**Responsive Web Design**

**Qué es Responsive Web Design**

Es la técnica que nos permite crear sitios adaptables a las condiciones del ordenador o dispositivo desde donde se van a acceder.

**Consejos para que el diseño web adaptable sea más fácil de producir**

No uses estilos in-line: Son aquellos que se colocan en el propio HTML, sobre todo en el atributo style de la etiqueta, como align="center".

**La web no necesariamente se debe ver igual en todos los dispositivos y navegadores.**

**No diseñes para una plataforma.**

**Usar el Javascript para lo que es;**

Por ejemplo, una animación Javascript en vez de una animación CSS. O un diseño de filas en formato cebra, alternando colores. Si se puede hacer con CSS es preferible que apliques CSS.

**¿Ya te hemos dicho que no uses tablas para maquetar?:**

No se debe maquetar con tablas. En vez de eso, usa contenedores como DIV, o si encajan todavía es mejor usar los semánticos como ARTICLE, SECTION, MAIN, etc.

DIV o cualquier otra etiqueta que englobe contenido, puede tener un comportamiento de tabla, si es que lo necesitas para tu maquetación para hacer cosas como alineado vertical. En ese caso, puedes mediante CSS asignar ese comportamiento con display: table y compañía.

**Unidades relativas:**

Unidades de CSS de las relativas, como %, em, rem. Esto te facilitará la asignación de espacios y tamaños más que las unidades absolutas como px, cm, pt…

Por ejemplo, si asignas a una caja una anchura de 600px ¿qué crees que pasará si la ves en un dispositivo que tiene una pantalla de 320px de ancho?. Si en vez de ello hubieses definido esa anchura como 80%, no habrías tenido quizás problemas.

**Los tres pilares de optimización del diseño responsive**

Cuando diseñamos una web debemos fijarnos en tres cosas fundamentales, diseño para todos los navegadores y sistemas, todas las resoluciones de pantalla y todas las velocidades de conexión.

**TODOS LOS NAVEGADORES Y SISTEMAS**

Hay multitud aspectos que deben ser tenidos en cuenta, pero en general nos debemos de asegurar que el contenido sea accesible en todos los browsers.

**TODAS LAS RESOLUCIONES Y TAMAÑOS DE PANTALLA**

Usar las media queries para que todos los ordenadores y móviles vean la web con los elementos dispuestos de manera que se facilite la lectura.

**TODAS LAS VELOCIDADES DE CONEXIÓN**

Las personas esperan de media tan solo 5 segundos para desistir en el acceso a una web.

**Procedimientos habituales en el diseño adaptable**

**Pasos para diseñar un sitio responsive**

**1) CREAR UN HTML CON EL CONTENIDO QUE DESEAMOS MOSTRAR**

Como sabes, el contenido se escribe con HTML y debe ser común para todos los tipos de dispositivos e incluso para los ordenadores con pantallas enormes.

**2) APLICAR CSS**

**A) Recursos de diseño estéticos:** Aquellos estilos que se agregan para mejorar la estética de un sitio, como fuentes tipográficas especiales, cajas con esquinas redondeadas, sombras en textos o cajas, etc. este tipo de elementos no requieren una especial adaptabilidad.

**B) Diseño de layout:** Son aquellos estilos que se encargan de presentar la información con una estructura definida, por columnas, cabecera, pie, etc.

**Progressive Enhancement Vs Graceful Degradation**

Bajo ese prisma podríamos decir que Progressive Enhancement mira hacia adelante en el flujo de desarrollo, mientras que Graceful Degradation empieza delante y mira hacia atrás. Por tanto, mirar hacia el futuro nunca está de más.

Técnicamente y en términos de diseño CSS añadir una capa por encima no es más que añadir al final del código CSS nuevos estilos. Al estar al final del código CSS, y debido a la cascada, esa capa de estilos nuevos mandarán sobre los anteriores, aunque solo los interpretarán los navegadores nuevos, que los conocen. Los antiguos simplemente los ignorarán.

**Mobile First**

Lo hacen en el sentido de diseño "Mobile First" o "Desktop First", osea, si se diseña pensando en el móvil al principio y luego en ordenadores de escritorio.

**CONTENIDO**

En un móvil no te cabe toda la información del mundo y quizás lo podemos sustituir por un icono de "opciones" y, una vez pulsado que se vean aquellas cosas que no entran en la página para móviles.

**DISEÑO**

Es más fácil diseñar en pequeñito al principio. Cuando tienes más espacio verás que tus cosas siguen cabiendo. Si diseñas en grande, cuando tengas menos espacio las cosas no te entrarán y seguramente tendrás que hacer muchos más esfuerzos para acomodarlas. En este punto todos los diseñadores están de acuerdo, así que no lo pienses mucho, comienza diseñando para móviles.

**MediaQueries**

**CSS Media Queries**

Sirven para definir estilos diferentes para distintos tamaños de la pantalla. Son sencillas de entender y aplicar, aunque el estándar es bastante sofisticado, con diversas posibilidades.

Una construcción condicional como esta es tan útil y básica que, aunque CSS no sea un lenguaje de programación, necesita incorporarlas. Ejemplos de casos en los que nos vendría bien un condicional:

- Si la pantalla del usuario tiene estas características, entonces aplica estos estilos.

- Si se imprime el documento en la impresora, aplica estos estilos.

- Si la pantalla del dispositivo tiene estas dimensiones y además está situado en posición horizontal (landscape), entonces aplica este CSS.

**Alternativa 1:**

La primera alternativa de las Media Queries es a través del atributo media de la etiqueta LINK. Como sabemos, esa etiqueta es la que se usa para enlazar una hoja de estilo con un documento HTML. En ese enlace podemos especificar condicionales que deben cumplirse para que los estilos enlazados se apliquen. Por ejemplo, que se esté imprimiendo un documento o que la pantalla tenga cierta anchura mínima. Recordamos, la etiqueta LINK tiene esta forma:

***<link rel="stylesheet" href="estilos-generales.css">***

Pues ahora simplemente le podemos agregar el atributo "media" indicando la condición que se debe cumplir para que estos estilos se apliquen:

***<link rel="stylesheet" href="estilo-imprimir.css" media="print">***

Este atributo media="print" quiere decir que los estilos deben aplicarse sólo cuando la página se están mostrando para la impresión.

***<link rel="stylesheet" href="estilo-pantallas-grandes.css" media="(min-width:1200px)">***

Quiere decir que esos estilos deben aplicarse sólo cuando la pantalla del usuario (En caso de ordenadores de escritorio, la ventana del navegador) tenga una anchura mínima de 1200 píxeles.

El problema de escribir tus Media Queries así es que tienes archivos de CSS separados. Es decir, aquellos estilos para impresión o para pantallas de 1200px están en archivos independientes, lo que es sencillo de administrar para nosotros, pero una mala práctica en términos de optimización de la web, puesto que se tienen que realizarse varias solicitudes al servidor distintas para traerse los CSS.

**Alternativa 2:**

Consiste en incorporar los estilos en una construcción @media donde se apliquen entre llaves todos los estilos que queremos para una consulta de medio dada.

**@media (min-width: 500px) {**

**h1{**

**margin: 1%;**

**}**

**.estiloresponsive{**

**float: right;**

**padding-left: 15px;**

**}**

**}**

La sentencia @media en la que podemos indicar entre paréntesis las condiciones que deben cumplirse para que se aplique esta media query. En este caso será para pantallas que tengan una anchura mínima de 500 píxeles.

Luego entre llaves colocamos todas las reglas y atributos de estilos CSS que necesitemos aplicar en esta situación. Las reglas de estilos son las mismas que pondrías fuera de la estructura condicional. Cuando la sentencia entre paréntesis se evalúe como verdadera, se aplicarán todas ellas.

**Operadores lógicos para las Media Queries**

\* Operador and: \* Operador not: \* Operador only: \* Operador or:

**@media (max-width: 600px) and (orientation: landscape) {**

**h1{**

**color: red;**

**}**

**}**

Esta regla se aplicaría para pantallas con una anchura máxima de 600 píxeles y cuando la orientación está en horizontal.

*Nota: Ten en cuenta que la mayoría de smartphones simulan tamaños de pantalla mayores, haciendo una especie de dimensiones virtuales que faciliten la lectura de webs que no están diseñadas para Responsive Web Design. Por ello, lo más seguro es que tengas que poner el "viewport" en el documento HTML a algo como:*

**<meta name="viewport" content="width=device-width, initial-scale=1.0" />**

Sin ese viewport tu móvil simularía que tiene unas dimensiones de unos 980 píxeles, cuando quizás la pantalla solo tenga una anchura real de 320 o 500 píxeles.

Para ver en funcionamiento la parte del orientation (landscape o portrait), tienes que usar un móvil o Tablet y cambiar la posición de la pantalla, para que esté en horizontal o vertical.

*Nota: Si tienes un ordenador que no reconozca el cambio de orientación la posición siempre será considerada será landscape.*

**@media tv and (min-width: 1200px){**

**h1{**

**margin: 10%;**

**}**

**}**

Esta regla aplicaría en dispositivos de tipo televisión y cuya resolución minima de anchura sea de 1200 píxeles.

**@media (min-width: 600px), handheld and (orientation: portrait) {**

**h1{**

**color: green;**

**}**

**}**

Este mediaquery servirá para pantallas de minimas dimensiones 600 píxel y también para todos aquellos dispositivos handheld que estén en posición vertical.

*Nota: Handheld es un término inglés que sirve para especificar aquellos dispositivos que son de mano. Pequeños ordenadores que se llevan en la mano, como los palm o agendas electrónicas que aparecieron antes de popularizarse los smartphones o tablets. Por mis pruebas handheld no aplica a los móviles o tablets.*

[*https://desarrolloweb.com/articulos/guias-uso-media-queries-rwd.html*](https://desarrolloweb.com/articulos/guias-uso-media-queries-rwd.html)

**Guías de uso de las Media Queries en Responsive Web Design**

**Configurar un buen Viewport**

[*https://desarrolloweb.com/articulos/etiqueta-meta-viewport.html*](https://desarrolloweb.com/articulos/etiqueta-meta-viewport.html)

De momento te puedes quedar con que el Viewport sirve para que la página no se muestre en pequeñito en el dispositivo (Habrás apreciado que cuando entras en un sitio que no es "responsive" se ve todo muy pequeño).

**El orden de colocación de los mediaqueries**

1. Colocar todos los estilos a aplicar de manera global, fuera de cualquier MQ. Los estilos que coloques sin MQ serán como "MQ globales" que afectarán a todos los dispositivos, con todas las anchuras de pantalla.

2. Luego colocarás, como primer MQ, el estilo necesario para los dispositivos de la siguiente anchura de pantalla que necesites.

3. Irás colocando los MQ en orden de anchuras de pantalla ascendentes. Terminarás siempre por el MQ que coloques para las anchuras de pantallas mayores, ordenadores con pantallas de alta resolución.

**Rangos de media queries**

Puedes perfectamente especificar rangos de media queries. Para ello usas min-width y max-width a la vez. Quizás no es algo muy habitual, pero lo mencionamos por si tuvieras necesidad de ello. La sintaxis sería la siguiente:

@media (min-width: 700px) and (max-width: 800px){

.lateral{

width: 33%;

float: right;

background-color: #6ee;

}

}

**Cómo saber el lugar donde colocar un breakpoint**

Si no haces breakpoints orientados a dispositivos, ¿cómo saber dónde colocarlos? La respuesta te la da tu diseño. En otras palabras, breakpoint debe estar orientado al diseño particular de tu web.

**Medidas fijas y medidas relativas**

**Las siguientes unidades son unidades fijas:**

px: Píxeles in: Pulgadas (1 in es igual a 96px) pt: Puntos (1 pt es igual a 1/72 in) cm: Centímetros mm: Milímetros pc: Picas

**Unidades relativas:**

Si quieres que un elemento tenga 200 píxeles de ancho y las dimensiones del contenedor en un momento dado son de 1000 píxeles, el valor en % será de 200/1000 = 0.2 \* 100 = 20%

Si quieres que un elemento tenga 200 píxeles y el tamaño de la fuente en su contenedor es de 16 píxeles, en em tendrías que darle el valor 200 / 16 = 12.5em

**Unidades de medida CSS más adecuadas para el Responsive Web Design**

**Cómo funcionan las unidades Rem**

Rem no funciona de manera relativa a su contenedor, sino de manera relativa al tamaño definido en la raíz.

De esa forma, cuando quiero calcular una medida de un elemento, no tengo que pensar en el tamaño de su padre o en el tamaño del padre del padre.

Si en el elemento BODY teníamos 16px de tamaño y quiero 24px en el H1, tengo que calcular 24/16 = 1.5rem. Si luego en el enlace que había dentro del H1 quiero que sea 12px, tengo que calcular como context 16px (que era el equivalente al 100% definido como tamaño de la raíz) y no el tamaño del H1. O sea, en este caso sería 12/16 = 0.75rem.

**Compatibilidad con las unidades de medida Rem y fallback**

Los IE antiguos no son compatibles con las unidades rem. La solución es sencilla y se realiza por lo que se denomina fallback, que se basa en especificar código alternativo en caso de que no funcione el original. En este caso el fallback es tan sencillo como especificar las medidas en dos unidades.

font size: 16px;

font size: 1rem;

**Unidades CSS viewportwidth y viewportheight**

viewport es igual a las dimensiones de la pantalla de tu móvil o tableta, mientras que en ordenadores de escritorio serían las dimensiones de la ventana del navegador.

viewportheight, bastante importante, es que solo funciona en dispositivos móviles y no en ordenadores de escritorio.

body{

font size: 2.5vw

}

h1{

font size: 4.8vh

}

Es una maravilla que podemos usar y al menos en dispositivos nos dará la tranquilidad de que esas fuentes se van a ver igual de grandes, relativamente al tamaño de la pantalla donde se esté mostrando la página.

**Practicando con CSS para Responsive Web Design**

El atributo box-sizing, aparecido en la especificación CSS3, nos permite indicar cómo deseamos que se interprete el modelo de caja, evitando que cada navegador lo entienda a su modo.

box-sizing: border-box

Ese atributo lo tendrás que especificar para todos los elementos de la página, así que puedes usar el selector \* para seleccionarlos a todos.

**Maquetación Flexbox**

Para maquetar de manera mucho más versátil, rápida y evitando muchos de los problemas habituales del posicionamiento de elementos, es usar las técnicas Flexbox.

**Atributos min-width y max-width / min-height y max-height**

Los contenedores se estiran y se encogen, para adaptarse a las dimensiones de las ventanas o pantallas.

.dimensiones-maximas{

max-width: 1600px;

}

Una imagen que se adapta al ancho de un contenedor pero no quieres que llegue a ser tan grande que se sobrepase su resolución.

.img-banner-aside{

width: 100%;

max-width: 300px;

}

La altura necesaria para que quepa la descripción del artículo, pero unos tiene más texto que otros y por tanto puede ocurrir que las alturas queden descompensadas.

.article-home{

background-color: #ddd;

min-height: 225px;

}

**Etiqueta meta Viewport**

El Viewport es una de las etiquetas más representativas de la web móvil, que nos permite configurar cómo debe interpretar una página el navegador web para móviles.

Es altamente recomendable que se altere la etiqueta viewport para conseguir que tu navegador se comporte como tú deseas, especialmente en el caso de las páginas que estamos diseñando para verse correctamente en pantallas pequeñas. Para ello disponemos de los siguientes parámetros en la etiqueta META.

Width: anchura virtual (emulada) de la pantalla o anchura del viewport.

Height: altura virtual de la pantalla o anchura del viewport.

Initial-scale: la escala inicial del documento.

Minimum-scale: la escala mínima que se puede poner en el documento.

Maximum-scale: la escala máxima configurable en el documento.

User-scalable: si se permite o no al usuario hacer zoom.

Como puedes ver, en la META viewport no se indica simplemente las dimensiones de la pantalla emulada, sino también el nivel de zoom que se puede estar configurando inicialmente y el nivel de zoom que se permitiría tener.

Un ejemplo de etiqueta viewport sería el siguiente:

<meta name="viewport" content="user-scalable=no, width=device-width, initial-scale=1">

**Imágenes responsive con la etiqueta Picture**

Cómo usar la etiqueta Picture para crear imágenes responsive, un nuevo elemento de HTML5 que permite que las imágenes y su resolución se adapten a todo tipo de pantallas.

El objetivo, en resumidas cuentas es el siguiente:

Poder indicar una imagen y varios archivos de alternativa para distintos escenarios: pantallas pequeñas, medianas o grandes, velocidades de conexión, etc.

Que el navegador sólo descargue una de las imágenes posibles y no todas. O sea, si una imagen tiene tres alternativas de tamaños, se le entregue al navegador solamente una de ellas, ahorrando la descarga de los otros tamaños alternativos que no necesita.

<picture>

<source media="(min-width: 900px)" srcset="grande.png">

<source media="(min-width: 550px)" srcset="media.png">

<img src="archivo-pequeno.jpg">

</picture>

**SOURCE element**

Básicamente nos permite indicar varios archivos gráficos. El navegador irá leyéndolos uno a uno, desde el primero hasta el último, en orden de colocación, hasta que encuentre uno que sea posible usar.

Usará el primero que sea posible, descartando los demás que pueda haber por debajo.

Incluye los siguientes atributos.

srcset: Contiene el archivo de imagen que debe visualizarse. Es un campo requerido.

<source media="(max-width: 300px)" srcset="archivo-hasta-300px.jpg">

<source media="(min-width: 900px)" srcset="foto.jpg, foto-hd.jpg x2"

Sabiendo esto, podría ser interesante quitar el mediaquery de algún SOURCE, para que se use ese de manera predeterminada. En ese caso lo lógico sería colocar el SOURCE sin mediaquery al final del todo.

<picture>

<source media="(min-width: 900px)" srcset="grande.png">

<source media="(min-width: 550px)" srcset="media.png">

<source srcset="peque.png">

<img src="predeterminado.png">

</picture>

**Sizes**: Este atributo nos ayudará a realizar una selección de la imagen apropiada de una manera distinta.

Este atributo se combina con el atributo de antes, "srcset" para indicar las distintas alternativas de archivos gráficos.

<picture>

<source media="(min-width: 600px)" sizes="65vw" srcset="peque.png 256w, media.png 456w, grande.png 856w, gigante.png 1280w">

<img src="media.png" sizes="65vw" srcset="peque.png 256w, media.png 456w, grande.png 856w, gigante.png 1280w">

</picture>

**type**: Este atributo permite indicar el tipo de una imagen, de modo que el navegador pueda seleccionarlo si es que conoce ese formato de archivo.