

DISEÑO RESPONSIVE

El diseño Responsive consiste en diseñar los sitios web de forma que se vean bien en todos los dispositivos desde los que accedamos a Internet. Hay que notar que decimos que se vea bien, y no que se vea igual. Por definición, resulta imposible que algo planificado para una resolución de 1200 píxeles pueda verse igual en una resolución de apenas 400 píxeles.

Por lo tanto el objetivo es el de lograr que con independencia de cual sea el dispositivo con el que accedemos a nuestro sitio, este se va a ver bien, de forma correcta. Por bien entendemos que todo el contenido será visible y legible, y que el formato del sitio mantendrá una coherencia en todos los dispositivos, a nivel de tipografías, colores e identidad visual.

Pensar en Responsive significa sobre todo pensar en diseño web y diseño gráfico, consiste en adaptar el contenido al dispositivo de forma que su presentación sea siempre la más apropiada para ese medio y manteniendo la coherencia visual de todo el sitio.

Es un planteamiento que se pone del lado del contenido de manera radical. Lo más importante que tenemos en una web es el contenido, ese es su sentido, poner a disposición del visitante una información que pensamos que es útil o necesaria para él y que por lo tanto tiene que estar siempre en primer plano. No tiene sentido el modificar el contenido dependiendo de desde dónde accedemos al sitio. Un usuario que acceda al sitio desde un pc en un caso y desde un móvil en otro, espera encontrar el mismo contenido en ambos casos y tiene poco sentido que no sea así.

Responsive significa pues modificar la presentación de ese contenido, o lo que es lo mismo modificar el CSS de nuestro sitio para que el HTML se vea siempre bien si tener que hacer cambios en él.

APLICACIÓN

Desde el lado técnico, lo que vamos a hacer es preparar diferentes estilos de CSS para las diferentes resoluciones en las que podamos ver nuestro contenido. Es importante decir para diferentes resoluciones y no para diferentes dispositivos. Los dispositivos van cambiando con el tiempo, y trabajar pensando en un dispositivo concreto no garantiza que el trabajo vaya a seguir siendo válido cuando ese dispositivo cambie, o cuando aparezcan nuevos dispositivos en el mercado. En cambio, si nos aseguramos de que el contenido se ve bien a cualquier resolución de pantalla, estamos seguros de que siempre se mostrará de forma óptima al margen de cuál sea el dispositivo.

ATRIBUTOS Y PROPIEDADES

Los elementos concretos con los que vamos a trabajar en CSS para conseguir este objetivo son realmente pocos. En primer lugar vamos a olvidarnos de medidas absolutas para nuestros elementos y vamos a trabajar sobre todo con medidas relativas. Trabajar con unidades como porcentajes, o em y rem para los textos nos va a brindar la flexibilidad necesaria para este trabajo.

Esto es especialmente importante en el caso de las imágenes. Tenemos que asegurarnos de que las imágenes nunca exceden o desbordan sus contenedores, efecto debido a que el tamaño asignado a la imagen en el html (o por las medidas en píxeles de la propia imagen cuando esta no lleva atributos de tamaño en la etiqueta img) es mayor que la anchura del documento, y que provoca la aparición de un scroll horizontal en el documento.

Otro elemento fundamental en este tipo de diseño es limitar los tamaños de los elementos mediante mínimos y máximos. Al definir los tamaños con unidades relativas, perdemos el control del tamaño exacto de cualquier elemento así definido, puesto que una anchura porcentual se refiere a su contenedor, y este en última instancia viene definido por la anchura del dispositivo, que va a ser siempre un valor variable. Para evitar que ese porcentaje nos de valores que quedan fuera del rango de nuestro diseño utilizamos las propiedades min-width y max-width, o min-height y max-height para el caso de la altura. De esta forma establecemos una valores de seguridad que actuarán como límites del tamaño del elemento.

Así, para conseguir que todas las imágenes del sitio se comporten de forma relativa podríamos hacer algo así: **img {width: 100%;}**. Lo cual hará que cualquier imagen de nuestro documento siempre tenga la misma anchura que su contenedor. En cambio si lo que buscamos es que una imagen tenga su tamaño original, dentro de lo posible, pero que no exceda a su contenedor cuando este sea menor que la imagen, haremos esto:

```
img {  
max-width: 100%;  
height: auto;  
}
```

Estos son los atributos que nos van a posibilitar trabajar de forma fluida o flexible, que es una de dos “patas” sobre las que se articula el diseño responsive. La otra es la posibilidad de ofrecer un conjunto de estilos CSS específicos para una resolución dada. Este CSS condicional es lo que llamamos media queries, o consultas de medios. Es una instrucción con la que le pedimos al navegador que compruebe una serie de condiciones sobre el dispositivo en el que está mostrando el contenido, y que aplique un formato dado en función de esas condiciones.

MEDIA

Desde CSS 2.1 existe la directiva media, que permite definir diferentes estilos CSS para diferentes tipos de dispositivos o soportes. Todos los dispositivos que reconoce la propiedad media son: all, braille, embossed, handheld, print, projection, screen, speech, tty, y tv.

De esta forma podemos seleccionar el dispositivo y ajustar la presentación para cada uno de ellos. El problema que tenemos en la actualidad es que dentro del mismo tipo de dispositivo existen tantas diferencias entre ellos que trabajar solamente con ese atributo no nos resuelve el problema de adaptar la presentación al dispositivo. El caso específico es el de media screen, que implica a todas las pantallas, tanto la de un monitor de 2500px de resolución como la de un teléfono móvil de 340px de resolución.

Es por este motivo por el que CSS 3 introduce los media query, o consulta de medios, con el cuál se puede filtrar no sólo por el dispositivo, sino que además podemos referirnos a propiedades concretas del dispositivo que sí nos permitan un CSS más personalizado.

Esta característica está soportada por todos los navegadores actuales, aunque no funcionara en versiones antiguas, IE8 y anteriores no dan soporte a esta función.

La forma de implementar esta consulta es incluyendo la directiva @media dentro del archivo CSS:

@media all and(min-width: 960px) { dentro de las llaves todo el CSS específico para esta condición}

En este caso estamos definiendo un estilo que se aplicará a cualquier dispositivo (media all) cuya anchura mínima del área de visualización, normalmente la ventana del programa, alcance los 960 píxeles. Hay que notar que la directiva siempre empieza con la @, y que las condiciones van siempre entre paréntesis.

SINTAXIS

Los Media Queries pueden contener una o más expresiones, expresadas como funciones multimedia, que se resuelven en true o false. El resultado del query o consulta devuelve true si el media type especificado en el Media Query coincide con el tipo de dispositivo en que el documento está siendo mostrado y todas las expresiones en el Media Query devuelven true.

Cuando un Media Query devuelve true, la correspondiente hoja de estilo es aplicada, siguiendo las reglas habituales de CSS. Las hojas de estilo con Media Queries adjuntos a los tags <link> seguirán descargándose, incluso si sus Media Queries resultan false (sin embargo, no se aplicarán).

Se pueden crear Media Queries complejos utilizando logical operators, incluyendo **not, and y only**.

El operador **and** es usado para combinar múltiples media features en un sólo Media Query, requiriendo que cada función devuelve true para que el Query también lo sea. Además, se usa para combinar múltiples media features, así como combinar éstos con media types.

El operador **not** se utiliza para negar un Media Query completo y el operador **only** se usa para aplicar un estilo sólo si el Query completo es correcto.

El keyword **only** previene a los navegadores que no soportan Media Queries

Además, se pueden combinar múltiples Media Queries **separados por comas** en una lista; si alguno de los Queries devuelve true, todo el *media statement devolverá true. Esto es equivalente a un operador **or**.

Cuando se utilizan las listas separadas por comas en los Media Queries, si algunas de las Media Queries devuelven true, los estilos se aplican. Cada Media Query separado por comas en la lista se trata como un Query individual, y cualquier operador aplicado a un Media Query no afecta a los demás. Esto significa que los Media Queries separados por comas puede dirigirse a diferentes media features, types o states.

Listado de características completo para media:

Propiedad	Definición	Admite prefijo min- y max-
width	El ancho del área de visualización (display area)	Sí
height	El alto del área de visualización (display area)	Sí
device-width	El ancho total del dispositivo (rendering surface)	Sí
device-height	El alto total del dispositivo (rendering surface)	Sí
orientation	Acepta los valores portrait o landscape	No
aspect-ratio	Relación de aspecto entre el ancho y alto del área de visualización	Sí
device-aspect-ratio	Relación de aspecto entre el ancho y alto del dispositivo	Sí
color	El número de bits de profundidad de color	Sí
color-index	El número de entradas en la tabla de colores del dispositivo	Sí
monochrome	El número de bits de profundidad de color, en dispositivos monocromáticos	Sí
resolution	Densidad de pixels en el dispositivo, medido en dpi	Sí

VIEWPORT

Una peculiaridad que tienen los dispositivos móviles es que sus navegadores falsean la resolución real de la pantalla del dispositivo, haciendo que muestren el doble o el triple de píxeles de los que realmente contienen, para conseguir de esta forma adaptar los diseños de webs pensadas para pantallas de escritorio al tamaño del dispositivo móvil.

En este caso, nuestro diseño con media queries no va a funcionar, porque el navegador lanzará la versión de escritorio al entender que la resolución del dispositivo es mayor de lo que realmente es. Para anular este comportamiento utilizamos la meta etiqueta viewport.

La etiqueta viewport nos permite definir el ancho, alto y escala del área usada por el navegador para mostrar contenido. Al fijar el ancho o alto del viewport, podemos usar un número fijo de píxeles (ej: 320px , 480px , etc) o usar dos

constantes, `device-width` y `device-height` respectivamente. Se considera una buena práctica configurar el viewport con algunas de estas dos constantes, en lugar de utilizar un ancho o alto fijo.

Las propiedades a tener en cuenta son las siguientes:

width: controla el ancho del área de visualización. Puede ser inicializado a un número concreto de pixels, como `width=600` o con el valor especial `device-width`, que indica el ancho total del dispositivo, en pixels en una escala del 100%.

initial-scale: controla el nivel de zoom inicial al cargarse la página. Las propiedades `maximum-scale`, `minimum-scale`, y `user-scalable` controlan como el usuario puede realizar zoom sobre la página.

Una configuración habitual de esta etiqueta puede ser:

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximumscale=1">
```

EJEMPLOS

Media para seleccionar sólo por el tipo de dispositivo:

```
<link rel="stylesheet" type="text/css" href="core.css" media="screen" />
<link rel="stylesheet" type="text/css" href="print.css" media="print" />
```

Ejemplo de media query enlazando a hoja de estilos externa:

```
<link rel="stylesheet" type="text/css" media="screen and (max-device-width: 480px)" href="shetland.css" />
```

Ejemplo de media query en la misma hoja de estilos:

```
@media screen and (min-width: 1024px) {
body {
font-size: 100%;
}
}
```

Con dos condiciones a cumplir:

```
@media (min-width: 700px) and (orientation: landscape) { ... }
```

Usando una coma para separar valores:

```
@media (min-width: 700px), handheld and (orientation: landscape) { ... }
```