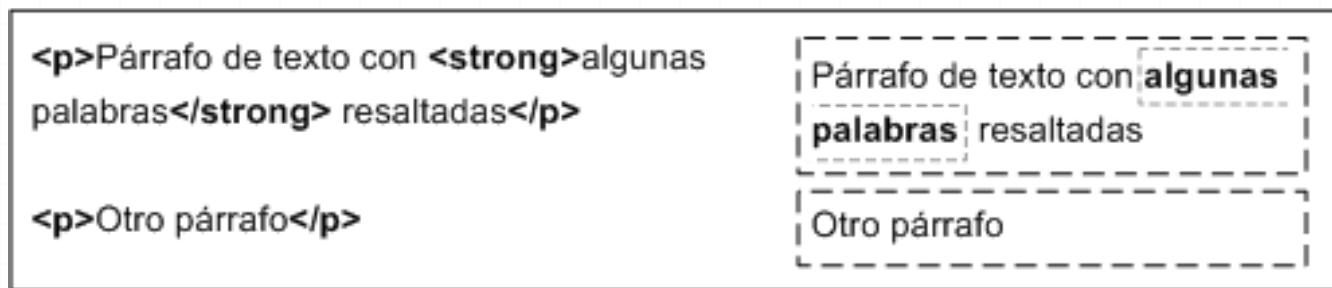


Formato de
fondos

Introducción

- Podemos pensar en que cada elemento HTML está encerrado en una caja rectangular.
- Las cajas de una página se crean automáticamente. Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento.



- Por defecto, las cajas de los elementos no son visibles a simple vista porque inicialmente no muestran ningún color de fondo ni ningún borde.
- Mediante CSS podemos modificar las características de estas cajas para que sean visibles y presenten diferentes aspectos.

background-color

- Esta propiedad nos permite mostrar un color de fondo sólido en la caja de un elemento.
 - Con esta propiedad no se pueden crear degradados ni ningún otro efecto avanzado.
- Su sintaxis es la siguiente:

background-color: color|transparent|initial|inherit;

- color. Especifica un color de fondo de cualquiera de las formas ya vistas
- transparent. Especifica que el color de fondo debería ser transparente. Este es el valor por defecto
- initial. Establece esta propiedad a su valor por defecto
- inherit. Hereda esta propiedad de su elemento padre.

```
h1 {background-color: #F5F5F5; }
```

- Si queremos establecer un color de fondo para toda la página, lo haremos definiendo la propiedad background-color en el elemento <body>. De esta manera, como el valor inicial del fondo de los elementos es transparente, todos los elementos de la página se visualizan con el mismo fondo a menos que algún elemento especifique su propio fondo.

Background-image

- Esta propiedad especifica una o varias imágenes para usar como fondo de un elemento.
- Su sintaxis es la siguiente:

```
background-image: url|none|initial|inherit;
```

- url. Es la url de la imagen, que puede ser absoluta o relativa. Si se quiere especificar más de una imagen se separarán las url por comas (,).

```
body {background-image: url("Mafalda.gif")}  
body {background-image: url("Mafalda1.gif", "Mafalda2.gif")}
```

- none. No se muestra ninguna imagen de fondo. Este es el valor por defecto.
- initial. Establece esta propiedad a su valor por defecto
- inherit. Hereda esta propiedad de su elemento padre.

Fondos con color e imagen

- CSS permite establecer de forma simultánea un color y una imagen de fondo.
- En este caso, la imagen se muestra delante del color, por lo que solamente si la imagen contiene zonas transparentes es posible ver el color de fondo.
- Además, en caso de que la imagen no se pueda mostrar o contenga errores, el navegador mostrará el color indicado (que debería ser, en lo posible, similar a la imagen) y la página no parecerá que contiene errores.

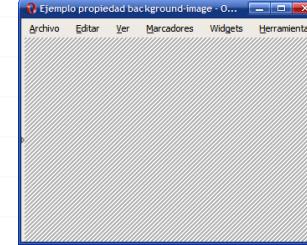
```
body {  
    background-image: url("mafalda.gif");  
    background-color: #cccccc;  
}
```

Ajuste del tamaño de la imagen de fondo

- Si la imagen que se quiere mostrar es demasiado grande para el fondo del elemento, solamente se muestra la parte de imagen comprendida en el tamaño del elemento. Si la imagen es más pequeña que el elemento, CSS la repite horizontal y verticalmente hasta llenar el fondo del elemento.



- Este comportamiento es útil para establecer un fondo complejo a una página web entera. El siguiente ejemplo utiliza una imagen muy pequeña para establecer un fondo complejo a toda una página:

Imagen original	Código css	Resultado
	<pre>body { background-image:url(imagenes/fondo.gif); }</pre>	

- Con una imagen muy pequeña (y que por tanto, se puede descargar en muy poco tiempo) se consigue cubrir completamente el fondo de la página, con lo que se consigue un gran ahorro de ancho de banda.

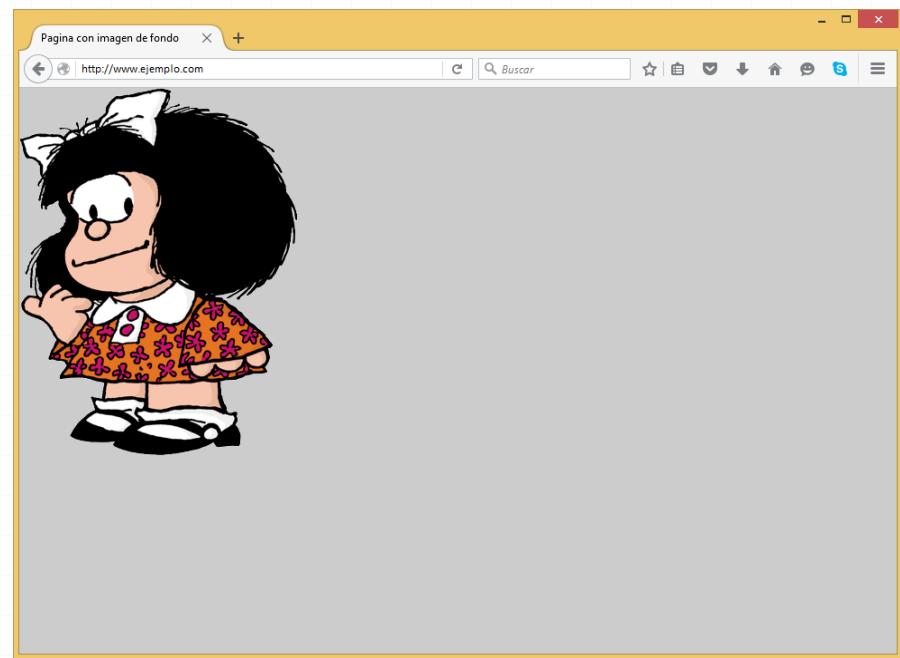
background-repeat

- Como hemos comentado anteriormente, si la imagen es más pequeña que el elemento, CSS la repite horizontal y verticalmente hasta llenar el fondo del elemento.
- En ocasiones, queremos evitar esta repetición de la imagen. Para ello, CSS introduce la propiedad background-repeat que permite controlar la forma de repetición de las imágenes de fondo.
- Su sintaxis es la siguiente:

```
background-repeat: repeat|repeat-x|repeat-y|no-repeat|initial|inherit;
```

- **repeat.** indica que la imagen se debe repetir tanto horizontal como verticalmente. Este es el comportamiento por defecto.
- **repeat-x.** La imagen se repetirá sólo horizontalmente.
- **repeat-y.** La imagen se repetirá solamente de forma vertical.
- **no-repeat.** La imagen no se repetirá.

```
body {  
background-image: url("mafalda.gif");  
background-color: #cccccc;  
background-repeat: no-repeat;  
}
```



background-position

- Si se pone una imagen de fondo, ésta comenzará en la esquina superior izquierda del elemento y se repetirá tanto horizontal como verticalmente.
- Si además utilizamos el atributo `no-repeat`, la imagen no se repetirá pero seguirá ubicándose en la esquina superior izquierda del elemento.
- La propiedad `background-position` permite controlar la posición de la imagen dentro del fondo del elemento.

`background-position: value;`

- Donde `value` puede tomar los valores que se muestran en la siguiente diapositiva.

Valor	Resultado
left top left center left bottom right top right center right bottom center top center center center bottom	Si sólo se especifica una palabra clave, el otro valor será "center"
x% y%	El primer valor es la posición horizontal y el segundo valor es el vertical. La esquina superior izquierda es 0% y 0%. La esquina inferior derecha es 100% 100%. Si sólo se especifica un valor, se entiende que es el desplazamiento horizontal y se asume que el vertical será del 50%. . El valor predeterminado es: 0% 0%
xpos ypos	El primer valor es la posición horizontal y el segundo valor es el vertical. La esquina superior izquierda es 0 0. Las unidades pueden ser píxeles (0px 0px) o cualquier otra unidad de CSS. Si sólo se especifica un valor, se entiende que es el desplazamiento horizontal y se asume que el vertical será del 50%. Se pueden mezclar % y posiciones
initial	Establece esta propiedad a su valor por defecto

```
body {  
background-image: url("mafalda.gif");  
background-color: #cccccc;  
background-repeat: no-repeat;  
background-position: 50% 200px;  
}
```



Background-attachment

- Esta propiedad permite controlar si una imagen de fondo se fija o se desplaza con el resto de la página cuando el usuario utiliza las barras de scroll.
- Su sintaxis es la siguiente:

`background-attachment: scroll|fixed|local|initial|inherit;`

- `scroll`. Es el comportamiento por defecto. En esta situación el fondo se desplaza junto con el elemento.
- `fixed`. Hace que la imagen de fondo se muestre fija al desplazar la ventana del navegador. Suele ser un efecto no deseable y que no todos los navegadores soportan correctamente.

background-size (CSS3)

- Esta propiedad permite especificar el tamaño de las imágenes de fondo.
- Su sintaxis es la siguiente:

`background-size: auto | length | cover | contain | initial | inherit;`

- `auto`. La imagen de fondo conserva su ancho y alto original. Esta es la opción por defecto.
- `length`. Establece el ancho y alto de la imagen de fondo. El primer valor establece el ancho, el segundo valor define la altura. Si sólo se da un valor, el segundo se establece en "auto".

`background-size: 100px 100px;`

- Se puede especificar el ancho y alto en %.

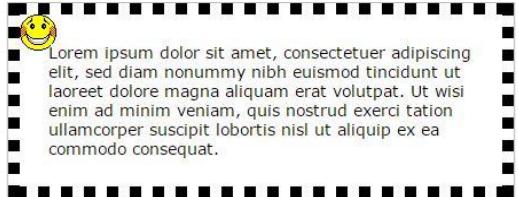
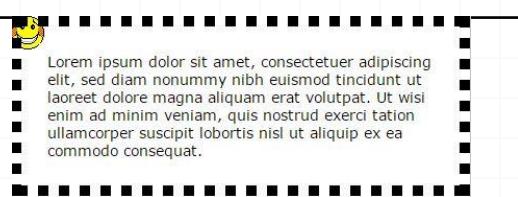
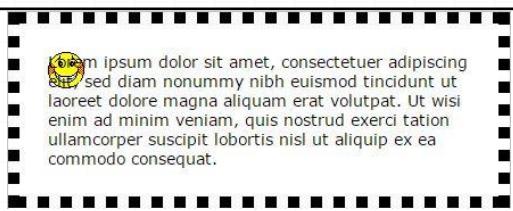
`background-size: 100% 100%;`

- `cover`. Escala la imagen de fondo al tamaño más grande posible para que cubra todo el área de fondo. Puede ocurrir que algunas partes de la imagen no se vean.
- `contain`. Escala la imagen al tamaño más grande posible pero de manera que toda la imagen sea visible.
- `initial`. Establece esta propiedad a su valor por defecto
- `inherit`. Hereda esta propiedad de su elemento padre.

background-origin (CSS3)

- Esta propiedad nos permite especificar el lugar donde se coloca la imagen de fondo.
- Su sintaxis es la siguiente:

```
background-origin: padding-box|border-box|content-box|initial|inherit;
```

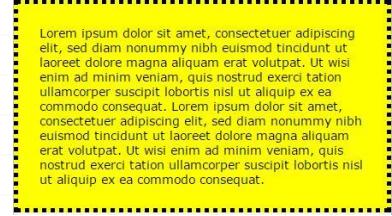
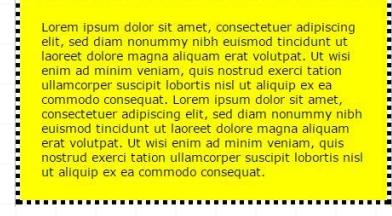
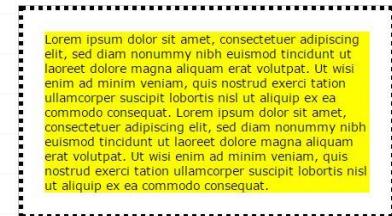
Valor	Efecto	Ejemplo
<code>padding-box</code>	<p>La imagen de fondo se inicia desde la esquina superior izquierda del límite del relleno.</p> <p>Este es el valor por defecto</p>	 Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. 
<code>border-box</code>	<p>La imagen de fondo se inicia desde la esquina superior izquierda del borde</p>	 Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. 
<code>content-box</code>	<p>La imagen de fondo se inicia a partir de la esquina superior izquierda del contenido</p>	 Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. 
<code>initial</code>	<p>Establece esta propiedad a su valor por defecto</p>	
<code>inherit</code>	<p>Hereda esta propiedad de su elemento padre.</p>	

Si la propiedad `background-attachment` está establecida en `fixed`, `background-origin` no tendrá efecto.

background-clip

- La propiedad background-clip especifica el área de la pintura del fondo.
- Su sintaxis es la siguiente:

background-clip: border-box|padding-box|content-box|initial|inherit;

Valor	Efecto	Ejemplo
Border-box	<p>El color de fondo rellena hasta el borde de la caja.</p> <p>Este es el valor por defecto</p>	
padding-box	El color de fondo rellena la parte interna de la caja, sin incluir el borde.	
content-box	El color de fondo rellena el contenido del elemento.	
initial	Establece esta propiedad a su valor por defecto	
inherit	Hereda esta propiedad de su elemento padre.	

background-blend-mode

- La propiedad CSS background-blend-mode describe cómo se deberán mezclar entre sí las imágenes de fondo y el color de fondo de un elemento.
- Su sintaxis es la siguiente:

```
background-blend-mode:blend-mode
```

- donde blend-mode puede tomar los valores: normal, multiply, screen, overlay, darken, lighten, color-dodge, saturation, color o luminosity.
- Pueden aplicarse varios modos de fusión a la vez.
- NOTA. Esta propiedad no la soportan los navegadores edge ni Internet Explorer

background

- CSS define una propiedad de tipo "shorthand" para indicar todas las propiedades de los colores e imágenes de fondo de forma directa. La propiedad se denomina `background` y es la que generalmente se utiliza para establecer las propiedades del fondo de los elementos.
- Su sintaxis es la siguiente:
background: color image position/size repeat origin clip attachment initial|inherit;
- El orden en el que se indican las propiedades es indiferente, aunque en general se sigue el formato indicado de color, url de imagen, repetición y posición.
- La propiedad `background` permite asignar todos o sólo algunos de todos los valores que se pueden definir para los fondos de los elementos

```
/* Color e imagen de fondo de la página mediante una propiedad shorthand */
body { background: #222d2d
url("./graphics/colorstrip.gif") repeat-x 0 0; }

/* La propiedad shorthand anterior es equivalente a las siguientes propiedades */
body {
    background-color: #222d2d;
    background-image: url("./graphics/colorstrip.gif");
    background-repeat: repeat-x;
    background-position: 0 0;
}
```



Manos a la obra

- ① Poner como fondo un logotipo de html5 a la página con el texto de html5 creado anteriormente. Centrarlo en el medio de la página para que no se repita. Poner también un color de fondo

Gradientes

Degrados

- Los degradados son un recurso muy utilizado por los diseñadores para decorar las webs. Hasta la llegada de CSS si se querían incluir degradados en las páginas web teníamos que implementarlos a través de imágenes creadas con un programa de tratamiento de imágenes e insertarlos en la página como imagen.
- Esto tiene varios inconvenientes:
 - Primero, las imágenes pueden llegar a ser muy pesadas lo cual puede ralentizar excesivamente la carga de la página.
 - En segundo lugar, si se quiere modificar algún parámetro del degradado (como por ejemplo, los colores del mismo) debe retocarse la imagen en un programa de retoque digital y volver a generar la imagen de nuevo.
- Con la utilidad gradiente de CSS3 se pueden incluir degradados en las páginas utilizando estilos CSS en lugar de necesitar imágenes.

Tipos de gradientes

- CSS3 permite definir dos tipos de gradientes:
 - Gradientes lineales (lineal gradients)
 - Este tipo de gradiente crea un degradado que va de un color a otro de manera lineal. Puede ser de arriba a abajo, de izquierda a derecha y viceversa. Incluso se puede conseguir un degradado con cualquier ángulo.
 - Gradientes radiales (Radial gradients). Este tipo de gradientes permite implementar un degradado que se distribuye radialmente, desde un punto del elemento hacia fuera, de manera circular que puede tener el mismo valor de radio (para hacer degradados en círculos perfectos) o con valores de radio variables (lo que generaría elipses).

Degrados
lineales

Crear un gradiente lineal

○ Para crear un gradiente lineal deberemos definir al menos dos colores. Estos colores serán el color inicial del que queremos partir y el color final al que queremos llegar haciendo transiciones suaves entre ellos.

○ Crearemos un gradiente lineal con el atributo:

`background: linear-gradient(direction, color-stop1, color-stop2, ...);`

○ direction. Permite indicar desde dónde comenzará el degradado. Podemos indicar varios valores según donde queramos que empiece: desde arriba o abajo, desde la izquierda o la derecha,...

○ lista de colores. Indicaremos todos los colores que queramos utilizar en el degradado separados por comas.

○ En lugar de direction podemos utilizar un ángulo para que el degradado no se realice en horizontal o vertical sino con alguna inclinación

Ejemplo 1

```
Body{background: linear-gradient(orange, pink);}
```

- 0 Cada navegador tiene una sintaxis ligeramente diferente para los gradientes por lo que para que un usuario vea correctamente los degradados que hayamos incluido en nuestra página independientemente del navegador que esté utilizando deben escribirse las etiquetas de estilos propietarias de cada navegador.

```
body{  
    /* Para Safari 5.1 a 6.0 */  
    background: -webkit-linear-gradient(orange, pink);  
    /* Para Firefox 3.6 a 15*/  
    background: -moz-linear-gradient(orange, pink);  
    /* Para Opera 11.1 a 12.0 */  
    background: -o-linear-gradient(orange, pink);  
    /* Sintaxis estándar*/  
    background: linear-gradient(orange, pink);  
    /* Para los navegadores que no admiten degradado*/  
    background: orange;  
}
```

Ejemplo 2 – Degradado lineal de izquierda a derecha

```
body {  
  
    /* Para Safari 5.1 a 6.0 */  
    background: -webkit-linear-gradient(left, orange, pink);  
  
    /* Para Firefox 3.6 a 15 */  
    background: -moz-linear-gradient(right, orange, pink);  
  
    /* Para Opera 11.1 a 12.0 */  
    background: -o-linear-gradient(right, orange, pink);  
  
    /* Sintaxis estándar */  
    background: linear-gradient(to right, orange, pink);  
  
    /* Para los navegadores que no admiten degradado */  
    background-color: orange; }  
}
```

Ejemplo 3 – Degradado lineal diagonal

SE puede crear un degradado lineal diagonal especificando las posiciones de inicio tanto vertical como horizontal

```
body {  
    /* Para Safari 5.1 a 6.0 */  
    background: -webkit-linear-gradient(left top, orange, pink);  
  
    /* Para Firefox 3.6 a 15*/  
    background: -moz-linear-gradient(bottom right, orange, pink);  
  
    /* Para Opera 11.1 a 12.0 */  
    background: -o-linear-gradient(bottom right, orange, pink);  
  
    /* Sintaxis estándar*/  
    background: linear-gradient(to bottom right, orange, pink);  
  
    /* Para los navegadores que no admiten degradado*/  
    background-color: orange; }  
}
```

Ejemplo 4 – Degradado especificando un ángulo

0 Podemos crear un degradado lineal especificando el ángulo en lugar de la dirección

```
background: linear-gradient(angle, color-stop1, color-stop2, ...);
```

```
body{  
    /* Para Safari 5.1 a 6.0 */  
    background: -webkit-linear-gradient(-90deg, orange, pink);  
    /* Para Firefox 3.6 a 15 */  
    background: -moz-linear-gradient(-90deg, orange, pink);  
    /* Para Opera 11.1 a 12.0 */  
    background: -o-linear-gradient(-90deg, orange, pink);  
    /* Sintaxis estándar */  
    background: linear-gradient(-90deg, orange, pink);  
    /* Para los navegadores que no admiten degradado */  
    background: orange;  
}
```



Manos a la obra

- 0 Definir un fondo para la página que estamos usando de ejemplo que consista en un gradiente lineal vertical que pase por todos los colores del arcoiris

Transparencias

Transparencias

- Los gradientes CSS3 permiten definir transparencias.
- Para añadir una transparencia a un degradado deberemos utilizar la función `rgba()` que permite definir el color a utilizar pero además nos deja añadirle una transparencia.
- La función `rgba` cuenta con cuatro parámetros. Los tres primeros son el código del color a utilizar en formato `rgb` y el último parámetro es un valor de 0 a 1 para indicar el grado de transparencia que se quiere utilizar (0=totalmente transparente, 1=ninguna transparencia)
 - `rgba(255,0,0,0)`--> Representaría un rojo totalmente transparente
 - `rgba(255,0,0,1)`--> Representaría un rojo sin transparencia

```
body {  
  
    /* Para Safari 5.1 a 6.0 */  
    background: -webkit-linear-gradient(left,  
        rgba(255,71,0,0),rgba(255,71,0,1));  
  
    /* Para Firefox 3.6 a 15*/  
    background: -moz-linear-gradient(right,  
        rgba(255,71,0,0),rgba(255,71,0,1));  
  
    /* Para Opera 11.1 a 12.0 */  
    background: -o-linear-gradient(right,  
        rgba(255,71,0,0),rgba(255,71,0,1));  
  
    /* Sintaxis estándar*/  
    background: linear-gradient(to right,  
        rgba(255,71,0,0),rgba(255,71,0,1));  
  
    /* Para los navegadores que no admiten degradado*/  
    background-color: white; }  
}
```

Degradados
radiales

- Un gradiente radial se define desde su centro.
- Para crear un gradiente radial utilizaremos la siguiente sintaxis:

background:

`radial-gradient(shape size at position, start-color, ..., last-color);`

- Shape es la forma del degradado. Puede ser circle o ellipse
- size. Permite especificar un tamaño para el degradado.
- Lista de colores. Lista de colores que se van a utilizar en el degradado.



Manos a la obra

○ Ejemplo 6.

○ Definir un degradado radial que vaya desde el color con código 5FFFFB al color FF1B09 y que acabe con el color 3ACC1B)

○ Ejemplo 7.

○ Definir un degradado radial que vaya desde el color con código 5FFFFB (15%) al color FF1B09 (%) (y que acabe con el color 3ACC1B

○ Ejemplo 8.

○ Sobre el ejercicio anterior probar las formas ellipse y circle.

Definir el tamaño con keywords

- El parámetro tamaño permite definir el tamaño del gradiente.
- Este tamaño se puede determinar mediante el uso de varias palabras clave que son:
 - closest-side
 - farthest-side
 - closest-corner
 - farthest-corner



Background: radial-gradient(*closest-side*, #5FFFFB, #FF1B09, #3ACC1B);

- Mediante porcentajes, podemos definir el desplazamiento del degradado en la ventana
- radial-gradient(60% 45%, *closest-side*, #5FFFFB, #FF1B09, #3ACC1B);

Repetir gradiente

Gradientes con repetición

- CSS3 nos permite generar gradados con repetición
- Estos degradados consisten en un degradado entre dos o más colores que se repite varias veces a lo largo del fondo del elemento.
- Los degradados con repetición se consiguen de manera muy similar a la que hemos visto anteriormente solo que tenemos que añadir el atributo
 - `repeating-linear-gradient` para el caso de que queramos generar un degradado con repetición lineal
 - `repeating-radial-gradient` para el caso de que queramos generar un degradado con repetición radial



Manos a la obra

- Abrir el primer ejercicio de degrados lineales realizado y cambiarlo para que el degradado se repita.

`repeating-linear-gradient(orange 5%, pink 10%);`

- Abrir el primer ejercicio de degrados radiales realizado y cambiarlo para que el degradado se repita.

`repeating-radial-gradient(#FFFFFB, #FF1B09 5%, #3ACC1B 15%);`



Formato de
bordes

Introducción

- CSS permite modificar el aspecto de cada uno de los cuatro bordes de la caja de un elemento.
- Las propiedades de borde de CSS permiten definir el estilo, ancho y color de los bordes de los diferentes elementos HTML.

Estilo de borde

- CSS permite establecer el estilo que deberán tener los bordes de una caja.
- Estos estilos pueden ser:

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

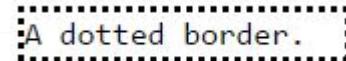
border-style

- La propiedad que nos permite definir cada uno de los estilos de borde vistos anteriormente es border-style.

selector {border-style: valor;}

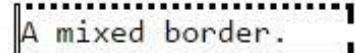
- Valor. Es el tipo de borde que queremos utilizar y puede ser: dotted, dashed, solid, double, groove, ridge, inset, outset, none o hidden.

h1 {border-style: dotted;}



- Si utilizamos el formato selector {border-style:valor;} se aplicará el mismo borde a los cuatro lados de la caja.
- Es posible asignar un tipo de borde a cada lado de la caja:

h1 {border-style: dotted dashed solid double;}



- En este caso se supone que el primer valor es el borde superior, el segundo el borde derecho, el tercero el borde inferior y el cuarto valor es el borde izquierdo.
- Si no configuramos el estilo de borde mediante la propiedad border-style ninguna de las otras propiedades que veremos posteriormente (ancho, color, etc.) tendrán efecto.

Funcionamiento de border-style

○ La propiedad `border-style` puede tener de 1 a 4 valores.

○ Si `border-style` tiene cuatro valores:

○ **border-style: dotted solid double dashed;**

- El borde superior se establece en dotted
- El borde derecho se establece en solid
- El borde inferior se establece en double
- El borde izquierdo se establece en dashed

○ Si `border-style` tiene tres valores:

○ **border-style: dotted solid double;**

- El borde superior se establece en dotted
- El borde derecho se establece en solid
- El borde inferior se establece en double

○ Si `border-style` tiene dos valores:

○ **border-style: dotted solid;**

- Los bordes superior e inferior se establecen en dotted
- Los bordes derecho e izquierdo se establecen en solid

○ Si `border-style` tiene un valor:

○ **border-style: dotted;**

- Los cuatro bordes se establecen en dotted

.....
| **Lorem Ipsum** |

Ancho del borde

- La propiedad `border-width` nos permite definir el ancho de los bordes de una caja.
- Su sintaxis es la siguiente:

`selector {border-width: valor;}`

- **Valor.** Es el ancho que se quiere dar a los bordes. Este ancho puede indicarse en cualquier unidad de medida (absoluta o relativa) o mediante las palabras clave `thin` (borde delgado), `medium` (borde normal) y `thick` (borde ancho)

`h1 {border-width: 5px;}`

La unidad de medida más habitual para establecer el grosor de los bordes es el píxel, ya que es la que permite un control más preciso sobre el grosor. Las palabras clave apenas se utilizan, ya que el estándar CSS no indica explícitamente el grosor al que equivale cada palabra clave, por lo que pueden producirse diferencias visuales entre navegadores

Funcionamiento de border-width

- Al igual que ocurría con border-style, la propiedad border-width permite indicar entre uno y cuatro valores:
 - Si se indica un solo valor, se aplica a los cuatro bordes.
 - Si se indican dos valores, el primero se aplica al borde superior e inferior y el segundo valor se aplica al borde izquierdo y derecho.
 - Si se indican tres valores, el primero se aplica al borde superior, el segundo se aplica al borde izquierdo y derecho y el tercer valor se aplica al borde inferior.
 - Si se indican los cuatro valores, el orden de aplicación es superior, derecho, inferior e izquierdo.

Expresión	Vista
<code>h1 {border-width: 5px;}</code>	 Lorem Ipsum
<code>h1 {border-width: 2px 5px;}</code>	 Lorem Ipsum
<code>h1 {border-width: 2px 4px 8px;}</code>	 Lorem Ipsum
<code>h1 {border-width: 2px 4px 8px 10px;}</code>	 Lorem Ipsum

Color de borde

- CSS permite establecer el color de los bordes con la propiedad border-color.

selector {border-color: color;}

- El color se puede especificar de cualquiera de las maneras vistas: nombre, código rgb o código hexadecimal.
- Al igual que con las propiedades anteriores, la propiedad border-color puede tomar de uno a cuatro valores.

Expresión	Vista
h1 {border-color: red;}	
h1 {border-color: red green;}	
h1 {border-color: red green blue;}	
h1 {border-color: red green blue yellow;}	

Propiedades de bordes individuales

- Cada una de las propiedades vistas anteriormente (estilos, ancho y color) ofrecen la posibilidad de configurar individualmente cada borde con las propiedades de borde individuales

Propiedad	Propiedad individual	Ejemplo
Border-style	border-top-style border-right-style border-bottom-style border-left-style	<pre>p { border-top-style: dotted; border-right-style: solid; border-bottom-style: dotted; border-left-style: solid; }</pre>
Border-width	border-top-width border-right-width border-bottom-width border-left-width	<pre>p { border-top-width: 2px; border-right-width: 4px; border-bottom-width: 8px; border-left-width: 10px; }</pre>
Border-color	border-top-color border-right-color border-bottom-color border-left-color	<pre>p { border-top-color: red; border-right-color: green; border-bottom-color: blue; border-left-color: yellow; }</pre>

border shorthand

- La propiedad shorthand border nos permite definir todas las propiedades de borde en una sola sentencia.
- Con la propiedad shorthand border podemos especificar:
 - border-width
 - border-style (requerido)
 - border-color

```
h1 {  
    border: 5px solid red;  
}
```

Dimensiones

Introducción

- 0 Las propiedades CSS dimensionales nos permiten establecer las dimensiones de los elementos.

Ancho y alto

Ancho y alto

- Las propiedades `width` y `height` permiten definir el ancho y el alto de los elementos.
- La sintaxis de estas dos propiedades es:

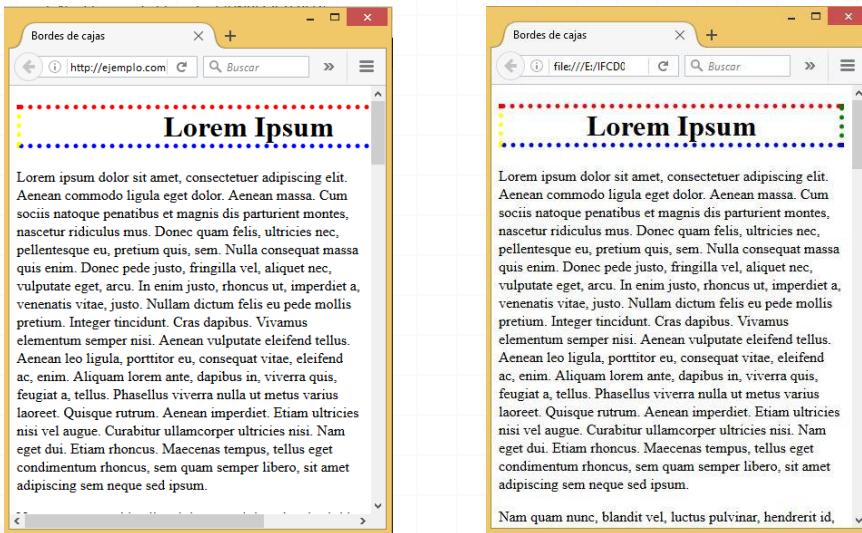
```
width: valor;  
height: valor;
```

- Por defecto el valor está establecido en `auto`, lo que significa que el navegador calculará automáticamente el ancho y alto del elemento.
- Podemos establecer este valor indicando una medida. Esta medida puede ser absoluta, relativa o un porcentaje.
 - Si se indica un porcentaje, éste se calcula en función de su elemento padre.

```
h1 {  
    width: 500px;  
    height: 100px;  
}
```

Ancho máximo

- La propiedad max-width nos permite establecer el ancho máximo que se quiere permitir a un elemento.
- Esta propiedad está pensada para resolver situaciones como la siguiente:
 - Supongamos que creamos un h1 al que ponemos un ancho de 500px. Si reducimos la ventana del navegador por debajo de los 500px, esto haría que el navegador ocultara la parte del h1 que no entra en la ventana y mostraría la barra de desplazamiento horizontal.
 - Esta barra en ocasiones resulta poco práctica por lo que sería mejor que en caso de reducción de la ventana del navegador, el cuadro se redujera para entrar completo en la ventana y que no hubiera que desplazar la ventana en horizontal. Esto se consigue mediante la propiedad max-width



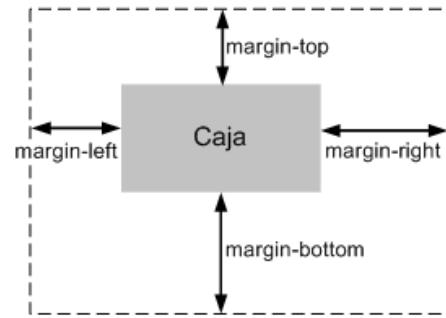
La sintaxis de max-width es `max-width=valor` donde valor es una medida que puede estar expresada de forma absoluta, relativa o porcentaje

También existe el comando max-height aunque su uso no es tan frecuente

Márgenes

Márgenes

- Las propiedad margin de CSS se usan para generar espacio alrededor de un elemento. Es decir, sirve para establecer el espacio en blanco fuera del borde.
- Los márgenes de un elemento son transparente, no tienen color de fondo.
- CSS tiene propiedades para especificar el margen para cada uno de los lados de un elemento:
 - margin-top
 - margin-right
 - margin-bottom
 - margin-left
- Cada uno de los márgenes anteriores puede tomar diferentes valores:
 - auto. El navegador calcula el margen.
 - valor. Especifica una medida para el margen. Esta medida puede ser absoluta o relativa.
 - %. Especifica el margen en un porcentaje de la anchura del elemento que la contiene
 - inherit



Ejemplo

```
h1 {  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 80px;  
}
```

- Los márgenes verticales (margin-top y margin-bottom) sólo se pueden aplicar a los elementos de bloque y las imágenes, mientras que los márgenes laterales (margin-left y margin-right) se pueden aplicar a cualquier elemento
- Cuando queremos definir una sangría de párrafo a la izquierda es más correcto cambiar el margen izquierdo del párrafo que la etiqueta <blockquote>. La etiqueta <blockquote> deberíamos utilizarla únicamente para citas.

margin shorthand

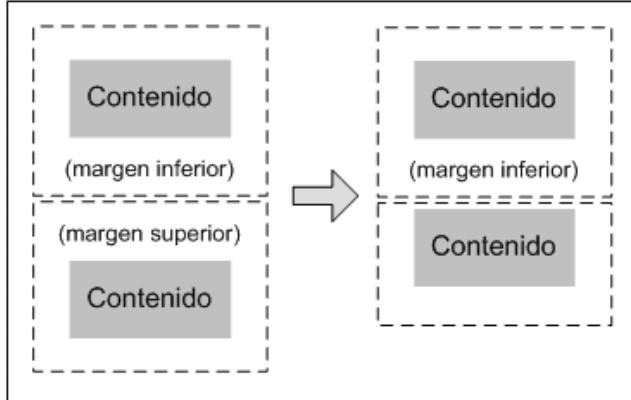
- Además de las cuatro propiedades que controlan cada uno de los márgenes del elemento, CSS define una propiedad shorthand que permite establecer los cuatro márgenes de forma simultánea: se trata de la propiedad margin.

```
selector {  
    margin: valor1 valor2 valor3 valor4;  
}
```

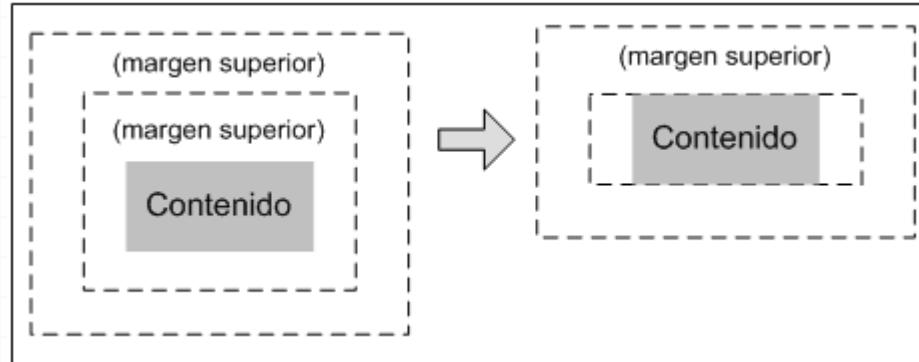
- Si observamos la sintaxis anterior vemos que la propiedad margin admite entre 1 y 4 valores
 - Si solo se indica un valor, todos los márgenes tienen ese valor.
 - Si se indican dos valores, el primero se asigna al margen superior e inferior y el segundo se asigna a los márgenes izquierdo y derecho.
 - Si se indican tres valores, el primero se asigna al margen superior, el tercero se asigna al margen inferior y el segundo valor se asigna a los márgenes izquierdo y derecho.
 - Si se indican los cuatro valores, el orden de asignación es: margen superior, margen derecho, margen inferior y margen izquierdo.
- Si se establece el valor de margin en auto, se centrará el elemento.

Comportamiento de los márgenes verticales

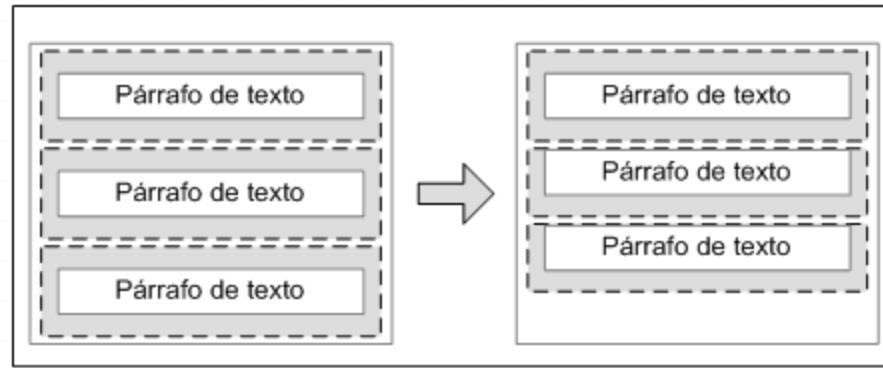
- Cuando se juntan dos o más márgenes verticales, se fusionan de forma automática y la altura del nuevo margen será igual a la altura del margen más alto de los que se han fusionado.



- Si un elemento está contenido dentro de otro elemento, sus márgenes verticales se fusionan y resultan en un nuevo margen de la misma altura que el mayor margen de los que se han fusionado

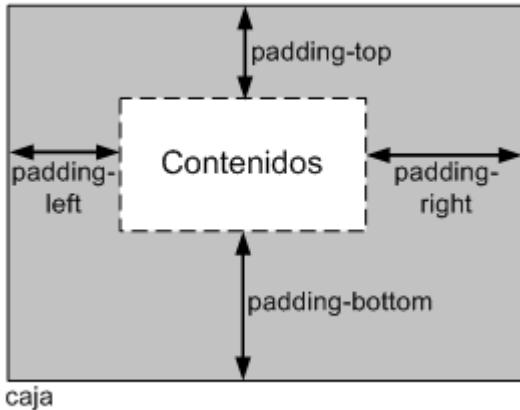


- 0 Aunque en principio puede parecer un comportamiento extraño, la razón por la que se propuso este mecanismo de fusión automática de márgenes verticales es el de dar uniformidad a las páginas web habituales. En una página con varios párrafos, si no se diera este comportamiento y se estableciera un determinado margen a todos los párrafos, el primer párrafo no mostraría un aspecto homogéneo respecto de los demás.



Padding

- Las propiedades de padding CSS definen el espacio en blanco entre el contenido del elemento y el borde del elemento



- Existen propiedades CSS para configurar el padding de cada uno de los lados de un elemento: superior, derecho, inferior e izquierdo: **padding-top, padding-right, padding-bottom, padding-left**
- Todas las propiedades padding anteriores tienen como atributo un valor que puede ser expresado en una unidad de medida absoluta o relativa, en un porcentaje o con el valor inherit.

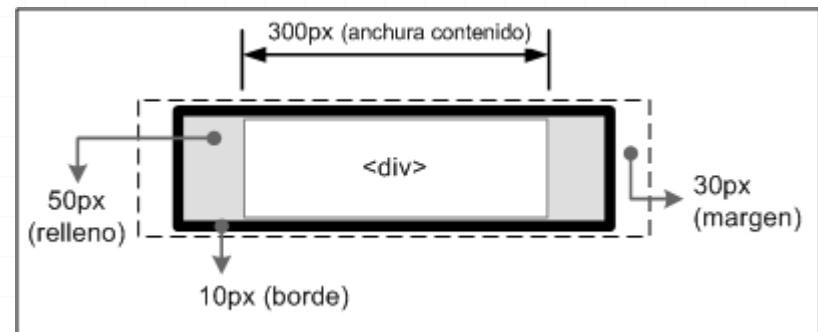
padding shorthand

- Como sucede con los márgenes, CSS también define una propiedad de tipo "shorthand" llamada padding para establecer los cuatro rellenos de un elemento de forma simultánea.
- La propiedad padding admite entre uno y cuatro valores, con el mismo significado que el de la propiedad margin.

```
h1 {  
    padding: 50px 30px 50px 80px;  
}
```

0 La anchura y altura de un elemento no solamente se calculan teniendo en cuenta sus propiedades width y height. El margen, el relleno y los bordes establecidos a un elemento determinan la anchura y altura final del elemento. En el siguiente ejemplo se muestran los estilos CSS de un elemento:

```
div {  
    width: 300px;  
    padding-left: 50px;  
    padding-right: 50px;  
    margin-left: 30px;  
    margin-right: 30px;  
    border: 10px solid black;  
}
```



Ancho total del elemento

$$30\text{px} + 10\text{px} + 50\text{px} + 300\text{px} + 50\text{px} + 10\text{px} + 30\text{px} = 480 \text{ píxel}$$

resumiendo





Manos a la obra

O Crea una página para presentar la serie que prefieras

TITULO DE LA SERIE

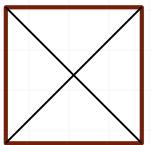
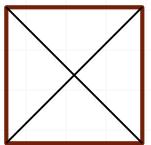
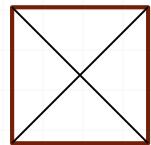
ARGUMENTO

Xxxxx x xxxxxxxx xxx xx xx
xxx xxx

PERSONAJES

X xxxxxxxx xxxxxxxx xxxx
x

IMÁGENES



BANDA SONORA

Enlace a video
youtube

VIDEO

Enlace a video
youtube

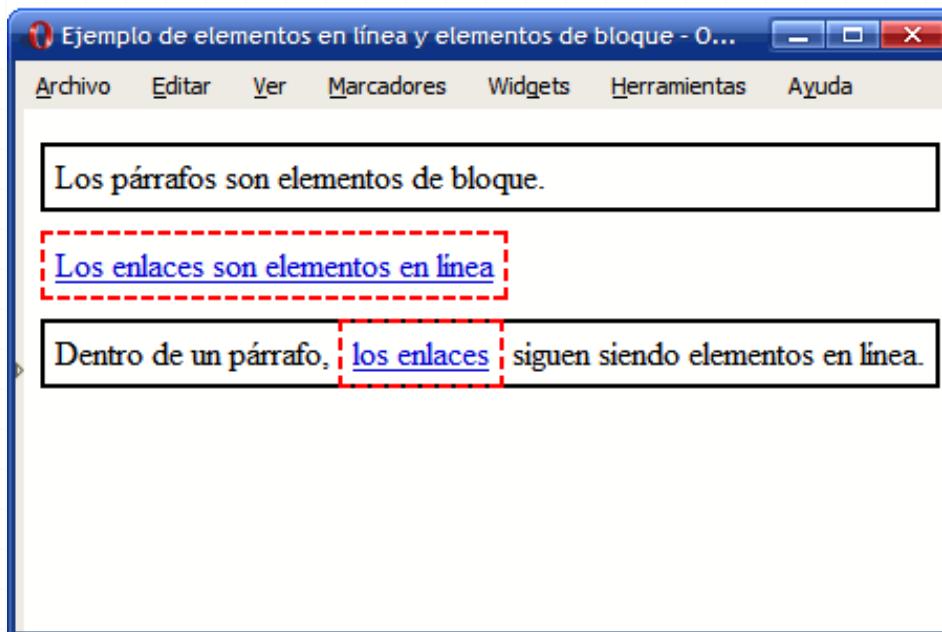
- El fondo de la página debe contener un degradado
- Los diferentes apartados deben estar formateados mediante un estilo que defina:
 - Relleno
 - Bordes (estilo, color y ancho)
 - Margen
 - padding
- Las imágenes deben formatearse mediante estilos para que todas tengan el mismo alto

Posicionamiento

- Cuando los navegadores descargan el contenido HTML y CSS de las páginas web, aplican un procesamiento muy complejo antes de mostrar las páginas en la pantalla del usuario.
- Para cumplir con el modelo de cajas presentado en el capítulo anterior, los navegadores crean una caja para representar a cada elemento de la página HTML. Los factores que se tienen en cuenta para generar cada caja son:
 - Las propiedades width y height de la caja (si están establecidas).
 - El tipo de cada elemento HTML (elemento de bloque o elemento en línea).
 - Posicionamiento de la caja (normal, relativo, absoluto, fijo o flotante).
 - Las relaciones entre elementos (dónde se encuentra cada elemento, elementos descendientes, etc.)
 - Otro tipo de información, como por ejemplo el tamaño de las imágenes y el tamaño de la ventana del navegador.

Tipos de elementos

- El estándar HTML clasifica a todos sus elementos en dos grandes grupos: **elementos en línea** y **elementos de bloque**.
 - Elementos en bloque (*block elements*). Siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea.
 - **Elementos en línea** (*inline elements*) no empiezan necesariamente en nueva línea y sólo ocupan el espacio necesario para mostrar sus contenidos.
- El tipo de un elemento influye de forma decisiva en la caja que el navegador crea para mostrarlo.



- Por sus características, los elementos de bloque no pueden insertarse dentro de elementos en línea y tan sólo pueden aparecer dentro de otros elementos de bloque.
- En cambio, un elemento en línea puede aparecer tanto dentro de un elemento de bloque como dentro de otro elemento en línea.
- Los elementos en línea definidos por HTML son: `a`, `abbr`, `acronym`, `b`, `basefont`, `bdo`, `big`, `br`, `cite`, `code`, `dfn`, `em`, `font`, `i`, `img`, `input`, `kbd`, `label`, `q`, `s`, `samp`, `select`, `small`, `span`, `strike`, `strong`, `sub`, `sup`, `textarea`, `tt`, `u`, `var`.
- Los elementos de bloque definidos por HTML son: `address`, `blockquote`, `center`, `dir`, `div`, `dl`, `fieldset`, `form`, `h1`, `h2`, `h3`, `h4`, `h5`, `h6`, `hr`, `isindex`, `menu`, `noframes`, `noscript`, `ol`, `p`, `pre`, `table`, `ul`.
- Los siguientes elementos también se considera que son de bloque: `dd`, `dt`, `frameset`, `li`, `tbody`, `td`, `tfoot`, `th`, `thead`, `tr`.
- Los siguientes elementos pueden ser en línea y de bloque según las circunstancias: `button`, `del`, `iframe`, `ins`, `map`, `object`, `script`.

Posicionamiento

Posicionamiento

- Los navegadores crean y posicinan de forma automática todas las cajas que forman cada página HTML. No obstante, CSS permite al diseñador modificar la posición en la que se muestra cada caja.
- Utilizando las propiedades que proporciona CSS para alterar la posición de las cajas es posible realizar efectos muy avanzados y diseñar estructuras de páginas que de otra forma no serían posibles.
- El estándar de CSS define cinco modelos diferentes para posicionar una caja: normal, relativo, absoluto, fijo y flotante.
- El posicionamiento de una caja se establece mediante la propiedad `position`:

```
position: static | relative | absolute | fixed | inherit
```

- La propiedad `position` no permite controlar el posicionamiento flotante, que se establece con otra propiedad llamada `float`.
- Además, la propiedad `position` solo indica cómo se posiciona la página pero no la desplaza.

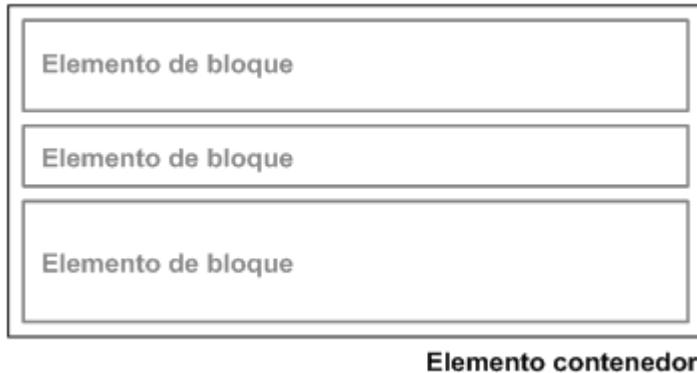
- Para desplazar la caja deberemos utilizar las propiedades
top / right / bottom / left
- En el caso del posicionamiento relativo, cada una de estas propiedades indica el desplazamiento del elemento desde la posición original de su borde superior / derecho / inferior / izquierdo.
- Si el posicionamiento es absoluto, las propiedades indican el desplazamiento del elemento respecto del borde superior / derecho / inferior / izquierdo de su primer elemento padre posicionado.
- En cualquiera de los dos casos, si el desplazamiento se indica en forma de porcentaje, se refiere al porcentaje sobre la anchura (propiedades right y left) o altura (propiedades top y bottom) del elemento.

1. Posicionamiento normal

- El posicionamiento normal o estático es el modelo que utilizan por defecto los navegadores para mostrar los elementos de las páginas.
- En este modelo, sólo se tiene en cuenta si el elemento es de bloque o en línea, sus propiedades width y height y su contenido.
- En este tipo de posicionamiento los elementos se comportarán de forma diferente según se trate de un elemento de bloque o de un elemento de línea generando lo que se denomina:
 - Contextos de formato de bloque
 - Contextos de formato en línea

Contextos de formato de bloque

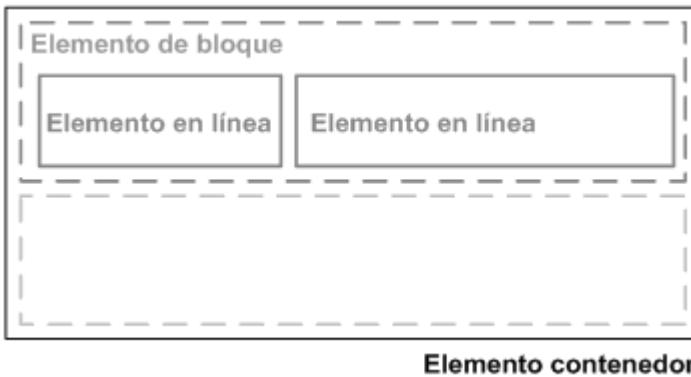
- En este tipo de contextos, las cajas se muestran una debajo de otra comenzando desde el principio del elemento contenedor. La distancia entre las cajas se controla mediante los márgenes verticales.



- Si un elemento se encuentra dentro de otro, el elemento padre se llama "elemento contenedor" y determina tanto la posición como el tamaño de todas sus cajas interiores.
 - Si un elemento no se encuentra dentro de un elemento contenedor, entonces su elemento contenedor es el elemento <body> de la página.
- Normalmente, la anchura de los elementos de bloque está limitada a la anchura de su elemento contenedor, aunque en algunos casos sus contenidos pueden desbordar el espacio disponible.

Contextos de formato en línea

- En este tipo de contextos, las cajas se muestran una detrás de otra de forma horizontal comenzando desde la posición más a la izquierda de su elemento contenedor.
- La distancia entre las cajas se controla mediante los márgenes laterales.



- Si las cajas en línea ocupan más espacio del disponible en su propia línea, el resto de cajas se muestran en las líneas inferiores. Si las cajas en línea ocupan un espacio menor que su propia línea, se puede controlar la distribución de las cajas mediante la propiedad text-align para centrarlas, alinearlas a la derecha o justificarlas.



Manos a la obra

- En un nuevo documento teclear el siguiente código seguido, sin pulsar intros.

```
<h1>Esto es el título</h1> <p>Esto es un párrafo</p> <Esto es otro párrafo>
```

```
<p>Este párrafo tiene un <a href="www.atcgijon.com">  
Enlace</a></p>
```

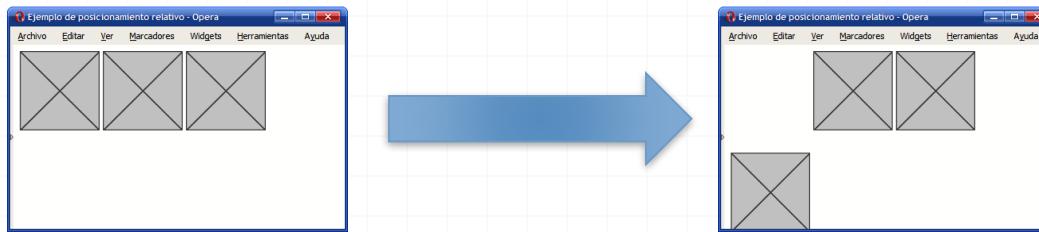
```
 
```

```
<h1>Esto es el título</h1> <p>Esto es un párrafo</p> 
```

```
<h1>Esto es el título</h1> <p>Esto es un párrafo  </p>
```

2. Posicionamiento relativo

- El posicionamiento relativo desplaza una caja respecto de su posición original establecida mediante el posicionamiento normal.
 - El desplazamiento de la caja se controla con las propiedades top, right, bottom y left.
- Supongamos que tenemos tres imágenes posicionadas de modo normal (imagen 1) y aplicando el posicionamiento relativo, desplazamos la primera de las imágenes (imagen2)



- Observamos que las otras dos imágenes no varían su posición y por tanto no ocupan el hueco dejado por la primera imagen, ya que el posicionamiento relativo no influye en el resto de elementos de la página.
- El principal problema de posicionar elementos de forma relativa es que se pueden producir solapamientos con otros elementos de la página.



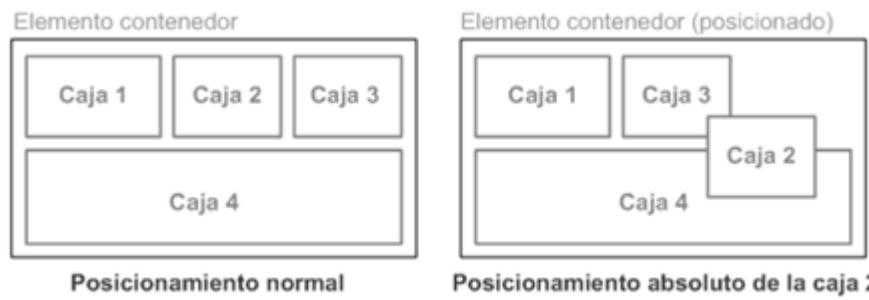
Manos a la obra

- Crear un archivo html que inserte las tres imágenes de Mafalda una al lado de la otra.
- Asignar un id a la primera imagen que llamaremos desplazada.
- Luego crear un archivo de estilos.css con el siguiente código:

```
#desplazada {  
    position: relative;  
    top: 30em;  
}
```

3. Posicionamiento absoluto

- El posicionamiento absoluto se emplea para establecer de forma exacta la posición en la que se muestra la caja de un elemento.
- La nueva posición de la caja se indica mediante las propiedades top, right, bottom y left.
- En este caso el posicionamiento del elemento depende del posicionamiento del elemento contenedor.
- Cuando una caja se posiciona de forma absoluta, el resto de elementos de la página se ven afectados y modifican su posición.



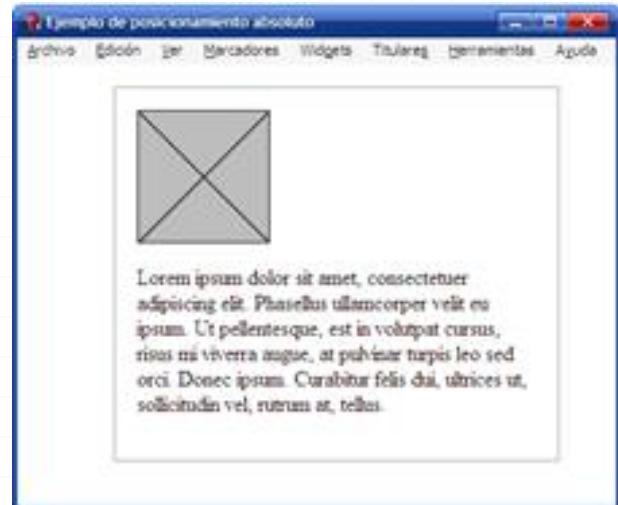
La caja 2 está posicionada de forma absoluta, lo que provoca que el resto de elementos de la página modifiquen su posición. En concreto, la caja 3 deja su lugar original y pasa a ocupar el hueco dejado por la caja 2.

0 Al igual que en el posicionamiento relativo, cuando se posiciona de forma absoluta una caja es probable que se produzcan solapamientos con otras cajas



Manos a la obra

```
div {  
    border: 2px solid #CCC;  
    padding: 1em;  
    margin: 1em 0 1em 4em;  
    width: 300px;  
}  
  
<div>  
      
    <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus ullamcorper velit eu ipsum. Ut pellentesque, est in volutpat cursus, risus mi viverra augue, at pulvinar turpis leo sed orci. Donec ipsum. Curabitur felis dui, ultrices ut, sollicitudin vel, rutrum at, tellus.</p>  
</div>
```



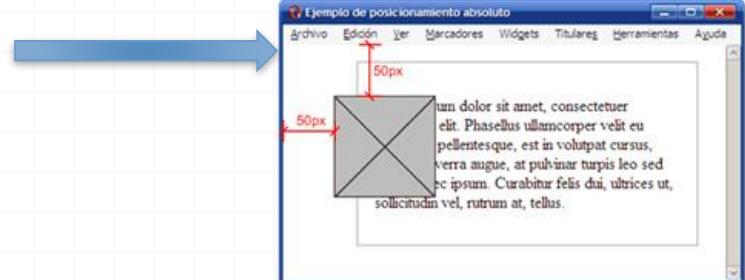
- 0 Vamos a posicionar la imagen de forma absoluta mediante la propiedad position y a definir su posición mediante las propiedades top y left.

```
div img {  
    position: absolute;  
    top: 50px;  
    left: 50px;  
}
```

Como la imagen se posiciona de forma absoluta, el resto de elementos de la página se mueven para ocupar el lugar libre dejado por la imagen. Por este motivo, el párrafo sube hasta el principio del <div> y se produce un solapamiento con la imagen posicionada que impide ver parte de los contenidos del párrafo.

El resultado visual será similar al siguiente:

La imagen posicionada de forma absoluta no toma como referencia su elemento contenedor <div>, sino la ventana del navegador



Para obtener la posición de la imagen el navegador:

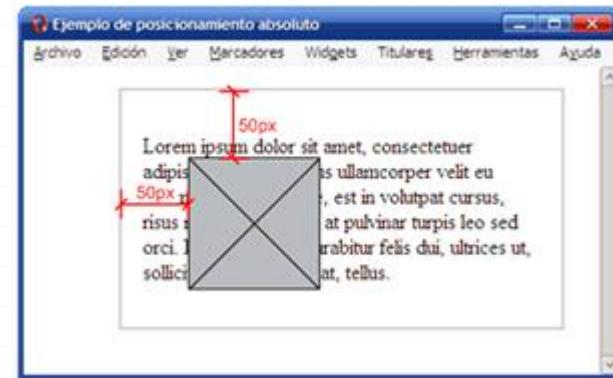
1. Obtiene la lista de elementos contenedores de la imagen: <div> y <body>.
2. Recorre la lista de elementos contenedores desde el más cercano a la imagen (el <div>) hasta terminar en el <body> buscando el primer elemento contenedor que esté posicionado (el primer elemento que tenga una position diferente a static)
3. El posicionamiento de todos los elementos contenedores es el normal o estático, ya que ni siquiera tienen establecida la propiedad position
4. Como ningún elemento contenedor está posicionado, la referencia es la ventana del navegador. Y desplaza la imagen 50px hacia la derecha y 50px hacia abajo.

0 Vamos a modificar el código CSS anterior de la siguiente forma:

```
div {  
    border: 2px solid #CCC;  
    padding: 1em;  
    margin: 1em 0 1em 4em;  
    width: 300px;  
    position: relative;  
}
```

```
div img {  
    position: absolute;  
    top: 50px;  
    left: 50px;  
}
```

En este caso, como el elemento contenedor de la imagen está posicionado, se convierte en la referencia para el posicionamiento absoluto.



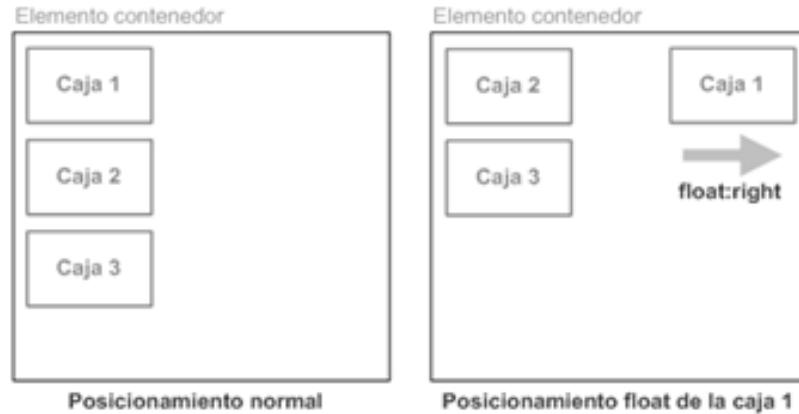
Por tanto, si se quiere posicionar un elemento de forma absoluta respecto de su elemento contenedor, es imprescindible posicionar este último. Para ello, sólo es necesario añadir la propiedad position: relative, por lo que no es obligatorio desplazar el elemento contenedor respecto de su posición original.

3. Posicionamiento fijo

- El estándar CSS considera que el posicionamiento fijo es un caso particular del posicionamiento absoluto, ya que sólo se diferencian en el comportamiento de las cajas posicionadas.
- Cuando una caja se posiciona de forma fija, la forma de obtener el origen de coordenadas para interpretar su desplazamiento es idéntica al posicionamiento absoluto. De hecho, si el usuario no mueve la página HTML en la ventana del navegador, no existe ninguna diferencia entre estos dos modelos de posicionamiento.
- La principal característica de una caja posicionada de forma fija es que su posición es inamovible dentro de la ventana del navegador. El posicionamiento fijo hace que las cajas no modifiquen su posición ni aunque el usuario suba o baje la página en la ventana de su navegador.
- Si la página se visualiza en un medio paginado (por ejemplo en una impresora) las cajas posicionadas de forma fija se repiten en todas las páginas. Esta característica puede ser útil para crear encabezados o pies de página en páginas HTML preparadas para imprimir.
- El posicionamiento fijo apenas se ha utilizado en el diseño de páginas web hasta hace poco tiempo porque el navegador Internet Explorer 6 y las versiones anteriores no lo soportan.

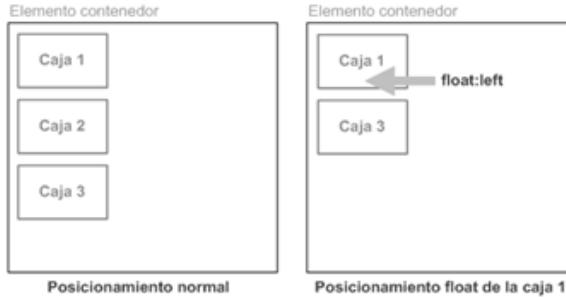
4. Posicionamiento flotante

- Es el más difícil de comprender pero también es el más utilizado.
- Cuando una caja se posiciona con el modelo de posicionamiento flotante, automáticamente se convierte en una caja flotante, lo que significa que se desplaza hasta la zona más a la izquierda o más a la derecha de la posición en la que originalmente se encontraba.
- La siguiente imagen muestra el resultado de posicionar de forma flotante hacia la derecha la caja 1:



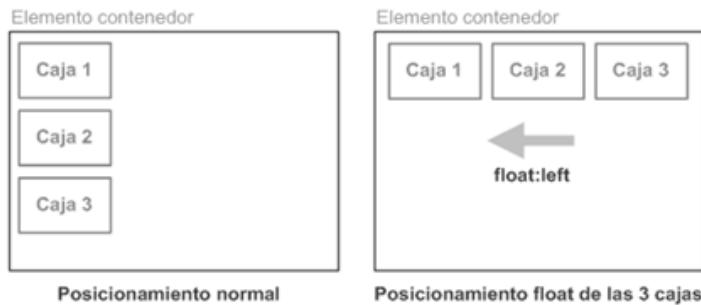
0 Cuando se posiciona una caja de forma flotante:

- 0 La caja deja de pertenecer al flujo normal de la página, lo que significa que el resto de cajas ocupan el lugar dejado por la caja flotante.
 - 0 La caja flotante se posiciona lo más a la izquierda o lo más a la derecha posible de la posición en la que se encontraba originalmente.
- 0 Supongamos que en la imagen siguiente posicionamos la caja1 de forma flotante hacia la izquierda:



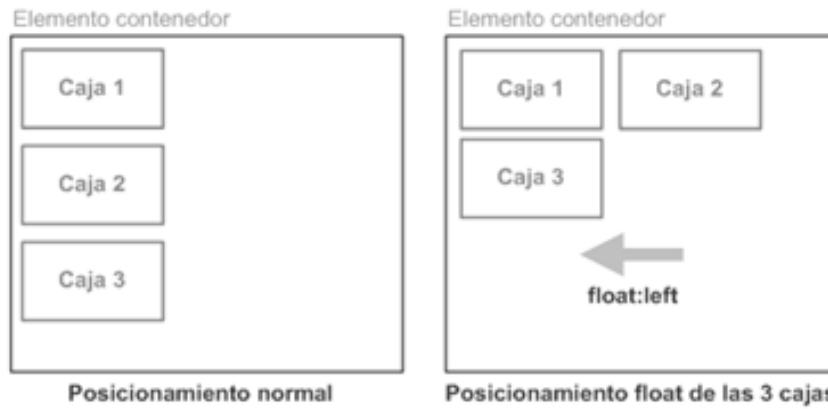
- 0 La caja 1 es de tipo flotante, por lo que desaparece del flujo normal de la página y el resto de cajas ocupan su lugar. El resultado es que la caja 2 ahora se muestra donde estaba la caja 1 y la caja 3 se muestra donde estaba la caja 2. Al ocupar la caja 2 la posición de la caja 1 pero ésta estar a la izquierda, caja1 tapa a la caja2

- Si existen otras cajas flotantes, al posicionar de forma flotante otra caja, se tiene en cuenta el sitio disponible.
- En el siguiente ejemplo se posicionan de forma flotante hacia la izquierda las tres cajas:



En el ejemplo anterior, las cajas no se superponen entre sí porque las cajas flotantes tienen en cuenta las otras cajas flotantes existentes. Como la caja 1 ya estaba posicionada lo más a la izquierda posible, la caja 2 sólo puede colocarse al lado del borde derecho de la caja 1, que es el sitio más a la izquierda posible respecto de la zona en la que se encontraba.

- Si no existe sitio en la línea actual, la caja flotante baja a la línea inferior hasta que encuentra el sitio necesario para mostrarse lo más a la izquierda o lo más a la derecha posible en esa nueva línea:



- La propiedad CSS que permite posicionar de forma flotante una caja se denomina float:
- Su sintaxis es la siguiente:

```
float: left | right | none | inherit;
```
- Si se indica un valor left, la caja se desplaza hasta el punto más a la izquierda posible en esa misma línea (si no existe sitio en esa línea, la caja baja una línea y se muestra lo más a la izquierda posible en esa nueva línea)
- El valor right tiene un funcionamiento idéntico, salvo que en este caso, la caja se desplaza hacia la derecha.
- El valor none permite anular el posicionamiento flotante de forma que el elemento se muestre en su posición original.



Los elementos que se encuentran alrededor de una caja flotante adaptan sus contenidos para que fluyan alrededor del elemento posicionado:

Uno de los principales motivos para la creación del posicionamiento float fue precisamente la posibilidad de colocar imágenes alrededor de las cuales fluye el texto.



Manos a la obra

```
div {  
    border: 2px solid #CCC;  
    padding: 1em;  
    margin: 1em 0 1em 4em;  
    width: 300px;  
}
```

```
img {  
    width: 125px;  
    float: left;  
}
```



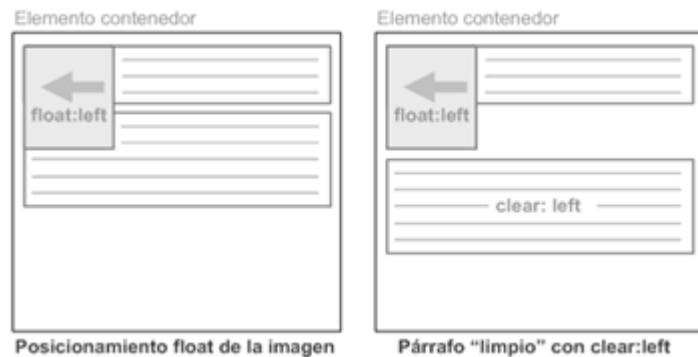
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus ullamcorper velit eu ipsum. Ut pellentesque, est in volutpat cursus, risus mi viverra augue, at pulvinar turpis leo sed orci. Donec ipsum. Curabitur felis dui, ultrices ut, sollicitudin vel, rutrum at, tellus.

propiedad clear

- CSS permite controlar la forma en la que los contenidos fluyen alrededor de los contenidos posicionados mediante float.
- De hecho, en muchas ocasiones es admisible que algunos contenidos fluyan alrededor de una imagen, pero el resto de contenidos deben mostrarse en su totalidad sin fluir alrededor de la imagen
- La propiedad `clear` permite modificar el comportamiento por defecto del posicionamiento flotante para forzar a un elemento a mostrarse debajo de cualquier caja flotante

- En este ejemplo, la regla CSS que se aplica al segundo párrafo será:

```
<p style="clear: left;">...</p>
```



- La sintaxis de clear es la siguiente:

```
clear: none | left | right | both | inherit
```

- La propiedad clear indica el lado del elemento HTML que no debe ser adyacente a ninguna caja posicionada de forma flotante. Si se indica el valor left, el elemento se desplaza de forma descendente hasta que pueda colocarse en una línea en la que no haya ninguna caja flotante en el lado izquierdo.
- Si se indica el valor right, el comportamiento es análogo, salvo que en este caso se tienen en cuenta los elementos desplazados hacia la derecha.
- El valor both despeja los lados izquierdo y derecho del elemento, ya que desplaza el elemento de forma descendente hasta que el borde superior se encuentre por debajo del borde inferior de cualquier elemento flotante hacia la izquierda o hacia la derecha.



Otras
propiedades

display

- La propiedad display es una de las propiedades CSS más importantes para controlar la disposición de los elementos
- Esta propiedad permite definir cómo se muestra un elemento
- Cada elemento HTML tiene un modo display por defecto dependiendo del tipo de elemento:
 - block
 - Los elementos block siempre empiezan en una nueva línea y ocupan todo el ancho disponible
 - Ejemplos de elementos en bloque: <div> <p> <h1>-<h6> <footer> <section> ...
 - inline.
 - Los elementos inline no empiezan en una nueva línea y sólo ocupan el ancho necesario
 - Algunos elementos en línea son: <a>
 - Podemos hacer que un elemento de tipo block se comporte como un elemento de tipo inline con la propiedad **display:inline**
 - Podemos hacer que un elemento de tipo inline se comporte como un elemento de tipo block con la propiedad **display:block**

Inline-block

- Existe un último valor que la propiedad display puede tomar y es **inline-block**
- Esta forma de visualizar los elementos crea cajas de son de bloque y de línea simultáneamente. Una caja de tipo inline-block se comporta como si fuera de bloque pero respecto a los elementos que la rodean se comporta como si fuera una caja.

```
.floating-box {  
    float: left;  
    width: 150px;  
    height: 75px;  
    margin: 10px;  
    border: 3px solid #73AD21;  
}  
  
.after-box {  
    clear: left;  
}
```



```
.floating-box {  
    display: inline-block;  
    width: 150px;  
    height: 75px;  
    margin: 10px;  
    border: 3px solid #73AD21;  
}
```

- Este comportamiento de bloque y línea a la vez hace que sea posible por ejemplo, establecer un ancho al elemento.

Vivamus fermentum, tortor id ornare ultrices, ligula ipsum tincidunt pede, et blandit sem pede suscipit

```
[display: inline-block,  
width: 25%] Quisque  
semper, magna sed  
pharetra tincidunt, quam  
urna dapibus dolor, a  
dignissim sem neque id  
purus. Etiam luctus
```

pede. Nulla cursus porta sem. Donec mollis nunc in leo. [viverra nisi.](#) Integer lobortis
accumsan felis. Cras venenatis. Morbi cursus, tellus vitae iaculis pulvinar, turpis nibh posuere nisl,
sed vehicula massa orci at dui. Morbi pede ipsum, porta quis, venenatis et, ullamcorper in, metus. Nulla
facilisi. Quisque laoreet molestie mi. Ut mollis elit eget urna.

z-index

- La propiedad z-index permite definir el orden de solapamiento.
- Un elemento con un orden de apilamiento mayor estará siempre delante de un elemento con un orden de apilamiento más bajo.
- z-index solo funciona con los elementos en con posición absolute, relative o fixed.

z-index: auto|number|initial|inherit;

- auto: Establece el orden de apilamiento igual que sus padres.
- number. Permite establecer el orden de apilamiento. Se permiten los números negativos.

overflow

- Esta propiedad permitirá especificar qué deberá ocurrir cuando el contenido rebosa sobre la caja de un contenido
 - overflow: visible|hidden|scroll|auto|initial|inherit;
 - **visible.** El desbordamiento no se recorta. Sale fuera de la caja del elemento (este es el comportamiento por defecto).
 - **hidden.** El contenido se recorta, y el contenido que no entra se vuelve invisible.
 - **scroll.** El contenido se recorta, pero se mostrará una barra de scroll para poder ver el resto del contenido.
 - **auto.** Si el contenido supera el tamaño de la caja éste se recortará y entonces aparecerá una barra de scroll para poder ver el resto del contenido.
- Las propiedades `overflow-x` y `overflow-y` funcionan similar pero refiriéndose sólo al sentido horizontal o vertical.

Links

Introducción

- Como se sabe, por defecto los navegadores muestran los enlaces no visitados de color azul y subrayados y los enlaces visitados de color morado
- Mediante CSS podremos modificar este formato predeterminado para los enlaces.
- Para dar formato a un enlace usaremos la sintaxis

```
a {  
    propiedad: valor;  
}
```

- Propiedad se refiere al atributo que podemos configurar.
- Las propiedades más típicas suelen ser: color, fondo, tipo de letra, etc.



Definir un estilo para que los enlaces se muestren en color verde.

Formato de un enlace según su estado

- CSS también permite aplicar diferentes estilos a un mismo enlace en función de su estado
 - De esta forma, es posible cambiar el aspecto de un enlace cuando por ejemplo el usuario pasa el ratón por encima o cuando el usuario pincha sobre ese enlace
- Los cuatro posibles estados de un enlace son:
 - **a:link**, aplica estilos a los enlaces que apuntan a páginas o recursos que aún no han sido visitados por el usuario.
 - **a:visited**, aplica estilos a los enlaces que apuntan a recursos que han sido visitados anteriormente por el usuario.
 - **a:hover**, aplica estilos al enlace sobre el que el usuario ha posicionado el puntero del ratón.
 - **a:active**, aplica estilos al enlace que está pinchando el usuario. Los estilos sólo se aplican desde que el usuario pincha el botón del ratón hasta que lo suelta, por lo que suelen ser unas pocas décimas de segundo.
- Puntos a tener en cuenta:
 - No puede haber espacio entre los dos puntos(:) y el estado especificado
 - Si se definen varias pseudo-clases sobre un mismo enlace, el único **orden** que asegura que todos los estilos de las pseudo-clases se aplican de forma coherente es el siguiente: **a:link, a:visited, a:hover y a:active**.



Manos a la obra

Crear un documento HTML que defina un enlace a la página web

Luego, definir los siguientes estilos para el enlace:

```
/* Formato para los enlaces sin visitar */
a:link {
    color: crimson;
}

/* Formato para los enlaces visitados */
a:visited {
    color: orange;
}

/* Formato para cuando se sitúa el ratón sobre un enlace*/
a:hover {
    color: pink;
}

/* Formato para cuando se hace clic sobre un enlace*/
a:active {
    color: blue;
}
```

Decoración personalizada

- La propiedad text-decoration permite añadir o eliminar el subrayado de los enlaces.
- Sin embargo, el aspecto del subrayado lo controla automáticamente el navegador, por lo que su color siempre es el mismo que el del texto del enlace y su anchura es proporcional al tamaño de letra.



```
a:link {  
    text-decoration: none;  
}  
a:visited {  
    text-decoration: none;  
}  
a:hover {  
    text-decoration: underline;  
}  
a:active {  
    text-decoration: underline;  
}.
```

- No obstante, utilizando la propiedad border-bottom es posible añadir cualquier tipo de subrayado para los enlaces de las páginas



```
<a class="color" href="#">Enlace un subrayado de otro color</a>

a { text-decoration: none; }
.color {border-bottom: medium solid red;}

<a class="ancho" href="#">Enlace con un subrayado muy ancho</a>

.ancho {border-bottom: thick solid;}

<a class="discontinuo" href="#">Enlace con borde punteado</a>

.discontinuo {border-bottom: thin dashed;}
```

Color de fondo

- 0 La propiedad background-color permite especificar un color de fondo para los enlaces.



```
a:link {  
    background-color: yellow;  
}  
  
a:visited {  
    background-color: cyan;  
}  
  
a:hover {  
    background-color: lightgreen;  
}  
  
a:active {  
    background-color: hotpink;  
}
```

Mostrar los enlaces como botones

- 0 Jugando con diversas propiedades, podemos hacer que nuestros enlaces se presenten como botones muy vistosos.



Código html

Menú del sitio

```
<a href="#">Inicio</a>
<a href="#">Galería</a>
<a href="#">Contacto</a>
```

Código CSS

```
a:link, a:visited {
    background-color: #f44336;
    color: white;
    padding: 14px 25px;
    text-align: center;
    text-decoration: none;
}

a:hover, a:active {
    background-color: pink;
    color: black;
}
```



Listas

Introducción

- Por defecto, los navegadores muestran los elementos de las listas con un formato predeterminado:
 - En el caso de listas no ordenadas se mostrará una viñeta formada por un pequeño círculo de color negro delante de cada elemento de la lista
 - En el caso de las listas ordenadas los elementos de la lista se muestran con la numeración decimal utilizada en la mayoría de países.
- No obstante, CSS define varias propiedades para controlar el tipo de viñeta y número que muestran las listas, además de poder controlar la posición de la propia viñeta.

list-style-type

- Esta propiedad nos permite controlar el tipo de viñeta que se muestra
- La sintaxis de este comando es:

list-style-type: valor;

- Donde valor puede tomar los siguientes:
 - **none**. permite mostrar una lista en la que sus elementos no contienen viñetas, números o letras. Se trata de un valor muy utilizado, ya que es imprescindible para los menús de navegación creados con listas.
 - El resto de valores de la propiedad list-style-type se dividen en tres tipos
 - Gráficos: **disc**, **circle** y **square**
 - Valores numéricos: **decimal**, **decimal-leading-zero**, **lower-roman**, **upper-roman**, **armenian** y **georgian**
 - Valores alfanuméricos: **lower-latin**, **lower-alpha**, **upper-latin**, **upper-alpha** y **lower-greek**

Ejemplo propiedad list-style-type

Archivo Edición Ver Marcadores Widgets Titulares Herramientas Ayuda

- list-style-type: disc
- Elemento
- Elemento

- list-style-type: circle
- Elemento
- Elemento

- list-style-type: square
- Elemento
- Elemento

- list-style-type: lower-roman
- ii. Elemento
- iii. Elemento

- list-style-type: lower-greek
- β. Elemento
- γ. Elemento

- list-style-type: upper-latin
- B. Elemento
- C. Elemento

- list-style-type: lower-latin
- b. Elemento
- c. Elemento

- list-style-type: armenian
- Բ. Elemento
- զ. Elemento

- list-style-type: upper-alpha
- B. Elemento
- C. Elemento

- list-style-type: decimal-leading-zero
- 02. Elemento
- 03. Elemento



Manos a la obra

Estilos de listas

file:///E:/IFCD0210/Ejercicio

Buscar

■ Elemento
■ Elemento
■ Elemento

i. Elemento
ii. Elemento
iii. Elemento

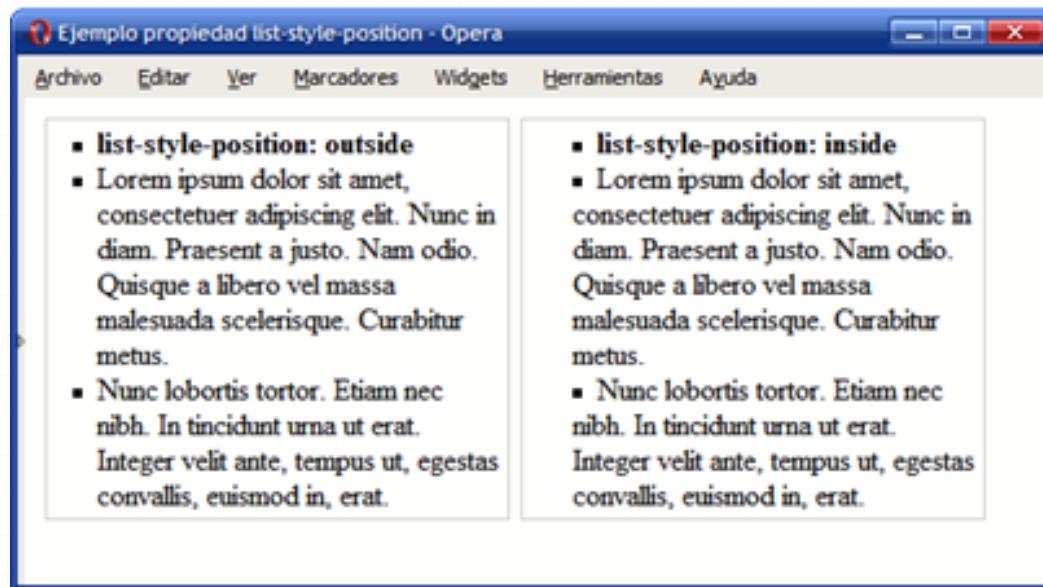
01. Elemento
02. Elemento
03. Elemento

list-style-position

- La propiedad list-style-position permite controlar la colocación de las viñetas.
- Su sintaxis es la siguiente:

list-style-position: valor;

- Donde `valor` puede tomar los valores: `inside`, `outside` o `inherit`



list-style-image

- Esta propiedad permite mostrar una imagen propia en vez de una viñeta automática
- Su sintaxis es la siguiente:

list-style-image: url("url_imagen");

- Mediante url indicaremos la ruta y nombre donde se encuentra la imagen personalizada.
- Si no se encuentra la imagen o no se puede cargar, se muestra la viñeta automática correspondiente (salvo que explícitamente se haya especificado otra con la propiedad list-style-type)

NOTA. Esta propiedad da poco juego a la hora de crear efectos adicionales, modificar tamaños, posiciones, etc. por lo que a veces en vez de usarla se coloca una imagen a modo de viñeta usando una imagen de fondo y anulando la viñeta propiamente dicha en lugar de usando esta propiedad.

Propiedad shorthand

○ Como es habitual, CSS define una propiedad de tipo "shorthand" que permite establecer todas las propiedades de una lista de forma directa. La propiedad se denomina `list-style`.

○ Sintaxis:

`list-style: list-style-type list-style-position list-style-image`

○ Ejemplo:

```
ul {  
    list-style: square inside url("circulo.png");  
}
```

○ Si no se especificara valor para una de las propiedades, se tomará el valor por defecto para esa propiedad

Formateando listas con colores

- Podemos aplicar colores a las listas para mejorar su aspecto y hacer que se vean algo más interesantes.
- Cualquier cosa añadida a la etiqueta o , afecta a toda la lista, mientras que las propiedades añadidas a la etiqueta afectará a los elementos individuales de la lista



```
ul {  
    background: #3399ff;  
    padding: 20px;  
}  
  
ul li {  
    background: #cce5ff;  
    padding: 5px;  
    margin: 5px;  
}
```

Listado de postres

- Tarta de queso
- Tarta de whisky
- Flan con nata



Manos a la obra

- Crea un sitio para la guardería pasitos que contendrá tres documentos html: Inicio, Personal y Contacto.
- Las tres páginas deberán tener la misma estructura:

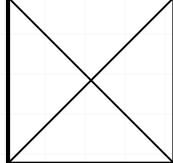
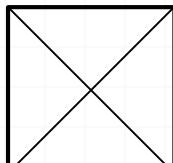




Manos a la obra

La section de Inicio deberá tener un pequeño texto de presentación de la guardería (sirven dos párrafos de Lorem Ipsum)

La section de personal deberá tener un aspecto similar al siguiente:

	Nombre
	Nombre

La section de Contacto deberá tener un aspecto similar al siguiente:

Nombre	<input type="text"/>
Email	<input type="text"/>
Déjanos tu comentario	
<input type="text"/>	