DISEÑO WEB RESPONSIVE: MEDIA QUERIES DE CSS3.

Si has trabajado anteriormente con la versión para imprimir de una página web, entonces ya habrás tenido una primera experiencia con los media queries. Con los media queries lo que hacemos básicamente es decidir cómo se va a mostrar una página web en función de los parámetros que indiquemos. En el caso tradicional, nos encontrábamos con dos versiones de una web, la que era para pantalla o screen y la que era para imprimir o print.

|  |  |
| --- | --- |
| 1 | @media screen { |
| 2 | body {  width: 70%; } | |

|  |  |  |
| --- | --- | --- |
| 3 | } | |
| 4 |  |

|  |  |
| --- | --- |
| 5 | @media print { |
| 6 | body {  width: 100%; } | |

|  |  |
| --- | --- |
| 7 | } |
|  |  |

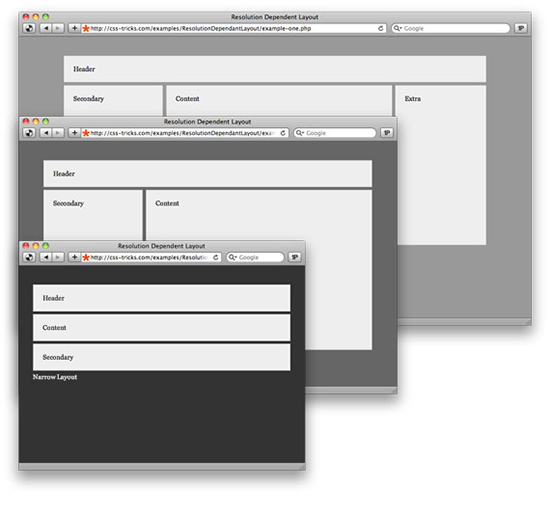
MEDIA QUERIES DE CSS3

Los **media queries de CSS3** llevan la idea de los media queries tradicionales un paso más allá para adaptar el diseño web a los diferentes dispositivos. Más que fijarnos en el tipo de dispositivo, lo que tendremos en cuenta es la capacidad del dispositivo en cuestión. Podemos así utilizar los media queries de CSS3 para comprobar diferentes factores, como por ejemplo son:

* ancho y alto de la ventana del navegador.
* ancho y alto del dispositivo.
* resolución del dispositivo.
* orientación (un dispositivo móvil puede tomar dos posiciones).

RESOLUCIÓN DE PANTALLA VERSUS ANCHO DE VENTANA

Aunque a primera vista pueda parecer lo mismo, hemos de distinguir entre la resolución de pantalla y el ancho de ventana del navegador. La manera más fácil de entenderlo es pensando en un PC. Imagina que tienes una resolución de pantalla de 1980px de ancho y abres tu navegador habitual. Si este ocupa toda la pantalla, entonces la resolución de pantalla coincide con el ancho de ventana. Pero imagínate ahora que reduces su tamaño, ahora el ancho de ventana del navegador es menor que la resolución de pantalla.



*Imagen de CSS Tricks*

Así podemos concluir que el ancho de ventana de navegador como máximo podrá ser el tamaño de la resolución de pantalla del dispositivo.  
Es recomendable que cuando utilicemos los media queries de CSS3 lo hagamos pensando en el**ancho de ventana del navegador** y no en la resolución de pantalla. ¿Por qué? Veamos un ejemplo. Imagínate que queremos utilizar un media query de CSS3 que tenga en cuenta la resolución de pantalla del dispositivo. En ese caso, siguiendo los estándares marcados por el W3C utilizaremos este código:

|  |  |
| --- | --- |
| 1 | <link rel="stylesheet" media="screen and (min-device-width: 960px)" href="960.css"/> |

Con este código lo que estamos haciendo es indicar al navegador web que utilice la hoja de estilos 960.css si el dispositivo en cuestión tiene una resolución de ancho de pantalla de 960px o más. ¿Cuál es el problema? Pues que si como veíamos por algún motivo el usuario decide reducir el ancho de ventana de navegador (por ejemplo, para ver la página web y su editor de textos a la vez), si reduce la ventana a pongamos un ancho de 600px, el navegador seguirá mostrando la página web con la resolución pensada para 960px o más, con lo que no se adaptará al nuevo ancho de ventana, generando una mala experiencia de usuario.

MEDIA QUERIES DE CSS3 EN ACCIÓN

Hemos visto que más que referirnos a la resolución del dispositivo lo haremos al ancho de ventana del navegador. Bien, ahora vamos a dar un paso más y ver los media queries de CSS3 en acción. La mejor manera es con un ejemplo. Veamos:

|  |  |  |
| --- | --- | --- |
| 1 | @media screen and (min-width:320px) { | |
| 2 | /\* Código CSS \*/ |

|  |  |
| --- | --- |
| 3 | } |

Esta es una expresión que indica que si se cumplen las condiciones especificadas, entonces y sólo entonces aplique el código CSS que se encuentra en su interior. Las condiciones indicadas son las siguientes:

* **screen**- indica que el dispositivo de salida es una pantalla (y no una impresora u otro dispositivo).
* **min-width:320px** – indica que el ancho de ventana ha de ser como mínimo de 320px.

Dicho de otra manera, lo que hace el media query de CSS3 es preguntar lo siguiente: ¿estamos mostrando este contenido en una pantalla Y además la ventana actual tiene un ancho de mínimo 320px? Si la respuesta es afirmativa, entonces se aplicará el código CSS indicado. En otro caso, pasará de largo a la siguiente parte del documento CSS.

Seguramente te estés preguntando por qué min-width va entre paréntesis y screen no. La razón es porque, en este caso, min-width es una característica de screen. Aquí es donde CSS3 entra en juego, anteriormente sólo podíamos definir el dispositivo de salida, como una pantalla o una impresora, pero no podíamos definir sus características, como el ancho de ventana, el color o la orientación del dispositivo.  
Así, por ejemplo, si queremos aplicar un determinado código CSS para impresoras en blanco y negro utilizaremos este media query de CSS3:

|  |  |
| --- | --- |
| 1 | @media print and (monochrome) { } |

Con el CSS tradicional no habríamos podido especificar que la impresora fuera en blanco y negro, cosa que ahora gracias a los media queries de CSS3 si podemos hacer. Lo mismo para impresoras de color:

|  |  |
| --- | --- |
| 1 | @media print and (color) { } |

Para separar las diferentes condiciones utilizamos la palabra and y podemos especificar tantas como creamos necesarias. En muchas ocasiones utilizaremos tres, una que especificará que la condición se aplica sólo a ventana, otra para especificar el ancho mínimo y otra para especificar el ancho máximo. Veamos:

|  |  |
| --- | --- |
| 1 | @media screen and (min-width: 480px) and (max-width: 767px) { } |

En este caso, se aplicará el código CSS entre corchetes sólo cuando la página web en cuestión se muestre por pantalla y la ventana del navegador tenga un ancho entre 480px y 767px, ambos incluidos.  
Si, por ejemplo, queremos especificar la orientación del dispositivo, entonces utilizaremos este media query de CSS3:

|  |  |
| --- | --- |
| 1 | @media all and (orientation:landscape) { } |

Fíjate que en este caso hemos puesto all, pues nos sirve para cualquier dispositivo de salida.

INTERNET EXPLORER Y LOS MEDIA QUERIES DE CSS

Internet Explorer no da soporte a los media queries excepto IE9+ sí, así como la versión móvil del navegador. Pero, ¿qué hacemos para el resto? Nos guste o no es un hecho que todavía son muchos los usuarios que navegan con Internet Explorer.

Por suerte, hay una solución. Necesitas descargarte [Respond](https://github.com/scottjehl/Respond" \t "_blank). Lo tienes que añadir a tu carpeta de archivos JavaScript y entonces añadir esta línea de código dentro del head de tu página web, modificando la URL del archivo a la de tu carpeta:

|  |  |
| --- | --- |
| 1 | <script type=”text/javascript” src=”js/respond.min.js”></script> |

Una manera de comprobar que el navegador encuentra este archivo es simplemente dando al botón derecho en la página web, ir a mostrar código fuente y una vez allí hacer clic en el enlace, en el ejemplo js/respond.min.js. Si no encuentra el archivo es que no has especificado la ruta correcta a la carpeta donde se encuentra.

Respond es un script que minimizado ocupa sólo 3kb, por lo que es muy rápido y eficiente. Y con su uso te aseguras que el usuario que navega con IE7 o IE8 vea tu página web tal y como tú la has diseñado.

La regla @media:

Nos permite especificar que cierto conjunto de reglas CSS solo deben aplicarse acierto tipo de dispositivo.

Así las definiciones dentro del bloque de la regla @media screen { ... }solo serían interpretadas por dispositivos conectados a monitores de ordenador y los de la regla@media projection { ... } solo se aplicaría a proyectores.

CSS3 añade importantes y nuevas capacidades que nos permiten definir conjuntos de estilos dependiendo de propiedades comunes de los dispositivos que acceden a nuestros sitios.

Propiedades como el alto y el ancho o la relación de aspecto o el número de colores disponible.

Las reglas @media pueden ser utilizadas para adaptar nuestras páginas, no solo para dispositivos comunes, sino para todo tipo de dispositivos que nuestros lectores usen para visitar nuestras páginas.

### Media queries

Hasta ahora, si necesitábamos conocer el tamaño actual de la ventana del navegador, debíamos usar JavaScript para recolectar datos de ese tipo desde el navegador y después darle un uso a esos datos a través de la modificación del DOM a través de métodos programados en JavaScript. Aunque dicho método es válido, no es realmente óptimo ni intuitivo.

CSS3 nos aporta las media queries que nos proveen de una forma de conocer bastantes propiedades comunes de los dispositivos que nos visitan que podemos utilizar en nuestros archivos de estilo para construir entornos dependiendo de los mismos sin ayuda de JavaScript.

Aunque media queries dispone de muchas propiedades diferentes, básicamente viene con cinco sabores:

Aspect-ratio: Mira las dimensiones relativas del dispositivo expresadas como una relación de aspecto: 16:9 por ejemplo.

Width y height: Mira las dimensiones del área de visualización. Además pueden ser expresadas en valores mínimos y máximos.

Orientation: Mira si el layout es panorámico ( el ancho es mayor que el alto ) o vertical ( el alto es mayor que el ancho ). Esto nos permite ajustar los diseños para dispositivos con propiedades de giro de la pantalla como el iPhone, y otros smartphones y los tablets.

Resolution: Mira la densidad de los pixeles en el dispositivo de salida. Esto es especialmente útil cuando queremos aprovecharnos de las ventajas de los dispositivos que tiene una resolución mayor a 72 dpi.

Color, Color-index y monochrome: Encuentran el número de color o bits por color. Esto nos permite crear diseños específicos para dispositivos monocromáticos.

| Propiedad | Valor |  |
| --- | --- | --- |
| aspect-ratio | <relación de aspecto> |  |
| max-aspect-ratio | <relación de aspecto> |  |
| min-aspect-ratio | <relación de aspecto> |  |
| device-aspect-ratio | <relación de aspecto> |  |
| max-device-aspect-ratio | <relación de aspecto> |  |
| min-device-aspect-ratio | <relación de aspecto> |  |
| color | <entero> |  |
| max-color | <entero> |  |
| min-color | <entero> |  |
| color-index | <entero> |  |
| max-color-index | <entero> |  |
| min-color-index | <entero> |  |
| device-height | <tamaño> |  |
| max-device-height | <tamaño> |  |
| min-device-height | <tamaño> |  |
| device-height | <tamaño> |  |
| max-device-height | <tamaño> |  |
| min-device-height | <tamaño> |  |
| device-width | <tamaño> |  |
| max-device-width | <tamaño> |  |
| min-device-width | <tamaño> |  |
| height | <tamaño> |  |
| max-height | <tamaño> |  |
| min-height | <tamaño> |  |
| monochrome | <entero> |  |
| max-monochrome | <entero> |  |
| min-monochrome | <entero> |  |
| orientation | <portrait, landscape> |  |
| resolution | <resolución> |  |
| max-resolution | <resolución> |  |
| min-resolution | <resolución> |  |
| scan | <progressive, interlaced> |  |
| width | <tamaño> |  |
| max-width | <tamaño> |  |
| min-width | <tamaño> |  |
| \*\* Supongo que IE9 también soporta estos media queries pero no he podido encontrar información | | |

Los media queries actúan por defecto sobre la vista del navegador con la excepción de aquellos que especifican en su nombre la palabra device que actúan sobre toda la ventana del navegador o área de salida completa.

Usando media queries para especificar estilos

Antes de nada debemos crear nuestra hoja de estilos. Creamos un estilo por defecto que incorporará todas las necesidades universales para nuestro diseño y lo guardamos. Si nos sentimos especialmente originales, podemos llamar al archivo default.css:

|  |  |
| --- | --- |
| */\*\**  *\* Estilo por defecto*  *\*/*  body {  **background:** white url(../media/fondo.jpg) **no-repeat** 0 0;  **margin:** 0;  **padding:** 100px 0 0 75px;  }    h1 {  **color:** black;  **font-style:** **italic**;  }    h2 {  **color:** #717171;  }    p {  **font:** **normal** 100**%/**1**.**5 Helvetica**,** Arial**,** Sans**-serif**;  **color:** #313131;  } |  |

[view raw](https://gist.github.com/DamnWidget/1062446/raw/3790374437878da952f4196b969b6611d94bd106/default.css)[default.css](https://gist.github.com/DamnWidget/1062446#file-default-css) hosted with ❤ by [GitHub](https://github.com/)

Creamos archivos de estilo para cada tipo de dispositivo o media para el que queramos diseñar un estilo específico. Por ejemplo, podemos incluir un estilo específico para impresoras, si nos seguimos sintiendo igual de originales a este podemos llamarle, no se, print.css por ejemplo:

|  |  |
| --- | --- |
|  |  |

[view raw](https://gist.github.com/DamnWidget/1062457/raw/271efc04a3d0721255f4d3361718f04ca502740e/print.css)[print.css](https://gist.github.com/DamnWidget/1062457#file-print-css) hosted with ❤ by [GitHub](https://github.com/)

Ya tenemos un estilo por defecto y otro para cuando la página va a ser impresa. Vamos a añadir otro para el iPhone, y para no perder la originalidad que nos caracteriza, vamos a llamarle iphone.css:

|  |  |
| --- | --- |
| **<!DOCTYPE html>**  <html lang="es">  <head>  <meta http-equiv="Content-Type" content="text/html" charset="UTF-8">  <meta name="viewport" content="width=device-width; initial-scale=1.0; maximum-scale=1.0; user-scalable=0;">  <title>El Señor de los Poyaques</title>  <link rel="stylesheet" media="all" href="/css/default.css">  <link rel="stylesheet" media="print" href="/css/print.css">  <link rel="stylesheet" media="screen and (max-device-width: 480px) and (min-device-width: 320px;)" href="iphone.css">  </head>  <body>  <h1>La Comunidad del Tresillo</h1>  <article>  <header>  <h1><strong>Capítulo I.</strong> Chandal el Shurmano</h1>  </header>  <p>Cuando Frodo Bolson de Pipas vio acercarse por el camino el carruaje de  Chandal el Shurmano no pudo evitar soltar un gritito de alegría como  <em>¡Vaaamono preeeehmoh!</em></p>  </article>  <footer>  <p class="author">by J.R.R. Torke Mada</p>  </footer>  </body>  </html> |  |

[view raw](https://gist.github.com/DamnWidget/1062462/raw/ea1066e6eacec9b988eca40e09831624268400ea/iphone.css)[iphone.css](https://gist.github.com/DamnWidget/1062462#file-iphone-css) hosted with ❤ by [GitHub](https://github.com/)

Ahora vamos a añadir el meta tag de la vista en el head del documento HTML:

|  |  |
| --- | --- |
| 1 | <meta name="viewport" content="width=device-width; initial-scale=1.0; maximum-scale=1.0; user-scalable=0;" /> |

[view raw](https://gist.github.com/DamnWidget/1062469/raw/eaa0d0a7216dca562e1c61eddad65ad8290b908c/meta_tag.html)[meta\_tag.html](https://gist.github.com/DamnWidget/1062469#file-meta_tag-html) hosted with ❤ by [GitHub](https://github.com/)

Eso previene de que dispositivos con una pantalla más pequeña —como el iPhone— redimensionen la página impidiendo que tus estilos se interpreten. Enlazamos nuestro estilo por defecto en el documento HTML a través de la etiqueta link y utilizando como regla media all.

Hacemos lo propio con la hoja de estilo print.css pero en esta ocasión utilizamos la regla print. Lo mismo con la hoja de estilo para el iPhone, pero esta vez utilizaremos la regla screen y añadimos nuestros primer media queries en paréntesis conectando múltiples consultas con la palabra and.

El código completo del documento HTML sería como el que sigue:

|  |  |
| --- | --- |
| 12345678910111213141516171819202122232425 | <!DOCTYPE html>  <html lang="es">  <head>  <meta http-equiv="Content-Type" content="text/html" charset="UTF-8">  <meta name="viewport" content="width=device-width; initial-scale=1.0; maximum-scale=1.0; user-scalable=0;">  <title>El Señor de los Poyaques</title>  <link rel="stylesheet" media="all" href="/css/default.css">  <link rel="stylesheet" media="print" href="/css/print.css">  <link rel="stylesheet" media="screen and (max-device-width: 480px) and (min-device-width: 320px;)" href="iphone.css">  </head>  <body>  <h1>La Comunidad del Tresillo</h1>  <article>  <header>  <h1><strong>Capítulo I.</strong> Chandal el Shurmano</h1>  </header>  <p>Cuando Frodo Bolson de Pipas vio acercarse por el camino el carruaje de  Chandal el Shurmano no pudo evitar soltar un gritito de alegría como  <em>¡Vaaamono preeeehmoh!</em></p>  </article>  <footer>  <p class="author">by J.R.R. Torke Mada</p>  </footer>  </body>  </html> |

[view raw](https://gist.github.com/DamnWidget/1062484/raw/cdb182f5d72b1a0a493e92b6065bab8a30f5e798/intro_css3_media.html)[intro\_css3\_media.html](https://gist.github.com/DamnWidget/1062484#file-intro_css3_media-html) hosted with ❤ by [GitHub](https://github.com/)

La regla @media

La regla @media permite que empotremos media queries directamente en nuestras hojas de estilo. De esta manera mejoramos el rendimiento de la página al no tener que cargar archivos CSSadicionales para los diferentes dispositivos.

Para utilizar las reglas @media lo único que tenemos que hacer es crear nuestra hoja de estilos y cuando queramos definir un conjunto de reglas de estilo específicas para un dispositivo determinado utilizar la sintaxis:  
@media screen and ( blablabla ) and ( bla bla bla ) { ... }

El efecto es el mismo pero minimizamos la carga de archivos a un único archivo que incluya todas nuestras reglas CSS.