



# TEMA 05

## CSS – Cascade Style Sheets: Parte II

**CSS**



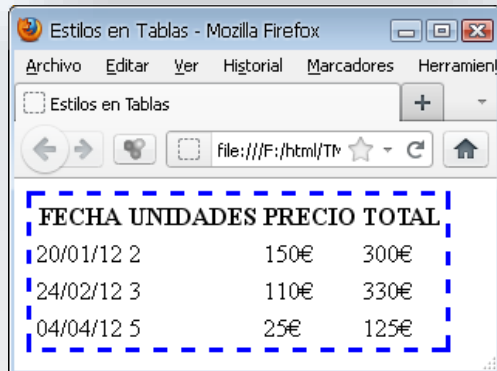
# TEMA 05

## Tablas

# Estilos en las Tablas

- Las tablas están definidas por diversos elementos: **`<table>`**, **`<tr>`**, **`<td>`**, **`<th>`**, **`<colgroup>`**, **`<caption>`**, **`<thead>`**, **`<tbody>`** y **`<tfoot>`**
- Los estilos pueden ser aplicados a toda una tabla o a todos los elementos que incluyan a otros elementos.
- Ejemplo:

```
<style>
table {border:dashed blue;}
</style>
```



Estilos en Tablas - Mozilla Firefox

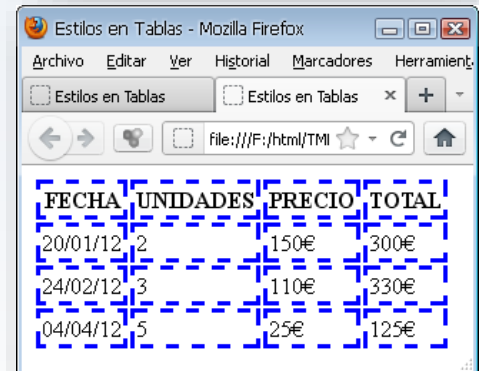
Archivo Editar Ver Historial Marcadores Herramientas

Estilos en Tablas

file:///F:/html/TV

FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	2	150€	300€
24/02/12	3	110€	330€
04/04/12	5	25€	125€

```
<style>
table * {border:dashed blue;}
</style>
```



Estilos en Tablas - Mozilla Firefox

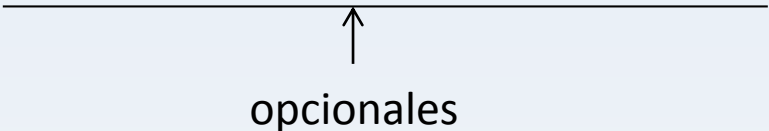
Archivo Editar Ver Historial Marcadores Herramientas

Estilos en Tablas

file:///F:/html/TMI

FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	2	150€	300€
24/02/12	3	110€	330€
04/04/12	5	25€	125€

# Bordes

- Para establecer las propiedades de los bordes de un elemento existen varias propiedades, incluidas varias propiedades resumen:
  - *border*-*borde\_específico*-*propiedad*  
  
opcionales
  - Por ejemplo se pueden especificar todas las propiedades del borde con ***border*** o todas las propiedades del borde derecho con ***border-right***.
  - También se pueden especificar individualmente como por ejemplo con ***border-right-color***, que establece el color del borde derecho.

# Bordes

- Propiedades comunes a cualquier borde:
  - **color**: color del borde.
  - **style**: estilo del borde .
  - **width**: ancho del borde.
- Ejemplo:

```
<style type="text/css">
table {
    border-color:#39C;
    border-width:3px;
    border-style: solid;
}
table * {
    border-color:#39C;
    border-width:2px;
    border-style: solid;
}
</style>
```



A screenshot of a web browser window titled 'Opera'. The address bar shows 'localhost/'. The page content displays a table with the title 'Estilos en Tablas'. The table has four columns: 'FECHA', 'UNIDADES', 'PRECIO', and 'TOTAL'. It contains three data rows. The table and its cells are styled with blue borders, demonstrating the CSS rules provided in the example.

FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	2	150€	300€
24/02/12	3	110€	330€
04/04/12	5	25€	125€

# Bordes

- Ejemplo de cómo utilizar una propiedad resumen como ***style***, donde los cuatro bordes se establecen siguiendo el movimiento de las agujas del reloj:

```
<style type="text/css">
table {
  border-color:#39C;
  border-width:3px;
  border-style: ridge double inset dotted;
}
table * {
  border-color:#39C;
  border-width:2px;
  border-style: solid;
}
</style>
```

En la dispositiva 25 hay un ejemplo similar con la propiedad padding



The screenshot shows a web browser window titled 'Opera' with a tab 'Estilos en Tablas'. The address bar shows 'localhost/'. The main content area displays a table with a blue border. The table has four columns: 'FECHA', 'UNIDADES', 'PRECIO', and 'TOTAL'. The data rows are: '20/01/12' with 2 units at 150€ (total 300€), '24/02/12' with 3 units at 110€ (total 330€), and '04/04/12' with 5 units at 25€ (total 125€).

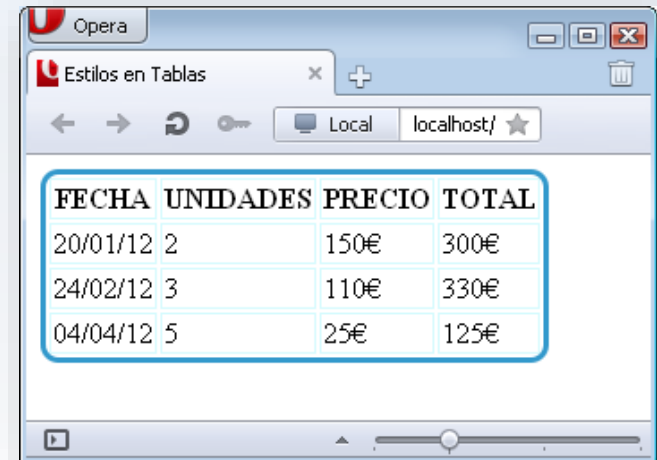
FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	2	150€	300€
24/02/12	3	110€	330€
04/04/12	5	25€	125€

# Bordes

- Para redondear los bordes existen cuatro propiedades nuevas en CSS3:
  - ***border-top-left-radius***
  - ***border-top-right-radius***
  - ***border-bottom-left-radius***
  - ***border-bottom-right-radius***

Existen más propiedades para los bordes en CSS3, pero las soportan pocos navegadores

```
<style type="text/css">
table {
  border-color:#39C;
  border-width:3px;
  border-style: solid;
  border-top-left-radius: 10;
  border-top-right-radius: 10;
  border-bottom-left-radius: 10;
  border-bottom-right-radius: 10;
}
table * {
  border-color: #D6FBFE;
  border-width:1px;
  border-style: solid;
}
</style>
```



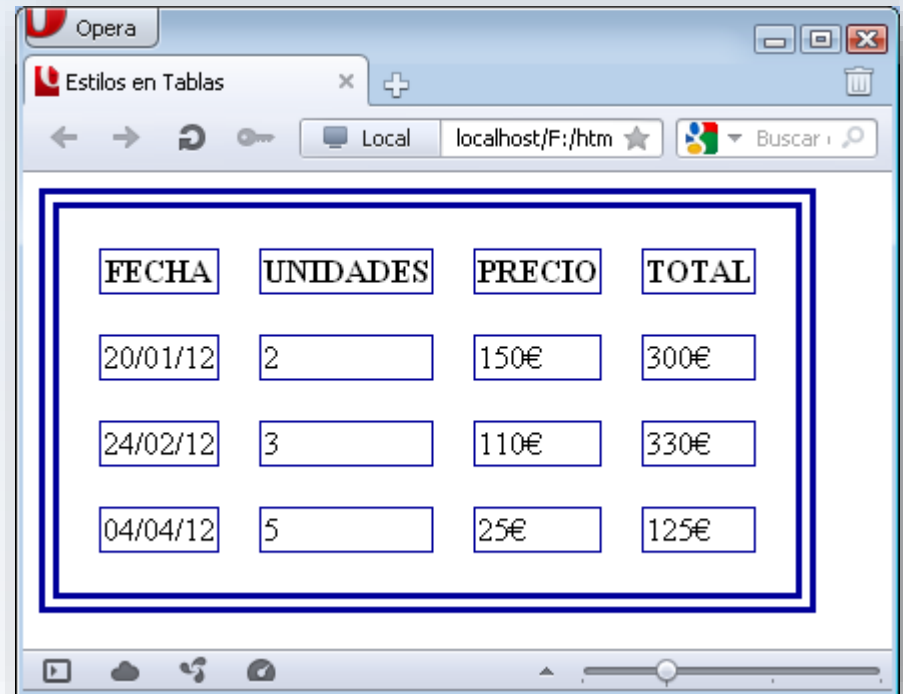
The screenshot shows a web browser window with a table titled 'Estilos en Tablas'. The table has four columns: FECHA, UNIDADES, PRECIO, and TOTAL. The table is styled with a blue border and rounded corners, as defined in the CSS code on the left. The table contains three rows of data.

FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	2	150€	300€
24/02/12	3	110€	330€
04/04/12	5	25€	125€

# Espaciado de los Bordes

- Se puede establecer un espacio para los bordes de la tabla con la propiedad ***border-spacing***.
- Ejemplo:

```
<style type="text/css">
table {
  border-width: 10px;
  border-style: double;
  border-color: #009;
  border-spacing: 20px 20px;
}
table * {
  border-width: 1px;
  border-style: solid;
  border-color: #009;
}
</style>
```

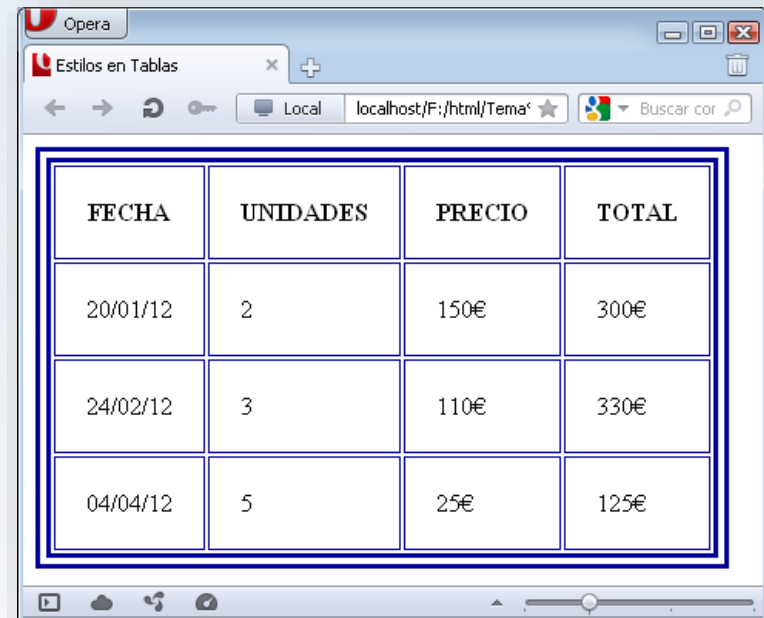




# Espaciado del Contenido

- La propiedad ***padding*** permite especificar el espaciado entre el contenido de una celda y los bordes de la misma.
- Ejemplo:

```
<style type="text/css">
table {
  border-width: 10px;
  border-style: double;
  border-color: #009;
}
table * {
  border-width: 1px;
  border-style: solid;
  border-color: #009;
  padding: 20px;
}
</style>
```

A screenshot of a web browser window (Opera) displaying a table. The table has four columns: FECHA, UNIDADES, PRECIO, and TOTAL. It contains three data rows. The table is styled with a double blue border for the outer frame and a solid blue border for the inner cells. Each cell has a significant amount of padding, making the text appear centered and well-separated from the borders. The browser's address bar shows 'localhost/F:/html/Tema5' and the title is 'Estilos en Tablas'.

FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	2	150€	300€
24/02/12	3	110€	330€
04/04/12	5	25€	125€

# Bordes de Celdas

- Por defecto, todas las celdas de una tabla tienen sus cuatro bordes. Para fusionar los bordes de una celda con las respectivas celdas adyacentes, se establece el valor ***collapse*** en la propiedad ***border-collapse***.
- Ejemplo:

```
<style type="text/css">
table {
  border-width: 3px;
  border-style: solid;
  border-color: #009;
  border-collapse: collapse;
}
table * {
  border-width: 2px;
  border-style: solid;
  border-color: #009;
}
</style>
```

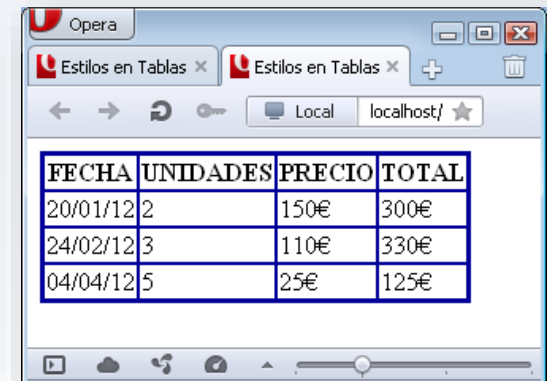


Opera

Estilos en Tablas x Estilos en Tablas x +

Local localhost/

FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	2	150€	300€
24/02/12	3	110€	330€
04/04/12	5	25€	125€



Opera

Estilos en Tablas x Estilos en Tablas x +

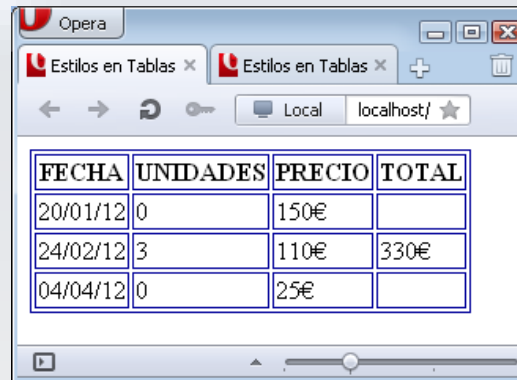
Local localhost/

FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	2	150€	300€
24/02/12	3	110€	330€
04/04/12	5	25€	125€

# Celdas Vacías

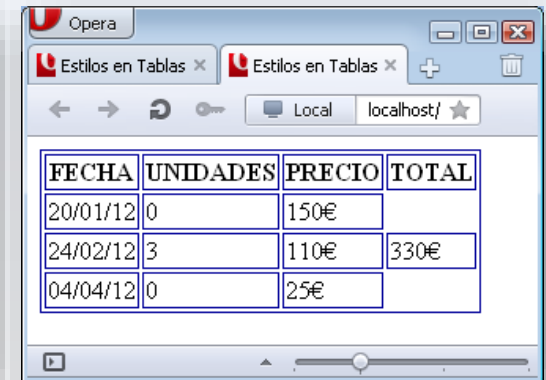
- Mediante la propiedad ***empty-cells*** se puede seleccionar si las celdas vacías se muestran o no.
  - Los valores posibles son ***show*** y ***hide***.
- Ejemplo:

```
<style type="text/css">
table {
  border-width: 1px;
  border-style: solid;
  border-color: #009;
  border-collapse: separate;
  empty-cells: hide;
}
table * {
  border-width: 1px;
  border-style: solid;
  border-color: #009;
}
</style>
```



A screenshot of the Opera browser window displaying a table. The table has four columns: FECHA, UNIDADES, PRECIO, and TOTAL. The data rows are: 20/01/12 with 0 units and 150€ price; 24/02/12 with 3 units and 110€ price (total 330€); and 04/04/12 with 0 units and 25€ price. The empty cells in the first and fourth columns of the first and third rows are not visible, demonstrating the effect of the 'empty-cells: hide;' CSS property.

FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	0	150€	
24/02/12	3	110€	330€
04/04/12	0	25€	



A screenshot of the Opera browser window displaying the same table as the previous one, but with the 'empty-cells: show;' CSS property. In this version, the empty cells in the first and fourth columns of the first and third rows are visible, showing the default behavior of the browser.

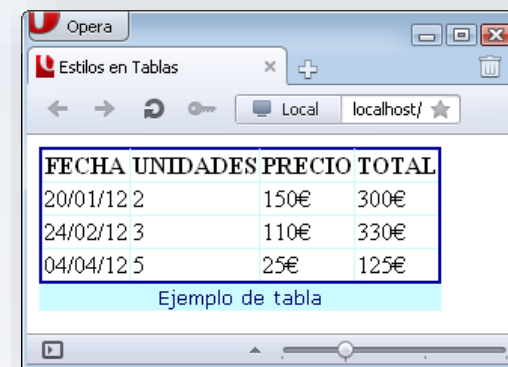
FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	0	150€	
24/02/12	3	110€	330€
04/04/12	0	25€	

# Título de Tabla

- El título de una tabla se establece mediante el elemento **<caption>**. Dado que es un elemento independiente se le pueden aplicar diversos estilos salvo su posicionamiento. Este se realiza mediante la propiedad **caption-side**, que permite los valores **top** y **bottom**.

```
<style type="text/css">
table {
  border-width: 2px;
  border-style: solid;
  border-color: #009;
  border-collapse: collapse;
  caption-side: bottom;
}
table * {
  border-width: 1px;
  border-style: solid;
  border-color: #CEFBFA;
}
caption {
  font-family: Verdana, Geneva, sans-serif;
  font-size: 0.8em;
  color: #006;
  background: #CCFBFF;
  caption {
    font-family: Verdana, Geneva
    font-size: 0.8em;
    color: #006;
    background: #CCFBFF;
  }
</style>

</head>
<body>
<div class="borde">
<table class="sample">
  <caption>Ejemplo de tabla</c
  <tr>
    <th>FECHA</th>
    <th>UNIDADES</th>
    <th>PRECIO</th>
    <th>TOTAL</th>
```



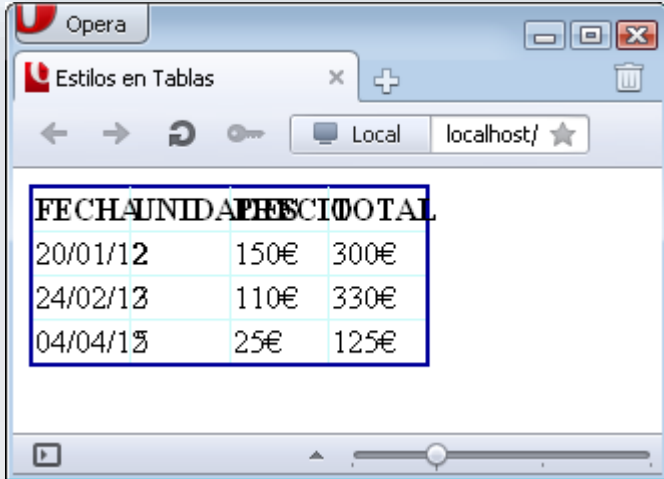
FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	2	150€	300€
24/02/12	3	110€	330€
04/04/12	5	25€	125€

Ejemplo de tabla

# Ajuste de Celdas

- Las tablas pueden adaptarse al contenido o respetar el tamaño establecido. Para ello se puede utilizar la propiedad ***table-layout***, que puede tomar los valores ***auto*** y ***fixed***.
- Ejemplo:

```
<style type="text/css">
table {
  border-width: 2px;
  border-style: solid;
  border-color: #009;
  border-collapse: collapse;
  caption-side: bottom;
  width: 200px;
  table-layout: fixed;
}
table * {
  border-width: 1px;
  border-style: solid;
  border-color: #CEFBFA;
}
</style>
```



FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	150€	300€	
24/02/12	110€	330€	
04/04/12	25€	125€	

# Celdas Pares e Impares

- Para modificar el estilo de todas las celdas pares y/o impares, se puede utilizar el selector ***:nth-child(n)***. Este selector permite aplicar un estilo al enésimo hijo de un elemento.
- Por ejemplo, ***p:nth-child(2)*** aplicaría un estilo al segundo párrafo.
- En el caso de tablas, el valor numérico se puede sustituir por ***odd*** o ***even***, para las filas impares o pares respectivamente.
- Por ejemplo, ***tr:nth-child(odd)***, modificaría todos los ***tr*** impares.

# Ejemplo de Tabla

- Ejemplo de diseño de tabla:



The screenshot shows the Opera web browser window. The address bar displays 'localhost/F:/t'. The page content features a table with the following data:

FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	2	150€	300€
24/02/12	3	110€	330€
04/04/12	5	25€	125€

# Ejemplo de Tabla

```
<style type="text/css">
table {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 12px;
  width: 400px;
  border-radius: 6px 6px 6px 6px;
  border-style: solid;
  border-width: 2px;
  border-color: #584521;
}
table th {
  padding: 5px;
  font-size: 16px;
  background-color: #83aec0;
  background-image: url(fondo01.png);
  background-repeat: repeat-x;
  color: #FFFFFF;
  border-right-width: 1px;
  border-bottom-width: 1px;
  border-right-style: solid;
  border-bottom-style: solid;
  border-right-color: #808000;
  border-bottom-color: #808000;
  font-family: "Trebuchet MS", Arial;
  text-transform: uppercase;
  text-align: center;
}
table td {
  font-size: 12px;
  font-weight: bold;
  background-color: #EED3AE;
  color: #34484E;
  font-family: "Trebuchet MS", Arial;
  text-align: right;
}
</style>
```

FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	2	150€	300€
24/02/12	3	110€	330€
04/04/12	5	25€	125€



# Ejemplo de Tabla

```
<table>
  <tr>
    <th>FECHA</th>
    <th>UNIDADES</th>
    <th>PRECIO</th>
    <th>TOTAL</th>
  </tr>
  <tr>
    <td>20/01/12</td>
    <td>2</td>
    <td>150€</td>
    <td>300€</td>
  </tr>
  <tr>
    <td>24/02/12</td>
    <td>3</td>
    <td>110€</td>
    <td>330€</td>
  </tr>
  <tr>
    <td>04/04/12</td>
    <td>5</td>
    <td>25€</td>
    <td>125€</td>
  </tr>
</table>
```

FECHA	UNIDADES	PRECIO	TOTAL
20/01/12	2	150€	300€
24/02/12	3	110€	330€
04/04/12	5	25€	125€

**CSS**



# TEMA 05

**Listas**

# Listas

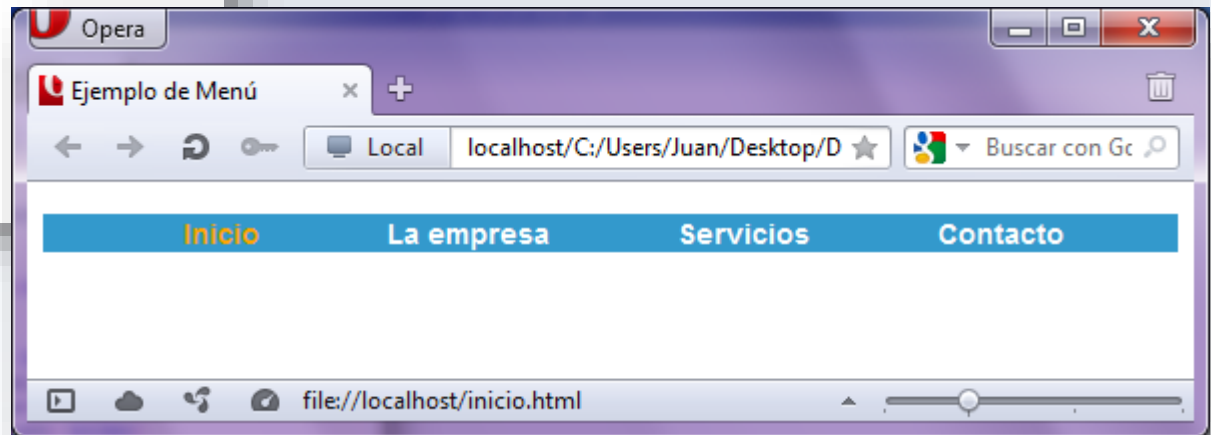
- Las listas aportan muchas funcionalidades a la hora de crear elementos iguales o que forman parte de un grupo.
- Cómo crear un menú con una lista:
  - Crear una lista desordenada con links.
  - Utilizar la propiedad **display** y asignarle el valor **inline**, para que los elementos se muestren seguidos en la misma línea.

```
<nav>
  <ul>
    <li><a href="/inicio.html" target="_blank">Inicio</a></li>
    <li><a href="/empresa.html" target="_blank">La empresa</a></li>
    <li><a href="/servicios.html" target="_blank">Servicios</a></li>
    <li><a href="/contacto.html" target="_blank">Contacto</a></li>
  </ul>
</nav>
```

# Listas

- CSS utilizados:

```
<style type="text/css">
nav {
  background:#39C;
}
nav li {
  display:inline;
}
nav li a{
  margin: 30px;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 0.9em;
  font-weight: bolder;
  color: white;
  text-decoration:none;
}
nav li a:hover {
  color: orange;
  text-decoration:none;
}
</style>
```



# Transformar en Lista con CSS

- Mediante CSS se pueden utilizar varios elementos como si formaran parte de una lista sin la necesidad de utilizar `<li>`. Para ello se establece el valor ***list-item*** en la propiedad ***display***:

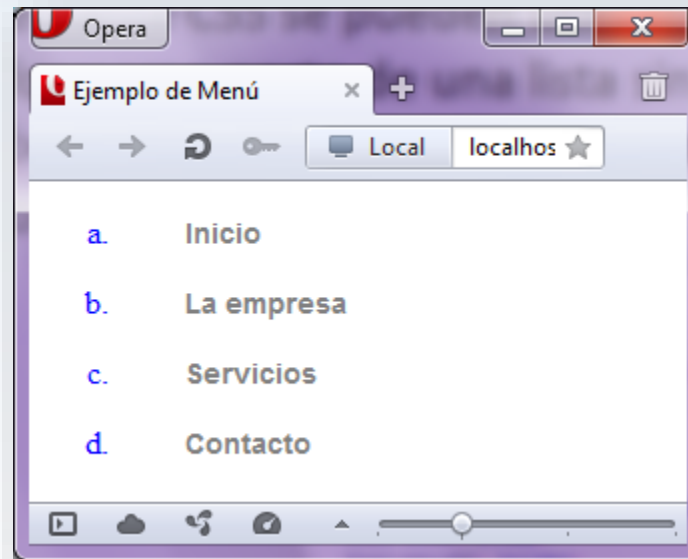
```
<style type="text/css">
.elemento {
    display: list-item;
    list-style-type: lower-alpha;
    color: blue;
}
.elemento a{
    margin: 30px;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 0.9em;
    font-weight: bolder;
    color: grey;
    text-decoration:none;
}
.elemento a:hover {
    color: orange;
    text-decoration:none;
}
</style>
```

La propiedad ***list-style-type*** permite seleccionar el tipo de lista. Tiene más de 15 posibles valores.

# Transformar en Lista con CSS

- Transformación de los párrafos en elementos de lista:

```
<nav>
  <ul>
    <p class="elemento"><a href="/inicio.html" target="_blank">Inicio</a></p>
    <p class="elemento"><a href="/empresa.html" target="_blank">La empresa</a></p>
    <p class="elemento"><a href="/servicios.html" target="_blank">Servicios</a></p>
    <p class="elemento"><a href="/contacto.html" target="_blank">Contacto</a></p>
  </ul>
</nav>
```



# Imágenes como Viñetas

- Existe la posibilidad de utilizar una imagen en una viñeta con la propiedad ***list-style-image***:

```
<style type="text/css">
.elemento {
  display: list-item;
  list-style-type: lower-alpha;
  list-style-image: url(tango.png);
}
.elemento a{
  margin: 30px;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 0.9em;
  font-weight: bolder;
  color: grey;
  text-decoration:none;
  vertical-align:10px;
}
.elemento a:hover {
  color: orange;
  text-decoration:none;
}
</style>
```





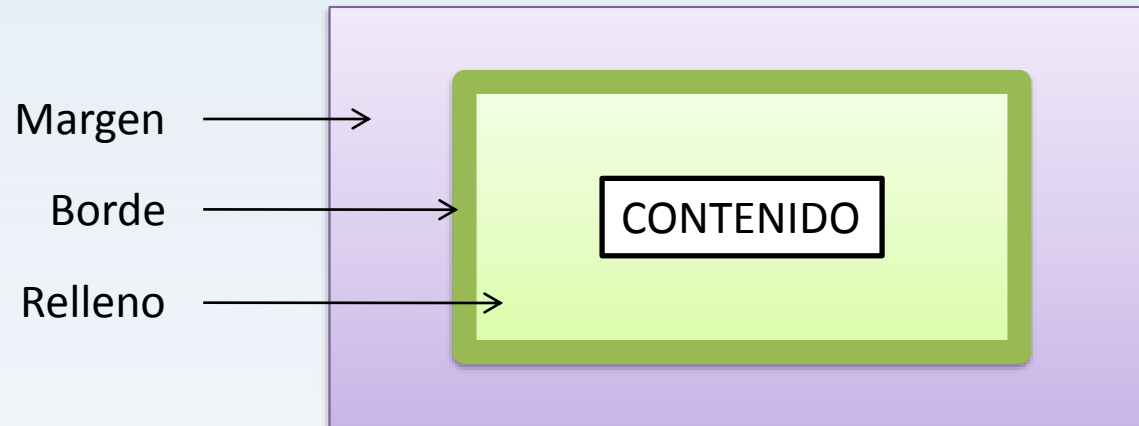
# TEMA 05

## Bordes, Márgenes, Rellenos y Foco



# Elementos en HTML

- Todos los elementos en HTML están incluidos en una caja, la cual tiene bordes, márgenes y rellenos.



Los bordes ya se trataron en la sección de tablas. Ahora se verán junto con los márgenes y los rellenos.

# Relleno

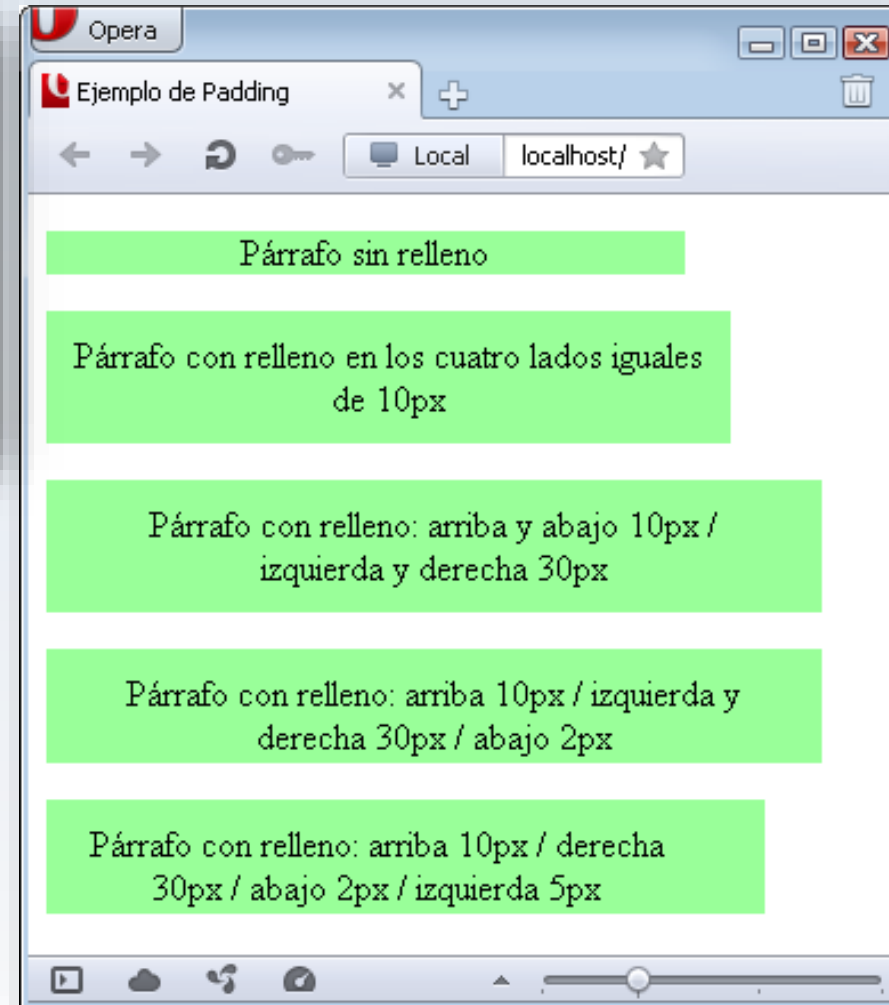
- El relleno especifica el espacio existente entre el contenido de un elemento y su borde.
- Se especifica mediante la propiedad ***padding***, que al igual que con los bordes, se puede especificar con propiedades específicas para cada lado o con propiedades resumen.
- Ejemplo utilizando sólo la propiedad resumen ***padding***:

```
<style type="text/css">
p {
  background-color:#9F9;
  text-align:center;
  width:280px;
}
p.a{
  padding: 10px;
}
p.b{
  padding: 10px 30px;
}
p.c{
  padding: 10px 30px 2px;
}
p.d{
  padding: 10px 30px 2px 5px;
}
</style>
```

# Relleno

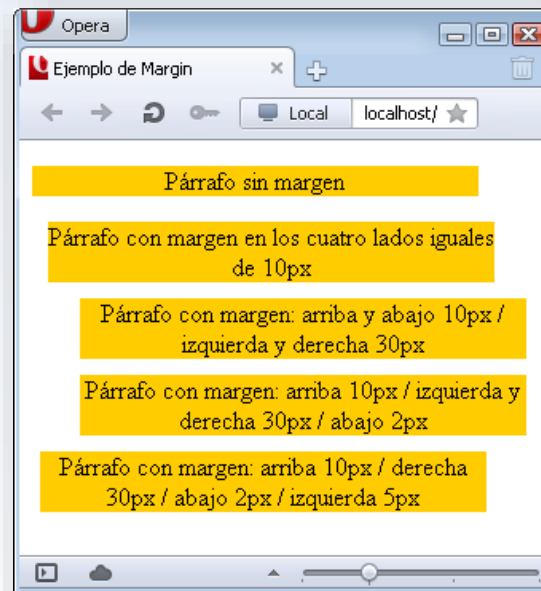
- Ejemplo:

```
<style type="text/css">
p {
  background-color:#9F9;
  text-align:center;
  width:280px;
}
p.a{
  padding: 10px;
}
p.b{
  padding: 10px 30px;
}
p.c{
  padding: 10px 30px 2px;
}
p.d{
  padding: 10px 30px 2px 5px;
}
</style>
```

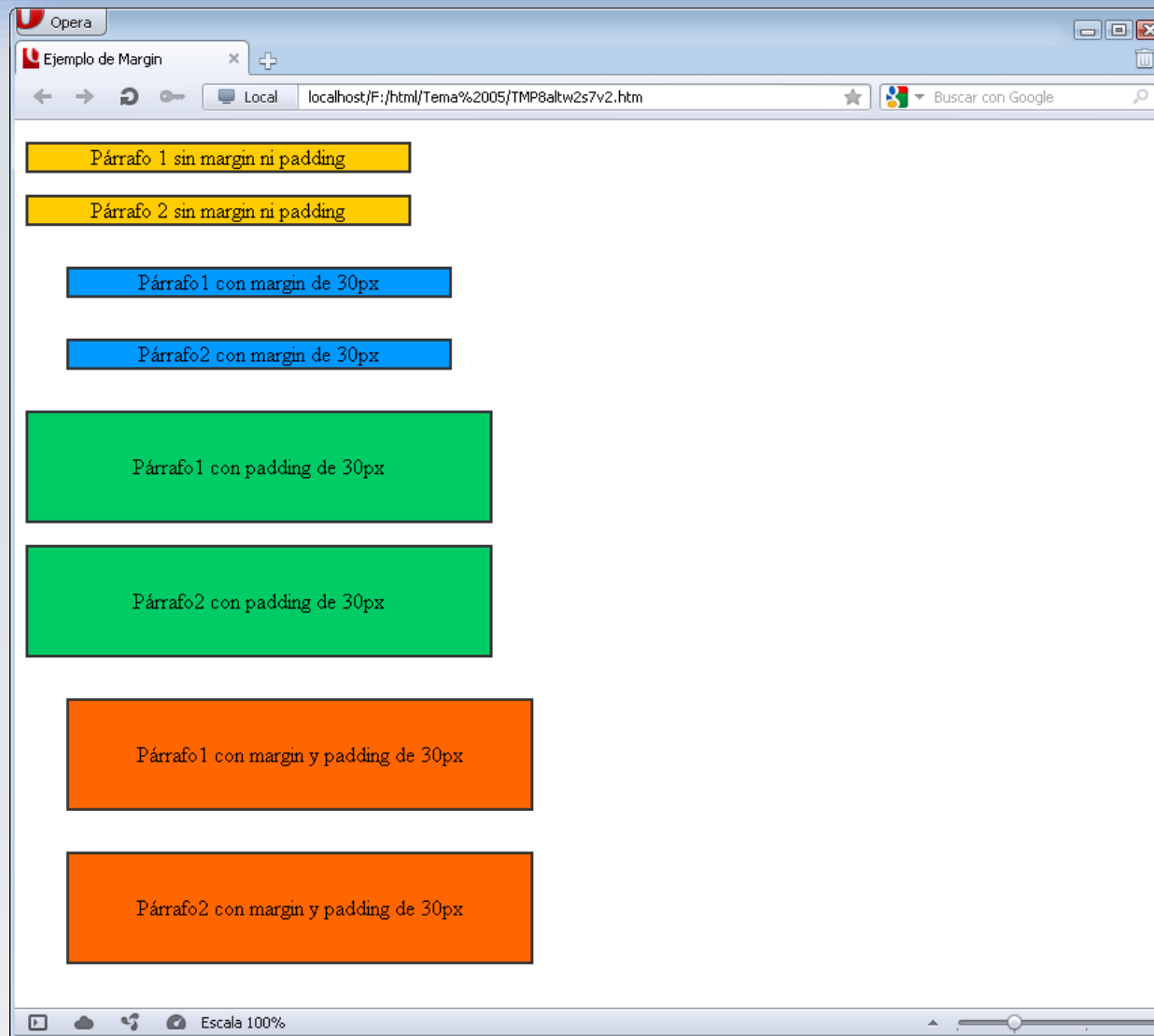


# Margen

- Un margen representa el espacio existente entre un elemento y un elemento adyacente.
- Se especifica mediante la propiedad ***margin***.
- Al igual que ocurre con los bordes y los rellenos, el margen también tiene propiedad resumen.



# Ejemplo



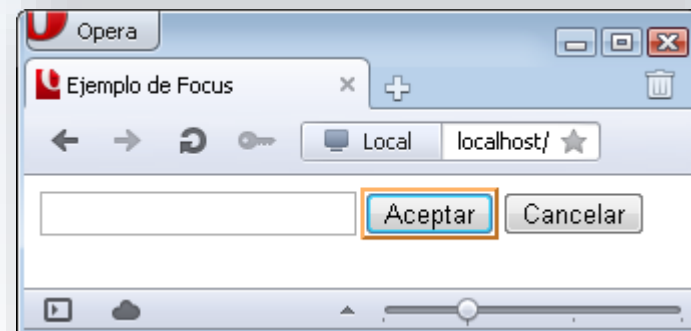
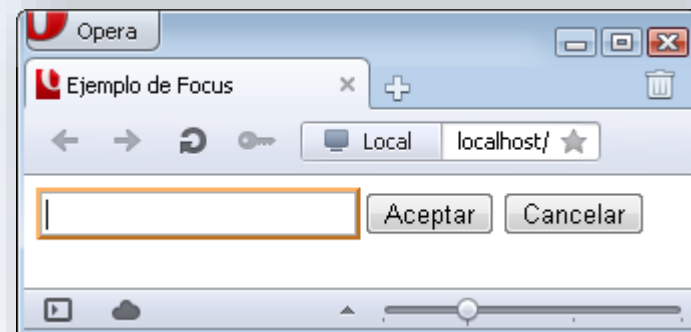
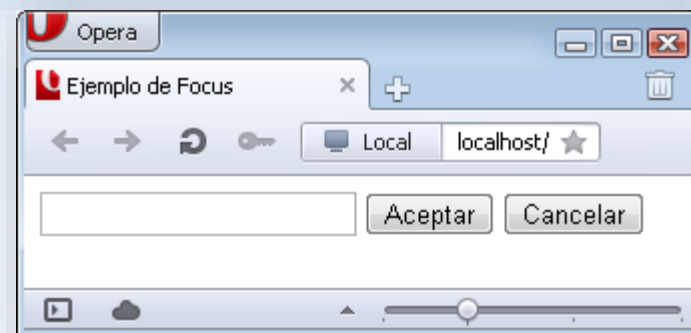
# Foco

- Mediante el pseudo-clase ***:focus***, podemos establecer un borde adicional a un elemento cuando éste tiene el foco, aunque no todos los navegadores lo representan en todos los elementos.
- En este caso todos los bordes tienen la misma apariencia.
- Las propiedades existentes son:
  - ***outline-color***
  - ***outline-style***
  - ***outline-width***
  - ***outline*** (propiedad resumen)

- Ejemplo:

```
<style type="text/css">
form *:focus {
    outline-color: #F93;
    outline-style: outset;
    outline-width: 2px;
}
</style>

</head>
<body>
<form>
    <input type="text">
    <button class="button">Aceptar</button>
    <button class="button">Cancelar</button>
</form>
</body>
</html>
```





# TEMA 05

## Colores y Fondos



# Tipos de Colores

- Los elementos de HTML actualmente pueden representar dos colores, el que corresponde al color del elemento y el que corresponde al color del fondo.
- El color del elemento (foreground color) se establece con la propiedad **color** como se vio en el tema anterior.
- Para el fondo (background color) existen varias propiedades para el color y para imágenes, algunas de ellas nuevas en CSS3.

# Propiedades para el Fondo

- Propiedades:
  - ***background-color***: especifica el color del fondo
  - ***background-attachment***: indica si la imagen del fondo es fija (***fixed***) o se mueve con el texto al desplazarse (**scroll**)
  - ***background-image***: especifica la url de la imagen del fondo
  - ***background-position***: especifica la posición de comienzo de la imagen. Permite palabras clave, porcentajes y posiciones en cualquier unidad
  - ***background-repeat***: indica cómo se repite la imagen. Acepta los valores **repeat**, **repeat-x**, **repeat-y**, **no-repeat**
  - ***background***: es la propiedad resumen

# Propiedades para el Fondo

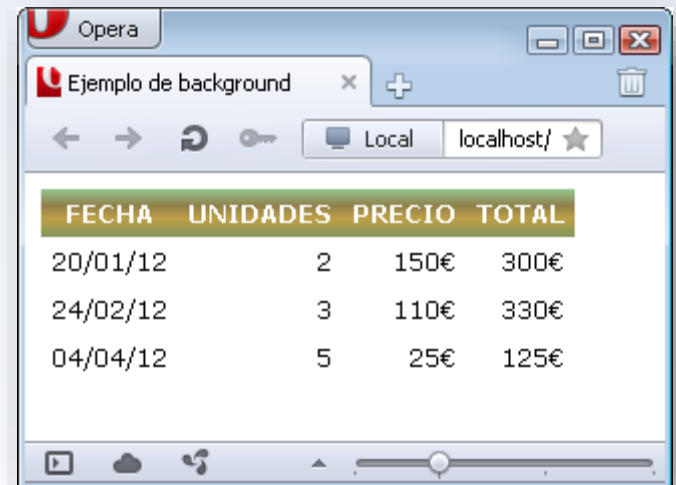
- Propiedades nuevas en CSS3:
  - ***background-clip***: establece el límite del elemento que mostrará el background. Acepta tres valores: ***border-box***, ***padding-box*** y ***content-box***
  - ***background-origin***: similar al anterior, pero para las imágenes
  - ***background-size***: especifica el tamaño de las imágenes del fondo. Se puede especificar el tamaño en píxeles o utilizar las palabras clave ***cover*** (*horizontal*) o ***contain*** (*vertical*), que ajustan la imagen al elemento

# Ejemplo 1

- En este ejemplo se utiliza una imagen para el fondo del **<th>** de una tabla.

```
<style type="text/css">
table {
  border-width:0px;
  border-spacing:0px;
}
table *{
  font-family:Verdana, Geneva, sans-serif;
  font-size:0.9em;
  padding:5px;
}
th {
  background-image:url(fondo03.png);
  background-repeat:repeat-x;
  font-weight:bolder;
  color:white;
  margin:0;
}
td {
  text-align:right;
}
</style>
```

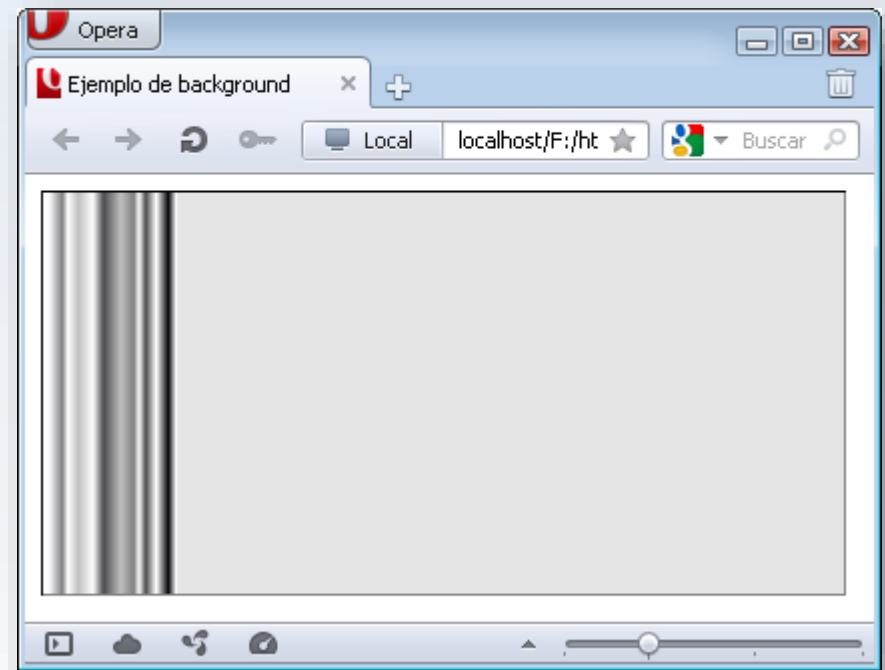
Imagen utilizada →



# Ejemplo 2

- Es este ejemplo se utiliza una imagen para el fondo de un **<div>** y se le aplica un borde

```
<style type="text/css">
div {
    background-image:url(fondo04.png);
    background-repeat:repeat-y;
    border-style:inset;
    border-width:1.5px;
    position:0 0;
    width:400px;
    height:200px;
}
</style>
```





# TEMA 05

## Posicionamiento

# Posicionamiento

- El posicionamiento de los elementos que componen un documento HTML es la parte esencial a la hora de maquetar el resultado final.
- Existen 5 tipos de posicionamiento, 4 de los cuales se pueden establecer mediante la propiedad ***position***:
  - ***static***: estático
  - ***relative***: relativo
  - ***absolute***: absoluto
  - ***fixed***: fijo

El posicionamiento flotante se establece mediante la propiedad ***float***

# Posicionamiento

- Otras propiedades:
  - ***top, right, bottom*** y ***left***: establecen la posición del elemento cuando no es *static*. Aceptan valores y porcentajes.
  - ***visibility***: indica si el elemento es visible o no. Acepta los valores ***visible*** y ***hidden***.
  - ***z-index***: especifica el valor para mostrar elementos superpuestos. Para que un elemento se coloque encima de otro tendrá que tener un ***z-index*** superior.



# Posicionamiento Estático

- Los elementos se muestran en el orden en el que han sido definidos.
- Ejemplo:



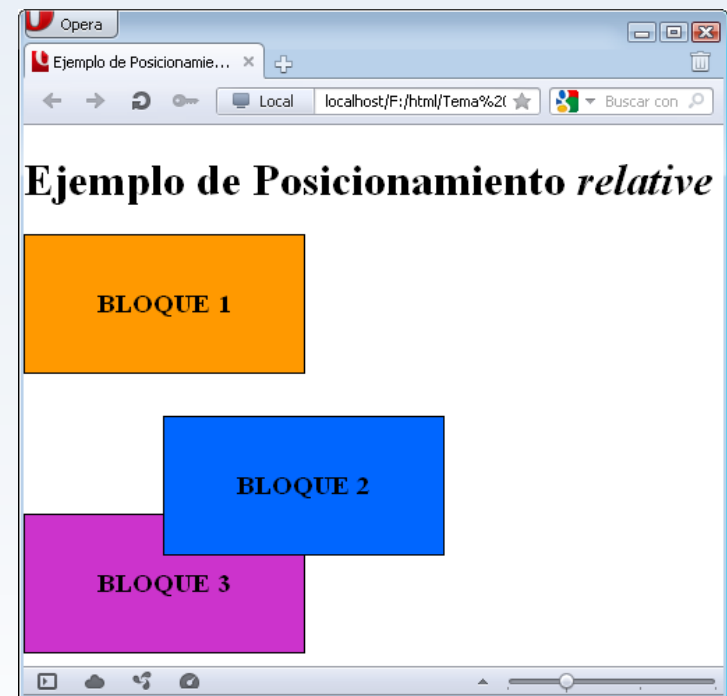
# Posicionamiento Relativo

- El elemento **se coloca en una posición relativa a la que tendría el elemento anterior a él con posición *static***, y en base a valores establecidos en otras propiedades (top, bottom, etc.).
- Un elemento con posición relativa no modifica la posición de otros elementos.

# Posicionamiento Relativo

- Ejemplo:
  - El bloque 2 tiene un posicionamiento relativo al bloque 1

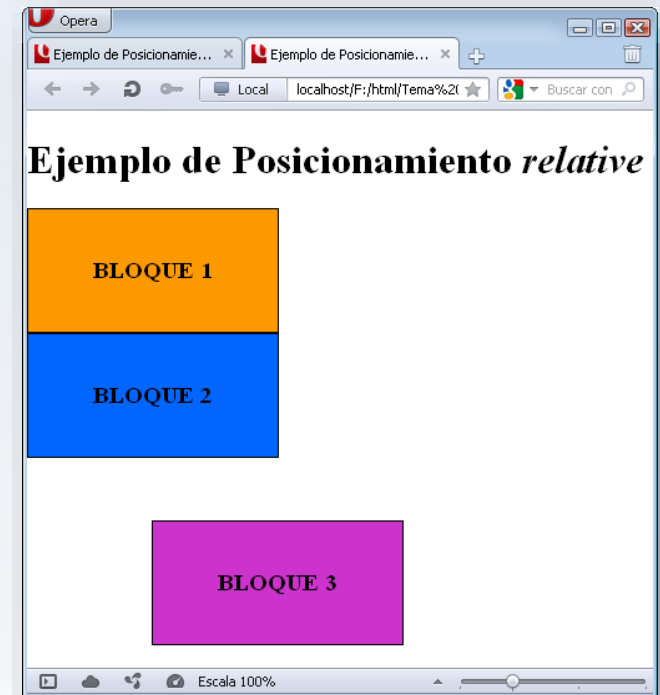
```
<style type="text/css">
html, body {
    margin: 0;
    padding: 0;
    height: 100%;
    overflow: hidden;
}
div {
    background: #0F0;
    border-style: solid;
    border-width: 1.5px;
    width: 200px;
    height: 100px;
    font: Verdana, Geneva, sans-serif;
    font-size: 1.2em;
    font-weight: bold;
    display: table;
}
.naranja {
    background: #F90;
}
.azul {
    background: #06F;
    position: relative;
    top: 30px;
    left: 100px;
}
.morado {
    background: #C3C;
}
p {
    display: table-cell;
    vertical-align: middle;
    text-align: center;
}
</style>
```



# Posicionamiento Relativo

- Ejemplo:
  - El bloque 3 tiene un posicionamiento relativo al bloque 2

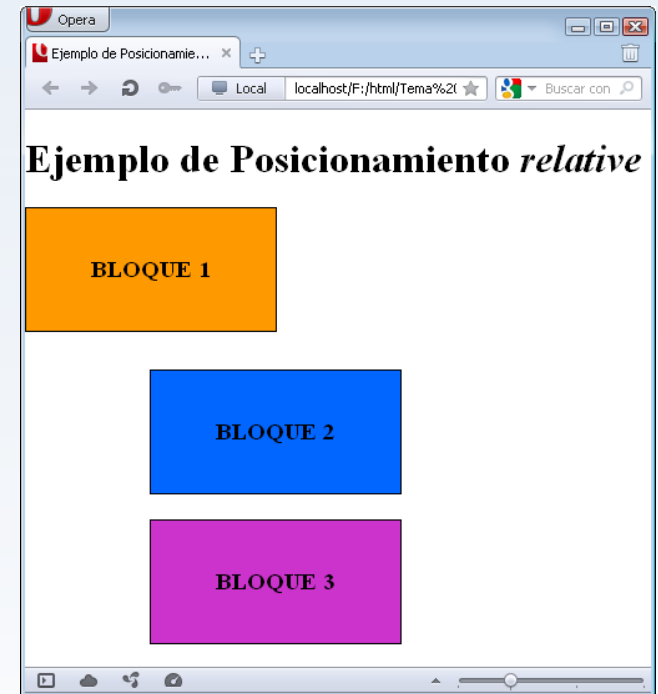
```
<style type="text/css">
html, body {
    margin: 0;
    padding: 0;
    height: 100%;
    overflow: hidden;
}
div {
    border-style: solid;
    border-width: 1.5px;
    width: 200px;
    height: 100px;
    font: Verdana, Geneva, sans-serif;
    font-size: 1.2em;
    font-weight: bolder;
    display: table;
}
.naranja {
    background: #F90;
}
.azul {
    background: #06F;
}
.morado {
    background: #C3C;
    position: relative;
    top: 50px;
    left: 100px;
}
p {
    display: table-cell;
    vertical-align: middle;
    text-align: center;
}
</style>
```



# Posicionamiento Relativo

- Ejemplo:
  - Los bloques 2 y 3 con posicionamiento relativo

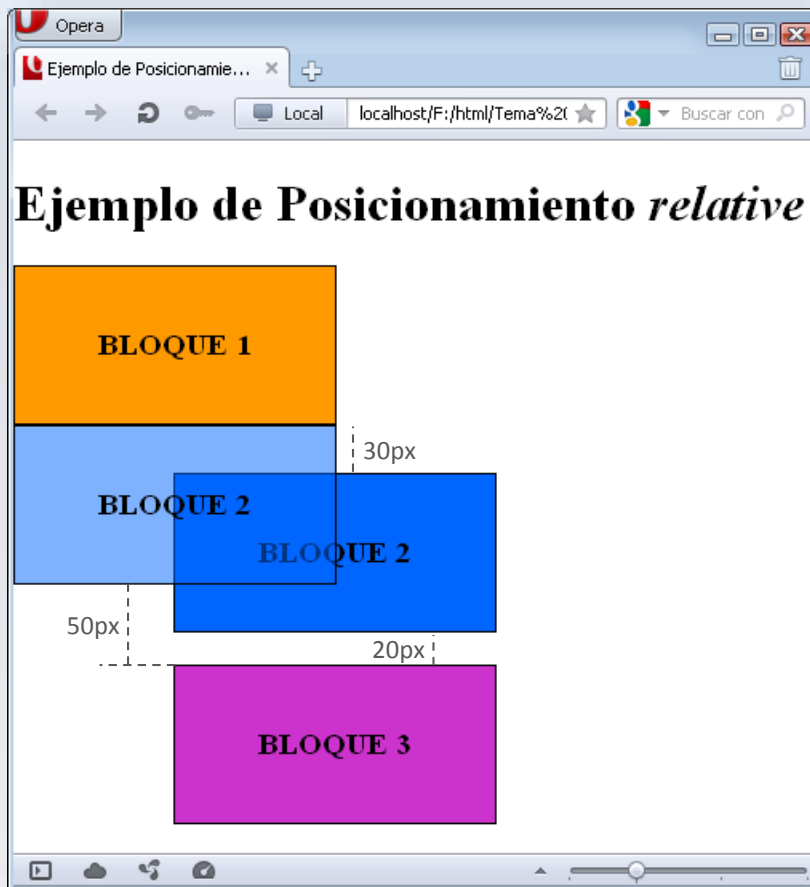
```
<style type="text/css">
html, body {
    margin: 0;
    padding: 0;
    height: 100%;
    overflow: hidden;
}
div {
    border-style: solid;
    border-width: 1.5px;
    width: 200px;
    height: 100px;
    font: Verdana, Geneva, sans-serif;
    font-size: 1.2em;
    font-weight: bolder;
    display: table;
}
.naranja {
    background: #F90;
}
.azul {
    background: #06F;
    position: relative;
    top: 30px;
    left: 100px;
}
.morado {
    background: #C3C;
    position: relative;
    top: 50px;
    left: 100px;
}
p {
    display: table-cell;
    vertical-align: middle;
    text-align: center;
}
</style>
```



¿Qué ocurre con el Bloque 3?

# Posicionamiento Relativo

- Explicación: el bloque 3 se coloca en la posición relativa a la que tendría el bloque 2 si fuera estático



El posicionamiento relativo de un elemento **no modifica la posición relativa de otros elementos**. Estos elementos se colocan de forma relativa a la posición que ocuparía el elemento anterior con posicionamiento estático.

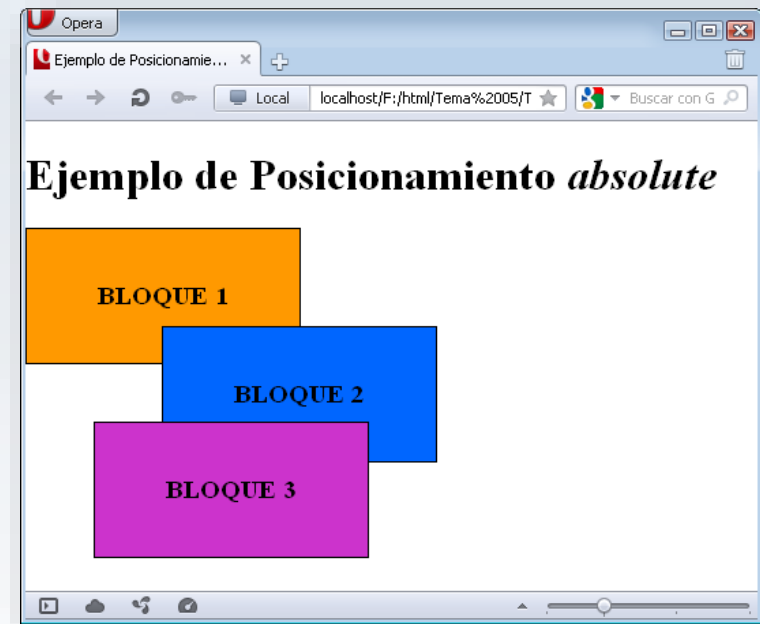
# Posicionamiento Absoluto

- El elemento **se coloca de forma absoluta a los bordes del elemento que lo contenga**. Puede ser un elemento HTML o la ventana del navegador. Al igual que ***relative***, su cambio de posición no afecta a otros elementos.
- Un elemento que utiliza este posicionamiento es **ignorado por el resto de elementos**, es decir, el elemento ignora su posición estática y se coloca en base a los bordes de su contenedor.

# Posicionamiento Absoluto

- Ejemplo:

```
<style type="text/css">
html, body {
    margin: 0;
    padding: 0;
    height: 100%;
    overflow: hidden;
}
div {
    border-style: solid;
    border-width: 1.5px;
    width: 200px;
    height: 100px;
    font: Verdana, Geneva, sans-serif;
    font-size: 1.2em;
    font-weight: bolder;
    display: table;
}
.naranja {
    background: #F90;
}
.azul {
    background: #06F;
    position: absolute;
    top: 150px;
    left: 100px;
}
.morado {
    background: #C3C;
    position: absolute;
    top: 220px;
    left: 50px;
}
p {
    display: table-cell;
    vertical-align: middle;
    text-align: center;
}
</style>
```



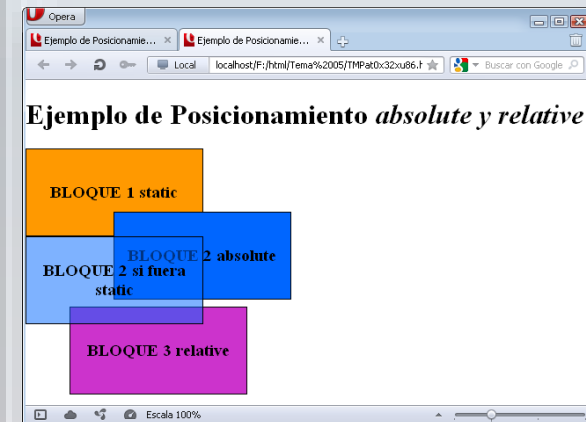


# Posicionamiento Absoluto

- Ejemplo de posicionamiento absoluto y relativo:

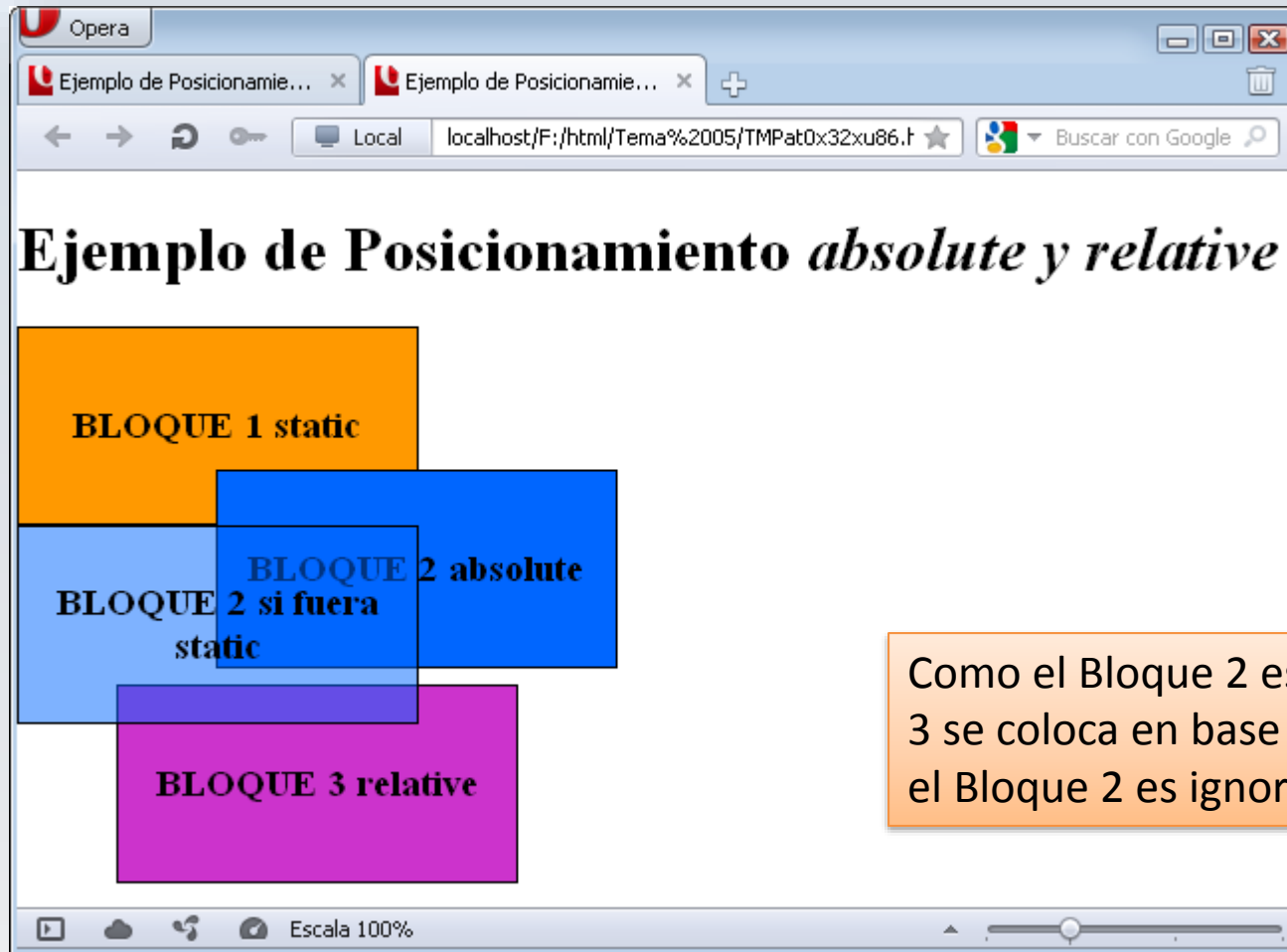
```
<style type="text/css">
html, body {
    margin: 0;
    padding: 0;
    height: 100%;
    overflow: hidden;
}
div {
    border-style: solid;
    border-width: 1.5px;
    width: 200px;
    height: 100px;
    font: Verdana, Geneva, sans-serif;
    font-size: 1.2em;
    font-weight: bolder;
    display: table;
}
.naranja {
    background: #F90;
}
.azul {
    background: #06F;
    position: absolute;
    top: 150px;
    left: 100px;
}

.azulito {
    background: rgba(0,102,255,0.5);
    position: fixed;
    top: 178px;
    left: 0px;
    z-index: 1;
}
.morado {
    background: #C3C;
    position: relative;
    top: 80px;
    left: 50px;
}
p {
    display: table-cell;
    vertical-align: middle;
    text-align: center;
}
</style>
```



# Posicionamiento Absoluto

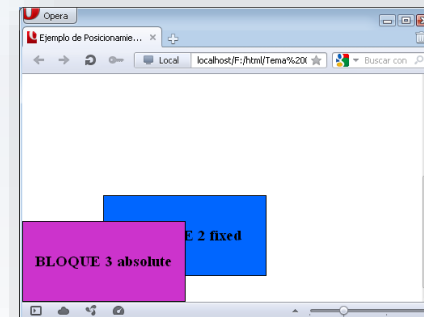
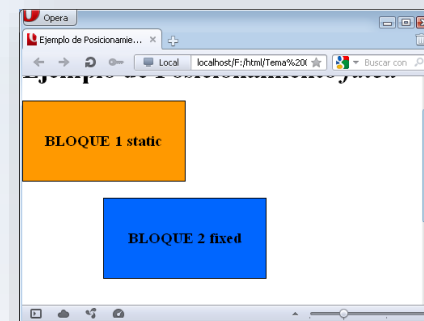
- Ejemplo de posicionamiento absoluto y relativo:

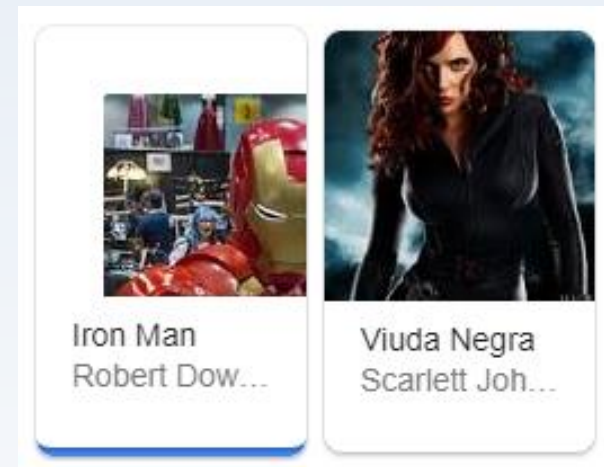
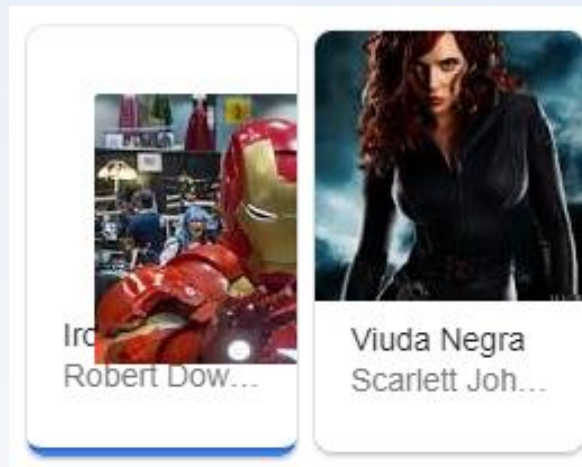
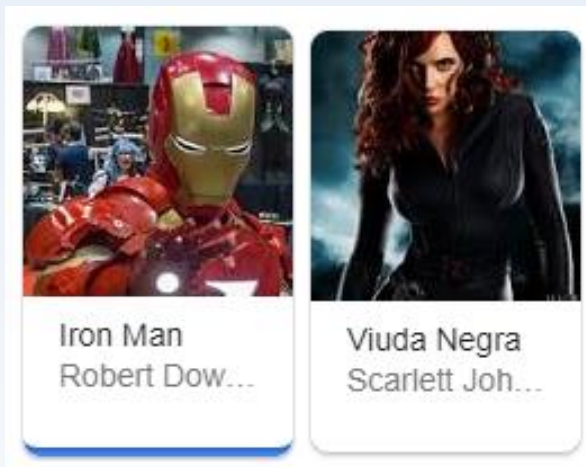


Como el Bloque 2 es *absolute*, el Bloque 3 se coloca en base al Bloque 1, ya que el Bloque 2 es ignorado por el Bloque 3.

# Posicionamiento Fijo

- El posicionamiento fijo es similar al absoluto, con la diferencia de que un elemento con posicionamiento fijo ignora el *scroll* de la ventana y permanece siempre en la misma posición.
- Ejemplo:





# Ejemplo

- Todos las imágenes están dentro de un *div* que las contiene a todas, y cada imagen en otro *div* con posicionamiento *static* o *relative*



# Ejemplo

- Modificamos el posicionamiento del *div* señalado a *absolute*.





# Ejemplo

- El *div* se coloca en base al contenedor de todas las imágenes. El resto de *div* ya no le tienen en cuenta y se recolocan siguiendo su posición *static*



# Posicionamiento *sticky* (pegajoso)

- Este posicionamiento es más reciente que los anteriores. Permite que un objeto se comporte como ***static*** o ***relative*** hasta que al hacer *scroll* y encontrarse a una distancia definida por la propiedad ***top***, se comporta como ***fixed***.
- Suele utilizarse mucho en menús o banners.



# Posicionamiento *sticky* (pegajoso)

- Ejemplo: <https://codepen.io/rachelandrew/pen/NbPamG>
- Los elementos con la letra se van quedando fijos y superpuestos al ir haciendo *scroll*.

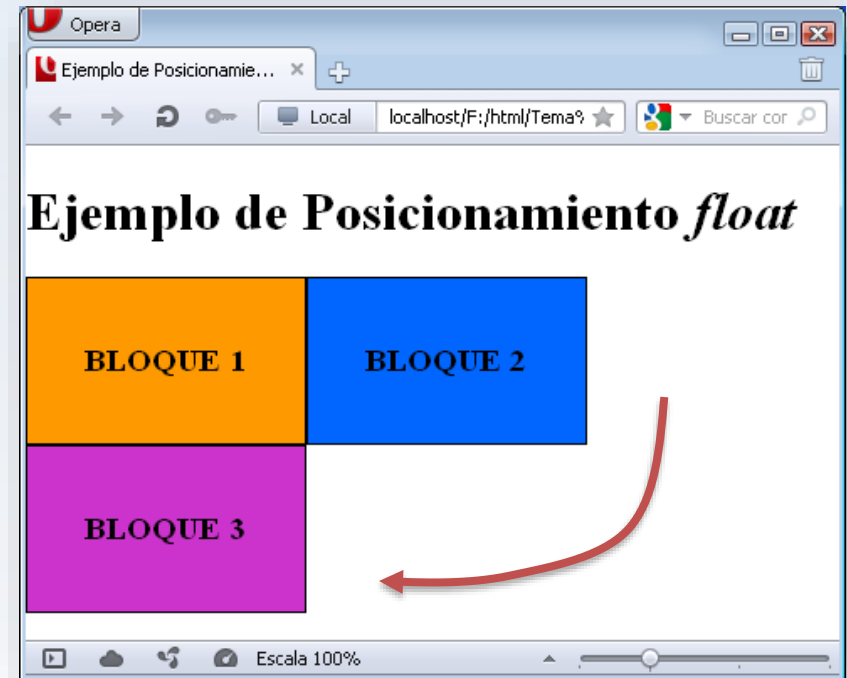


# Posicionamiento Flotante

- El posicionamiento flotante es el más complejo de todos, sin embargo es muy utilizado.
- Permite que un elemento se coloque lo más pegado posible a uno de los bordes laterales.
- Cuando un elemento utiliza este posicionamiento es ignorado por el resto de los elementos, salvo por los flotantes.
- Varios elementos flotantes se colocarán unos junto a otros, salvo que utilicen la propiedad ***clear*** para inicializar el posicionamiento.

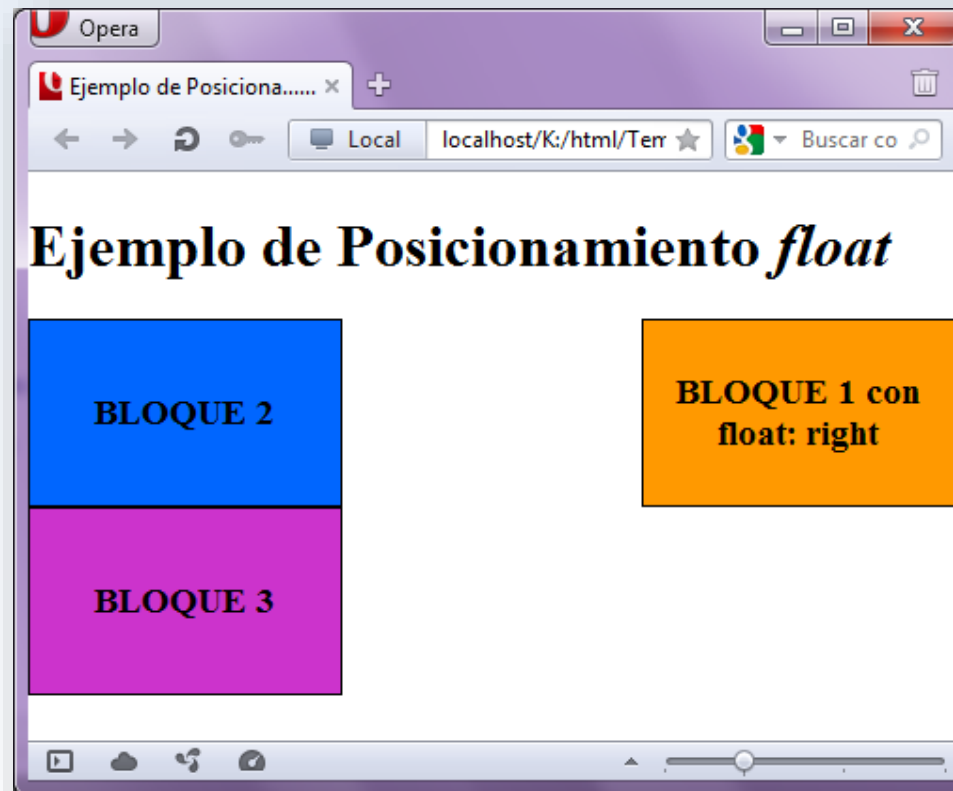
# Posicionamiento Flotante

- Ejemplo de 3 elementos flotantes:
  - Los elementos se colocan automáticamente al disminuir el ancho de la ventana.



# Posicionamiento Flotante

- Ejemplo aplicando flotación derecha al Bloque 1:
  - El resto de elementos ignoran al Bloque 1 y se colocan siguiendo su posición estática.

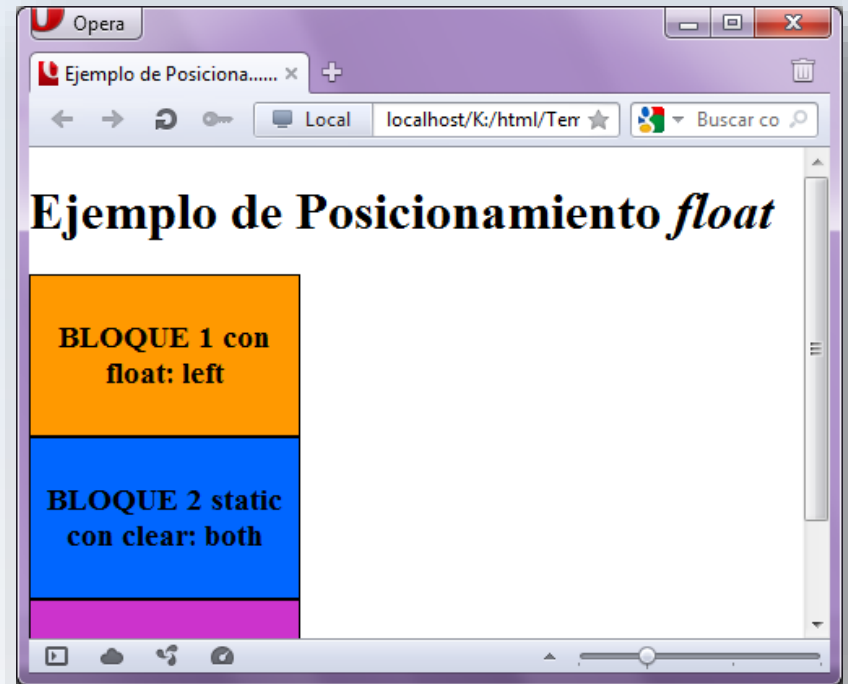
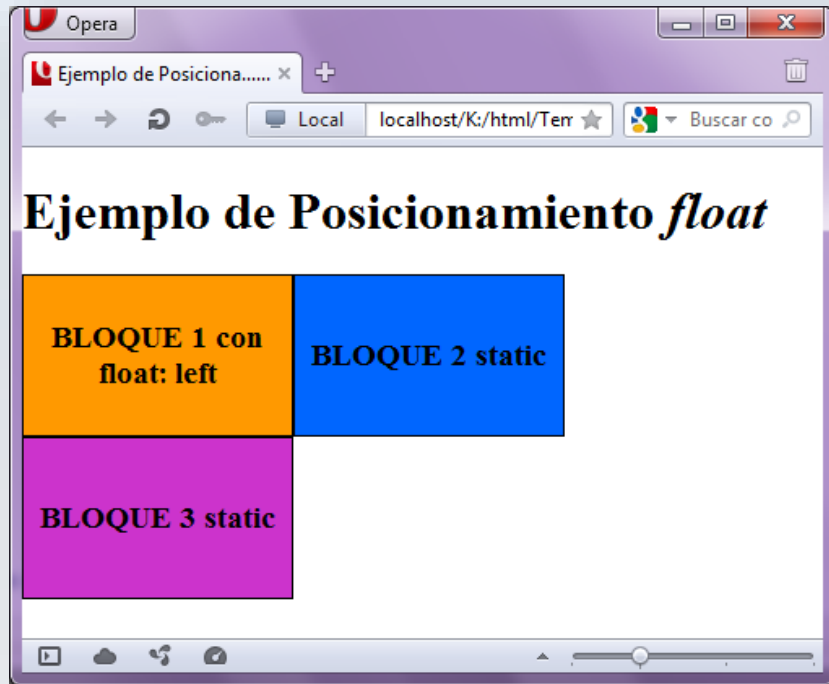


# Posicionamiento Flotante

- **Inicialización de la flotación**
  - Existe la posibilidad de inicializar la flotación mediante la propiedad ***clear***.
  - Cuando un elemento la inicializa, habrá un salto de línea y la flotación comenzará de nuevo.
  - ***clear*** tiene tres posibles valores:
    - ***right***: inicializa la flotación sólo para los elementos que tengan la flotación a la derecha
    - ***left***: inicializa la flotación sólo para los elementos que tengan la flotación a la izquierda
    - ***both***: inicializa la flotación para cualquier elemento que tenga flotación

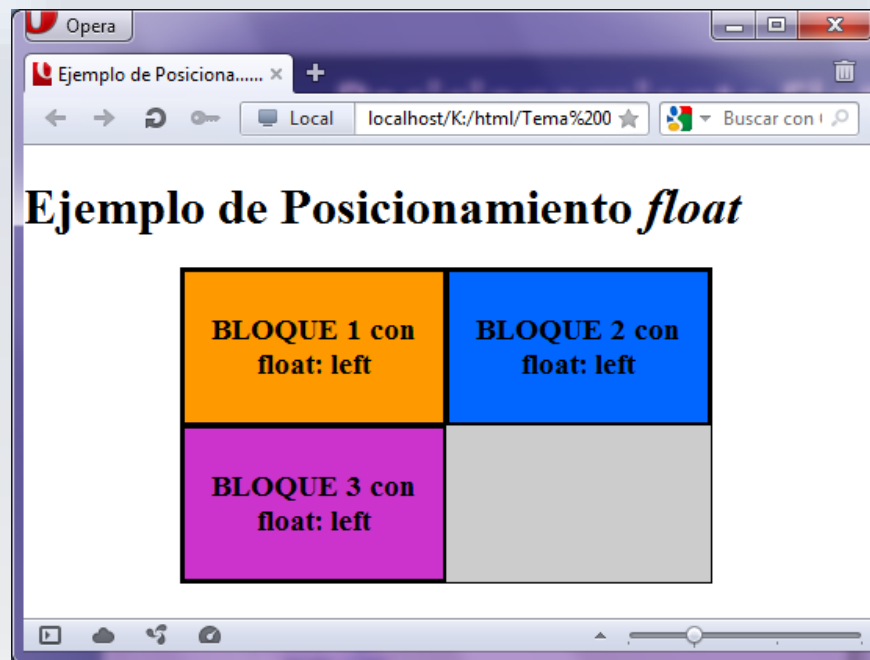
# Posicionamiento Flotante

- Ejemplo:



# Posicionamiento Flotante

- Evitar la descolocación de elementos flotantes
  - Un método para controlar mejor la colocación de elementos flotantes es incluyendo a los elementos dentro de otro bloque del mismo ancho que los bloques a los que incluye.

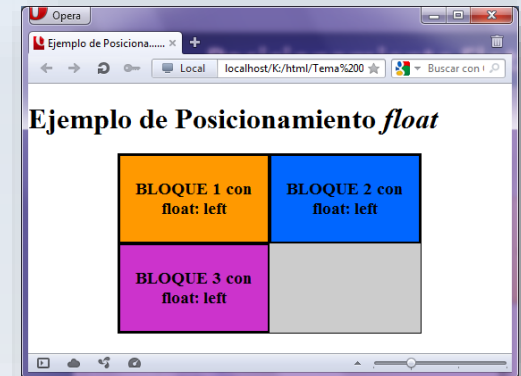


# Posicionamiento Flotante

- Evitar la descolocación de elementos flotantes

```
<style type="text/css">
html, body {
    margin: 0 auto;
    padding: 0;
    height: 100%;
    overflow: auto;
}
.grupo {
    margin: auto;
    border-style:solid;
    border-width:1.5px;
    width: 338px;
    height:200px;
    background: #CCC;
    clear:both;
    text-align:center;
}
.bloque {
    border-style:solid;
    border-width:2px;
    width:165px;
    height:100px;
    font:Verdana, Geneva, sans-serif;
    font-size:1.2em;
    font-weight:bold;
    display:table;
}

.naranja {
    background: #F90;
    float:left;
}
.azul {
    background: #06F;
    float:left;
}
.morado {
    background:#C3C;
    float:left;
}
p {
    display: table-cell;
    vertical-align: middle;
    text-align:center;
}
</style>
```



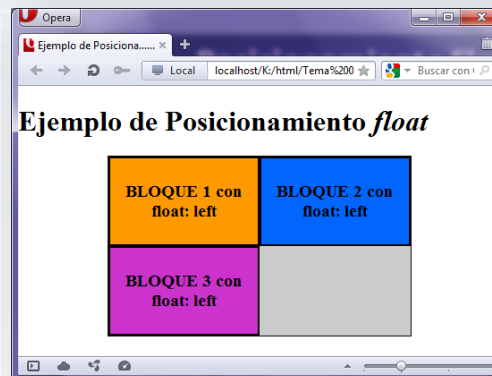
Para centrar un div en la página o en otro elemento, se establece el valor de ***margin: auto*** o ***0 auto*** y un valor para el ancho con ***width***.



# Posicionamiento Flotante

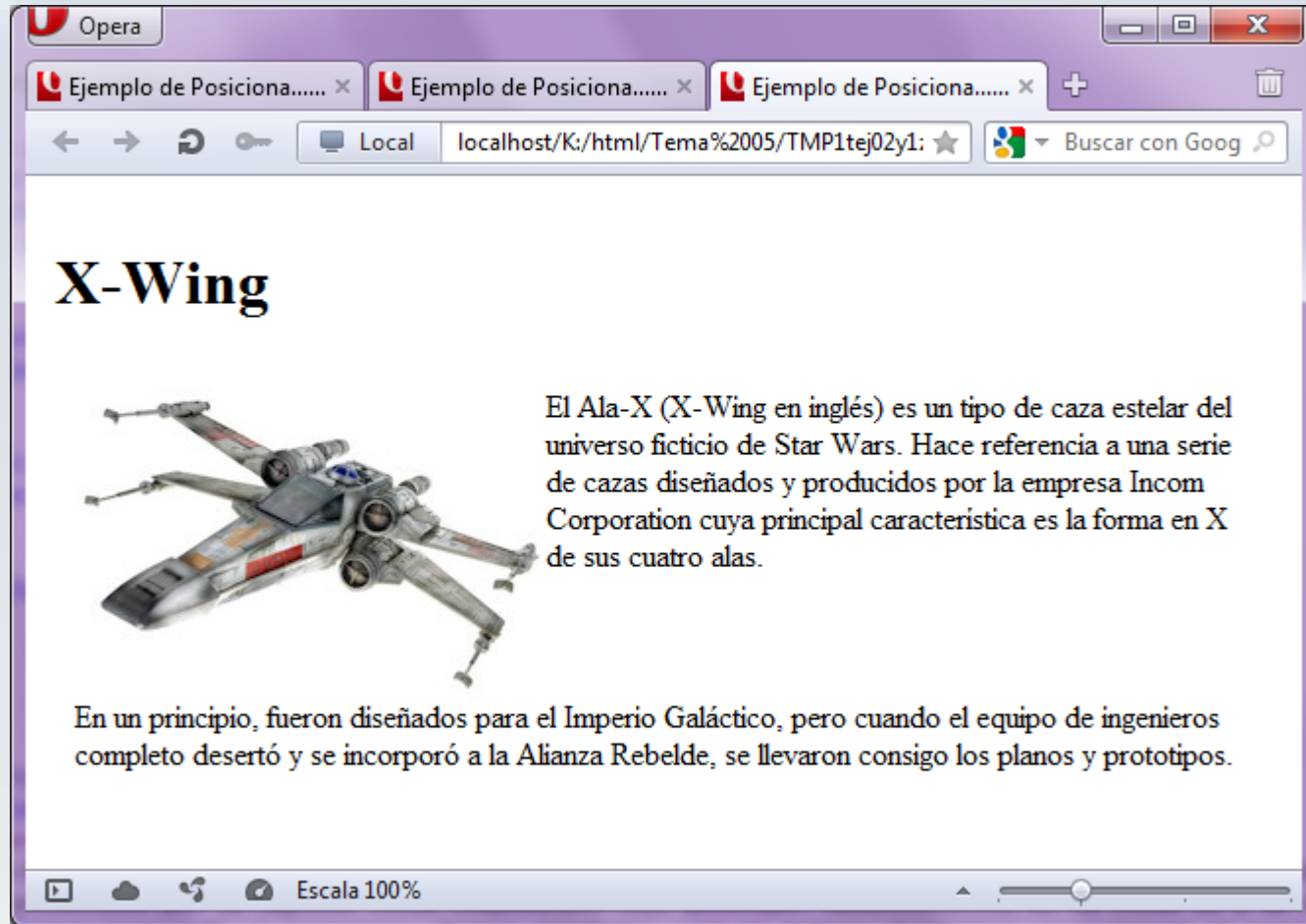
- Evitar la descolocación de elementos flotantes

```
<body>
<h1>Ejemplo de Posicionamiento <span style="font-weight:bold; font-style:italic;">float</span></h1>
<div class="grupo">
  <div class="bloque naranja">
    <p>BLOQUE 1 con float: left</p>
  </div>
  <div class="bloque azul">
    <p>BLOQUE 2 con float: left</p>
  </div>
  <div class="bloque morado">
    <p>BLOQUE 3 con float: left</p>
  </div>
</div>
</body>
</html>
```



# Posicionamiento Flotante

- Ejemplo de texto con una imagen



# Posicionamiento Flotante

- Ejemplo de texto con una imagen

```
<style type="text/css">
```

```
html, body {  
    margin: 0 auto;  
    padding: 0;  
    height: 100%;  
    overflow: auto;  
}
```

```
.grupo {  
    margin: auto;  
    width: 600px;  
}
```

```
.imagen {  
    float: left;  
    border: 0px;  
}
```

```
h1 {  
    padding: 15px;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>X-Wing</h1>
```

```
<div class="grupo">
```

```
    
```

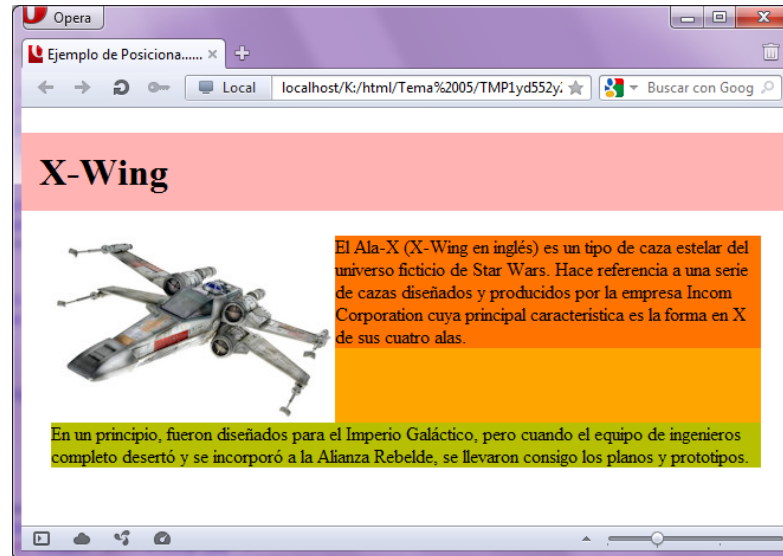
```
    <p>El Ala-X (X-Wing en inglés) es un tipo de caza estelar del universo ficticio de Star Wars. Hace  
    diseñados y producidos por la empresa Incom Corporation cuya principal característica es la forma en X
```

```
    <p style="clear: both;">En un principio, fueron diseñados para el Imperio Galáctico, pero cuando el  
    desertó y se incorporó a la Alianza Rebelde, se llevaron consigo los planos y prototipos.</p>
```

```
</div>
```

```
</body>
```

```
</html>
```



# Limitación de Tamaños

- Las propiedades ***width*** y ***height*** permiten especificar un ancho y una altura fija respectivamente.
- Pero también existen valores para mínimos y máximos, que permiten controlar el tamaño variable en función del tamaño de la ventana. Estas propiedades son:
  - ***min-width***
  - ***max-width***
  - ***min-height***
  - ***max-height***

**CSS**



# TEMA 05

## Media queries

# Media Queries

- Reglas de CSS3 que permiten aplicar ciertas reglas en función de las condiciones del dispositivo.
- Permiten modificar hojas de estilo completas o propiedades individuales.
- Sintaxis para propiedades:

```
@media not|only <medio> and (<propiedad medio>) {  
    Propiedades CSS;  
}
```

- Sintaxis para hojas:

```
<link rel="stylesheet" media=" <medio> and|not|only  
(<propiedad medio>)" href="<hoja de estilos">
```

# Media Queries

- **Medios:** son los dispositivos en los que se va a visualizar la página y para los que se va a aplicar la condición.
- **Lista de medios:**
  - all: todos los dispositivos
  - print: impresoras
  - screen: pantallas
  - speech: lectores de pantalla

# Media Queries

- **Propiedades del dispositivo:** existen 36 propiedades que se pueden consultar. Las más utilizadas son:
  - device-height
  - device-width
  - height
  - **max-device-height**
  - **max-device-width**
  - max-height
  - max-width
  - **min-device-width**
  - **min-device-height**
  - **orientation**
  - resolution
  - width

Cuando se hace referencia a **device** es porque se obtiene del dispositivo físico (pantalla de ordenador, móvil o tablet), no del área de visualización como la ventana del navegador.



# Media Queries

- Ejemplos:

```
@media only screen and (max-width: 767px) {  
  #logo {  
    margin-top: -20px;  
    margin-left: 0px;  
  }  
}  
  
@media only screen and (min-device-width : 768px) and (max-device-width : 1023px) {  
  #logo {  
    margin-top: -55px;  
    margin-left: 30px;  
  }  
}
```

# Media Queries

- Ejemplos:

```
@media only screen and (min-device-width : 1024px) and (max-device-width : 1279px)
and (orientation : portrait) {
    #logo {
        margin-top: -30px;
        margin-left: 30px;
    }
    #logo img {
        width: 70%;
    }
}

@media only screen and (min-device-width : 1024px) and (max-device-width : 1279px)
and (orientation : landscape) {
    #logo {
        margin-top: 0px;
        margin-left: 10px;
    }
}
```



# TEMA 05

**JavaScript y CSS**

# JavaScript y CSS

- Antes de entrar en el tema de JavaScript, vamos a ver una forma muy sencilla de modificar los estilos utilizando JavaScript.
- Para ello habrá que crear:
  - Una función en JavaScript que modificará o añadirá alguna propiedad
  - Adición de un evento a algún elemento para lanzar la función

# Función en JavaScript

- Para crear una función en JavaScript es necesario declararla dentro del elemento **<script>**
- La función tendrá un identificador y unos parámetros que serán útiles para obtener el objeto a modificar y la propiedad
- Para declarar la función se utiliza la palabra clave **function**
- Ejemplo:

```
<script type="text/javascript">  
function CambiarTamano(id){  
    obj=document.getElementById(id);  
    obj.style.fontSize="2em";  
}  
</script>
```

# Función en JavaScript

- Explicación del ejemplo:
  - La función tiene el identificador **CambiarTamano**
  - y el parámetro **id**, el cual servirá para recibir el elemento
  - **document.getElementById(id);**
    - Este método se utiliza en JS para obtener el objeto del documento, el cual se almacena en la variable **obj**
  - **obj.style.fontSize="2em";**
    - Aplica el valor 2em a la propiedad font-size

```
<script type="text/javascript">
function CambiarTamano(id){
    obj=document.getElementById(id);
    obj.style.fontSize="2em";
}
</script>
```

# Propiedades CSS en JavaScript

- JavaScript distingue entre mayúsculas y minúsculas
- Sólo admite el **guión bajo** para usarlo en identificadores
- Dado que muchas propiedades de CSS contienen guiones:
  - JS respeta el nombre de la mayoría de las propiedades en **minúsculas**
  - En aquellas que contienen un guión, se elimina el guión y se convierte a mayúsculas la letra que sigue al guión
    - *margin-left* será *marginLeft*

# Llamada a una función de JS

- Existen varias maneras de llamar a una función en JS, como por ejemplo mediante un **link** o mediante la respuesta a un **evento**.
- **Mediante un link:**

```
<a href="javascript:CambiarTamano('idtexto')">Agrandar texto desde link</a>
```

- En la propiedad **href** del elemento **<a>** se escribe la llamada con la siguiente sintaxis:

```
"javascript:<nombre_de_la_función>(<lista_de_parámetros>)"
```



# Llamada a una función de JS

- Como respuesta a un evento:

```
<button onClick="CambiarTamano('idtexto')">Agrandar texto</button>
```

- Los eventos se añaden a la etiqueta inicial del elemento HTML siguiendo la siguiente sintaxis:

```
<evento>="<nombre_de_la_función>(<lista_de_parámetros>)"
```

# JavaScript y CSS

- Ejemplo completo:

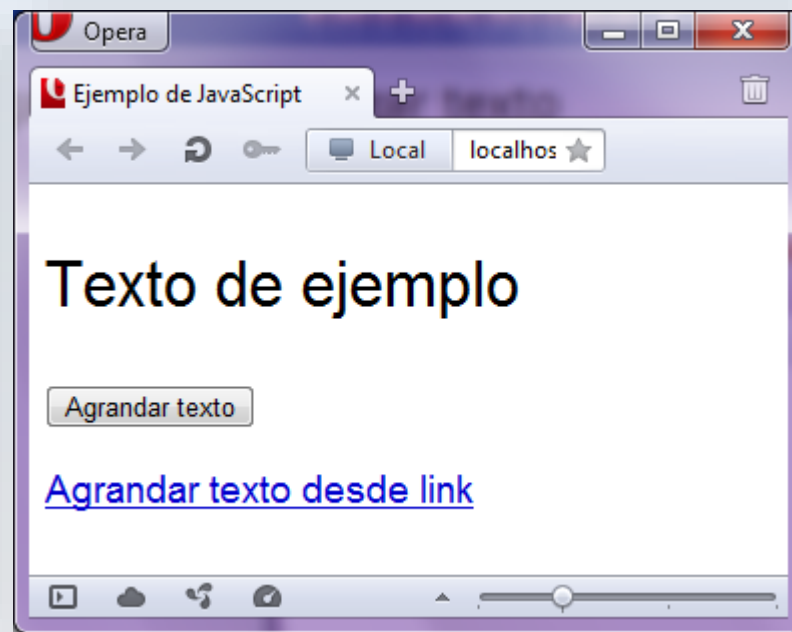
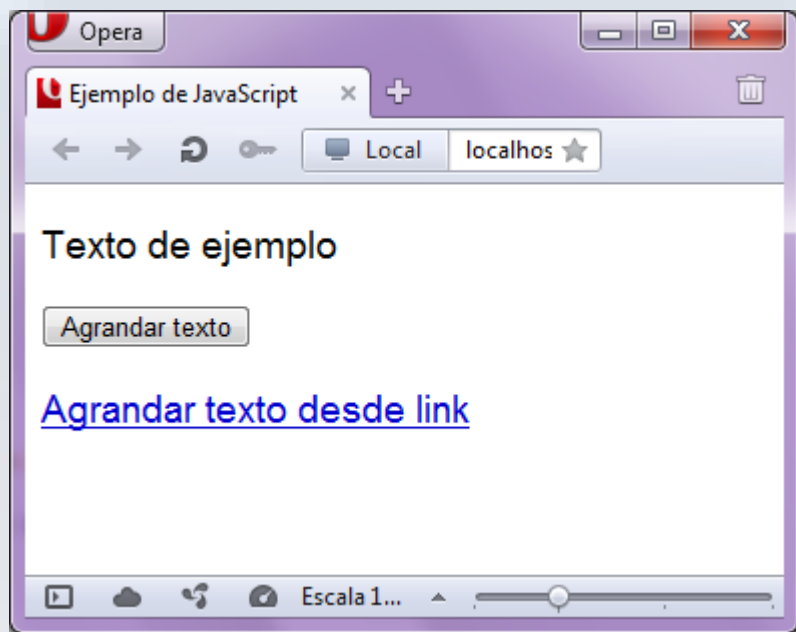
```
<script type="text/javascript">
function CambiarTamano(id){
    obj=document.getElementById(id);
    obj.style.fontSize="2em";
}
</script>

<style type="text/css">
p {
    font-family:Arial, Helvetica, sans-serif;
    font-size:1.2em;
}
</style>

</head>
<body>
<p id=idtexto>Texto de ejemplo</p>
<button onClick="CambiarTamano('idtexto')">Agrandar texto</button>
<p>
    <a href="javascript:CambiarTamano('idtexto')">Agrandar texto desde link</a>
</p>
</body>
</html>
```

# JavaScript y CSS

- Ejemplo completo:



# Modificar una Clase

- Para modificar una clase, el código de la función de JS sería la siguiente:

```
<script type="text/javascript">
function CambiarTamano(id){
    obj=document.getElementById(id);
    obj.className="otraclase";
}
</script>

<style type="text/css">
p {
    font-family:Arial, Helvetica, sans-serif;
    font-size:1.2em;
}
.otraclase {
    background:black;
    color:white;
}
</style>
```

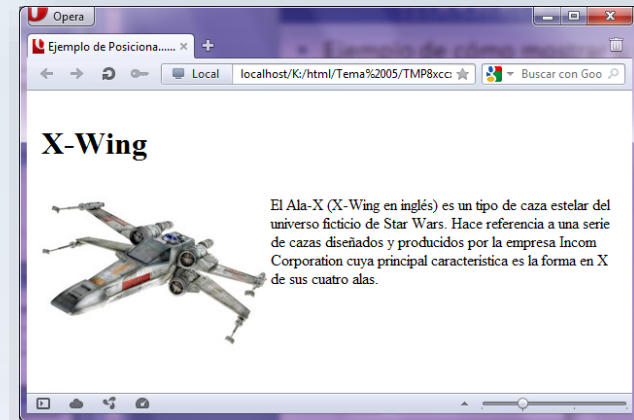
```
<script type="text/javascript">
function CambiarTamano(id){
    obj=document.getElementById(id);
    obj.className="otraclase amarillo";
}
</script>

<style type="text/css">
p {
    font-family:Arial, Helvetica, sans-serif;
    font-size:1.2em;
}
.otraclase {
    background:black;
    color:white;
}
.amarillo {
    color:yellow;
}
</style>
```

# Mostrar / Ocultar elementos

- Ejemplo de cómo mostrar u ocultar elementos:

```
<script type="text/javascript">
function MostrarTexto(id,mostrar){
  obj=document.getElementById(id);
  if (mostrar==true)
    obj.style.visibility="visible";
  else
    obj.style.visibility="hidden";
}
</script>
<style type="text/css">
html, body {
  margin: 0 auto;
  padding: 0;
  height: 100%;
  overflow: auto;
}
.grupo {
  margin: auto;
  width: 600px;
}
.imagen {
  float: left;
  border: 0px;
}
h1 {
  padding: 15px;
}
</style>
```



# Mostrar / Ocultar elementos

- Ejemplo de cómo mostrar u ocultar elementos:

```
</head>
<body>
<h1>X-Wing</h1>
<div class="grupo">

  <p>El Ala-X (X-Wing en inglés) es un tipo de caza estelar del universo ficticio de Star Wars. Hace referencia a una serie de cazas diseñados y producidos por la empresa Incom Corporation cuya principal característica es la forma en X de sus cuatro alas. </p>
  <p style="clear:both;" id=idtexto>En un principio, fueron diseñados para el Imperio Galáctico, pero cuando el equipo de ingenieros completo desertó y se incorporó a la Alianza Rebelde, se llevaron consigo los planos y prototipos.</p>
</div>

</body>
</html>
```

# Zoom en Imágenes

- Ejemplo de cómo hacer un zoom en imágenes:

```
<script type="text/javascript">
function Zoom(id,activar){
  obj=document.getElementById(id);
  if(activar){
    obj.style.width="360px";
    obj.style.height="237px";
  }
  else{
    obj.style.width="240px";
    obj.style.height="158px";
  }
}
</script>
```

