CSS Media Queries

Reglas especiales para establecer excepciones en CSS

Una vez nos adentramos en el mundo del **Responsive Design**, nos damos cuenta en que hay situaciones en las que determinados aspectos o componentes visuales deben aparecer con ciertas diferencias dependiendo del dispositivo donde se están visualizando, ya que no todos los dispositivos tienen los mismos tamaños o características.

¿Qué son las Media Queries?

Existe un concepto denominado **media queries**, mediante el cual podemos hacer excepciones para que unos determinados estilos de diseño sólo se apliquen si se cumplen una serie de condiciones, generalmente relacionadas con el dispositivo o navegador mediante el cual se está viendo la página.

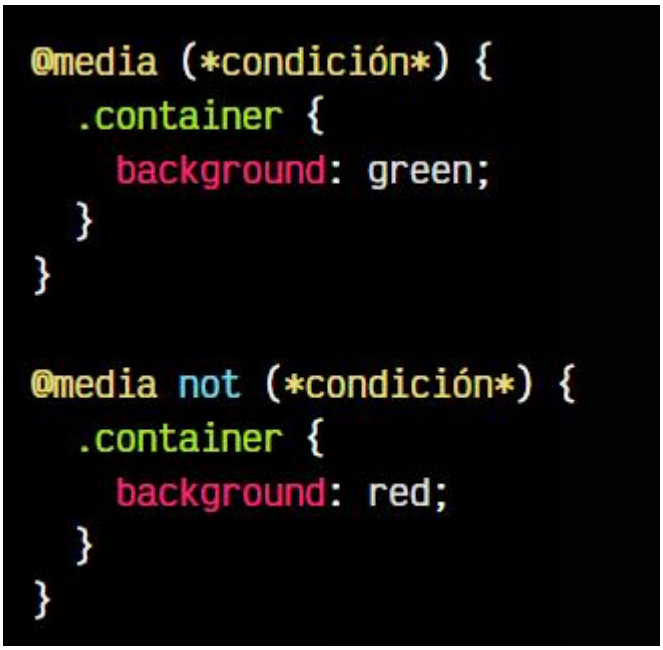
A fin de cuentas, se trata de una especie de condicionales de diseño, en las que si se cumple la condición se aplicarán unos estilos, y en caso contrario, se aplicarán otros.

La regla @media

Las reglas **media queries** (abreviadas como **MQ**) se indican en el código mediante la regla @media, indicando la condición en cuestión entre paréntesis. La sintaxis sería la siguiente:

Regla	Descripción
@media (<condición>)	Si se cumple la condición, se aplican los estilos de su interior.
@media not (<condición>)	Si no se cumple la condición, se aplican los estilos de su interior.
@media only (<condición>)	Si se cumple la condición y es un navegador moderno, se aplican los estilos.
@media (<condición>) and (<condición>)	Se se cumplen ambas condiciones, se aplican los estilos.

Observa que, de utilizar la palabra clave not antes de los paréntesis, invertimos la condición. Así pues, veamos un pequeño ejemplo donde escribimos las dos opciones anteriores, pero sin entrar en detalles aún en la condición:



En el ejemplo anterior, si se cumple la condición establecida, se aplicará un color verde. Sin embargo, si no se cumple, se aplicará un color rojo. Recuerda que es similar al funcionamiento de un if / else en programación.

No olvides que al escribir una regla @media podrías estar sobreeescribiendo los estilos CSS en otro fragmento posterior. Una buena forma de empezar a escribir **MQ** sería escribir las reglas @media siempre al final, como excepciones al código anterior.

Si lo deseas, es posible establecer múltiples condiciones en las reglas @media. De esta forma, se pueden conseguir situaciones mucho más específicas y flexibles. Ten mucho cuidado si aplicas el not en las condiciones, no sea que niegues de forma incorrecta los casos deseados:

Condiciones de Media Queries

En los ejemplos anteriores hemos indicado *condiciones* en el interior de los paréntesis, pero no hemos visto como definir ninguna. Vamos a profundizar en esto.

Media Queries (range syntax)

Aunque existen otras formas, hoy en día la forma preferida de escribir Media Queries es utilizando la modalidad de rangos de condiciones (Media Query Range Syntax), mucho más versátiles que las sintaxis anteriores, y mucho menos tediosas.

Para ello, vamos a escribir las condiciones utilizando operadores de comparación como <, <=, > o >=:

Ejemplo 1:

Html

```
<div class="container">
  <p>Redimensiona el navegador o mueve abajo a la derecha.</p>
  <ul>
    <li>Dispositivos ⩽ 550px → Lila</li>
    <li>Dispositivos entre 551px y 749px → Gris</li>
    <li>Dispositivos ⩾ 750px → Verde</li>
  </ul>
  <p>¡No te olvides de seguir a Manz en sus redes sociales!</p>
</div>
```

Css

```css
.container {
  background: var(--color, grey);
  min-height: 200px;
  padding: 1rem 2rem;
}

@media (width ⩽ 550px) {
  .container {
    --color: indigo;
  }
}

@media (width ⩾ 750px) {
  .container {
    --color: green;
  }
}
```

Observa que en el caso de que la ventana del navegador del dispositivo sea igual o más pequeña que 550px, se verá el fondo de color lila, y en el caso de que sea mayor o igual que 750px se verá verde.

Si la ventana tuviera entre 551px y 749px, no se aplicaría ninguna media query (no cumplen las condiciones) y se quedaría de color gris, que es el valor que tiene indicado por defecto.

```css
.container {
  background: var(--color, grey);
  min-height: 200px;
  padding: 1rem 2rem;

  @media (width ⩽ 550px) { --color: indigo; }
  @media (width ⩾ 750px) { --color: green; }
}
```

Como puedes ver, exactamente equivalente al código anterior, pero más compacto y legible. Ten en cuenta que con la sintaxis de rangos también puedes hacer condiciones múltiples de este estilo, en el que el código CSS se aplica sólo si el navegador tiene un ancho de pantalla entre 400px y 800px:

```css
@media (400px ⩽ width ⩽ 800px) {
  /* Código CSS */
}
```

Esta serie de características de sintaxis junto a poder utilizar los operadores >, >=, <, <= en las condiciones de los media queries, hace mucho más intuitivo el escribir MQ CSS.

Tipos de medios

En algunas ocasiones, queremos indicar que las reglas @media sólo se pongan en funcionamiento en determinados tipos de dispositivos. Son los llamados **tipos de medios**, que pueden utilizarse en las condiciones de los media queries. Existen los siguientes:

Tipo de medio	Significado
all	Todos los dispositivos o medios. El que se utiliza **por defecto**.
screen	Monitores o pantallas de ordenador. Es el más común.
print	Documentos de medios impresos o pantallas de previsualización de impresión.

Tipo de medio	Significado
speech	Lectores de texto para invidentes (Antes aural, el cuál ya está obsoleto).

Estos tipos de medios se pueden indicar como una condición más, de modo que podría quedar de la siguiente forma:

```
/* Pantallas menores de 600px */
@media screen and (width ⩽ 600px) {
  /* ... */
}

/* Previsualizaciones y formatos de impresión */
@media print {
  /* ... */
}
```

Viewport (Región visible)

Cuando hablamos de **Responsive Design** muchas veces haremos referencia al **viewport** (región visible del navegador). Recordemos que con el siguiente fragmento de código HTML estamos indicando que hay que preparar el navegador para el **Responsive** y que el nuevo ancho de la pantalla será **el ancho del dispositivo**, por lo que el aspecto del viewport se va a adaptar consecuentemente:

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

Con esto conseguiremos preparar nuestra web para dispositivos móviles y prepararnos para la introducción de reglas **media query** en el documento CSS. Es importante no olvidar este paso.

```
<link rel="stylesheet"
      href="mobile.css"
      media="(max-width: 640px)">

<link rel="stylesheet"
      href="tablet.css"
      media="(min-width: 640px) and (max-width: 1280px)">

<link rel="stylesheet"
      href="desktop.css"
      media="(min-width: 1280px)">
```

Observa, sin embargo, que en este caso, el código CSS de las diferentes condiciones queda en un archivo .css diferente, sin embargo, el navegador al cargar la página los descargará todos y los aplicará cuando sea necesario, al igual que lo hace en los ficheros .css.

Ejemplo 2:

Html

```html
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" name="viewport" content="width=device-width, initial-scale=1">
        <link rel="stylesheet" href="est.css">
        <title>Responsivo</title>
    </head>
    <body>
        <h2>Cambiar el tamaño de fuente de un elemento en diferentes tamaños de pantalla</h2>
        <div class="example">Ejemplo DIV.</div>
        <p>Si el ancho del navegador es de 600 px o menos, configure el tamaño de fuente del DIV en 30 px. Si es de 601 px o más,
            configure el tamaño de fuente en 80 px. Redimensione la ventana del navegador para ver el efecto.</p>
    </body>
</html>
```

Css

```css
div.example {
  background-color: lightgrey;
  padding: 20px;
}

@media screen and (min-width: 600px) {
  div.example {
    font-size: 80px;
    color: white;
    background-color: blue;
  }
}

@media screen and (max-width: 600px) {
  div.example {
    font-size: 30px;
    color: white;
    background-color: black;
  }
}
```