

2

CASCADING STYLE SHEETS



2.1 SELECTORES CSS

SELECTORES CSS

2.1 SELECTORES CSS

En CSS 3 se han introducido nuevos selectores mediante las especificaciones *Selectors Level 3* y *Selectors Level 4*.

En este apartado repasaremos los selectores ya existentes en CSS 2.1 y también veremos los nuevos selectores.

2.1 SELECTORES CSS

Selector descendente

```
a b { .. }
```

Selecciona todos los elementos de la página de tipo **b** que se encuentran contenidos en un elemento de tipo **a**.

2.1 SELECTORES CSS

Selector de hijos

```
a > b { .. }
```

Selecciona todos los elementos de la página de tipo **b** que son hijos inmediatos de un elemento de tipo **a**.

2.1 SELECTORES CSS

Selector adyacente

$$a + b \{ \dots \}$$

Selecciona todos los elementos de la página de tipo **b** que son directamente precedidos de un selector de tipo **a**, es decir, que ambos tienen un padre común y el elemento **b** es adyacente a continuación de **a**.

2.1 SELECTORES CSS

Selector general de elementos hermanos



```
a ~ b {..}
```

Selecciona todos los elementos de la página de tipo **b** que son hermanos de un elemento de tipo **a** el cual les precede en el código.

2.1 SELECTORES CSS

Selector de atributo

`[nombre_atrib] {...}`

Selecciona todos los elementos que tienen un atributo llamado *nombre_atrib*, independientemente de su valor.

```
/* Se muestran de color azul todos los enlaces que tengan  
un atributo "class", independientemente de su valor */  
a[class] { color: blue; }
```


2.1 SELECTORES CSS

Selector de atributo

`[nom_atrib=val] {..}`

Selecciona todos los elementos que tienen un atributo llamado *nombre_atrib* con un valor igual a *val*.

```
/* Se muestran de color azul todos los enlaces que tengan  
un atributo "class" con el valor "externo" */  
a[class="externo"] { color: blue; }
```

2.1 SELECTORES CSS

Selector de atributo

`[nom_atrib~=val] {..}`

Selecciona todos los elementos que tienen un atributo llamado *nombre_atrib* con una lista de palabras separadas por espacios en blanco en la que al menos una de ellas es exactamente igual a *val*.

```
/* Se muestran de color azul todos los enlaces que tengan  
un atributo "class" en el que al menos uno de sus valores  
sea "externo" */  
a[class~= "externo"] { color: blue; }
```

2.1 SELECTORES CSS

Selector de atributo

`[nom_atrib|=val] {..}`

Selecciona todos los elementos que tienen un atributo llamado *nombre_atrib* con una lista de palabras separadas por guiones, pero que comienza por *val*.

```
/* Selecciona todos los elementos de la página cuyo atributo  
   "lang" empiece por "es", es decir, "es", "es-ES", "es-AR", etc. */  
*[lang|=es] { color : red }
```

2.1 SELECTORES CSS



Selector de atributo

`[nom_atrib^=val] {..}`

Selecciona todos los elementos que tienen un atributo llamado *nombre_atrib* y cuyo valor comienza exactamente por la cadena *val*.

```
/* Selecciona todos los enlaces que apuntan a una dirección de correo electrónico */  
a[href^="mailto:"] { ... }
```

2.1 SELECTORES CSS



Selector de atributo

`[nom_atrib$=val] {..}`

Selecciona todos los elementos que tienen un atributo llamado *nombre_atrib* y cuyo valor termina exactamente por la cadena *val*.

```
/* Selecciona todos los enlaces que apuntan a una página HTML */  
a[href$=".html"] { ... }
```

2.1 SELECTORES CSS



Selector de atributo

`[nom_atrib*=val] {..}`

Selecciona todos los elementos que tienen un atributo llamado *nombre_atrib* y cuyo valor contiene la cadena *val*.

```
/* Selecciona todos los títulos h1 cuyo atributo title contenga la palabra "capítulo" */  
h1[title*="capítulo"] { ... }
```

2.1 SELECTORES CSS

Pseudo-clase

`a:first-child {..}`

Selecciona todos los elementos de tipo **a** que sean el primer hijo de otro elemento.

```
p em:first-child {  
  color: red;  
}
```

```
<p>Lorem <span><em>ipsum dolor</em></span> sit amet, consectetur adipiscing elit. Praesent odi  
o sem, tempor quis, <em>auctor eu</em>, tempus at, enim. Praesent nulla ante,  
<em>ultrices</em> id, porttitor ut, pulvinar quis, dui.</p>
```

2.1 SELECTORES CSS

Pseudo-clase

```
a:last-child {...}
```

Selecciona todos los elementos de tipo **a** que sean el último hijo de otro elemento.

2.1 SELECTORES CSS

Pseudo-clase

```
a:link{..}
```

Selecciona todos los enlaces que todavía no han sido visitados.

Pseudoclase

```
a:visited{..}
```

Selecciona todos los enlaces que todavía ya han sido visitados.

2.1 SELECTORES CSS

Pseudo-clase

```
a:hover {..}
```

Se activa cuando el usuario pasa el ratón o cualquier otro elemento apuntador por encima del elemento.

2.1 SELECTORES CSS

Pseudo-clase

```
a:active{..}
```

Se activa cuando el usuario activa un elemento, por ejemplo cuando hace click sobre él. El elemento solo está activo desde que pulsa el botón del ratón hasta que lo suelta.

2.1 SELECTORES CSS

Pseudo-clase

```
a:focus{..}
```

Se activa cuando el elemento mantiene el foco del navegador. Habitualmente se utiliza en los elementos `<input>` cuando el cursor está en ellos.

2.1 SELECTORES CSS

Pseudo-clase

`a:lang{..}`

Se emplea para seleccionar elementos en función de su idioma. El idioma está determinado por los atributos lang.

```
p { color: blue; }  
p:lang(es) { color: red; }
```

```
<p lang="en">Lorem ipsum dolor sit amet...</p>  
<div lang="fr">  
<p>Lorem ipsum dolor sit amet...</p>  
<p lang="es_ES">Lorem ipsum dolor sit amet...</p>  
</div>
```

2.1 SELECTORES CSS



Pseudo-clase

```
a:nth-child(n){..}
```

Selecciona el elemento indicado pero con la condición de que sea el hijo enésimo de su padre.

Admite parámetros de la forma:

- $(2n)$: hijos múltiplos de 2
- $(3n+4)$: múltiplos de 3 a partir del cuarto
- (odd) : impares
- (even) : pares

2.1 SELECTORES CSS



Pseudo-clase

```
a:nth-last-child(n){..}
```

Selecciona el elemento indicado pero con la condición de que sea el hijo enésimo de su padre pero comenzando a contar desde el final.

2.1 SELECTORES CSS



Pseudo-clase

`a:empty{..}`

Selecciona el elemento indicado pero con la condición de que no tenga ningún hijo. Esto indica también que no debe contener ningún texto.

2.1 SELECTORES CSS



Pseudo-clase

`a:nth-of-type(n){..}`

Selecciona el elemento indicado pero con la condición de que sea el enésimo elemento hermano de ese tipo.

2.1 SELECTORES CSS



Pseudo-clase

`a:nth-last-of-type(n){..}`

Selecciona el elemento indicado pero con la condición de que sea el enésimo elemento hermano de ese tipo pero comenzando a contar por el final.

2.1 SELECTORES CSS



Pseudo-clase

`:target{..}`

Cuando tenemos varios enlaces en la propia página del tipo `#primero, ...` selecciona la porción de documento que corresponde a ese fragmento.

2.1 SELECTORES CSS



Pseudo-clase

:not(a){..}

Selecciona todos los elementos que no sean del tipo **a**.

```
:not(p) { ... } /* selecciona todos los elementos de la página que no sean párrafos */  
:not(#especial) { ... } /* selecciona cualquier elemento cuyo atributo id no sea "especial" */
```

2.1 SELECTORES CSS

Pseudo-elemento



`a::first-line{..}`

Selecciona la primera línea de texto del elemento **a**.

```
div:first-line { color: red; }  
p:first-line { text-transform: uppercase; }
```

```
<div>  
  <p>Lorem ipsum dolor sit amet...</p>  
  <p>Lorem ipsum dolor sit amet...</p>  
  <p>Lorem ipsum dolor sit amet...</p>  
</div>
```

2.1 SELECTORES CSS

Pseudo-elemento



```
a::first-letter{..}
```

Selecciona la primera letra del elemento **a**. Los signos de puntuación que se encuentran antes y después de la primera letra también son afectados por este pseudo-elemento.

2.1 SELECTORES CSS

Pseudo-elemento



```
a::before{..}
```

```
a::after{..}
```

Se utilizan en combinación con la propiedad *content* para añadir contenidos antes o después del contenido original.

```
h1:before { content: "Capítulo - "; }  
p:after   { content: "."; }
```

2.1 SELECTORES CSS

Junto con el contenido se pueden especificar más propiedades CSS.

```
p.box {  
    width: 300px;  
    border: solid 1px blue;  
    padding: 20px;  
}  
  
p.box:before {  
    content: "#";  
    border: solid 1px red;  
    padding: 2px;  
    margin: 0 10px 0 0;  
}
```


2.1 SELECTORES CSS

Aunque en los ejemplos anteriores hemos insertado cadenas de texto también es posible insertar una imagen.

```
p:before {  
    content: url(image.jpg);  
}
```

2.1 SELECTORES CSS



Pseudo-elemento

`a::selection{..}`

Selecciona el texto que ha seleccionado el usuario con el ratón o el teclado.

Con este selector solo funcionan las propiedades:

- color
- background-color
- cursor
- outline
- text-decoration
- text-emphasis-color
- text-shadow

MODELO DE CAJAS

En CSS todo es una caja.

De acuerdo con este modelo, todos los elementos de una página web generan una caja rectangular alrededor llamada caja del elemento.

Hay dos tipos de cajas que determinan el comportamiento de la caja:

- Cajas de bloque
- Cajas *inline*

Cada elemento HTML es por defecto de un tipo u otro, pero también se puede cambiar mediante CSS.

Cajas de bloque:

- Crean nuevos bloques de contenido.
- Son renderizadas verticalmente siguiendo el orden en el código.
- Se expanden para ocupar todo el ancho del elemento contenedor (salvo tablas)
- Es lo que se conoce como **flujo normal** de la página

2.2 MODELO DE CAJAS

Cajas *inline*:

- No forman nuevos bloques de contenido
- Ocupan el ancho que necesiten para su contenido.
- No exceden el ancho del contenedor, si es necesario pasan a ocupar varias líneas

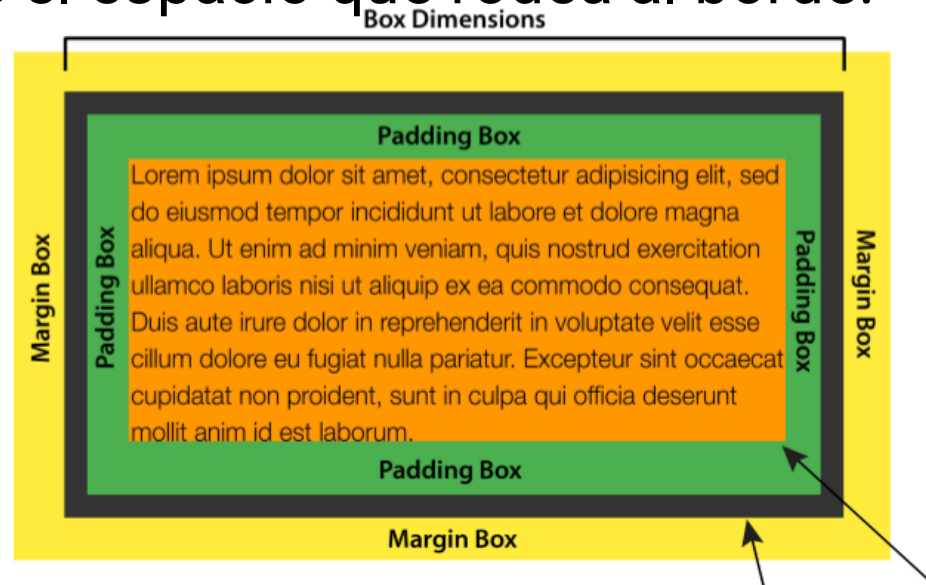
Fusce odio leo, sollicitudin vel mattis eget, iaculis sit amet turpis. Quisque porta suscipit erat. Nulla sit amet neque eleifend diam aliquam rhoncus. Donec id congue est. Aliquam sagittis euismod tristique. Nunc aliquet iaculis ex, rutrum interdum nulla ultricies at. Curabitur turpis est, feugiat sit amet risus quis, mollis elementum libero. Cras vitae aliquet felis. Nam efficitur semper arcu a blandit. Proin condimentum elit sapien, ut tempor nisl porta quis. Pellentesque vel quam efficitur, pharetra urna in, suscipit tortor.

Pellentesque vel nisl sed tellus eleifend efficitur. Sed imperdiet diam sem, nec sodales libero lobortis non. Nunc semper felis sem, id pharetra lacus pellentesque tempor. Ut vitae scelerisque mauris. Nunc interdum ultricies laoreet. Nam a dui luctus, egestas enim sit amet, blandit quam. Phasellus lorem libero, sollicitudin id hendrerit a, faucibus eget nisi.

2.2 MODELO DE CAJAS

En cada caja podemos distinguir varios elementos que son los que determinan su tamaño final:

- Área de contenido
- Padding o relleno
- Borde o parte que perfila el relleno
- Margen es el espacio que rodea al borde.



2.2 MODELO DE CAJAS

1. El relleno, los bordes y márgenes son opcionales. Si ajustas a cero se eliminarán de la caja
2. Cualquier color o imagen que apliques de fondo al elemento se extenderán al relleno.
3. Los bordes se generan con propiedades que especifican su estilo, grosor y color.
4. Los márgenes son transparentes, es decir, el color del elemento padre se verá a través de ellos).
5. Cuando definas el largo de un elemento estás definiendo del área del contenido (los largos de relleno, borde y margen se sumarían).
6. Se puede cambiar el estilo del lado de arriba, derecho, bajo, izquierdo de forma independiente.

2.2 MODELO DE CAJAS –CONTENIDO

Propiedad **width**

sintaxis CSS

```
width: auto|value|initial|inherit;
```

Valores de propiedad

Value	Description
auto	Default value. The browser calculates the width
<i>length</i>	Defines the width in px, cm, etc.
%	Defines the width in percent of the containing block
initial	Sets this property to its default value. Read about <i>initial</i>
inherit	Inherits this property from its parent element. Read about <i>inherit</i>

2.2 MODELO DE CAJAS-CONTENIDO

Propiedad **height**

sintaxis CSS

```
height: auto|length|initial|inherit;
```

Valores de propiedad

Value	Description
auto	The browser calculates the height. This is default
<i>length</i>	Defines the height in px, cm, etc.
%	Defines the height in percent of the containing block
initial	Sets this property to its default value. Read about <i>initial</i>
inherit	Inherits this property from its parent element. Read about <i>inherit</i>

2.2 MODELO DE CAJAS

Estas propiedades no se aplican a todos los elementos, solamente a:

- Todos los elementos de bloque
- Imágenes
- Filas y grupos de fila de tabla (`width`)
- Columnas y grupos de columna de tabla (`height`)

2.2 MODELO DE CAJAS- CONTENIDO

Relacionadas con estas propiedades tenemos:

sintaxis CSS

```
min-width: length|initial|inherit;
```

```
min-height: length|initial|inherit;
```

Value	Description
<i>length</i>	Default value is 0. Defines the minimum width in px, cm, etc.
%	Defines the minimum width in percent of the containing block
initial	Sets this property to its default value. Read about <i>initial</i>
inherit	Inherits this property from its parent element. Read about <i>inherit</i>

2.2 MODELO DE CAJAS

Propiedad **margin**

sintaxis CSS

```
margin: length|auto|initial|inherit;
```

Valores de propiedad

Value	Description
<i>length</i>	Specifies a margin in px, pt, cm, etc. Default value is 0
%	Specifies a margin in percent of the width of the containing element
auto	The browser calculates a margin
initial	Sets this property to its default value. Read about <i>initial</i>
inherit	Inherits this property from its parent element. Read about <i>inherit</i>

2.2 MODELO DE CAJAS

Admite de 1 a 4 valores:

- **1 valor:**
 - Se aplica a los 4 márgenes
- **2 valores:**
 - Primero: margen superior e inferior
 - Segundo: margen izquierdo y derecho
- **3 valores:**
 - Primero: margen superior
 - Segundo: márgenes izquierdo y derecho
 - Tercero: margen inferior
- **4 valores:**
 - El orden es: superior, derecho, inferior e izquierdo

2.2 MODELO DE CAJAS

También se pueden indicar individualmente:

```
margin-top: length|auto|initial|inherit;
```

```
margin-right: length|auto|initial|inherit;
```

```
margin-bottom: length|auto|initial|inherit;
```

```
margin-left: length|auto|initial|inherit;
```

2.2 MODELO DE CAJAS

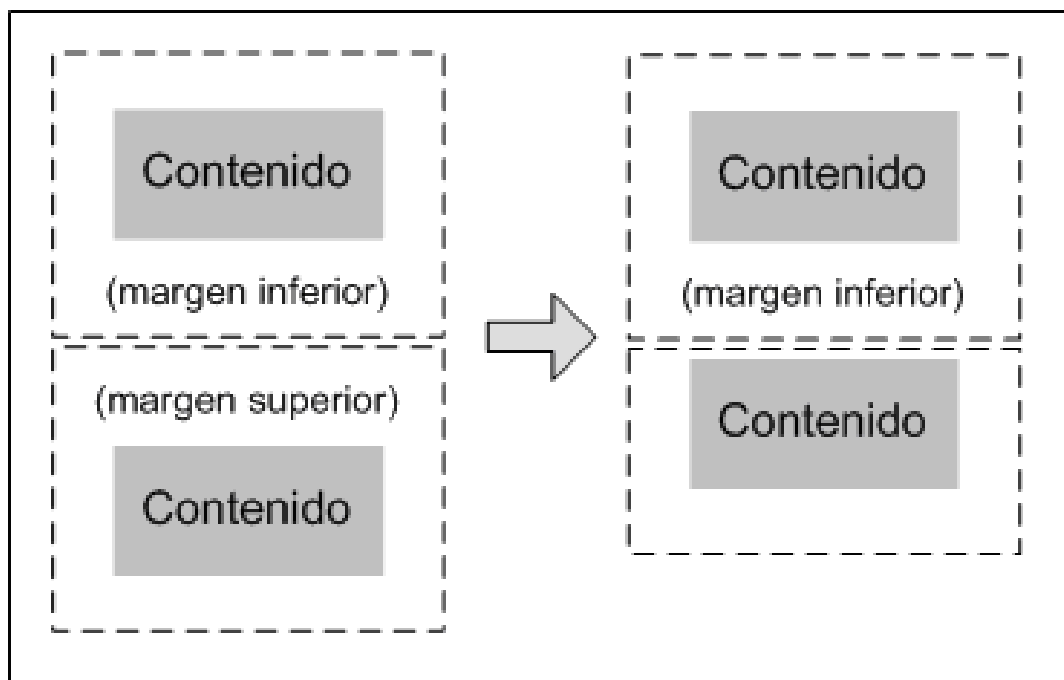
Los **márgenes superior e inferior** solo se pueden aplicar a elementos de bloque.

Los **márgenes izquierdo y derecho** se pueden aplicar a cualquier elemento.

2.2 MODELO DE CAJAS

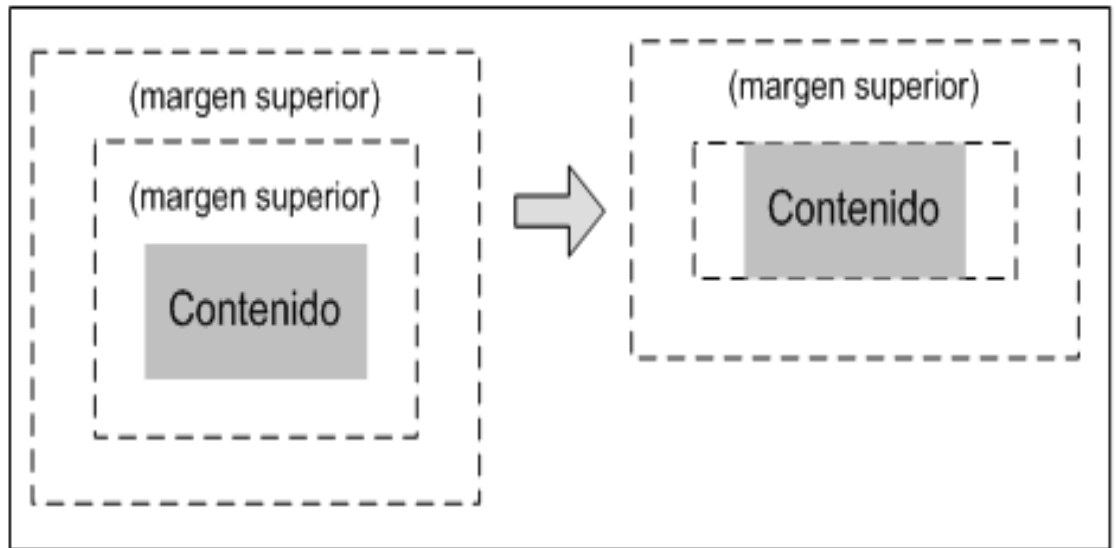
El comportamiento de los márgenes verticales es más complejo:

Cuando **se juntan dos o más márgenes verticales** se fusionan y la altura del nuevo margen será la del mayor de ambos márgenes.



2.2 MODELO DE CAJAS

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



2.2 MODELO DE CAJAS

Propiedad **padding**

sintaxis CSS

```
padding: length|initial|inherit;
```

Valores de propiedad

Value	Description
<i>length</i>	Specifies the padding in px, pt, cm, etc. Default value is 0
%	Specifies the padding in percent of the width of the containing element
initial	Sets this property to its default value. Read about <i>initial</i>
inherit	Inherits this property from its parent element. Read about <i>inherit</i>

2.2 MODELO DE CAJAS

Al igual que en el caso de `margin` admite entre 1 y 4 valores.

E igualmente se puede indicar por separado cada uno de los lados.

```
padding-left: length|initial|inherit;
```

```
padding-bottom: length|initial|inherit;
```

```
padding-top: length|initial|inherit;
```

```
padding-right: length|initial|inherit;
```

2.2 MODELO DE CAJAS

Propiedad **border**

sintaxis CSS

```
border: border-width border-style border-color|initial|inherit;
```

Valores de propiedad

Value	Description
<u>border-width</u>	Specifies the width of the border. Default value is "medium"
<u>border-style</u>	Specifies the style of the border. Default value is "none"
<u>border-color</u>	Specifies the color of the border. Default value is the color of the element
initial	Sets this property to its default value. <u>Read about initial</u>
inherit	Inherits this property from its parent element. <u>Read about inherit</u>

2.2 MODELO DE CAJAS

Propiedades	border-top-width, border-right-width, border-bottom-width, border-left-width
Valores	(unidad de medida thin medium thick) inherit

Propiedades	border-top-color, border-right-color, border-bottom-color, border-left-color
Valores	color transparent inherit

Propiedades	border-top-style, border-right-style, border-bottom-style, border-left-style
Valores	none hidden dotted dashed solid double groove ridge inset outset inherit

2.2 MODELO DE CAJAS

```
<!DOCTYPE html>
<html>
<head>
    <style>
    div {
        background-color: lightblue;
    }
    </style>
</head>
<body>

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

</body>
</html>
```

Añadir:

- Anchura 200 px,
- padding 25px,
- border: 25px solid navy
- Margin: 25px

2.2 MODELO DE CAJAS

Propiedad **background**

```
background: bg-color bg-image position/bg-size bg-repeat bg-origin bg-clip bg-attachment initial|inherit;
```

Property Values

Value	Description	CSS
<i>background-color</i>	Specifies the background color to be used	1
<i>background-image</i>	Specifies ONE or MORE background images to be used	1
<i>background-position</i>	Specifies the position of the background images	1
<i>background-size</i>	Specifies the size of the background images	3
<i>background-repeat</i>	Specifies how to repeat the background images	1
<i>background-origin</i>	Specifies the positioning area of the background images	3
<i>background-clip</i>	Specifies the painting area of the background images	3
<i>background-attachment</i>	Specifies whether the background images are fixed or scrolls with the rest of the page	1
initial	Sets this property to its default value. Read about <i>initial</i>	3
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	2

2.2 MODELO DE CAJAS

Propiedad	background-image
Valores	url none inherit

- La imagen se muestra delante del color.
- Las imágenes se indican a través de su url

```
body { background-image: url("imagenes/fondo.png") }
```

2.2 MODELO DE CAJAS

```
<!DOCTYPE html>
<html>
<head>
    <style>
    div {
        background-color: lightblue;
    }
    </style>
</head>
<body>

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

</body>
</html>
```

Añadir:

- Añadir color de fondo azul claro
- Imagen la del fichero árbol.png

2.2 MODELO DE CAJAS

- Si la imagen es más grande que el contenedor se recorta.
- Si es más pequeña se repite tanto horizontal como verticalmente.



2.2 MODELO DE CAJAS

Propiedad	background-repeat
Valores	repeat repeat-x repeat-y no-repeat inherit

- Con esta propiedad controlamos en qué eje se repite y en cuál no.

2.2 MODELO DE CAJAS

Propiedad	background-position
Valores	<code>((porcentaje unidad de medida left center right) (porcentaje unidad de medida top center bottom)?) ((left center right) (top center bottom)) inherit</code>

- Si no repetimos la imagen podemos escoger la posición de la misma dentro del contenedor.
- Valor por defecto: 0% 0%
- El valor de porcentaje es poco intuitivo: si se indica (x%, y%) el navegador coloca el punto (x%, y%) de la imagen) en el punto (x% y%) del elemento.

2.2 MODELO DE CAJAS

- Las palabras claves son:
 - top: 0%
 - left: 0%
 - center: 50%
 - bottom: 100%
 - right: 100%
- Se pueden combinar palabras clave con unidades.
- Si solo se utilizan palabras clave el orden es indiferente.

2.2 MODELO DE CAJAS

¿Cuál es el ancho real de un elemento?

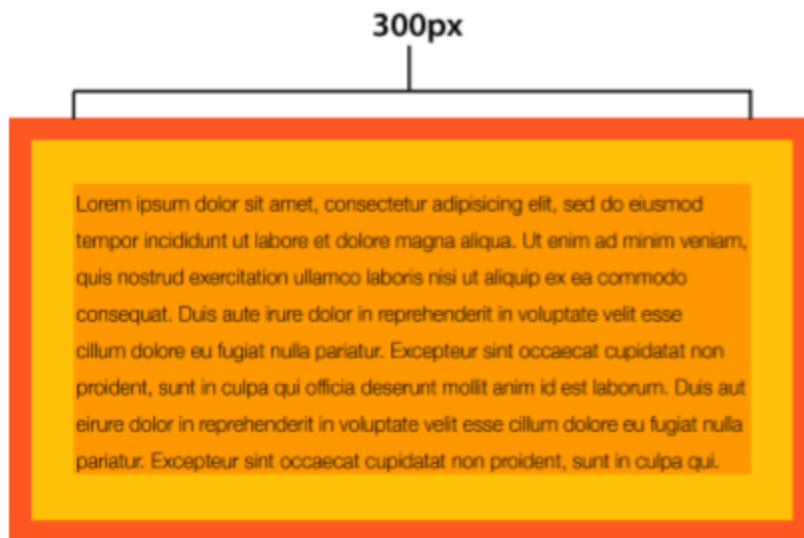
El ancho que ocupa una caja está determinado por:

- el ancho del contenido
- más el tamaño del padding
- más el tamaño de la línea de borde

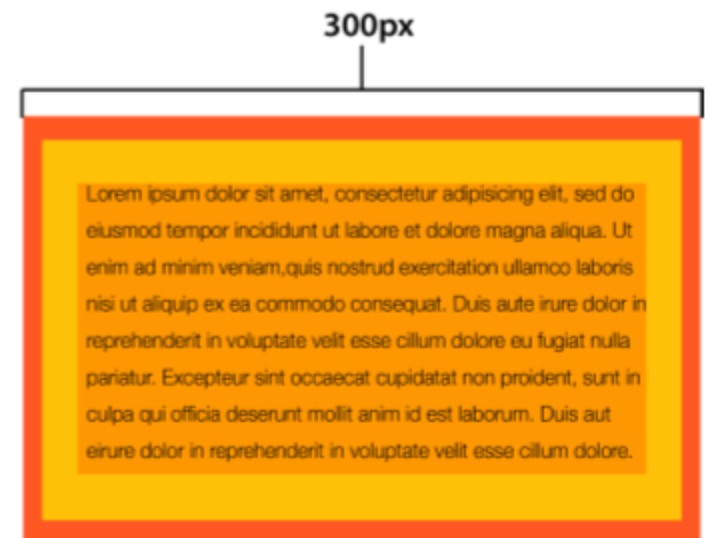
Sin embargo, como no podía ser de otra manera, en Internet Explorer 5.5 y anteriores **tanto el padding como el borde se incluyen dentro del ancho indicado por width.**

2.2 MODELO DE CAJAS

En las siguientes imágenes se ha aplicado la propiedad `{width: 300px}`



Box Dimensions in CSS 2.1 Model



Box Dimensions in IE 5.5

2.2 MODELO DE CAJAS

El modelo de *IExplorer* realmente es más cómodo, sobre todo cuando queremos crear diseño fluidos ya que simplifica los cálculos.

Por ello ahora se puede escoger cuál de los dos modelos queremos utilizar, y se hace mediante la propiedad **box-sizing**.

Propiedad **box-sizing**

sintaxis CSS

```
box-sizing: content-box|border-box|initial|inherit;
```

Valores de propiedad

Value	Description
content-box	Default. The width and height properties (and min/max properties) includes only the content. Border, padding, or margin are not included
border-box	The width and height properties (and min/max properties) includes content, padding and border, but not the margin
initial	Sets this property to its default value. Read about <i>initial</i>
inherit	Inherits this property from its parent element. Read about <i>inherit</i>

POSICIONAMIENTO

2.2 POSICIONAMIENTO

Los factores que se tienen en cuenta para generar cada caja son:

- Las propiedades width y height.
- El tipo de cada elemento (bloque o inline)
- Posicionamiento de la caja (normal, relativo, absoluto, fijo y flotante).
- Las relaciones entre elementos (dónde se encuentra cada elemento, elementos descendientes, etc.)
- Otro tipo de información como el tamaño de las imágenes y el tamaño de la ventana del navegador

2.2 POSICIONAMIENTO

El estándar de CSS define cinco modelos para posicionar una caja:

- Posicionamiento **normal o estático**
- Posicionamiento **relativo**
- Posicionamiento **absoluto**
- Posicionamiento **fijo**
- Posicionamiento **flotante**

Se establece con la propiedad `position`

Propiedad	position
Valores	static relative absolute fixed <u>inherit</u>

2.2 POSICIONAMIENTO

Se establece con la propiedad `position`

Propiedad	position
Valores	<code>static</code> <code>relative</code> <code>absolute</code> <code>fixed</code> <code>inherit</code>

También se utilizan estas propiedades para determinar la posición (salvo para posicionamiento estático)

Propiedades	top, right, bottom, left
Valores	<code>unidad de medida</code> <code>porcentaje</code> <code>auto</code> <code>inherit</code>

Posicionamiento normal

- Es el modelo utilizado por defecto.
- Los elementos de bloque se colocan uno debajo del otro.
- Si un elemento se encuentra dentro de otro, el padre se llama **elemento contenedor**
- El contenedor determina el tamaño y posición de todas las cajas hijo.
- Los elementos en línea se sitúan uno detrás de otro de forma horizontal.

Posicionamiento relativo

- Este modelo desplaza la caja respecto a la ubicación que tendría en el **posicionamiento estático**.
- El desplazamiento se controla con `top`, `bottom`, `right` y `left`.
- Se admiten valores negativos.
- El desplazamiento de una caja **no afecta a las cajas adyacentes**.
- Este tipo de posicionamiento produce solapamiento de objetos.

```
<!DOCTYPE html>
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
      img.desplazada {
        position: relative;
        top: 10em;
      }

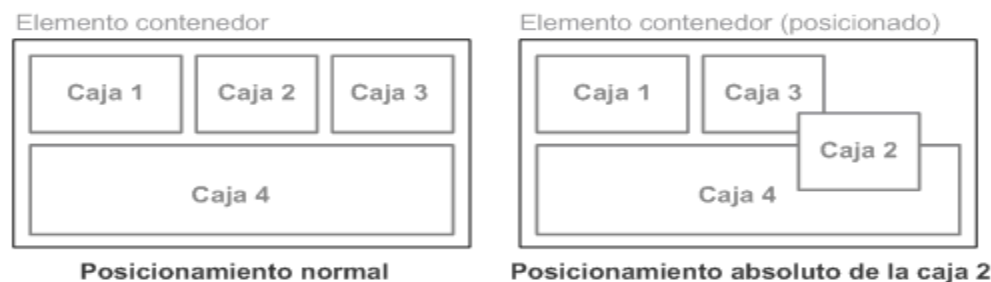
    </style>

  </head>
  <body>
    
    
    
    <p>Esto es una prueba</p>
    <p>Esto es una segunda prueba</p>
  </body>
</html>
```

2.2 POSICIONAMIENTO ABSOLUTO

Posicionamiento absoluto

- Establece la posición exacta en que se muestra la caja de un elemento.
- El elemento se extrae del flujo normal de la página, por lo que los elementos detrás de él se desplazarán.



- El cálculo del origen de las coordenadas es más complejo que en el caso anterior.

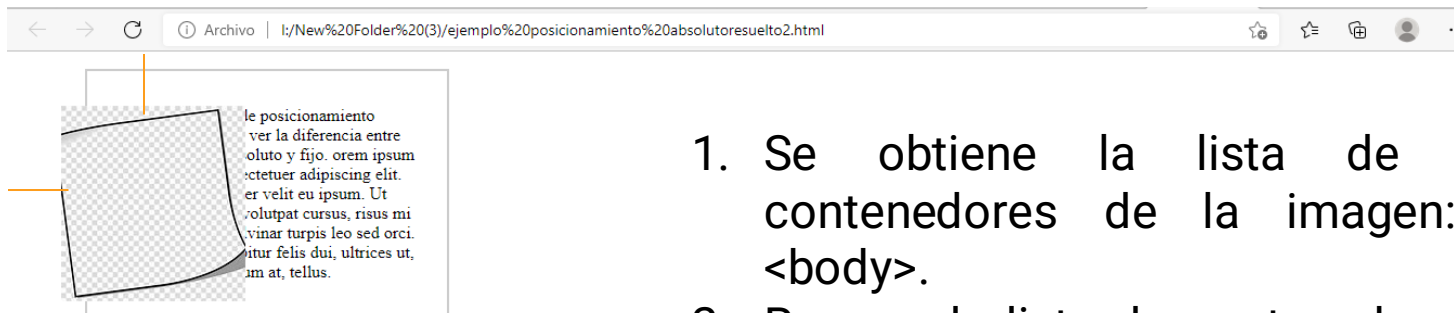
2.2 POSICIONAMIENTO ABSOLUTO

- Los pasos para calcularlas son:
 - Se buscan todos los elementos contenedores hasta llegar al elemento `<body>`
 - Se recorren todos estos elementos empezando por el más cercano a la caja y llegando hasta el body
 - El primer elemento contenedor que esté posicionado de cualquier forma diferente a `static` se convierte en la referencia que determina la posición de la caja.
 - Si ningún elemento contenedor está posicionado, la referencia es la ventana del navegador.

2.2 POSICIONAMIENTO ABSOLUTO

- Piensa que esto implica que el posicionamiento por defecto es respecto a la ventana del navegador.
- Si quieres que la posición sea respecto a un contenedor solamente tienes que aplicarle a este la propiedad `position: relative`, sin indicar ninguna distancia.

EJERCICIO



1. Se obtiene la lista de elementos contenedores de la imagen: `<div>` y `<body>`.
2. Recorre la lista de contenedores desde el más cercano a la imagen `<div>` hasta terminar en el `<body>` buscando el primer elemento contenedor posicionado.
3. En este caso el posicionamiento es estático. Entonces se toma la referencia de la ventana del navegador.
4. A partir de ahí se desplaza de forma descendente y a la derecha

Posicionamiento fijo

- La forma de calcular el origen de coordenadas es igual que en el posicionamiento absoluto.
- Solo varía su comportamiento cuando se hace scroll ya que la posición del elemento es inamovible dentro de la ventana del navegador.

2.2 POSICIONAMIENTO FLOTANTE

Posicionamiento flotante

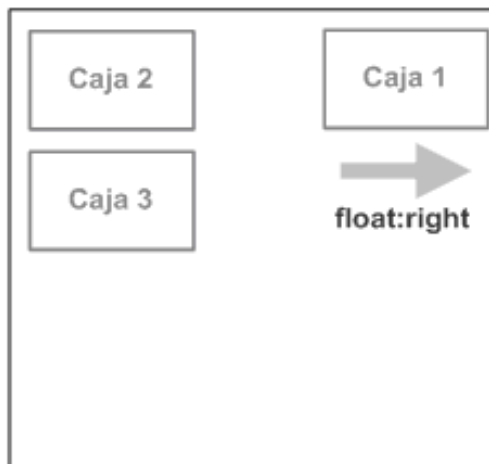
- Cuando una caja tiene posicionamiento flotante se desplaza hasta la zona más a la izquierda o derecha de donde se encuentra.

Elemento contenedor



Posicionamiento normal

Elemento contenedor

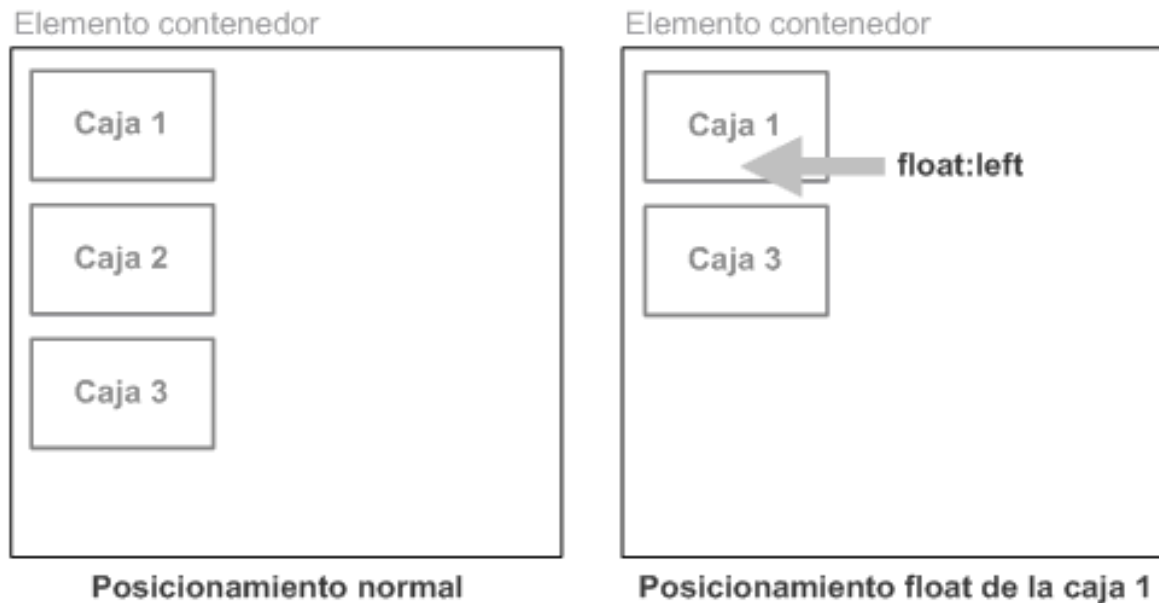


Posicionamiento float de la caja 1

Desplazar
Caja1 a la derecha

2.2 POSICIONAMIENTO FLOTANTE

- Cuando una caja es flotante deja de pertenecer al flujo normal de la página, por lo que el resto de cajas pasarán a ocupar su lugar.



Si se vuelve a desplazar la caja 1 a la izquierda la caja 2 será tapada por la caja 1

2.2 POSICIONAMIENTO FLOTANTE

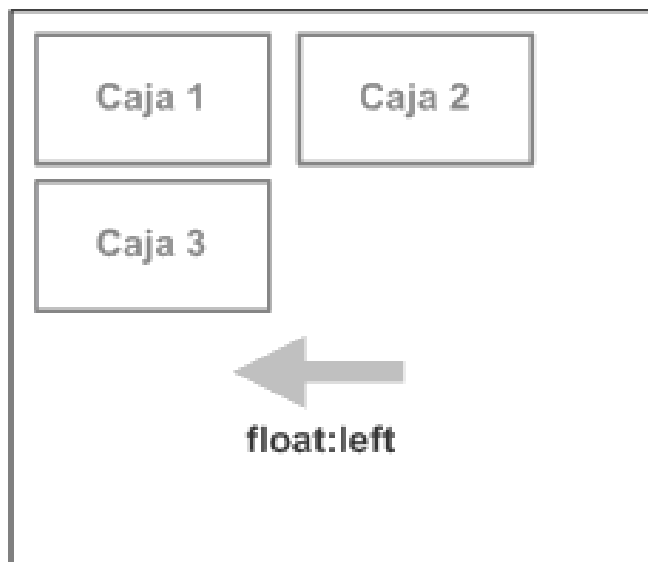
- Si no hay sitio a continuación baja a la línea inferior hasta encontrar sitio.

Elemento contenedor



Posicionamiento normal

Elemento contenedor



Posicionamiento float de las 3 cajas

2.2 POSICIONAMIENTO FLOTANTE

- Hay que tener en cuenta que las cajas flotantes influyen en la disposición del resto de cajas.
- Los **elementos en línea** hacen sitio a las cajas flotantes adaptando su anchura al espacio libre dejado por la caja desplazada.
- Los **elementos en bloque** no les hacen sitio, pero adaptan sus contenidos para que no se solapen con las cajas flotantes.

2.2 POSICIONAMIENTO FLOTANTE

La propiedad que indica que un elemento es flotante es:

Propiedad	float
Valores	left right none inherit

2.2 POSICIONAMIENTO FLOTANTE

Cuando tenemos un elemento flotante el contenido de la página fluye alrededor de él.

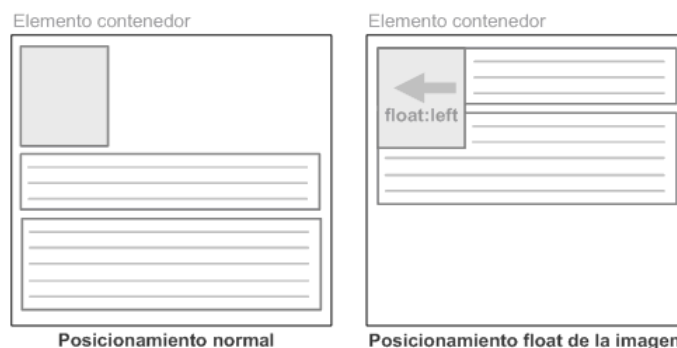


Figura 5.17 Elementos que fluyen alrededor de un elemento posicionado mediante float

Si queremos evitar ese comportamiento tenemos a nuestra disposición la propiedad **clear**.

Propiedad	clear
Valores	none left right both <u>inherit</u>

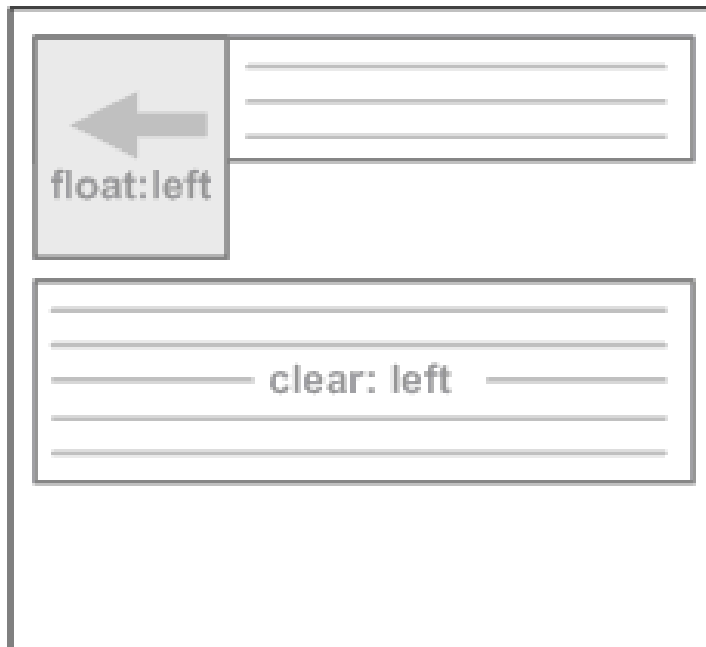
2.2 POSICIONAMIENTO FLOTANTE

Elemento contenedor



Posicionamiento float de la imagen

Elemento contenedor



Párrafo “limpio” con clear:left

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

2.2 VISIBILIDAD

Muy relacionado con el posicionamiento de cajas en CSS tenemos las propiedades que influyen en la visibilidad de las cajas.

Estas propiedades son:

- `display`
- `visibility`
- `overflow`
- `z-index`

Propiedades `display` y `visibility`

- Ambas sirven para ocultar elementos de la página.
- Se suelen utilizar junto con JavaScript para crear efectos dinámicos

2.2 PROPIEDADES DISPLAY Y VISIBILITY

Propiedad	display
Valores	<code>inline</code> <code>block</code> <code>none</code> <code>list-item</code> <code>run-in</code> <code>inline-block</code> <code>table</code> <code>inline-table</code> <code>table-row-group</code> <code>table-header-group</code> <code>table-footer-group</code> <code>table-row</code> <code>table-column-group</code> <code>table-column</code> <code>table-cell</code> <code>table-caption</code> inherit

- Esta propiedad determina la forma en que se visualiza el elemento.
- Los valores `inline` y `block` sirven para indicar que el elemento se muestre como *en línea* o de *bloque* independientemente de su tipo.
- El valor `inline-block` hace que el elemento se muestre como *en línea* pero permite modificar su altura (*height*) y anchura (*width*)

2.2 PROPIEDADES DISPLAY Y VISIBILITY

- El valor `none` hace que el elemento desaparezca de la página, es decir, que no se muestre.
- El espacio que ocuparía el elemento pasa a ser ocupado por otros elementos.
- Es decir, es como si el elemento no existiera.

2.2 PROPIEDADES DISPLAY Y VISIBILITY

Propiedad	visibility
Valores	visible hidden collapse inherit

- **visible**: es el comportamiento por defecto.
- **hidden**: el elemento no se muestra. Se reserva el espacio donde iría el elemento, es decir, dejaría un hueco en blanco.
- **collapse**: solo se puede utilizar en filas, grupos de filas, columnas y grupos de columnas de tablas. Hace que no se muestren y otros elementos ocupan su lugar.

2.2 PROPIEDAD OVERFLOW

Propiedad	overflow
Valores	visible hidden scroll auto inherit

Esta propiedad determina el comportamiento de la caja cuando se ha modificado su alto o ancho y el contenido ya no cabe en ella.

- **visible**: el contenido no se corta y se muestra fuera de la caja. Es el comportamiento por defecto.
- **hidden**: el contenido sobrante no se muestra.
- **scroll**: se muestra el contenido que cabe en la caja y aparecen unas barras de scroll para acceder al resto del contenido

2.2 PROPIEDAD OVERFLOW

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed at ante sit amet tellus fringilla auctor eget in mi. Fusce venenatis dictum dolor, vitae egestas erat fringilla a. Maecenas quam est, varius ac efficitur eu, vehicula auctor mauris. Nullam at facilisis justo, id maximus lorem. Donec metus sem, condimentum eget vulputate at, ornare quis justo. In et metus vitae urna tristique tincidunt eu nec felis. Phasellus sit amet hendrerit ipsum, id sodales sem. Duis in bibendum urna, nec accumsan erat. Duis quis porttitor erat. Donec eu arcu leo. Vivamus vel metus est. Phasellus laoreet velit at erat fermentum auctor. Fusce elit mauris, sagittis at nibh congue, mattis auctor sem. Integer eu magna ac nibh viverra malesuada vel eu dui. Morbi suscipit porta dignissim. Nam interdum, quam quis interdum faucibus, odio tortor dapibus sapien, sit amet pellentesque risus odio nec augue. Quisque eu malesuada urna. Praesent in enim purus. Nunc in dolor tempor turpis ultrices euismod. Donec a mauris in magna sodales auctor. Nulla hendrerit porta arcu, at efficitur turpis lacinia non. Morbi sollicitudin, odio ornare porta sodales, ex ipsum accumsan mi, nec tincidunt est augue sit amet tortor. Praesent est nunc, pretium id urna id, mollis lobortis tortor. Aliquam lectus urna, aliquet sit amet ex sit amet, maximus eleifend eros. Curabitur eu enim pellentesque tortor blandit vestibulum. Aenean a quam mattis, venenatis orci et, malesuada massa. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam non ornare lectus. Nullam ultrices sit amet libero non rutrum. Suspendisse gravida consequat turpis ut tincidunt. Aenean hendrerit, dolor et pretium varius, turpis urna ultricies orci, vitae pulvinar nibh tortor eu magna. Donec sit amet tincidunt metus. Curabitur mollis leo et felis tristique, ut sollicitudin ante viverra. Morbi tristique orci mauris, in convallis nibh ornare ut. Aenean egestas nibh vel sagittis luctus. Donec risus nibh, maximus sit amet gravida eget, efficitur in urna. Nam vel quam quam. Morbi est risus, suscipit nec libero vel, accumsan finibus sapien. Donec dignissim porta metus. Sed luctus sem nisl, eu gravida felis fermentum at. Integer laoreet, tortor id condimentum interdum, neque nisl porttitor lacus, at dictum ex tortor sed nunc. Praesent venenatis sagittis mollis. Nam sagittis efficitur arcu, sed vestibulum justo lacinia non. Vestibulum vitae gravida arcu. Integer dui nec euismod diam quis fringilla tristique. Ornare id rutrum urna, non

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed at ante sit amet tellus fringilla auctor eget in mi. Fusce venenatis dictum dolor, vitae egestas erat fringilla a. Maecenas quam est, varius ac efficitur eu, vehicula auctor mauris. Nullam at facilisis justo, id maximus lorem. Donec metus sem, condimentum eget vulputate at, ornare quis justo. In et metus vitae urna tristique tincidunt eu nec felis. Phasellus sit amet hendrerit ipsum, id sodales sem. Duis in bibendum urna, nec accumsan erat. Duis quis porttitor erat. Donec eu arcu leo. Vivamus vel metus est. Phasellus laoreet velit at erat fermentum auctor. Fusce elit mauris, sagittis at nibh congue, mattis auctor sem. Integer eu magna ac nibh viverra malesuada vel eu dui. Morbi suscipit porta dignissim. Nam interdum, quam quis interdum faucibus, odio tortor dapibus sapien, sit amet pellentesque risus odio nec augue. Quisque eu malesuada urna. Praesent in enim purus. Nunc in dolor tempor turpis ultrices euismod. Donec a mauris in magna sodales auctor. Nulla hendrerit porta arcu, at efficitur turpis lacinia non. Morbi sollicitudin, odio ornare porta sodales, ex ipsum accumsan mi, nec tincidunt est augue sit amet tortor. Praesent est nunc, pretium id urna id,

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed at ante sit amet tellus fringilla auctor eget in mi. Fusce venenatis dictum dolor, vitae egestas erat fringilla a. Maecenas quam est, varius ac efficitur eu, vehicula auctor mauris. Nullam at facilisis justo, id maximus lorem. Donec metus sem, condimentum eget vulputate at, ornare quis justo. In et metus vitae urna tristique tincidunt eu nec felis. Phasellus sit amet hendrerit ipsum, id sodales sem. Duis in bibendum urna, nec accumsan erat. Duis quis porttitor erat. Donec eu arcu leo. Vivamus vel metus est. Phasellus laoreet velit at erat fermentum auctor. Fusce elit mauris, sagittis at nibh congue, mattis auctor sem. Integer eu magna ac nibh viverra malesuada vel eu dui. Morbi suscipit porta dignissim. Nam interdum, quam quis interdum faucibus, odio tortor dapibus sapien, sit amet pellentesque risus odio nec augue. Quisque eu malesuada urna. Praesent in enim purus. Nunc in dolor tempor turpis ultrices euismod. Donec a mauris in magna sodales auctor. Nulla hendrerit porta arcu, at efficitur turpis lacinia non. Morbi sollicitudin, odio ornare porta

2.2 PROPIEDAD Z-INDEX

Propiedad	z-index
Valores	auto numero inherit

Esta propiedad nos permite seleccionar el orden en que se solapan los elementos de una página web. Nos permite controlar la posición tridimensional de las cajas posicionadas.

El elemento `html` crea lo que se denomina ***root stacking context***.

Algunos elementos CSS lanzan lo que se denomina *local stacking context*.

Dentro de un contexto los elementos se apilan en el orden en que se encuentran en el código, de abajo a arriba.

2.2 PROPIEDAD Z-INDEX

El orden en que se apilan es:

- Contextos locales con un valor negativo en z-index
- Elementos no posicionados
- Contextos locales con valor z-index 0 ó auto
- Contextos locales con valores positivos.

Si dos elementos tienen el mismo valor se apilan según su orden en el código.

Cuanto más alto sea el valor numérico, más cerca del usuario se muestra la caja. Un elemento con z-index: 10 se muestra por encima de los elementos con z-index: 8 y z-index: 9