

Diseño web adaptable: un ejemplo universal

Autor: Alvarez Amada, María Paz (Licenciada en Ciencias Físicas. Técnico superior en Desarrollo de Aplicaciones Informáticas y en Desarrollo de Aplicaciones Web. Postgrado en Informática, Profesora de FP de la especialidad de Sistemas y Aplicaciones Informáticas).

Público: Programadores de páginas web. Ciclos formativos de Informática. **Materia:** Programación web. **Idioma:** Español.

Título: Diseño web adaptable: un ejemplo universal.

Resumen

En este artículo se presentan dos ejemplos de código web adaptable, que cumplen la característica de adaptarse a cualquier tipo de tamaño de pantalla. Permiten mostrar el contenido centrándolo en la pantalla independientemente de las dimensiones de pantalla. El código expuesto va a ser muy útil para aquellas personas interesadas en la programación de diseños web. Es un código sencillo pero a la vez muy potente. Los ejemplos mostrados pueden ser utilizados en la docencia de diseños de páginas web puesto que son muy clarificadores.

Palabras clave: Diseño web adaptable, html, css, navegador, resolución, tamaño de pantalla.

Title: Responsive web design: a universal example.

Abstract

In this article is presented two examples of web responsive design, that has the property of adapt it to any type of screen size. It shows the content centered on the screen regardless of the screen dimensions. The specified code will be very useful for people interested in the programming of web designs. It is a simple but a very powerful code. The shown examples can be used in the teaching of web design since they are very clarifiers.

Keywords: Responsive web design, html, css, browser, resolution, screen size.

Recibido 2017-03-23; Aceptado 2017-03-28; Publicado 2017-04-25; Código PD: 082077

1. INTRODUCCIÓN

La evolución de las tecnologías de red, ha permitido el acceso a Internet a muchos usuarios. El principal uso de Internet es el la visualización de páginas web.

Las tecnologías para la construcción de páginas web evolucionan rápidamente y se van adaptando a la evolución del hardware.

A fecha de hoy existen muchos tipos de dispositivos, que tienen tamaños específicos de pantalla para acceder a la visualización de páginas web. Estos tamaños van desde pocas pulgadas (teléfonos móviles) a decenas de pulgadas (equipos de sobremesa). La visualización de las páginas web en tan amplio rango de tamaños de pantalla representa un problema técnico a la hora de programar los diferentes diseños. Ya en el año 2008, el World Wide Web Consortium (W3C) publicó una recomendación llamada "Mobile Web Best Practices" bajo el subtítulo "One Web" para especificar las mejores prácticas para el diseño.

2. BREVE EVOLUCIÓN HISTÓRICA

La primera especificación de página web fue creada por Tim Berners Lee en el año 1991. Las páginas web creadas en esta época se implementaban mediante un editor de texto en el que se introducían las etiquetas html con sus diferentes atributos. Esto suponía que el fichero estaba formado tanto por contenido de información como por diseño.

A partir del año 1998 se empezó a crear la especificación CSS por el W3C con el propósito de separar contenido de diseño. Mediante la especificación se consiguió sacar a un fichero exterior todas las configuraciones de diseño de la página (estilos), permitiendo mantener una independencia entre el contenido y el diseño, siendo el código de diseño independiente del código html.

Tanto en HTML como en CSS han existido diferentes versiones, que han ido evolucionando para proveer de más posibilidades de creación de diseños complejos y su control. Las últimas versiones son HTML5 y CSS3.

El contenido de las páginas web puede ser fijo, o puede obtenerse de ficheros o de bases de datos (contenido dinámico). Para ello, dentro del código de la página web (u otro fichero enlazado) hay que situar el código de programación para tener acceso a un servidor o para que el propio navegador interprete el código en función de si se emplean lenguajes de servidor (php, asp, jsp, ...) o de cliente (javascript, ...).

Esto supone que el contenido del código de una página puede llegar a ser bastante complejo si contamos con la parte de código de la página html, de la de diseño css y la parte de programación de contenido dinámico.

Si a esto sumamos la escasez de herramientas de programación web (entornos de desarrollo) que cumplan con todos los requisitos exigibles para llevar una gestión de los tres tipos de código simultáneamente permitiendo al programador un control absoluto de todo su trabajo, nos encontramos ante una dificultad técnica.

Por ello, un buen desarrollador web que cree páginas web desde cero, deberá ser muy escrupuloso en la realización del análisis previo, estudiando que partes del código y diseño serán comunes a las páginas, que estructura visual tendrán las páginas, etc., teniendo siempre un control de todo lo que genere.

Existen herramientas específicas para diseño de interfaces web, que permiten realizar el diseño seleccionando objetos visuales que se desplazan a la página en formato gráfico, pero al final, suelen requerir trabajo adicional para ser integradas en un proyecto web.

3. ESTRUCTURA BÁSICA DE UNA PÁGINA WEB CON ARCHIVO CSS VINCULADO

Por lo dicho anteriormente, una página web elemental estará formada por dos ficheros, uno de tipo html y otro de tipo css.

Vamos a analizar a fecha actual qué contenido vamos considerar como básico en el fichero html indicando que el contenido está escrito en idioma español. Supongamos que los ficheros que vamos a emplear, para contextualizar a un ejemplo concreto, se llaman fichero.html y fichero.css respectivamente. El código de fichero.html básico que vamos a considerar es el siguiente:

```
<!doctype html>
<html lang="sp">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" type="text/css" href="fichero.css">
  </head>
  <body>
  </body>
</html>
```

Sobre este código base vamos a ir incluyendo las aportaciones.

4. DISEÑO WEB UNIVERSAL

Se va a presentar como realizar un diseño elemental de página web con diseño adaptable a cualquier tipo de pantalla utilizando el mínimo de programación, sin tener que definir resoluciones de pantalla específicas.

Es decir, vamos a evitar entrar en especificaciones de media queries en el fichero.css

Una media query permite personalizar el código de diseño css para una resolución o conjunto de resoluciones específicas.

La técnica empleada va a ser la utilización de etiquetas div para definir el contenido que interese, sin tener que especificar en ningún lugar de los estilos css la resolución de pantalla. El diseño se adaptará automáticamente al tamaño de la pantalla.

Para rellenar de contenido, se utilizará un texto de tipo “Lorem ipsum”.

En diseño gráfico, se llama texto “Lorem ipsum” a un texto que se utiliza para probar el diseño visual de un contenido gráfico (en nuestro caso página web) antes de insertar el texto definitivo.

Existen en internet cientos de generadores de contenido de tipo “Lorem ipsum”. Un ejemplo de generador de código “Lorem ipsum” está en la página:

<http://www.blindtextgenerator.com/es>

Para que los ejemplos sean pedagógicos, en el diseño css, haremos que se muestren los bordes de las etiquetas divs, para que al cambiar el tamaño de la ventana del navegador mientras visualizamos la página, podamos ver claramente el comportamiento de la página.

Dada la limitación de la extensión para el presente artículo, cuando se muestre el contenido del código del fichero html, se empleará el contenido “Texto Lorem ipsum” para representar el contenido, que deberá ser sustituido por un texto “Lorem ipsum” largo.

Si lo obtenemos a través del generador que se ha indicado anteriormente, sustituiremos “Texto Lorem ipsum” por:

“Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue.”

5. EJEMPLOS DE CÓDIGO

Ejemplo 1.

Contenido del fichero fichero.html

```
<!doctype html>
<html lang="sp">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" type="text/css" href="fichero.css">
  </head>
  <body>
    <div class="wrapper">
      <div class="div1">
        Texto Lorem ipsum
      </div>
    </div>
  </body>
</html>
```

Contenido del fichero fichero.css

```
.wrapper {
    border: 1px solid green;
    overflow: hidden;
    height: auto;
    font-family: Arial,Helvetica Neue,Helvetica,sans-serif;
    color: #00CCCC;
}
/* Estilo genérico para los divs */
.wrapper div {
    max-width: 100%;
    height: auto;
    overflow: hidden;
    text-align: justify;
}
.div1{
    margin: 0 auto;
    border: 1px solid blue;
    width: 40em;
}
```

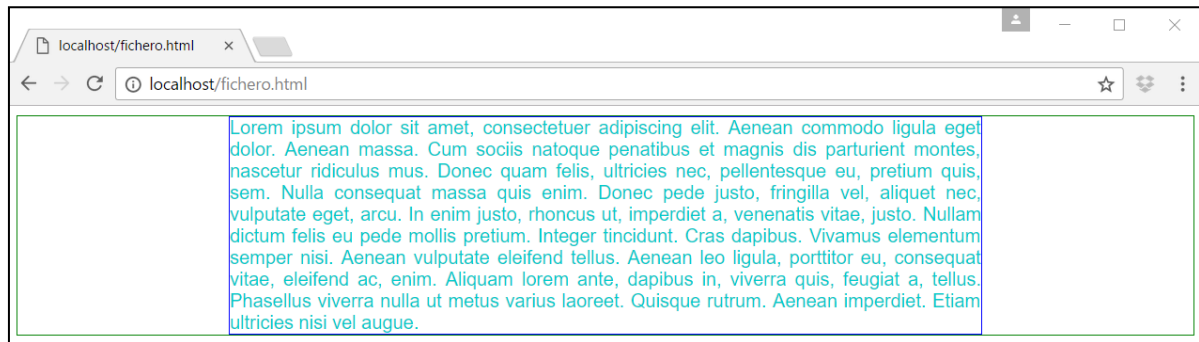
El código html, contiene un div genérico llamado wrapper que se adaptará a todo el contenido que muestre la página y otro div interior llamado div1 que será en el que se muestre el contenido del texto. Cuando digo que un elemento se llama de alguna forma particular, me estoy refiriendo a que el elemento tiene definida una clase con ese nombre. El div div1 tendrá un tamaño determinado, se centrará automáticamente en la pantalla, y en caso de que la pantalla sea pequeña, se adaptará para no cortar nada del contenido y que pueda ser visualizado completamente.

El centrado automático se consigue mediante el estilo margin: 0 auto. Por otro lado se ha puesto bordes a los div, se ha definido el tipo de letra, el color de la letra, etc. para tener más claridad en su visualización. El punto más importante de reseñar es que si el div wrapper ocupa todo el contenido de las etiquetas de su interior, es porque se le ha introducido el estilo height: auto. Si esto no se especifica, el contenido del div será muy pequeño y no contendrá a todo lo que hay en su interior. Recomiendo hacer la prueba de borrar height: auto y ver el comportamiento tan nefasto que se produce en el borde de este div.

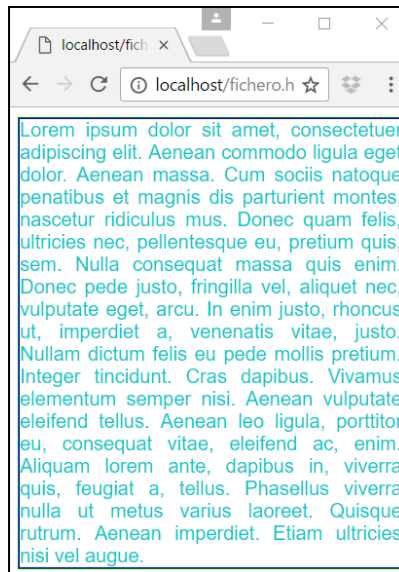
Para probar el comportamiento de la página en otras resoluciones, lo que podemos hacer es reducir y estirar la ventana del navegador. Esto simulará los diferentes tipos de pantalla. Existen páginas web en Internet que permiten visualizar el contenido de una página web para diferentes tipos de pantallas.

Los aspectos que generará el código anterior en diferentes tamaños de pantalla serán (no se especifican tamaños específicos porque lo que se está estudiando es el comportamiento global):

Tamaño grande de pantalla



Tamaño pequeño de pantalla



Como puede observarse, el contenido se ha adaptado plenamente.

Ejemplo 2

Contenido del fichero fichero.html

```
<!doctype html>
<html lang="sp">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" type="text/css" href="fichero.css">
  </head>
  <body>
    <div class="wrapper">
      <div class="contenedor">
        <div class="div1">
          Texto Lorem ipsum
        </div>
      </div>
    </div>
  </body>
</html>
```

```
        <div class="div2">
            Texto Lorem ipsum
        </div>
    </div>
    <div class="div3">
        Texto Lorem ipsum
    </div>
</body>
</html>
```

Contenido del fichero fichero.css

```
.wrapper {
    border: 3px solid green;
    overflow: hidden;
    height: auto;
    font-family: Arial,Helvetica Neue,Helvetica,sans-serif;
    color: #00CCCC;
}
/* Definición genérica para los div*/
.wrapper div {
    max-width: 100%;
    height: auto;
    overflow: hidden;
}
.contenedor{
    border: 3px solid red;
    margin: 0 auto;
    width: 41em;
}
.div1{

    border: 3px solid blue;
    width: 20em;
    float:left;
}
.div2{
```

```
margin: 0 auto;
border: 3px solid orange;
width: 20em;
float: left;
}
.div3{
clear: both;
border: 3px solid pink;
}
```

En este caso, lo que hemos hecho ha sido establecer dos divs para que se muestren horizontalmente (sin tener que definir tablas ni utilizar ningún otro tipo de mecanismo) y un div en la parte inferior, que se comporte a modo de pie de página. Todos estos divs se adaptarán.

Se ha modificado el código del ejemplo 1 para añadir el resto de divs.

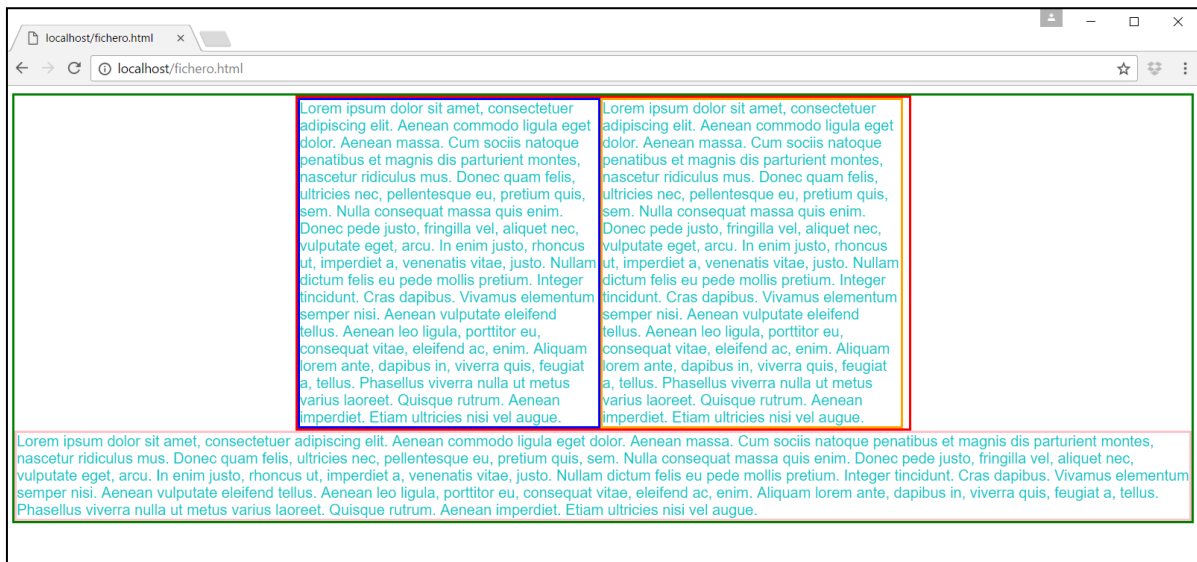
Lo más importante de resaltar en el código añadido en el fichero fichero.css, es que en div1 y div2 se ha añadido el estilo float: left para que sean flotantes y se coloquen a la izquierda cuando no quepan en la pantalla, siempre manteniendo una visualización centrada en la pantalla.

El div div3 que se comporta a modo de pie de página contiene el código clear: both. Este código tiene el mismo comportamiento que el punto y aparte en un procesador de texto, pero a modo de divs.

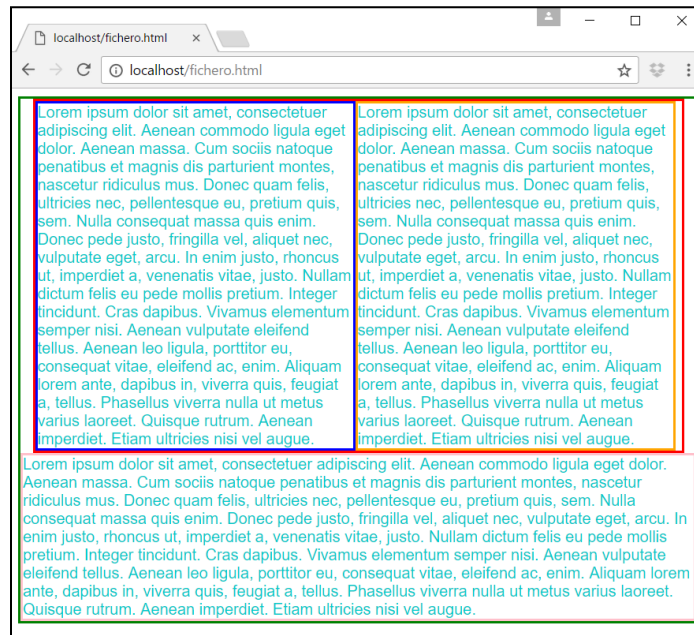
Veamos cual es el comportamiento del contenido en función de los diferentes tamaños de pantalla.

Visualización del contenido.

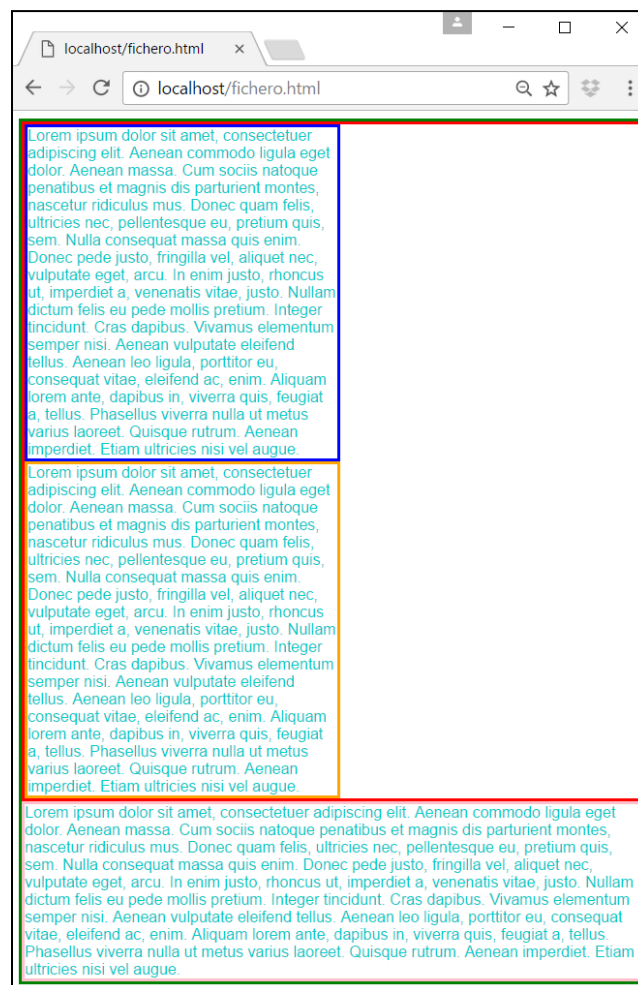
Tamaño grande de pantalla



Tamaño intermedio de pantalla

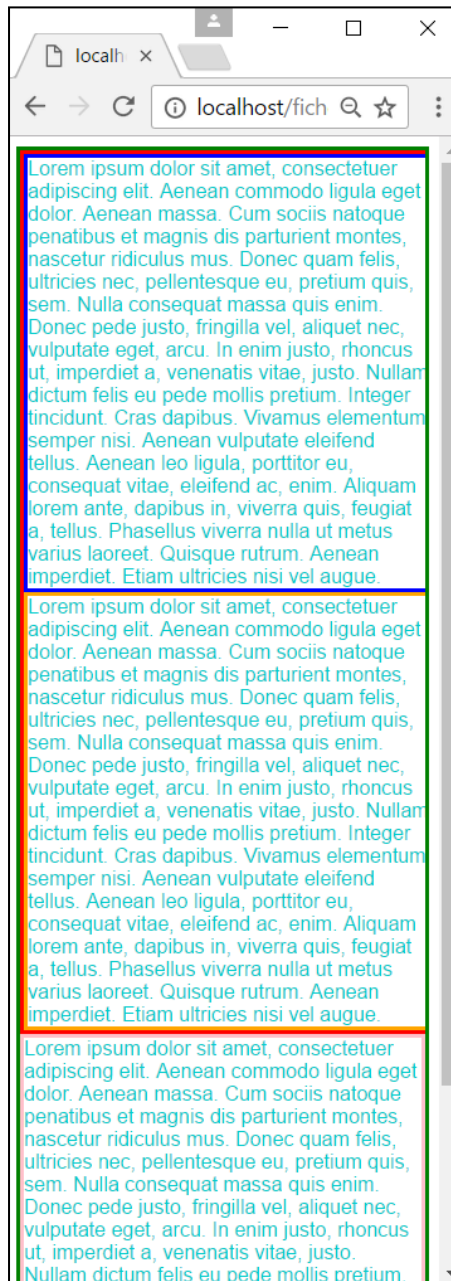


Tamaño un poco más pequeño de pantalla



Observar en este tamaño que como los dos divs horizontales no han cabido, el segundo div ha pasado directamente a la izquierda automáticamente (float: left;).

Tamaño pequeño de pantalla



En este caso, cada uno de los divs se están adaptando a la limitación de la pantalla tan pequeña, permitiendo visualizar siempre el contenido completo de los mismos.

6. CONCLUSIÓN

Existen posibilidades de diseñar páginas web adaptativas sencillas sin tener que tener en cuenta en el código los diferentes tamaños de pantalla usuales existentes en el mercado.

Si se busca en internet información sobre lo expuesto en este artículo se verá que en general, las soluciones presentadas por la comunidad programadora de páginas web, en general, son mucho más complejas que la presentada.

El código mostrado en ese artículo es la manera más sencilla de conseguir los comportamientos deseados de centrados y adaptaciones a tamaños de pantalla sin tener que especificar en el código el tamaño específico de la pantalla.

El resultado es un código sencillo, simple y limpio, y por ello es fácil de reutilizar.

A partir de él, se pueden realizar elaboraciones más complejas, teniendo como límite sólo la imaginación del programador.

Bibliografía

- Diseño web adaptable – https://es.wikipedia.org/wiki/Dise%C3%B1o_web_adaptable
- HTML - <https://es.wikipedia.org/wiki/HTML>
- CSS – https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada
- HTML5 – <https://es.wikipedia.org/wiki/HTML5>
- Media query - https://es.wikipedia.org/wiki/Media_query
- Lorem ipsum - https://es.wikipedia.org/wiki/Lorem_ipsum