

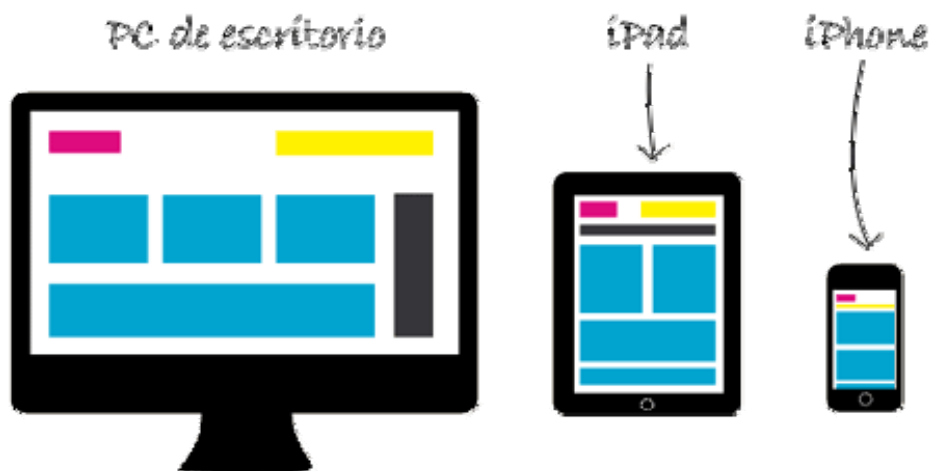
El concepto de media queries lo podemos definir en español como “consulta a medios”, y eso es lo que realmente trabajamos con esta funcionalidad de css3. Hoy en día vemos la gran evolución de la comunicación, en especial la web, últimamente se ha incrementado de una manera considerable la manera en que personas acceden a la web, hoy se puede acceder desde televisores, computadores, celulares, tabletas e incluso consolas de videojuegos. Lo que se convierte en un reto para el diseñador o desarrollador web, hacer que la web se visualice bien en los diferentes dispositivos y diferentes resoluciones de pantalla, desde un Smartphone hasta un televisor de gran resolución.

Esto lo podemos resolver gracias a los media queries, que nos permiten personalizar los estilos basándonos en las características del dispositivo del visitante de nuestra web.

Soporte de media queries en los diferentes navegadores.



Visualización de un diseño web utilizando media queries



Pero veamos cómo podemos añadir esos estilos condicionales y la consulta queries que si se da la condición desatara dichos estilos.

Una de las formas para hacer esto es añadiendo la regla dentro de nuestro css principal o también creando una hoja independiente y luego adjuntarla. Veamos la regla condicional que en caso de cumplirse ejecutara ciertos estilos css.

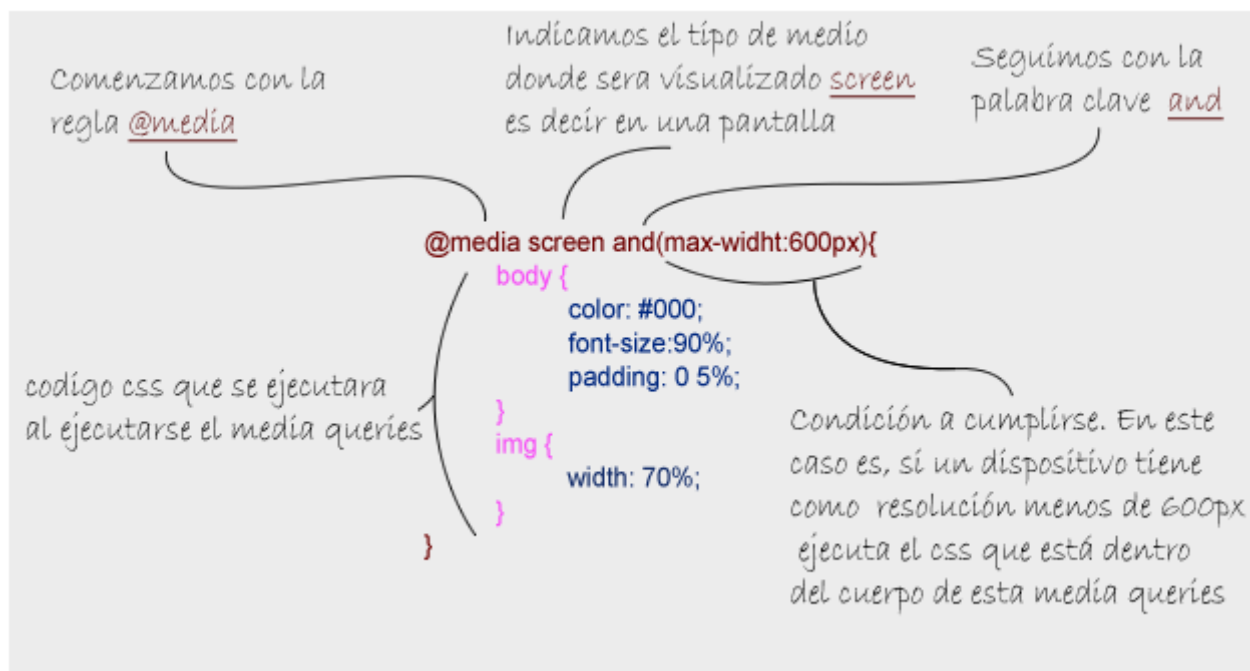
```
1 @media screen and(max-width:600px) {  
2   body {  
3     color: #000;
```

```

4    font-size:90%;
5    padding: 0 5%;
6  }
7  }

```

Analicemos la sintaxis de la ccs media querie.



Lo que en este caso debemos de tener presente es la función o condicional que al cumplirse desatara y ejecuta el css preparado para el dispositivo en cuestión.

En el caso anterior era `max-width: 600px`, es decir cualquier persona que habrá la web desde un dispositivo que la resolución sea menor de 600px automáticamente ejecutara el css que está en su cuerpo. Ahora, hay muchas más opciones de condicionales que funcionan con media queries, y es lo que denominamos los **Media features**, miremos cuales podemos utilizar para ejecutar css condicionales.

1 – width

```
1 @media screen and (min-width: 400px) and (max-width: 700px) { ... }
```

media queries que funcionara si la web se abre en un dispositivo entre los rangos de 400 y 700 px. `max-width` es el ancho maximo y `min-width` el ancho minimo.

2 – height

```
1 @media screen and (height: 700px) { ... }
```

Con esta función especificamos la altura de visualización del dispositivo, acepta los prefijos min/max.

3 – device-width

1@media screen and (device-width: 800px) { ... }

Especificamos el ancho del área de representación, la anchura de la hoja web, acepta los prefijos min/max.

4 – device-height

1@media screen and (device-height: 600px) { ... }

Especificamos la altura del área de representación, la altura de la hoja web, acepta los prefijos min/max.

5 – orientation

1@media all and (orientation: portrait) { ... }

2@media all and (orientation: landscape) { ... }

Especificamos la Orientación del dispositivo, podemos tener como valores **portrait** o **landscape**.

La orientación es **portrait** cuando el media feature width es igual o mayor a la media feature height, de lo contrario la orientación es **landscape**.

6 – aspect-ratio

1@media screen and (aspect-ratio: 1280/720)

Especificamos la relación del dispositivo entre la media feature width y la media feature height, acepta los prefijos min/max.

7 – device-aspect-ratio

1@media screen and (device-aspect-ratio: 1280/720) { ... }

Especificamos la relación del dispositivo entre la media feature device-width y la media feature device-height, acepta los prefijos min/max.

8 – color

1@media all and (min-color: 2) { ... }

especificamos el numero de bits por componentes del color,

La anterior media queries aplica para los dispositivos con 2 o más bits como componentes de color, acepta los prefijos min/max.

9 – color-index

1@media all and (min-color-index: 1) { ... }

Con esto especificamos el número de entradas en la tabla de conversión de color del dispositivo de salida, si el dispositivo no utiliza una tabla de conversión de color, su valor será cero, acepta los prefijos min/max.

10 – monochrome

```
1@media all and (min-monochrome: 2) { ... }
```

Básicamente con esta función especificamos que el dispositivo es a blanco y negro, y lo especificamos por el número de bits por pixel, acepta los prefijos min/max.

```
1@media all and (monochrome) { ... }
```

```
2@media all and (min-monochrome: 1) { ... }
```

En el anterior código especificamos 2 maneras de aplicar los css especiales por media queries a todos los dispositivos monochrome es decir a blanco y negro.

11 – resolution

```
1@media print and (min-resolution: 300dpi) { ... }
```

Especificamos la densidad de pixeles de un dispositivo de salida, acepta los prefijos min/max.

Por ejemplo en el código anterior especificamos los css para los dispositivos con una resolución mínima de 300 ppp “puntos por pulgadas”

12 – scan

```
1@media tv and (scan: progressive) { ... }
```

```
2/* o */
```

```
3@media tv and (scan: interlace) { ... }
```

Especificamos los css para los procesos de escaneados utilizados por los dispositivos de salida de televisión, los valores pueden ser progressive o interlace.

En resumen.

En css 2.1 utilizamos los media types screen y print para aplicar estilos diferentes a la web, ahora en CSS3 tenemos la opción de Media queries, que es mucho más abundante en funcionalidades, ya que nos va a permitir crear estilos css alternos para los diferentes dispositivos que pueden conectarse a internet y visualizar una web, ya no hay motivos para no hacer que nuestra web se vea bien en todas las pantallas.

Para más información sobre este tema: <http://www.w3.org/TR/css3-mediaqueries/>

<http://screensiz.es>

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



320px



768px



1024px



1068px

Smartphone

Tablets

Portátiles

Monitores