

Hojas de Estilo CSS

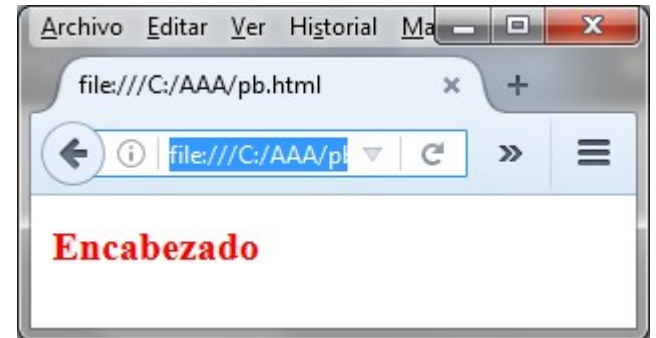
- HTML puede ser usado para representar
 - Semántica: `<H1>` significa que un elemento es un encabezado principal
 - Presentación: los elementos `<H1>` se ven de cierta manera
- Es mejor separar la semántica de la presentación para:
 - Presentar los documentos en diferentes plataformas (navegadores, teléfonos, etc.)
 - Generar documentos con una apariencia consistente
 - Hacer cambios en la semántica y presentación por separado.

Semántica vs Presentación

```
<!DOCTYPE html>
<html>
<head> <meta charset="utf-8"> </head>
<body>
<H1 align="center">Encabezado</H1>
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<head> <meta charset="utf-8">
    <link rel="stylesheet" title="present1"
        type="text/css" href="present1.css">
</head>
<body>
<H1>Encabezado</H1>
</body>
</html>
```



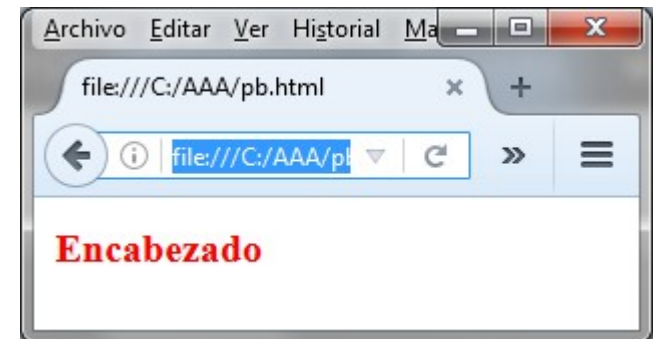
```
H1 {
    text-align: left;
    color: red;
    font-size: 15px;
}
```

- Archivo que contiene los estilos que se aplicarán a los elementos HTML
- Debe ser “importada” en el archivo HTML por medio de una etiqueta “link”.

```
<!DOCTYPE html>
<html>
<head> <meta charset="utf-8">
  <link rel="stylesheet" title="present1"
    type="text/css" href="present1.css">
</head>
<body>
<H1>Encabezado</H1>
</body>
</html>
```

present1.css

```
H1 {
  text-align: left;
  color: red;
  font-size: 15px;
}
```



Hojas para Distintos Medios de Presentación

5

- Un documento HTML puede indicar distintas hojas de estilo (css) para distintos medios: pantalla, impresión, teléfono, etc.
- Cada etiqueta “link” puede indicar a cuales medios (media) será aplicada la hoja referida.

```
<!DOCTYPE html>
<html>
<head> <meta charset="utf-8">
  <link rel="stylesheet" title="present1"
    type="text/css" href="present1.css"
    media="screen, tv, projection" />

  <link rel="stylesheet"
    type="text/css" href="present2.css"
    media="handheld, print" />
</head>
<body>
<H1>Encabezado</H1>
</body>
</html>
```

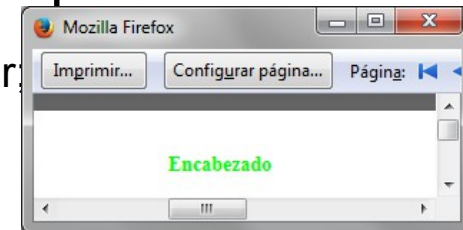
present1.css

```
H1 {
  text-align: left;
  color: red;
  font-size: 15px;
}
```

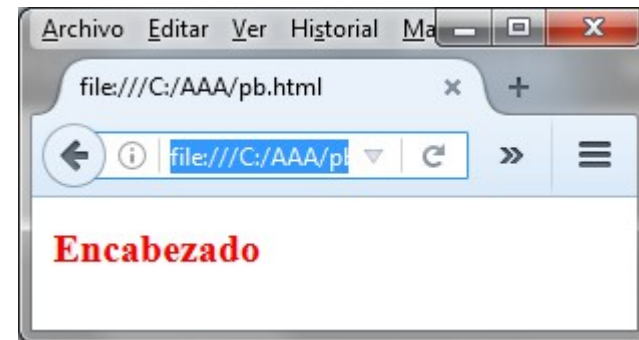
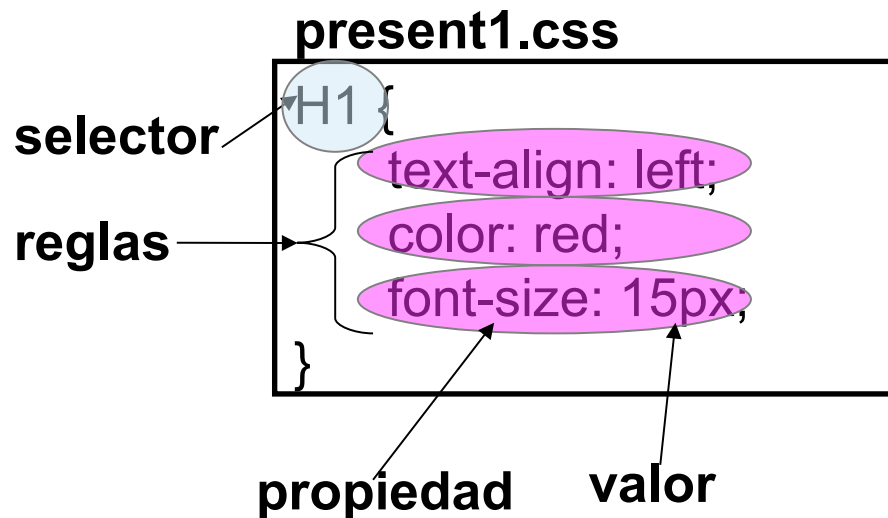


present2.css

```
H1 {
  text-align: center;
  color: lime;
  font-size: 20px;
}
```

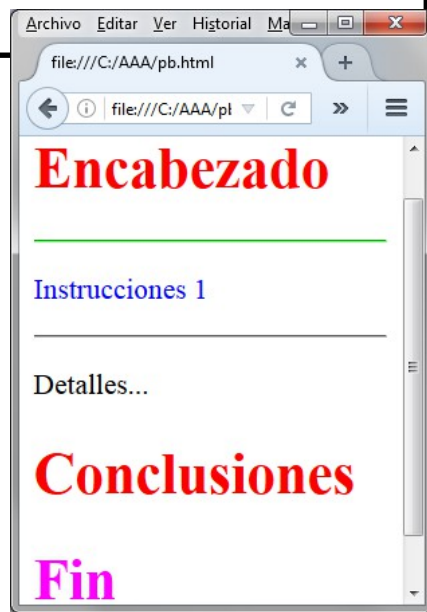


- Una Hoja de estilo define reglas de estilo con la siguiente estructura:
 - Selector: indica a qué elementos del documento html aplicará esa regla.
 - Declaraciones: indican la propiedad (atributo) y el valor que le corresponde.



- Por **tipo** de elemento html: h1{ ...}, etc.
- Por **Id** de un elemento: #id1{ ...}
- Por la **clase** (class) adscrita a elementos: .c1 { ...}
- Por **clase y tipo**: h1.c1 {...}

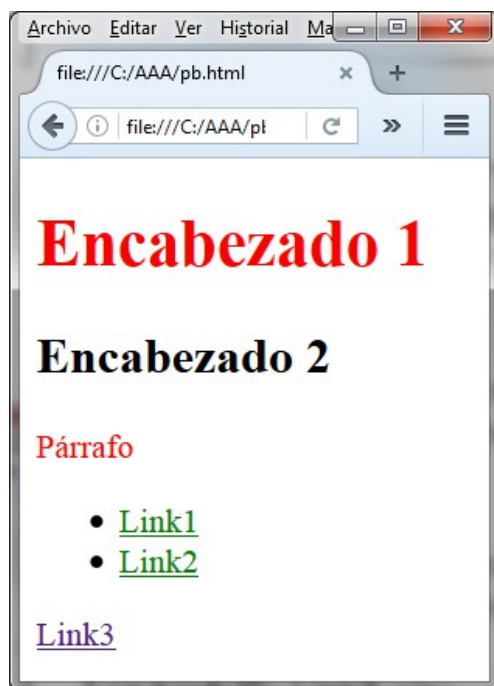
```
h1 { color: red; }  
#id1 { color: lime; }  
.c1 { color: blue; }  
h1.c1 { color: magenta; }
```



```
<!DOCTYPE html>  
<html>  
  <head> <meta charset="utf-8">  
    <link rel="stylesheet" title="present1"  
      type="text/css" href="present1.css" />  
  </head>  
  <body>  
    <h1>Encabezado</h1>  
    <hr id="id1">  
    <p class="c1"> Instrucciones 1</p>  
    <hr>  
    <p> Detalles... </p>  
    <h1>Conclusiones</h1>  
    <h1 class="c1"> Fin</h1>  
  </body>  
</html>
```

- Agrupados: $s_1, s_2 \{ \dots \}$: aplica a elementos de cualquiera s_1 o s_2
- Descendientes: $s_1 s_2 \{ \dots \}$: aplica a elementos seleccionados por s_2 que estén dentro de elementos seleccionados por s_1 .

```
h1,p {color: red; }  
ul li a {color: green; }
```



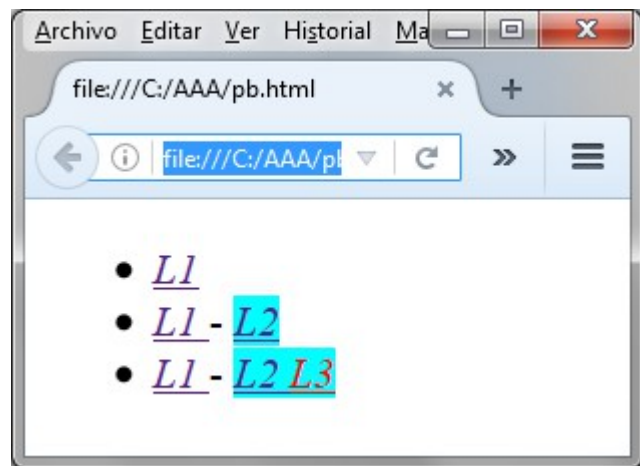
```
<!DOCTYPE html>  
<html>  
  <head> <meta charset="utf-8">  
    <link rel="stylesheet" title="present1"  
      type="text/css" href="present1.css" />  
  </head>  
  <body>  
    <h1>Encabezado 1</h1>  
    <h2>Encabezado 2</h2>  
    <p> Párrafo </p>  
    <ul>  
      <li> <a href=""> Link1 </a> </li>  
      <li> <a href=""> Link2 </a> </li>  
    </ul>  
    <a href=""> Link3 </a>  
  </body> </html>
```


Selectores: hijos y hermanos

9

- Hijos: $s_1 > s_2 \{ \dots \}$: s_2 que sean hijos de s_1
- Hermanos: $s_1 \sim s_2 \{ \dots \}$: s_2 hermanos a la derecha de s_1 .
- Gemelos: $s_1 + s_2 \{ \dots \}$: s_2 que sean hermanos inmediatos de s_1 .

```
li > a {font-style: italic;}  
a ~ a {background-color: cyan;}  
a + a {color: red;}
```



```
<!DOCTYPE html>  
<html>  
<head> <meta charset="utf-8">  
  <link rel="stylesheet" title="present1"  
    type="text/css" href="present1.css" />  
</head>  
<body>  
<ul>  
  <li> <a href=""> L1 </a> </li>  
  <li> <a href=""> L1 </a> <span>-</span>  
    <a href=""> L2 </a> </li>  
  <li> <a href=""> L1 </a> <span>-</span>  
    <a href=""> L2 </a>  
    <a href=""> L3 </a> </li>  
</ul>  
</body> </html>
```

Selectores: atributos y/o valor

10

- Atributo: `s[a]{ ...}`: **s** que tengan el atributo **a**.
- Valor: `s[a=v]{ ...}`: **s** que tengan en el atributo **a** el valor **v**.

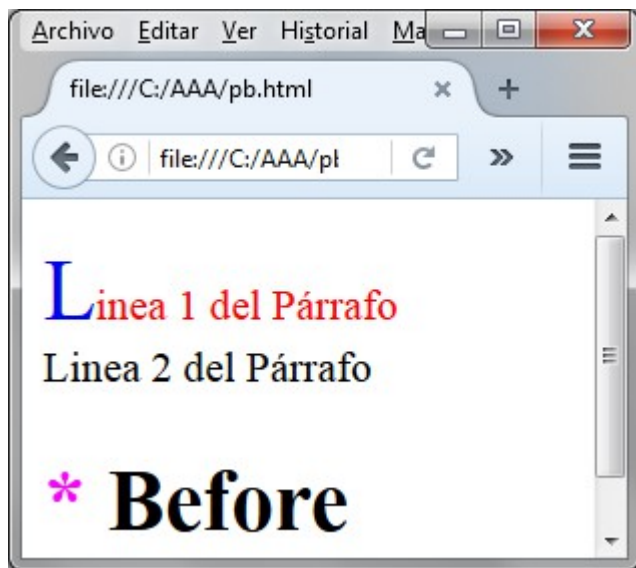
```
input[title]{
  text-decoration: underline;}
input[type=text]{
  background: pink;}
input[type=submit]{
  color: blue;}
```

```
<!DOCTYPE html>
<html> <head> <meta charset="utf-8">
  <link rel="stylesheet" title="present1"
    type="text/css" href="present1.css" /></head>
<body>
<form>
  Month<input type="text" size="1">
  $<input type="text" size="1">
  Pass<input type="password" size="1">
  <input type="submit" value="Add"
    title="add a month"><br>
</form>
</body></html>
```



- Son utilizados para estilizar **una parte** específica de un elemento: `selector::pseudo-element { ... }`

```
p::first-line {color: red; }  
p::first-letter { color: blue;  
                  font-size: 200%; }  
h1::before {content: "* ";  
            color: magenta;}
```



```
<!DOCTYPE html>  
<html>  
<head> <meta charset="utf-8">  
      <link rel="stylesheet" title="present1"  
            type="text/css" href="present1.css" />  
</head>  
<body>  
<p> Linea 1 del Párrafo <br>  
    Linea 2 del Párrafo  
</p>  
<h1>Before</h1>  
</body>  
</html>
```

- Seleccionan elementos basándose en parte de su **estado**:
selector:pseudo-class { ... } . (*hover, disabled, etc.*)

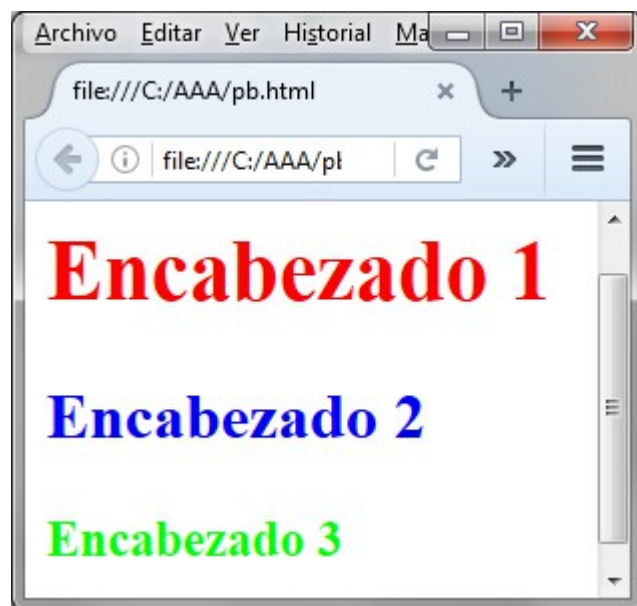
```
input[type=submit]:disabled{
  color: lightgray;}
input[type=text]:disabled {
  background: red;}
tbody tr:nth-child(odd) {
  background: #66ff00;}
tbody tr:nth-child(even){
  background: #ccff66;}
tbody tr:hover {
  background: cyan;}
```

Month	Savings	Month	Savings
January	\$100	January	\$100
February	\$80	February	\$80
March	\$90	March	\$90
M <input type="text"/>	\$ <input type="text"/>	M <input type="text"/>	\$ <input type="text"/>
<input type="button" value="Add"/>		<input type="button" value="Add"/>	

```
...
<body>
<table>
  <thead>
    <tr><th>Month</th> <th>Savings</th></tr>
  </thead>
  <tbody>
    <tr><td>January</td> <td>$100</td></tr>
    <tr><td>February</td> <td>$80</td></tr>
    <tr><td>March</td> <td>$90</td></tr>
  </tbody>
</table>
<form> M<input type="text" size="1">
  $<input type="text" size="1">
  <input type="submit" value="Add" disabled>
</form>
</body></html>
```

- **Externos**: en un archivo aparte y se “importan” mediante “link”
- **Internos**: se declaran con la etiqueta “style” dentro del “head”
- **Atributo**: en el atributo “style” de un elemento html específico

```
h1 { color: red; }
```



```
<html>
<head> <meta charset="utf-8">
  <link rel="stylesheet" title="present1"
    type="text/css" href="present1.css" />
  <style> h2 { color: blue; } </style>
</head>
<body>
<h1>Encabezado 1</h1>
<h2>Encabezado 2</h2>
<h3 style="color: lime;">Encabezado 3</h3>
</body>
</html>
```

- Múltiples selectores seleccionan (“match”) un elemento
- Muchos estilos “caen” sobre él en cascada
- Cuál estilo aplica?
 - Encontrar todas las declaraciones que aplican al elemento y propiedad específicos para el tipo de medio (media)
 - Las declaraciones más específicas prevalecen:
 - id
 - clase/pseudo-clase
 - tipo de elemento
 - universal
 - Atributo “style” es el más específico
 - Reglas con selector, luego
 - id
 - clase/pseudo-clase
 - tipo de elemento
 - universal
 - Atributo en el HTML, menos importante





- Atributo “style” más específico
- Reglas con selector
 - Id
 - clase/pseudo-clase
 - tipo de elemento
 - universal
- Atributo HTML

```
<head> <meta charset="utf-8">
<style>
  td { vertical-align: middle; background-color: blue; text-align: center;}
  .cl1 { background-color: lime; text-align: left;}
  #id1 { text-align: right;}
</style>
</head>
<body>
<table>
<tr height="100px">
<td width="100px" valign="bottom" bgcolor="red" align="left">hola</td>
<td width="200px" valign="bottom" bgcolor="red" align="left" class="cl1" >hola</td>
<td width="150px" valign="bottom" bgcolor="red" align="left" class="cl1" id="id1">hola</td>
</tr>
</table>
</body>
```



- Atributo “style” más específico
- Reglas con selector
 - Id
 - clase/pseudo-clase
 - tipo de elemento
 - universal
- Atributo HTML

```
<style>
td    { vertical-align: middle; background-color: blue; text-align: center;}
.cl1  {
      background-color: lime; text-align: left;}
#id1  {
      text-align: right;}
</style>
```

```
...
<td width="150px" valign="bottom" bgcolor="red" align="left" class="cl1" id="id1">hola</td>
...

```

HTML	:	width="150px"	valign="bottom"	bgcolor="red"	align="left"
Element	:	vertical-align: middle; background-color: blue; text-align: center;			
Class	:	background-color: lime; text-align: left;			
Id	:	text-align: right;			

width="150px" vertical-align: middle; background-color: lime; text-align: right;

- </body>



- Todos los elementos de html pueden considerarse como cajas.
- Cada caja puede definir un margen (*margin*), un borde (*border*), un “relleno” (*padding*) y su contenido

```
<!DOCTYPE html>
```

```
<html>
```

```
<head> <meta charset="utf-8">
```

```
<style>
```

```
  p { margin: 20px; border: 10px solid lime; padding: 20px;  
        background-color: lightgrey; display: inline-block; }
```

```
  span { margin: 10px; border: 20px solid yellow; padding: 10px;  
          background-color: pink; display: inline-block; }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Hola</p> <p>mundo</p> <span>Hello</span> <span>world</span>
```

```
</body> </html>
```



El Modelo de Cajas (espaciado flexible)

19

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<style>
```

```
  p { margin-left: 100px; margin-top: 20px;
```

```
    border-top: 30px solid lime; border-bottom: 10px solid yellow;
```

```
    padding: 20px;
```

```
    background-color: lightgrey; display: inline-block; }
```

```
  span { margin: 10px;
```

```
    border: 20px solid yellow;
```

```
    padding: 10px 30px 50px 70px;
```

```
    background-color: pink; display: inline-block; }
```

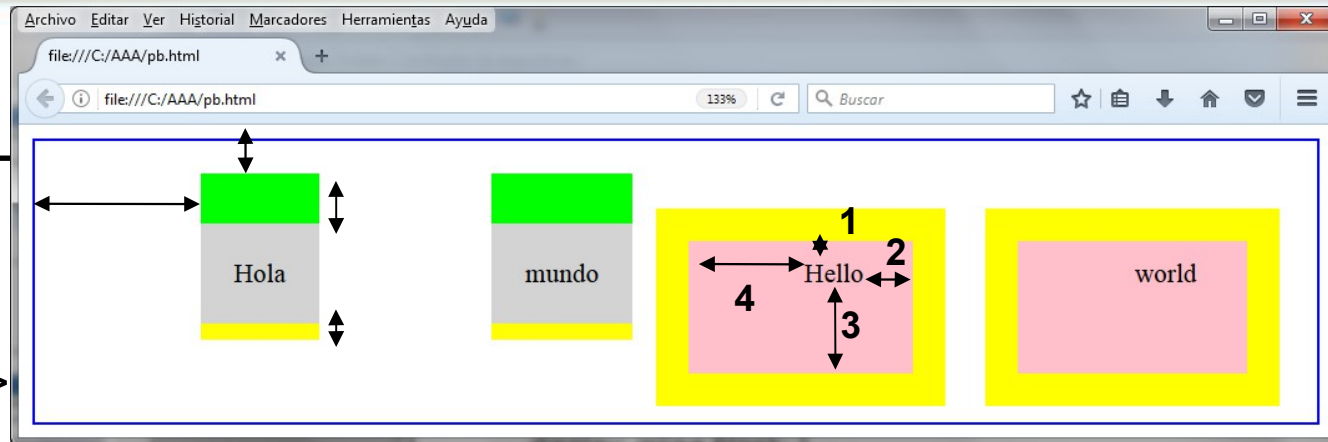
```
  div { border: 2px solid blue;}
```

```
</style> </head >
```

```
<body>
```

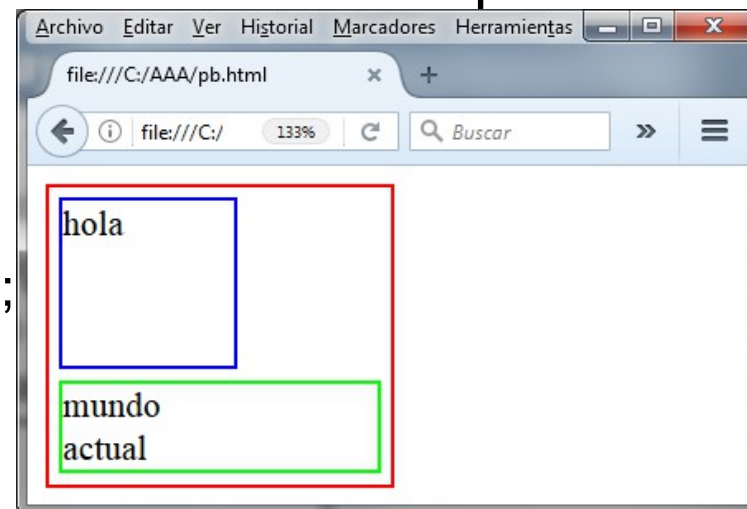
```
<div><p>Hola</p><p>mundo</p><span>Hello</span><span>world</span></div>
```

```
</body> </html>
```



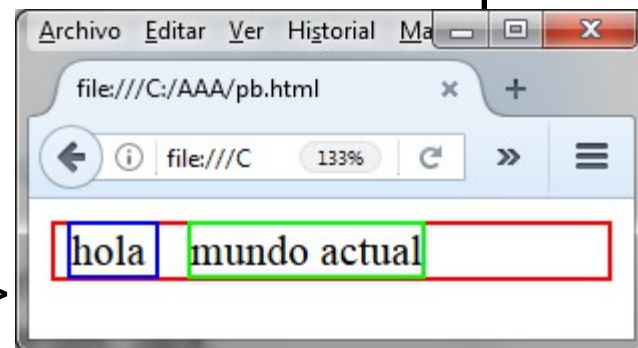
- Especifica el tipo de “caja” utilizada para un elemento
- **block**: se usa una nueva caja que inicia en una nueva línea.
 - Es default para algunos elementos como <div>, <p>, etc.
 - Usa todo el ancho disponible, o lo que se indique con *width*.
 - El alto lo determina su contenido, o lo que se indique con *height*.

```
<!DOCTYPE html>
<html>
<head> <meta charset="utf-8"></head>
<body>
<div style="display: block; border: 2px solid red;
           width:50%; ">
  <div style="display: block; margin: 5px;
           border: 2px solid blue; height:75px;
           width:50%;"> hola </div>
  <div style="display: block; margin: 5px; border: 2px solid lime; ">
    mundo <br> actual </div>
</div>
</body> </html>
```



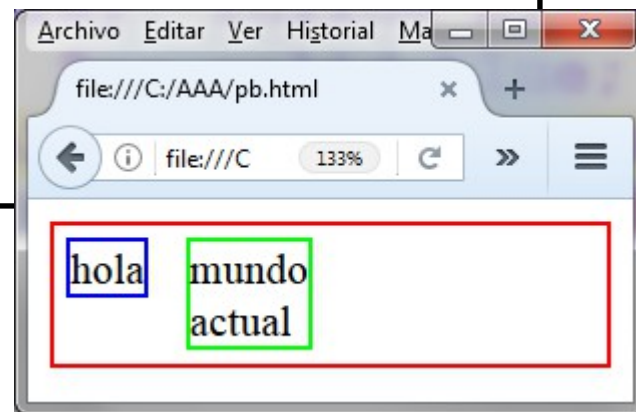
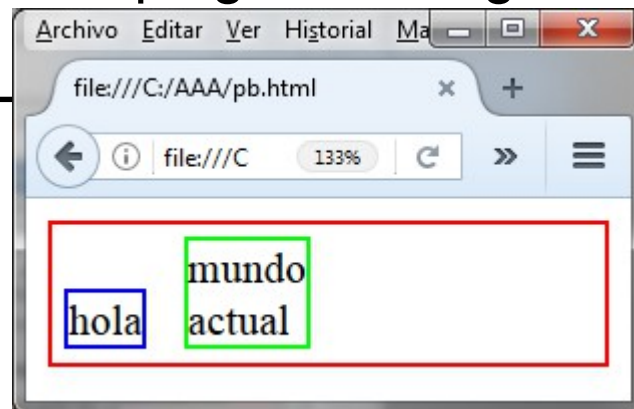
- ***inline***: se despliega contiguo al contenido anterior.
 - Es default para algunos elementos como ``, `<input>`, etc.
 - Usa solo el ancho requerido
 - Las propiedades *margin* y *padding* solo aplican horizontalmente
 - El alto de la línea se puede ajustar con *line-height*

```
<!DOCTYPE html>
<html>
<head> <meta charset="utf-8"></head>
<body>
<div style="display: block; border: 2px solid red;">
  <span style="display: inline; margin: 5px;
    border: 2px solid blue;"> hola </span>
  <span style="display: inline; margin: 5px; border: 2px solid lime;">
    mundo actual </span>
</div>
</body> </html>
```



- ***inline-block***: se usa una nueva caja (*box*) pero desplegada contiguo al contenido anterior (*inline*)

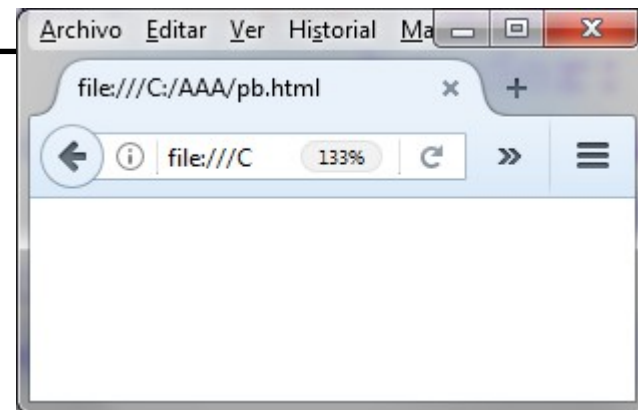
```
<!DOCTYPE html>
<html>
<head> <meta charset="utf-8"></head>
<body>
<div style="display: block; border: 2px solid red;">
  <div style="display: inline-block; margin: 5px;
    border: 2px solid blue;"> hola </div>
  <div style="display: inline-block; margin: 5px; border: 2px solid lime;">
    mundo <br>actual </div>
</div>
</body> </html>
```



vertical-align: top

- ***none***: el contenido no se despliega

```
<!DOCTYPE html>
<html>
<head> <meta charset="utf-8"></head>
<body>
<div style="display: none; border: 2px solid red;">
  <div style="display: inline-block; margin: 5px;
    border: 2px solid blue;"> hola </div>
  <div style="display: inline-block; margin: 5px; border: 2px solid lime;">
    mundo <br>actual </div>
</div>
</body> </html>
```



- Indica si un elemento debe flotar (hacia la izquierda o derecha) y dejar que el resto del contenido se acomode contiguo a él.

```
<!DOCTYPE html>
<html>
<head> <meta charset="utf-8"></head>
<body>
<div>
  
  La imagen flota hacia la izquierda, entonces
  el texto la rodea por la derecha.
</div>
</body> </html>
```

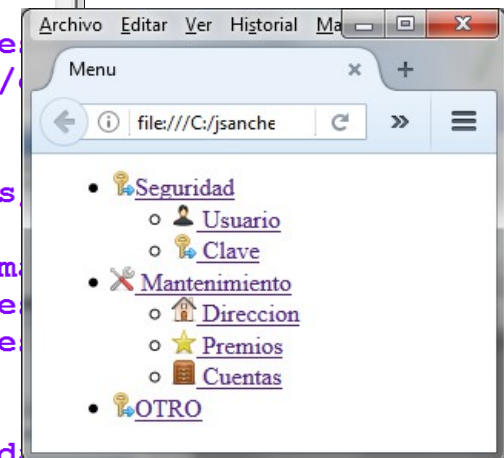
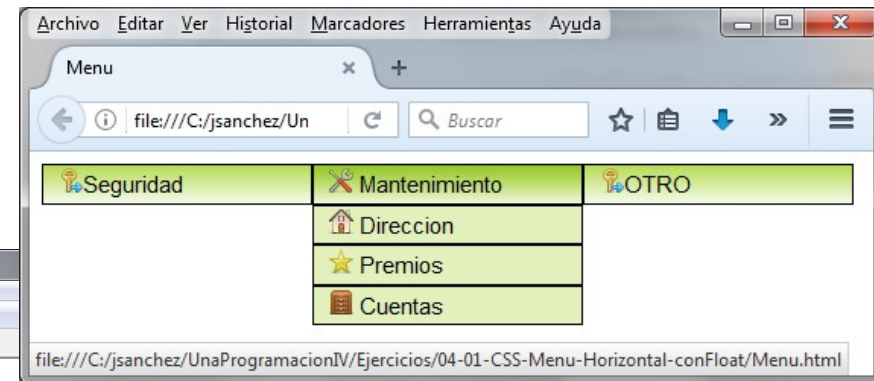


- Indica si un elemento debe flotar (hacia la izquierda o derecha) y dejar que el resto del contenido se acomode contiguo a él.

```
<!DOCTYPE html>
<html>
<head> <meta charset="utf-8"></head>
<body>
<div>
  
  La imagen flota hacia la derecha, entonces
  el texto la rodea por la izquierda.
</div>
</body> </html>
```



- Un menú construido como una lista de links, con sublistas como submenús.

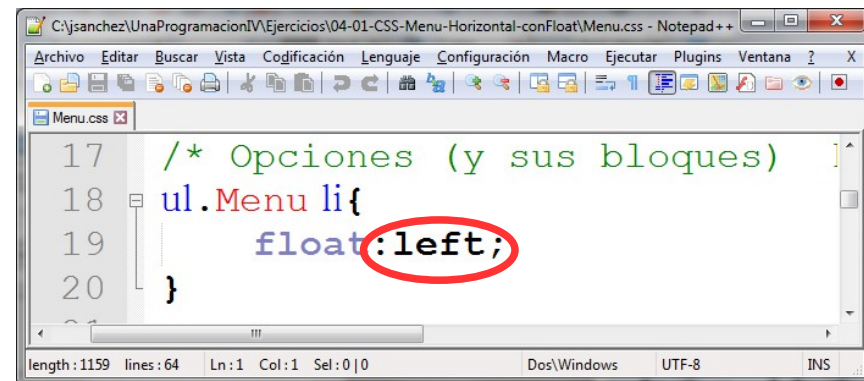
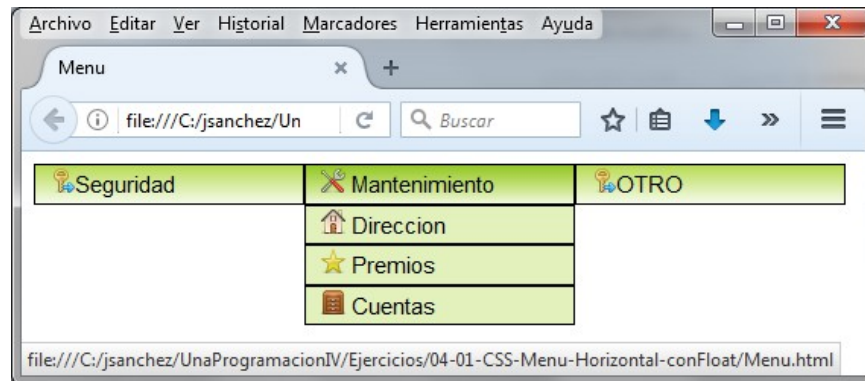
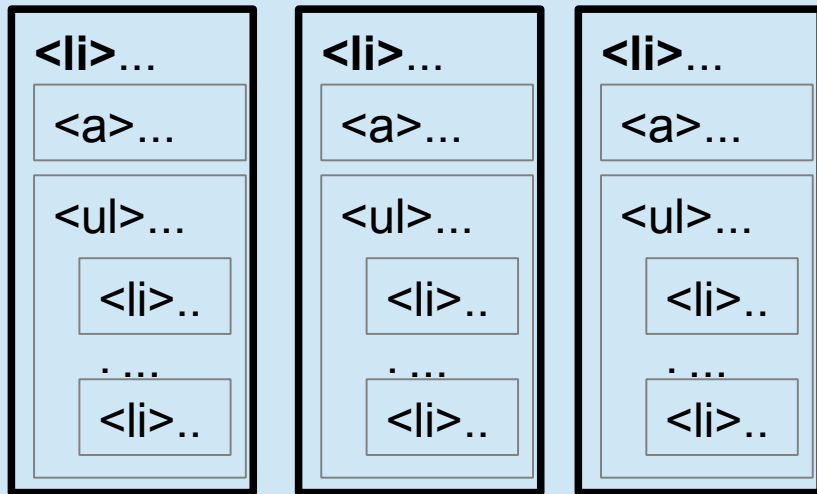


```
8 <ul class="Menu">
9 <li><a href="" title="Seguridad" > <image src="./images/segu
10 <ul>
11 <li><a href="" title="Usuario"><image src="./image
12 <li><a href="" title="Clave"><image src="./images/
13 </ul>
14 </li>
15 <li><a href="" title="Mantenimiento" ><image src="./images
16 <ul>
17 <li><a href="" title="Direccion"><image src="./im
18 <li><a href="" title="Premios"><image src="./image
19 <li><a href="" title="Cuentas"><image src="./image
20 </ul>
21 </li>
22 <li><a href="" title="OTRO" > <image src="./images/segurid
23 </ul>
```

Hyper Text Markup Language file length: 1302 lines: 34 Ln: 24 Col: 22 Sel: 0 | 0 Dos/Windows UTF-8 INS

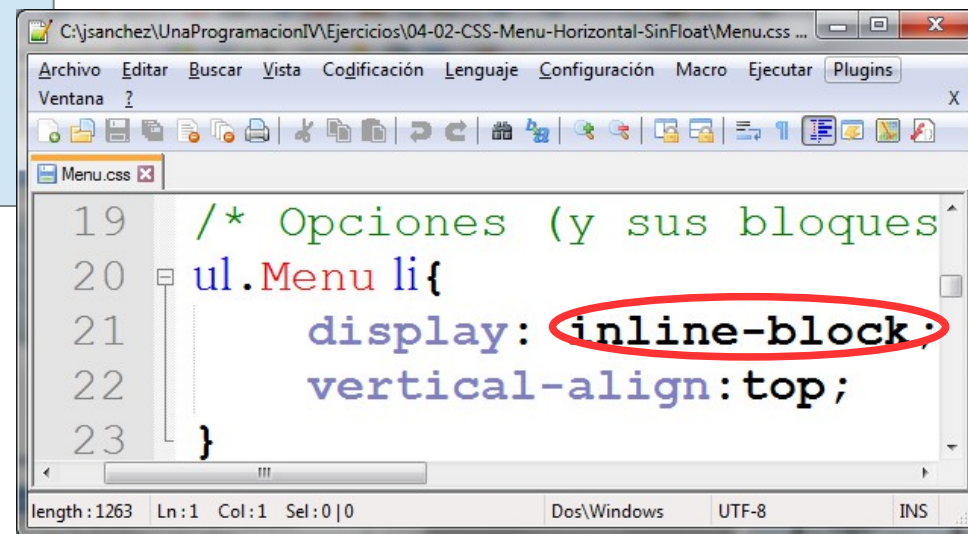
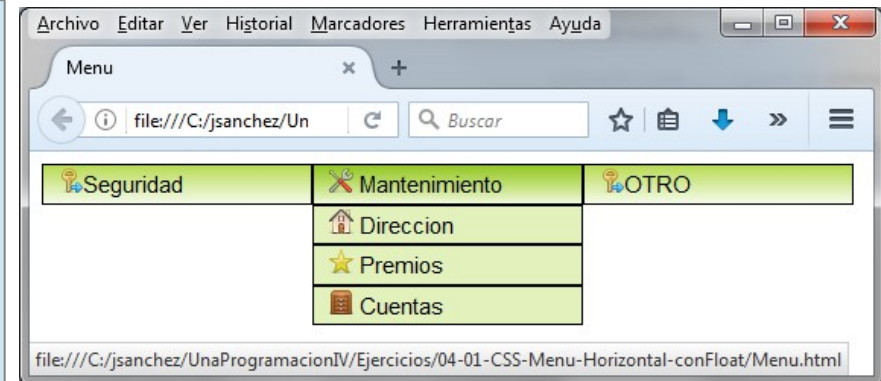
- Cada Item () del menú es un bloque que contiene un enlace (<a>) y una sublista (). Estos bloques flotan a la izq.

<ul class= "Menu" ...



- Los bloques se despliegan contiguos usando inline-block

`<ul class= "Menu" ...`



- Los bloques se despliegan verticalmente usando *block*

`<ul class= "Menu" ...`

`...`

`<a>...`

`...`

`...`

`<a>...`

`...`

`..`

`....`

`...`

`...`

`<a>...`

`...`

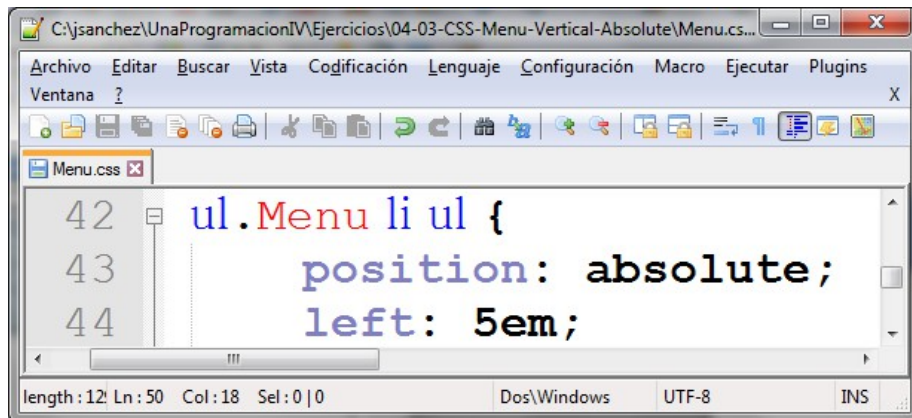
```
C:\AAA\04-03-CSS-Menu-Vertical-Absolute\Menu.css - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro  Ejecutar  Plugins  Ventana  ?
Menu.css x
19  /* Opciones (y sus bloques)
20  ul.Menu li{
21      display: block;
```



```
C:\AAA\04-03-CSS-Menu-Vertical-Absolute\Menu.css - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro  Ejecutar  Plugins  Ventana  ?
Menu.css x
25  ul.Menu li a {
26      display: inline-block;
27      vertical-align: top;
```

```
C:\AAA\04-03-CSS-Menu-Vertical-Absolute\Menu.css - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro  Ejecutar  Plugins  Ventana  ?
Menu.css x
47  ul.Menu li: hover ul{
48      display: inline-block;
49  }
```


- Determina el tipo de posicionamiento de un elemento:
 - *static*: según corresponda por el orden (default)
 - *absolute*: relativo a su contenedor, pero NO considera su espacio original
 - *relative*: relativo a su contenedor, pero considera su espacio original
 - *fixed*: relativo a la pantalla del browser



A screenshot of a text editor window titled 'C:\jsanchez\UnaProgramacionIV\Ejercicios\04-03-CSS-Menu-Vertical-Absolute\Menu.cs...'. The editor shows a CSS file named 'Menu.css' with the following code:

```
42 ul.Menu li ul {  
43     position: absolute;  
44     left: 5em;  
}
```

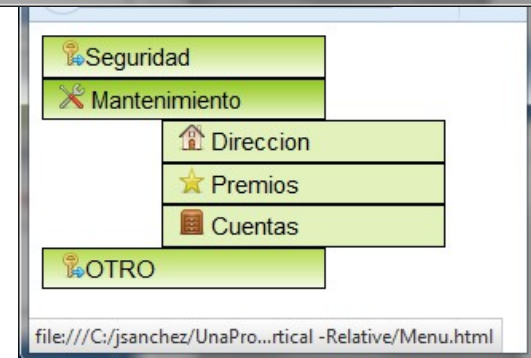
The status bar at the bottom indicates 'length: 12 Ln: 50 Col: 18 Sel: 0 | 0', 'Dos\Windows', 'UTF-8', and 'INS'.



A screenshot of a text editor window titled 'C:\jsanchez\UnaProgramacionIV\Ejercicios\04-04-CSS-Menu-Vertical-Relative\Menu.cs...'. The editor shows a CSS file named 'Menu.css' with the following code:

```
42 ul.Menu li ul {  
43     position: relative;  
44     left: 5em;  
}
```

The status bar at the bottom indicates 'length: 12 Ln: 43 Col: 27 Sel: 0 | 0', 'Dos\Windows', 'UTF-8', and 'INS'.



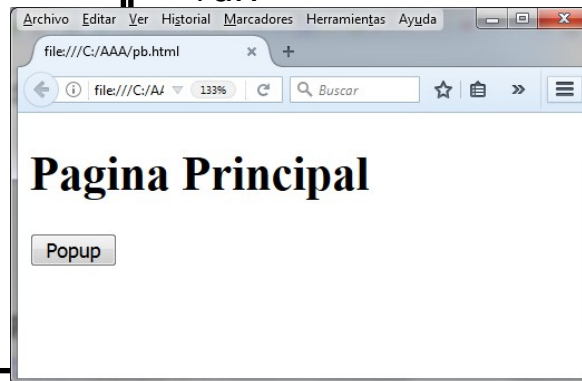
- Indica la “profundidad” del elemento. Elementos con valores mas grandes quedan encima de otros con valores menores.

```
<!DOCTYPE html>
<html>
<head> <meta charset="utf-8"></head>
  <style>
    img {
      position: absolute;
      left: 40px;
      top: 10px;
      z-index: -1;
    }
    * {color: lime;}
  </style>
</head>
<body>
  <h1>Esto es un encabezado</h1>
  
  <p>La imagen queda detras del texto por tener un z-index de -1.</p>
</body></html>
```

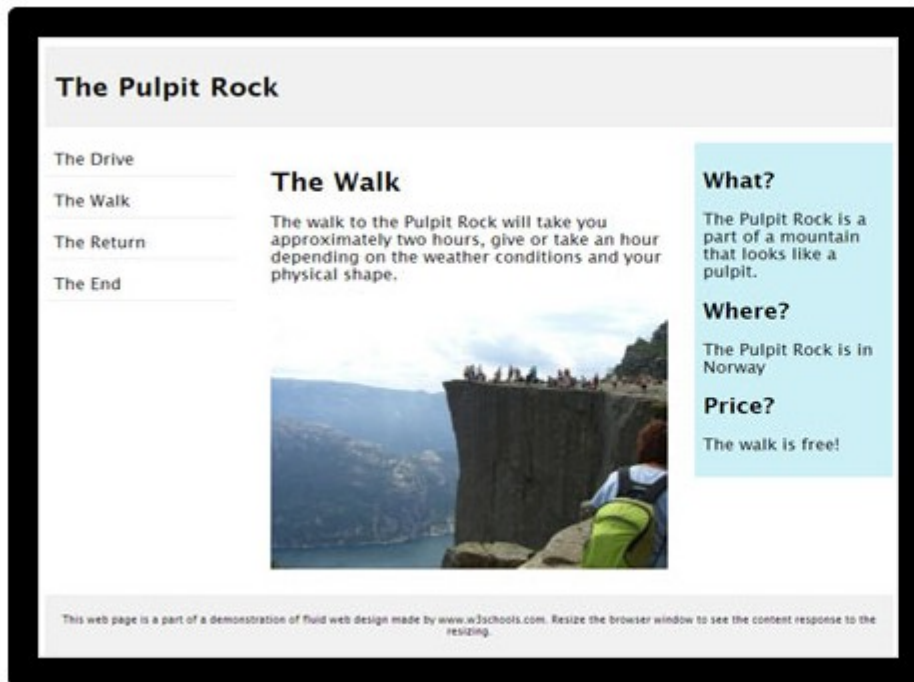


```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"></head>
  <style>
    .overlay{
      display: none; position: absolute;
      top: 0%; left: 0%;
      width: 100%; height: 100%;
      background-color: lightgray;
      z-index:1; opacity:.70;
    }
    .popup {
      display: none; position: absolute;
      top: 10%; left: 25%;
      width: 50%; height: 50%;
      padding: 5px;
      border: 5px solid lightgray;
      background-color: white;
      z-index:2;
    }
  </style>
</head>
```

```
<body>
  <div>
    <h1>Pagina Principal</h1>
    <button onclick =
      "document.getElementById('o').style.display='block';
      document.getElementById('p').style.display='block';">
      Popup</button>
    </div>
    <div id="o" class="overlay"></div>
    <div id="popup" class="popup">
      <h2>Esto es un pop-up</h2>
      <button onclick =
        "document.getElementById('o').style.display='none';
        document.getElementById('p').style.display='none';">
        Regresar</button>
    </div>
```



- Un diseño sensible hace que una página web se vea bien en todos los dispositivos (desktops, tables, teléfonos, ...)
- Consiste en usar HTML y CSS para ajustar (encontrar, alargar, ocultar, etc.) el contenido de la página a cualquier pantalla.



- “Consultar” las características del medio de despliegue (ej. ancho de la pantalla) para aplicar distintos estilos según dichas características

```
<!DOCTYPE html>
<html> <head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<style>
  body {
    background-color: lightgreen;
  }
  @media only screen and (max-width: 500px) {
    body {
      background-color: lightblue;
    }
  }
</style>
```

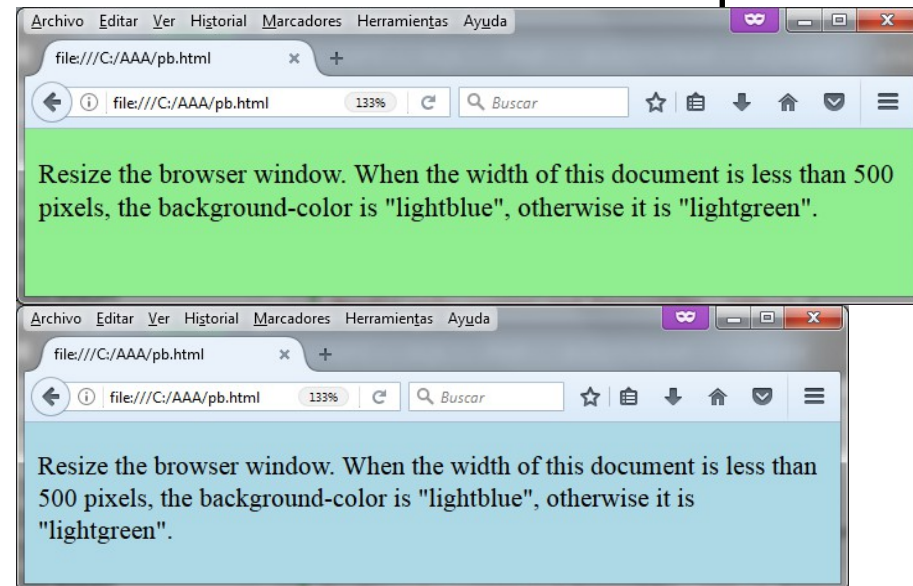
```
</head>
```

```
<body>
```

```
  <div>Resize the browser window. When the width of this document is less than 500
    pixels, the background-color is "lightblue", otherwise it is "lightgreen".</div>
```

```
</body>
```

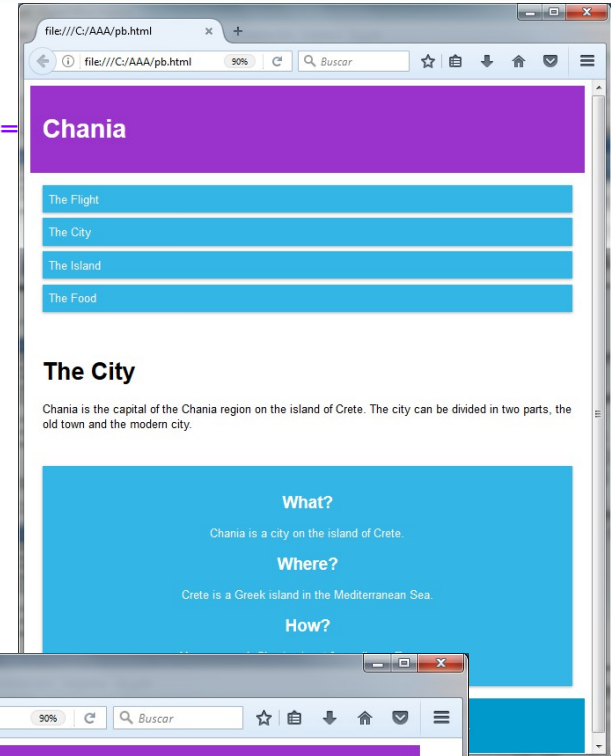
```
</html>
```



Media Queries (ejemplo)

35

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta name="viewport" content="width=device-width, initial-scale=1" />
5 </head>
6 <body>
7   /* Phones <768 by default container 100%, columns 100% */
8   .container {max-width: 100%;}
9   .col-izq {width: 100%;}
10  .col-cen {width: 100%;}
11  .col-der {width: 100%;}
12
13  @media screen and (min-width: 768px) { /* Tablets 768-992 */
14    .container {max-width: 90%;}
15    .col-izq {width: 25%;}
16    .col-cen {width: 40%;}
17    .col-der {width: 35%;}
18  }
19
20  <div class="container">
21    <div class="header">
22      <div class="row">
23        <div class="col-izq menu">
24          <div class="col-cen">
25            <div class="col-der">
26              <div class="header">
27                Chania
28              </div>
29              <div class="row">
30                <div class="col-izq">
31                  The Flight
32                  The City
33                  The Island
34                  The Food
35                </div>
36                <div class="col-cen">
37                  The City
38                  Chania is the capital of the Chania region on the island of Crete. The city can be divided in two parts, the old town and the modern city.
39                </div>
40                <div class="col-der">
41                  What?
42                  Chania is a city on the island of Crete.
43                  Where?
44                  Crete is a Greek island in the Mediterranean Sea.
45                  How?
46                  You can reach Chania airport from all over Europe.
47                </div>
48              </div>
49            </div>
50          </div>
51        </div>
52      </div>
53    </div>
54  </div>
```



```
C:\jsanchez\UnaProgramacionIV\Ejercicios\04-07-CSS-Responsive-bootstrap\dist\css\bootstrap.css - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro  Ejecutar  Plugins  Ventana  ?
bootstrap.css
1777 @media (min-width: 768px) {
1778   .col-sm-1, .col-sm-2, .col-sm-3, .col-sm-4, .
   .col-sm-5, .col-sm-6, .col-sm-7, .col-sm-8, .
   .col-sm-9, .col-sm-10, .col-sm-11, .col-sm-12 {
1779     float: left;
1780   }
1781   .col-sm-12 {
1782     width: 100%;
1783   }
1784   .col-sm-11 {
```

```
*C:\jsanchez\UnaProgramacionIV\Ejercicios\04-07-CSS-Responsive-bootstrap\pagina.htm - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro  Ejecutar  Plugins  Ventana  ?
pagina.htm
43 <body>
44   <div class="container">
45     <div class="page-header header">
48     <div class="row">
49       <div class="col-xs-12 col-sm-3 menu">
57       <div class="col-xs-12 col-sm-6">
61       <div class="col-xs-12 col-sm-3">
71     </div><!-- /row -->
72     <div class="footer">
75   </div> <!-- /container -->
76 </body>
```

