le corresponde por defecto, es decir, empieza en la coordenada 0,0 de la página. El truco está en inicializar la propiedad margin-left con un valor mayor a 200, que es el ancho de la columna1.

El resto de propiedades que inicializamos son el margin-top, para que sea igual que la primera columna y el margin-right, para que no quede el texto pegado a la derecha.