

Práctica 1



Sesión 3

Objetivos

- Comprender cómo HTML5 separa la semántica de los contenidos del aspecto de la página.
- Entender el concepto de regla de CSS.
- Conocer la filosofía de trabajo de CSS y su funcionalidad básica.
- Ser capaz de dar formato a un documento web, plasmándolo en reglas CSS.
- Conocer los fundamentos del diseño adaptable

Cascading Style Sheets (CSS)

HTML5 ha sido diseñado para crear páginas web desde el punto de vista lógico, no visual (a pesar de que en las versiones anteriores de HTML había elementos que tenían connotaciones eminentemente visuales y que han sido eliminados en la versión actual). Así, en lugar de tener elementos especiales que dieran formato al documento, se optó en su momento por crear un nuevo lenguaje que presentara prestaciones para el formateo de documentos HTML, dando lugar así al *Cascading Style Sheets (Hojas de estilo en cascada)* y diferenciando la semántica del aspecto.

Por tanto, CSS se encarga de controlar el estilo y el diseño de los documentos HTML. Actualmente, la recomendación existente es la 3 (CSS3).

CSS trabaja mediante la creación de reglas que se aplican a ciertas partes de la página (a un elemento concreto, a todos los de un tipo concreto o a todos los que compartan ciertas características particulares) y tienen la forma atributo:valor. Por ejemplo:

```
1      <!DOCTYPE HTML>
2      <html lang = "en">
3          <head>
4              <title>cssColors.html</title>
5              <meta charset = "UTF-8" />
6              <style type = "text/css">
7                  body {
8                      background-color: yellow;
9                  }
```

En este ejemplo vemos cómo se indica qué aspecto tendrán en el navegador ciertos elementos. La especificación del estilo se ha incrustado directamente en la página web mediante el elemento *style* y el valor "*text/css*" del atributo *type*. Las línea 7 indica que el elemento que se quiere modificar es *body* y seguidamente se establece que el color de fondo para ese elemento será amarillo.

The diagram illustrates the components of a CSS rule. It shows a selector 'h1' in a green rounded rectangle, followed by a declaration '{color:blue; font-size:12px;}' in a green rounded rectangle. Above the selector is the label 'Selector'. Above the declaration is the label 'Declaration'. Below the declaration, arrows point to the individual parts: 'color' is labeled 'Property', 'blue' is labeled 'Value', 'font-size' is labeled 'Property', and '12px' is labeled 'Value'.

```
graph LR; S(h1) --- D1({color:blue; font-size:12px;}); S --- S_label[Selector]; D1 --- D_label[Declaration]; D1 --> P1[color]; P1 --- P_label[Property]; D1 --> V1[blue]; V1 --- V_label[Value]; D1 --> P2[font-size]; P2 --- P_label; D1 --> V2[12px]; V2 --- V_label;
```

```
/*cambio de color del fondo de la página*/
body{
    background-color: blue;
}
```

CSS – Práctica 2 de Tecnologías Web – 2/26

Por otro lado, comentar que la gran mayoría de elementos de HTML tienen un atributo denominado *style* donde se pueden incluir directamente reglas CSS, aunque su uso no está totalmente recomendado por los inconvenientes que aparecen al querer realizar cambios de estilo:

```
<tr>
    <td>Amarillo</td>
    <td style="background-color: yellow;"><br /></td>
</tr>
```

En este otro ejemplo vemos cómo se ha cambiado el fondo de una celda para ponerlo amarillo.

Si quisiéramos añadir un estilo a todos los elementos de un tipo particular (por ejemplo, no que todos los párrafos de una página concreta que tengan un color determinado, sino sólo unos cuantos de ellos), debemos emplear los atributos globales *id* o *class* de los elementos de HTML para crear reglas sobre ellos. Por ejemplo:

```
1  <!DOCTYPE HTML>
2  <html lang = "en">
3  <head>
4      <title>Ejemplo de aplicación de estilos mediante id y class</title>
5      <meta charset = "UTF-8" />
6      <style type = "text/css">
7          p {
8              background-color: lightblue;
9              color: blue;
10         }
11         #modificado {
12             background-color: black;
13             color: white;
14         }
15         .alternativo {
16             background-color: blue;
17             color: lightblue;
18         }
19         section p {
20             background-color: green;
21             color: white;
22         }
23     </style>
24 </head>
25 <body>
26 <h2>Selectores CSS</h2>
27 <p id = "modificado"> Este párrafo tiene un estilo modificado. </p>
```

```

28      <p> Este párrafo tienen un estilo estándar </p>
29      <p class = "alternativa"> Este párrafo tiene un estilo alternativo. </p>
30      <p> Y este es de nuevo estándar. </p>
31      <p class = "alternativa"> Otra vez alternativo</p>
32      <section>
33          <p> Este párrafo depende de la sección </p>
34      </section>
35  </body>
36  </html>

```

Se han creado en este ejemplo tres estilos para tres situaciones particulares:

- En la línea 7, se ha especificado el color de fondo y el de fuente para los párrafos.
- En la línea 11, aplicable sólo para el elemento que tenga el *id modificado* (línea 23), se han indicado otros colores. Nótese que en estos casos, se especifica un # y el identificador de un elemento seguidamente.
- En la línea 15 se han creado unas reglas para todos aquellos elementos que tengan *alternativa* como valor del atributo *class*. En este caso, se precede la clase mediante un punto (.).
- En la línea 19 se ha creado una regla específicamente para los párrafos que estén dentro de las secciones (selector descendiente). Los dos elementos están separados por un espacio en blanco, indicando que existe una relación de descendencia, sin considerar la profundidad del anidamiento. Si queremos indicar que la relación es de descendencia directa (de padre a hijo), se hace mediante el símbolo > (mayor): *section > p {...};*.

Estos ejemplos anteriores corresponden a cuatro tipos de selectores, pero existe una gran cantidad de ellos. Una lista detallada se puede encontrar en http://www.w3schools.com/cssref/css_selectors.asp.

La tercera forma en que se puede añadir código CSS a una página es mediante un documento externo donde se incluyen las reglas. En el ejemplo anterior, podríamos extraer el contenido del elemento *style* de la cabecera del documento y crear un fichero que se llamara *ejemploPárrafos.css*:

```

p {
    background-color: lightblue;
    color: blue;
}
#modificado {
    background-color: black;
    color: white;
}
.alternativo {
    background-color: blue;
    color: lightblue;
}

```

Como vemos, no ha sido necesario introducir las reglas dentro del elemento *style*. Por otro lado, el fichero html del ejemplo quedaría como sigue:

```
1    <!DOCTYPE HTML>
2    <html lang = "en">
3    <head>
4        <title>Ejemplo de aplicación de estilos mediante id y class</title>
5        <meta charset = "UTF-8" />
6        <link rel = "stylesheet" type = "text/css" href = "ejemploParrafos.css" />
7    </head>
8    <body>
9        <h2>Selectores CSS</h2>
10       <p id = "modificado"> Este párrafo tiene un estilo modificado. </p>
11       <p> Este párrafo tienen un estilo estándar </p>
12       <p class = "alternativo"> Este párrafo tiene un estilo alternativo.</p>
13       <p> Y este es de nuevo estándar. </p>
14       <p class = "alternativo"> Otra vez alternativo</p>
15    </body>
16    </html>
```

Se incluye, como se puede apreciar, un elemento *link*, con atributos que nos permitirán indicar que el enlace va a ser una hoja de estilos CSS y el nombre de la misma. Esta forma de incluir estilos en una página web es la preferida porque nos permite aplicar estilos a páginas diferentes, teniendo que modificar sólo un fichero en caso de querer realizar algún cambio en el estilo.

En este punto cabe destacar que las propiedades establecidas a varios elementos se heredan a todos los elementos que están incluidos en ellos (pasan en cascada de padres a hijos). En caso de que haya conflictos de declaraciones de elementos, las reglas más específicas se aplican antes que las más generales (por ejemplo, el selector *header p* es más específico que *p*). En caso de tener la misma especificidad, la regla que esté antes en el fichero .css es la que se aplica.

Formato de texto

En primer lugar, indicar que los colores se suelen especificar en hexadecimal, indicando la cantidad de rojo, verde y azul, mediante dos dígitos para cada color. Así, por ejemplo:

```
h1 {
    color: #FF0000;
    background-color: #FFFF00;
}
```

Estamos indicando que el color para el titular *h1* tendrá un valor *FF* para rojo, y *0* para el verde y en azul, por tanto, es rojo puro. Por el contrario, el color de fondo, tendrá un valor *FF* de rojo y el mismo para el verde, sin embargo, no habría azul. También se pueden expresar mediante su valor decimal en rgb, por medio de la función *rgb(cantidad rojo, cantidad verde, cantidad azul)*, o bien

mediante el nombre del color (http://www.w3schools.com/cssref/css_colornames.asp).

Existen varios atributos que nos permitirán dar formato al texto a nuestro gusto. Aquí se muestran algunos de ellos, así como sus valores más comunes (http://www.w3schools.com/css/css_text.asp):

- color**: propiedad empleada para establecer el color de un elemento. El valor es un color, expresado en hexadecimal, en rgb (rgb(255,0,0)) ó con su nombre (red, white, blue, etc.).
- font-style**: especifica el tipo de estilos de la fuente tipográfica (*italic*, *normal* y *oblique*).
- font-weight**: necesario para indicar el grosor que se aplica al texto (*bold* y *normal*).
- text-decoration**: normalmente se emplea para añadir una línea al texto (*underline*, *overline*, *line-through*, *blink*, *none*).
- text-align**: utilizado para establecer el alineamiento del texto (center, left, right y justify)
- font-family**: se emplea para indicar la fuente del texto (*p{font-family:"Times New Roman", Times, serif;}*). Véase http://www.w3schools.com/css/css_font.asp y http://www.w3schools.com/cssref/css_websafe_fonts.asp para más detalles.
- font-size**: mediante un porcentaje (la forma más habitual y segura), se especifica cuánto de grande se mostrará el texto, con respecto al tamaño base (*body {font-size:100%;}* ó *body{font-size:2em;}* – En este último caso se está indicando que se duplique el tamaño de la fuente establecida en este momento).
- text-transformation**: para especificar que el texto sea en mayúscula o minúscula (*uppercase*, *lowercase*, *capitalize*).
- text-indent**: sangrado de la primera línea de un texto. Se suele indicar un valor en píxels (50px, por ejemplo).
- Otras propiedades: *direction*, *letter-spacing*, *line-height*, *text-shadow*, *vertical-align*, *white-space* ó *word-spacing*.

Ejercicio:

Crea un documento HTML5 y un fichero CSS para probar los atributos anteriores.

Efectos de fondo

Los efectos relacionadas con el fondo de un elemento se pueden modificar con el siguiente conjunto de propiedades:

- background-color**: se especifica un color mediante su nombre o mediante su valor en hexadecimal. También se puede usar *rgb(255,0,0)*, para poder hacerlo en decimal.
- background-image**: establece una imagen como fondo. El valor de esta propiedad es *url('fichero.jpg')*.
- background-repeat**: indica la forma en que se repite la imagen. Por defecto, se hace tanto horizontalmente como verticalmente. Pero, se puede indicar que sólo se haga horizontalmente (*repeat-x*), verticalmente (*repeat-y*) o no se repita (*no-repeat*).
- background-attachment**: establece si la imagen de fondo es fija o se mueve con el resto de la página (*fixed*, *scroll* ó *inherit* -la forma de proceder se hereda de su elemento padre).
- background-position**: permite establecer dónde situar la imagen (combinaciones de *left*, *right* y

center, con *top*, *bottom* y *center*): `body {background-position: left right;}`.

De forma resumida, y mediante una única propiedad (*background*), se pueden reunir varias propiedades. Por ejemplo:

```
body {background:#ffffff url('img_tree.png') no-repeat right top;}
```

Cuando usemos esta propiedad, los valores se indican según el orden establecido anteriormente.

Ejercicio:

Añade nuevos elementos al documento HTML5 anterior y al fichero CSS que has creado, para probar los atributos relacionados con los efectos de fondo.

Enlaces

A los enlaces se les puede modificar su fuente, el color, el fondo, etc., al igual que cualquier otro texto, aunque en este caso podremos cambiar su aspecto según hayan sido visitados, si se ha pasado el cursor por ellos o si se ha seleccionado. En este caso, se emplean los siguientes selectores: *a:visited*, *a:link*, *a:hover*, y *a:active*:

```
a:link {color:#FF0000;text-decoration:none;}
a:visited {background-color:#FFFF85; text-decoration:none;}
a:hover {text-decoration:underline;}
a:active {text-decoration:underline;}
```

Ejercicio:

Prueba los atributos anteriores para enlaces.

Formato de listas

Podemos cambiar los marcadores, tanto de las listas ordenadas, como de las no ordenadas, incluso establecer una imagen como marcador para las segundas. Para lo primero, se emplea la propiedad *list-style-type*, que puede tomar los valores *circle* y *square*, si la lista es no ordenada; mientras que si es ordenada, puede tomar *upper-roman* ó *lower-alpha*. También puede tomar el valor *none* para indicar que no se desea ninguno. Y poner una imagen, para lo cual se usa la propiedad *list-style-image* en combinación con el *url* de la imagen que queremos emplear. Veamos un ejemplo:

```
1    ul { list-style-image: url('flecha.gif'); }
2    ol.a {list-style-type: upper-roman;}
3    ol.b {list-style-type: lower-alpha;}
```

En la línea 1, se pone como marcador de todas las listas ordenadas la imagen flecha.gif. En la línea 2 se indica que, para la lista ordenada con *class="a"*, cada ítem se numerará con números romanos en mayúsculas. En la línea 3 se indica que las listas ordenadas con *class="b"* se numerarán con letras minúsculas.

Una última propiedad es *list-style-position*, con la que especificamos si los marcadores deben estar dentro o fuera del texto (http://www.w3schools.com/cssref/pr_list-style-position.asp), por lo que puede tomar *outside* (valor por defecto) ó *inside*.

Finalmente, comentar que también se puede emplear una única propiedad para establecer el estilo de las listas, que es *list-style* y el orden de valores corresponde a los atributos *list-style-type*, *list-style-position* y *list-style image*.

Ejercicio:

Añade dos lista, una ordenada y otra no ordenada, y modifica el fichero CSS para probar los atributos anteriores.

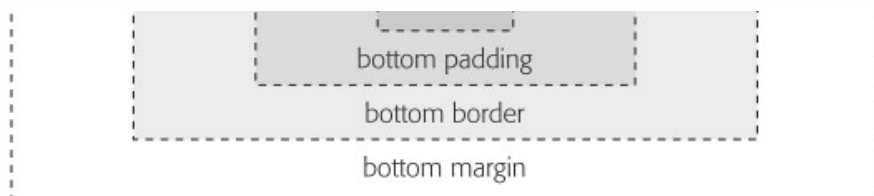
El modelo de caja de CSS

Todos los elementos en HTML se pueden considerar cajas. El modelo de caja asume la existencia de una "caja" que envuelve elementos y consta de:

- Márgenes (*margin*): permite dejar un espacio sin ocupar alrededor del borde. No queda afectado por el color de fondo y es transparente.
- Bordes (*border*): el borde del elemento propiamente dicho. Sí queda afectado por el color de fondo de la caja.
- Área alrededor del contenido (*padding*): libera un espacio alrededor del contenido. Sí queda afectado por el color de fondo de la caja.
- Contenido (*content*): el contenido de la caja donde aparece texto e imágenes.

El ancho (*width*) total de un elemento es igual al tamaño del contenido, más el área de la izquierda y de la derecha (*padding*), más el ancho del borde de la izquierda y el de la derecha, más los márgenes izquierdo y derecho. Análogamente, su altura (*height*) se calculará como la suma de la altura del contenido, más el área superior e inferior, más el ancho de los bordes superior e inferior, más los márgenes inferior y posterior.

El ancho y alto se pueden indicar mediante píxeles (5px) o porcentajes (80%) (echa un vistazo a http://www.w3schools.com/cssref/css_units.asp, donde podrás ver las unidades de medida que existen y a <http://kyleschaeffer.com/user-experience/css-font-size-em-vs-px-vs-pt-vs/> para una explicación de las diferencias). También se puede poner *auto* y en ese caso la caja del elemento se expandirá hasta el máximo que pueda según las cajas de los otros elementos donde esté incluida.



En cuanto a las propiedades que definen el modelo de caja, algunas de ellas son:

- Bordes (http://www.w3schools.com/css/css_border.asp):
 - *border-style*: *none, dashed, solid, double, groove, ridge, inset, outset* (puede tener de uno a cuatro valores coincidiendo con los cuatro bordes).
 - Bordes individuales: *border-top-style, border-right-style, border-bottom-style, border-left-style*.
 - *border-width*: especificado en píxeles, o mediante *thin, medium* ó *thick*.
 - *border-color*: como siempre, mediante un nombre de color, en rgb ó en hexadecimal.
 - *border*: propiedad para especificar todo → tamaño, estilo, color (*border:5px solid red;*).
- Márgenes (http://www.w3schools.com/css/css_margin.asp):
 - *margin-top, margin-bottom, margin-right, margin-left* (tamaño en píxeles, porcentajes, etc.).
 - *margin*: en una única propiedad (*top, right, bottom, left*)
- Área alrededor del contenido (*padding*) (http://www.w3schools.com/css/css_padding.asp):
 - *padding-top, padding-bottom, padding-right, padding-left* (tamaño en píxeles, porcentajes, etc.).
 - *padding*: en una única propiedad (*top, right, bottom, left*)

Ejercicio:

Formatea una tabla, cambiando bordes, márgenes y áreas alrededor del contenido.

Un elemento puede ser tipo *block*, *inline* o *inline-block*:

- Los primeros son aquellos que toman todo el ancho disponible y generan una nueva línea, tanto antes como después de ellos (por ejemplo, *h1* y *p*).
- Segundo, los elementos *inline* sólo toman el ancho que sea necesario en función de su contenido, y no fuerzan nuevas líneas (por ejemplo, *a* ó *span*). No podemos especificarle altura (*height*) ni anchura (*width*), pues estos parámetros se toman automáticamente en función de la altura y anchura de su contenido. En cuanto a los estilos *margin* y *padding*, sólo respetan los lados izquierdo y derecho, pero no los lados superior e inferior.
- Por último, los elementos *inline-block* siguen el mismo patrón anterior, con la diferencia de que sí respetan tanto las indicaciones altura y anchura, como los *margin* y *padding* en cualquier dirección.

Se puede cambiar la forma en que un elemento se muestre mediante el atributo *display*, el cual puede tomar los valores *block*, *inline*, o *inline-block*:

```
li {
```

```
        display:inline;
    }
    span {
        display:block;
    }

    p {
        display:inline-block;
    }
```

Agrupamiento de selectores

Se pueden agrupar varios selectores, para aplicarles el mismo estilo:

```
h1
{ color:green; }
h2
{ color:green; }
p
{ color:green; }
<!-- O lo que es lo mismo -->
h1,h2,p
{ color:green; }
```

Y encadenar varios selectores:

```
p
{color:blue;
 text-align:center;
}
.marked
{ background-color:red; }
.marked p
{color:white;}
```

Habrán un estilo para párrafos, otros para aquellos que tengan como valor del atributo *class*, *marked*, y otros para párrafos con *marked* como atributo.

Flujo de elementos

Se emplea para ubicar elementos según un diseño del documento que hayamos hecho. Si no se indica lo contrario, los elementos se irán apilando como bloques unos a continuación de otros. Para poder cambiar ese diseño se emplea el atributo *float*, el cual toma como atributos *left* o *right*. Así, al indicarle que se vaya a la izquierda, o a la derecha, el elemento se moverá a uno de esos

lados tanto como pueda. Los elementos que van después del elemento flotante lo cubrirán, mientras que los que vayan antes, no quedarán afectados:

```
img
{
  float:right;
}
```

En este caso, las imágenes irán todas a la derecha y si hay texto, este quedará a su izquierda. Para volver al diseño por defecto (bloques unos a continuación de otros), se emplea el atributo *clear*, el cual puede tomar los valores *left*, *right*, *both*, *none* e *inherit*, indicando qué justificación es la que se libera.

Ejercicio:



Escribe un documento HTML que contenga un elemento header, tres section y un footer. Escribe también un fichero css que sitúe una cabecera en la parte superior del documento; debajo, tres secciones de igual tamaño, una al lado de otra, y en la parte final, ocupando todo el ancho, un pie de documento. Visualiza los bordes de cada elemento y deja espacio entre cada elemento.

Diseño adaptable ('responsive')

Decimos que una página web es adaptable o 'responsive' si ha sido diseñada e implementada de forma que la disposición de sus contenidos se adapta al tamaño de pantalla del dispositivo que se utiliza para acceder a ella (teléfono, tableta, sobremesa, etc.). Sería posible diseñar páginas completamente diferentes según el tipo de dispositivo empleado y acceder la más adecuada en cada caso, pero esto aumentaría los costes y la redundancia de código.

CSS3 ofrece soporte para especificar cómo deben formatearse y organizarse los elementos de una página –es decir, qué estilos se deben aplicar– según el tamaño del visor del navegador. En general, seguiremos las siguientes recomendaciones (por móvil nos referimos genéricamente a un dispositivo con un tamaño de pantalla reducido, en posición horizontal o vertical):

- Ajuste de columnas: Evitaremos contenidos en varias columnas en las versiones para móvil, desactivando esta selección de formato en estos dispositivos para que se muestren una a continuación de otra, en lugar de en paralelo.
- Elementos flotantes: Evitaremos elementos flotantes en las versiones móviles.
- Anchura de contenidos flexible: Definiremos la anchura de los contenidos en tamaños relativos a la ventana del navegador.
- Reducir espacios en blanco: Reduciremos los valores de *margin* y *padding* en las versiones móviles.
- Ajustar tamaños de letra: Seleccionaremos distintos tamaños de letra según el dispositivo.
- Disposición de los menús de navegación: Cambiaremos la forma en que se presentan los menús de navegación. Por ejemplo, una barra de navegación ocupando toda la parte superior de la ventana puede ser difícil de utilizar en un dispositivo móvil. En su lugar, será preferible un botón con un menú desplegable de opciones.
- Ocultación de contenido: Podemos ocultar parte de los contenidos en las versiones móviles.
- Utilización de imágenes de fondo: Si tenemos un fondo de pantalla que queremos que se

muestre completo en la página, será necesario utilizar versiones de las imágenes con diferente tamaño.

Estilos dependientes del tamaño de ventana

Para realizar un diseño adaptable necesitamos conocer en primer lugar cómo podemos definir estilos dependientes del tamaño de ventana del navegador. En CSS tenemos instrucciones para obtener esta información del propio navegador, conocidas como ‘media queries’. Por ejemplo, podemos activar la hoja de estilos *small.css* cuando la ventana del navegador es menor o igual a 480 píxeles de la siguiente forma:

```
<link href="css/small.css" rel="stylesheet" media="(width: 480px)">
```

De forma similar, podemos utilizar la consulta (*min-width: 720px*) para activar la hoja de estilos cuando la ventana del navegador tiene una anchura igual o mayor a 720 píxeles.

También es posible utilizar la directiva *@import* en lugar de la etiqueta *link*. Esta directiva debe utilizarse al principio de la hoja de estilos.

```
@import url(css/small.css) (width: 480px);
```

Una última opción es emplear una consulta dentro de la hoja de estilos, agrupando un conjunto de especificaciones de formato:

```
@media (width: 480px) {  
  body {  
    font-size: small  
  }  
  .style1 {  
    /*propiedades de la clase style1 */  
  }  
}
```

Normalmente utilizaremos los siguientes valores habituales de límites de tamaño (llamados ‘breakpoints’):

- Móvil: *width* <= 480
- Tablet: 481 <= *width* <= 768
- Monitor: *width* > 768

Para evitar que en un dispositivo móvil pueda hacerse zoom en nuestra página, definimos la siguiente etiqueta en la cabecera de nuestra página:

```
<meta name="viewport" content="width=device-width">
```

Bloques flexibles

Aunque utilicemos los límites de tamaño anteriores, no es conveniente utilizar valores absolutos de tamaño o posicionamiento de los elementos de la página. Por ello, emplearemos valores de tamaño y posición relativos (con %).

Los bloques flexibles permiten dividir de forma elástica el espacio de la pantalla, en el sentido de que los elementos no tienen un valor o una posición determinado y pueden reorganizarse automáticamente dentro de la misma (pueden “fluir”). Existen diversas formas de implementar bloques flexibles en CSS; las principales están basadas en el uso de elementos *float* y de elementos *flexbox*. Veremos el primero de ellos. Para dividir los bloques de la cuadrícula se utilizan las marcas de contenedores de texto explicadas anteriormente `<section>`, `<header>`, `<footer>`, etc., así como la marca de división genérica `<div>`.

Por ejemplo, podemos crear una disposición de texto en dos columnas con reorganización automática de la siguiente forma: (redimensiona el tamaño de la pantalla para ver cómo cambian los estilos aplicados)

```
/* CSS: responsive.css */
@media (min-width: 481px) {
  .columns {
    width: 100%;
    /* same as 100% */
  }
  .columns:after {
    content: "";
    display: table;
    clear: both;
  }
  .first-column {
    background-color: #cccccc;
    float: left;
    width: 33%;
  }
  .second-column {
    float: left;
    width: 67%;
  }
}

@media (max-width: 480px) {
  .columns {
    width: 100%;
    /* same as 100% */
  }
  .columns:after {
    content: "";
```

```
display: table;
clear: both;
}
.first-column {
    background-color: #ff0000;
    width: 100%;
}
.second-column {
    width: 100%;
}
}
```

<!-- HTML: columnas.html -->

<html>

<head>

<link rel="stylesheet" type="text/css" href="responsive.css">

</head>

<body>

<section class="columns">

<article class="first-column">

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc hendrerit pellentesque diam quis lobortis. Morbi vel eleifend quam. Nullam urna nisi, vehicula nec maximus ut, gravida ac tellus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse vitae egestas neque, a consequat est. Suspendisse fermentum maximus fringilla. Vestibulum ultricies eros quis elit vulputate blandit. Aenean blandit sapien quam, nec ullamcorper tortor convallis at. Fusce id pretium lectus, vel finibus augue. Proin maximus pellentesque ultricies. Vestibulum sed metus id magna aliquet facilisis.

</article>

<article class="second-column">

Integer tincidunt nibh purus, a pulvinar lectus luctus ac. Suspendisse porttitor porttitor metus. Etiam bibendum lectus at ante varius sodales. Donec congue aliquet dui et eleifend. Etiam nec turpis eget quam tristique pretium. Vestibulum neque ante, lobortis sed urna blandit, vestibulum venenatis nisi. Curabitur ut arcu efficitur, scelerisque neque eu, hendrerit magna. Suspendisse tristique accumsan sapien, ut placerat ante.

</article>

</section>

</body>

</html>

Ejercicio:

Cambia el ejemplo anterior para utilizar <div> en lugar de <section> y <article>..

Por extensión, este enfoque puede seguirse no sólo para estilos básicos, sino también para

modificar otros aspectos del aspecto de la página, incluyendo diferentes vistas de las barras de navegación (que pueden pasar de ser listas a menús desplegables), elementos visibles/no visibles, etc.

Contenidos multimedia

Además de ajustar las dimensiones de los bloques de texto con CSS, también es necesario gestionar cómo se presentarán los elementos multimedia (principalmente imágenes) en la página. Para ello, seguiremos dos pasos:

- Ajustar en el CSS el tamaño máximo de los contenidos al 100% del tamaño del bloque que las contiene:

```
img, video, embed, object {  
    max-width: 100%;  
}
```

- Eliminar los atributos de anchura y altura que pudieran tener las imágenes en el HTML, para evitar que aparezcan distorsionadas al haber fijado su anchura máxima.

En caso de necesitar diferentes tamaños de imágenes, no necesariamente ocupando todo su contenedor, podemos definir varias clases de imágenes.

Puedes encontrar más información sobre diseño flexible en:
http://www.w3schools.com/html/html_responsive.asp.

Ejercicio:

Crea un documento HTML5 y un fichero CSS adaptable que incluya una barra de navegación superior, un cuerpo a tres columnas (20%, 40%, 40%) con texto e imágenes, y un pie de página con el ancho de la página.

Frameworks

Existen numerosos *frameworks* para CSS con hojas de estilos que definen clases más sofisticadas para implementar cuadrículas flexibles donde posicionar los elementos. Por ejemplo, W3.css establece un modelo de contenedores, un conjunto de estilos y diversos componentes configurables aplicables a cualquier página web. Otros frameworks ampliamente utilizados son Bootstrap (basado en HTML, CSS y jQuery), Skeleton (solo CSS) y HTML5 Boilerplate (HTML, CSS, JavaScript).

El siguiente ejemplo muestra cómo crear una disposición de cabecera más 3 columnas utilizando W3.css. Observa también cómo se utilizan otras clases para definir elementos de formato y componentes configurables:

```
<!DOCTYPE html>  
<html>  
<title>Demo Newspaper</title>  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<link rel="stylesheet" href="http://www.w3schools.com/lib/w3.css">  
<body>
```

```
<header class="w3-light-grey w3-padding-jumbo w3-center">
  <h1 class="w3-jumbo">Daily Ping</h1>
</header>

<section class="w3-row-padding">
  <article class="w3-container w3-light-blue w3-border w3-third w3-justify">
    <h2>Empty Notebook Found</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
  </article>
  <article class="w3-third w3-justify">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
  </article>
  <article class="w3-third w3-justify">
    <header class="w3-container w3-teal"><h1>Very New News!</h1></header>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et
```


dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

</article>

</section>

</body>

</html>

Puedes encontrar más información sobre W3.css en:
<http://www.w3schools.com/w3css/default.asp>.

Más información

Otras características interesantes que debes mirar para tener una idea completa de CSS:

- Posicionamiento de elementos: http://www.w3schools.com/css/css_positioning.asp.
- Visibilidad de elementos: http://www.w3schools.com/css/css_display_visibility.asp.
- Alineamiento horizontal: http://www.w3schools.com/css/css_align.asp.
- Pseudoclases: http://www.w3schools.com/css/css_pseudo_classes.asp.
- Pseudoelementos: http://www.w3schools.com/css/css_pseudo_elements.asp.
- Transparencia de imágenes: http://www.w3schools.com/css/css_image_transparency.asp.
- Nuevos selectores de CSS3: http://www.w3schools.com/css/css_attribute_selectors.asp.
- Barras de navegación: <http://css.maxdesign.com.au/> y <http://www.htmldog.com/articles/tabs/>.
- Flexboxes: http://www.w3schools.com/css/css3_flexbox.asp

Ejemplo completo

```
<!-- Ejemplo tomado del libro: Foundation Website Creation with HTML5, CSS3, and JavaScript. J. Lane, T. Barker, J. Lewis, M. Moscovitz, Friendsoft. 2012. -->
<!-- Fichero: inscripcion.html-->
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Summer Smash 2010 Registration</title>
  <link rel="stylesheet" href="style.css">
  <script src="js/javascript.js"></script>
</head>

<body>
  <section id="pagewrap">
    <header>
      <a href="/">
        <h1>Summer Smash 2012</h1>
        <p>This summer's most smashing event!</p>
      </a>
    </header>
    <section id="main">
      <section id="registration_form">
        <form action="/registration/">
          <fieldset>
            <legend>Registrant Information</legend>
            <ol>
              <li class="formitem">
                <label for="name">Registrant name:</label>
                <input type="text" name="name" id="name" required autofocus />
              </li>
              <li class="formitem">
                <label for="email">Email address:</label>
                <input type="email" name="email" id="email" required />
              </li>
              <li class="formitem">
                <label for="phone">Phone number:</label>
                <input type="tel" name="phone" id="phone" />
              </li>
              <li class="formitem">
                <label for="party">How many people do you have in your party (including yourself)?</label>
                <input type="number" name="party" id="party" min="1" max="10" />
              </li>
            </ol>
          </fieldset>
        </form>
      </section>
    </section>
  </section>
</body>
```

```

        <label for="dob">Your date of birth:</label>
        <input type="date" name="dob" id="dob" />
    </li>
</ol>
</fieldset>
<fieldset>
    <legend>A few quick questions</legend>
    <ol>
        <li class="formitem">
            Is this your first Summer Smash event?
            <ul>
                <li>
                    <input type="radio" name="yes_first" id="yes_first" value="1" />
                    <label for="yes_first">Yes, this is my first</label>
                </li>
                <li>
                    <input type="radio" name="no_first" id="no_first" value="0" />
                    <label for="no_first">No, I've been to one before
                    </label>
                </li>
            </ul>
        </li>
        <li class="formitem">
            <label for="how_hear">How did you hear about Summer Smash 2012?</label>
            <input type="text" name="how_hear" id="how_hear" list="media">
            <datalist id="media">
                <option value="Google search">
                <option value="Magazine ad">
                <option value="A friend told me">
                <option value="An enemy told me">
                <option value="My dog told me">
                <option value="My dead uncle Henry told me">
            </datalist>
        </li>
    </ol>
</fieldset>
    <input type="submit" value="Register now!" />
</form>
</section>
<aside>
    <h2>Questions about whether Summer Smash is right for you?</h2>
    <p>We're here to help! Our team of smashtastic smashers will shatter any doubt in your mind. Why not give us a call, or send us an email?
</p>
    <ul>
        <li>Call: 1-800-555-5555</li>
        <li>Email: <a href="mailto:smash@example.com">smash@example.com</a></li>
    </ul>

```

```
</ul>
</aside>
</section>
<footer>
  <p>Summer Smash 2012. A Barker and Lane Production.</p>
</footer>
</section>
</body>
```

```
</html>
<!-- Fichero style.css-->
/* RESET STYLES */
html, body, div, span, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
abbr, address, cite, code,
del, dfn, em, img, ins, kbd, q, samp,
small, strong, sub, sup, var,
b, i,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, figcaption, figure,
footer, header, hgroup, menu, nav, section, summary,
time, mark, audio, video {
    margin: 0;
    padding: 0;
    border: 0;
    outline: 0;
    font-size: 100%;
    vertical-align: baseline;
    background: transparent;
}

body {
    line-height: 1;
}

article,
aside,
details,
figcaption,
figure,
footer,
header,
hgroup,
menu,
```

```
nav,
section {
    display: block;
}

nav ul {
    list-style: none;
}

blockquote,
q {
    quotes: none;
}

blockquote:before,
blockquote:after,
q:before,
q:after {
    content: none;
}

a {
    margin: 0;
    padding: 0;
    font-size: 100%;
    vertical-align: baseline;
    background: transparent;
}

/* change colours to suit your needs */

ins {
    background-color: #ff9;
    color: #000;
    text-decoration: none;
}

/* change colours to suit your needs */

mark {
    background-color: #ff9;
    color: #000;
    font-style: italic;
    font-weight: bold;
}

del {
    text-decoration: line-through;
```

```
}

abbr[title],
dfn[title] {
    border-bottom: 1px dotted;
    cursor: help;
}

table {
    border-collapse: collapse;
    border-spacing: 0;
}
/* change border colour to suit your needs */

hr {
    display: block;
    height: 1px;
    border: 0;
    border-top: 1px solid #cccccc;
    margin: 1em 0;
    padding: 0;
}

input,
select {
    vertical-align: middle;
}
/* END RESET */
/* Layout Block level elements */

section#pagewrap {
    width: 80%;
    margin: 4em auto;
    font-family: Verdana, Arial, sans-serif;
}

header {
    padding: 1em 0 2em;
}

section#main {
    float: left;
    margin: 2em 2%;
}

section#registration_form {
```

```
width: 74%;  
float: left;  
padding: 0 1% 0 0;  
}  
  
aside {  
width: 21%;  
float: right;  
}  
  
footer {  
clear: both;  
padding: 2em 3%;  
}  
/* Header */  
  
header {  
background-color: #333;  
}  
  
header a {  
text-decoration: none;  
color: white;  
text-align: center;  
}  
  
header h1 {  
font-size: 3.5em;  
font-family: "Frijole", Arial, sans-serif;  
color: #eee;  
}  
  
header p {  
font-size: 1.2em;  
font-family: "Frijole", Arial, sans-serif;  
color: #ccc;  
margin: 0.7em 0 0 0;  
letter-spacing: 1px;  
}  
/* Form */  
  
fieldset {  
padding: 0.5em 1% 1em;  
}  
  
legend {
```

```
text-transform: uppercase;
font-size: 1.2em;
font-weight: bold;
padding-bottom: 3px;
border-bottom: 1px solid #333;
width: 100%;
}

form ol,
form ul {
  list-style-type: none;
  margin: 0 2%;
}

form li {
  margin: 0.5em 0;
}

label {
  font-size: 0.8em;
}

form ol li label {
  display: block;
}

form ul li label {
  display: inline;
}

input[type="text"],
input[type="number"],
input[type="tel"],
input[type="email"],
input[type="date"] {
  width: 50%;
  padding: 0.5em 0.7%;
  margin-top: 0.4em;
  font-size: 1em;
  color: #eee;
  border: none;
  background-color: rgba(0, 0, 0, 0.6);
  -webkit-border-radius: 4px;
  -moz-border-radius: 4px;
  -o-border-radius: 4px;
  border-radius: 4px;
```



```
}

input[type="number"],
input[type="date"] {
    width: 20%;
}

input[type="submit"] {
    background-color: rgba(0, 0, 0, 0.6);
    color: #eee;
    padding: 0.5em 0.7%;
    font-size: 1em;
    border: none;
    -webkit-border-radius: 4px;
    -moz-border-radius: 4px;
    -o-border-radius: 4px;
    border-radius: 4px;
    margin-left: 2%;
}

/* Aside */

aside {
    background-color: #ccc;
    padding: 1em 2%;
    -webkit-border-radius: 4px;
    -moz-border-radius: 4px;
    -o-border-radius: 4px;
    border-radius: 4px;
}

aside h2 {
    font-size: 1em;
    line-height: 1.3em;
    text-transform: uppercase;
    margin-bottom: 1em;
}

aside p {
    font-size: 0.8em;
    line-height: 1.2em;
    margin-bottom: 1.5em;
}

aside ul {
    list-style-type: none;
    font-size: 0.8em;
```

```
    line-height: 1.5em;
}

aside a {
    color: #333;
    background-color: transparent;
    text-decoration: none;
    -webkit-transition: all 3s ease;
    -moz-transition: all 3s ease;
    -o-transition: all 3s ease;
    transition: all 3s ease;
}

aside a:hover {
    color: #c00;
    background-color: black;
    font-weight: bold;
}
```

Bibliografía

- <http://www.w3schools.com/css3/>
- <http://css.maxdesign.com.au/>
- <http://www.htmldog.com/guides/>
- <http://librosweb.es/css/>