

Diseño de Interfaces Web

UD - 5

Mobile First

Contenidos











ÍNDICE

Introducción	3
El viewPort	
Regla @media	4
EJEMPLOS @ media	
¿Qué significa Mobile First?	
¿Que puntos de ruptura utilizar? - Breakpoint	
Hojas de estilo diferentes	<i>t</i>
Grid View	
¿Cómo construir un Grid View?	7

1. INTRODUCCIÓN

El diseño web resposivo hace que su página web se vea bien en todos los dispositivos, utilizando las tecnologías HTML y CSS, para realizar estas adaptaciones.

Las páginas web se pueden ver usando muchos dispositivos diferentes:

Escritorio vs Tabletas vs Teléfonos.

La página web debe verse bien y ser fácil de usar, independientemente del dispositivo. No deben dejar de lado información para adaptarse a dispositivos más pequeños, sino adaptar su contenido para adaptarse a cualquier dispositivo:







1.1 El viewPort

La ventana gráfica es el área visible del usuario en una página web, varía según el dispositivo y será más pequeña en un teléfono móvil que en la pantalla de una computadora. HTML5 introdujo un método para permitir que los diseñadores web tomen el control de la ventana gráfica a través de la etiqueta <meta>.

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

Esto le da al navegador instrucciones sobre cómo controlar las dimensiones y la escala de la página.

- width=device-width
 - Establece el ancho de la página para seguir el ancho de pantalla del dispositivo.
- Initial-scale=1.0

Establece el nivel de zoom inicial cuando el navegador carga la página por primera vez.

Este es un ejemplo de una página web **sin/con** la meta-etiqueta de ventana gráfica:







Algunas **reglas importantes** a seguir:

- 1. NO use elementos grandes de ancho fijo: por ejemplo, si una imagen se muestra con un ancho mayor que la ventana gráfica, puede hacer que la ventana gráfica se desplace horizontalmente. Recuerde ajustar este contenido para que se ajuste al ancho de la ventana gráfica.
- 2. NO permita que el contenido dependa de un ancho de ventana en particular para mostrarse bien : dado que las dimensiones de la pantalla y el ancho en píxeles CSS varían ampliamente entre dispositivos, el contenido no debe depender de un ancho de ventana en particular para mostrarse bien.
- 3. Use consultas de @media para aplicar diferentes estilos para pantallas pequeñas y grandes: establecer anchos CSS absolutos grandes para elementos de página hará que el elemento sea demasiado ancho para la ventana gráfica en un dispositivo más pequeño. En su lugar, considere usar valores de ancho relativo, como ancho: 100%. Además, tenga cuidado al usar valores de posicionamiento absolutos grandes. Puede hacer que el elemento quede fuera de la ventana gráfica en dispositivos pequeños.

2. REGLA @MEDIA

Media query es una técnica CSS introducida en CSS3. Utiliza para incluir un bloque de propiedades CSS solo si se cumple una determinada condición.

```
@media only screen and (max-width: 600px){
body{
background-color:lightblue;
}
}
```

Si la ventana del navegador tiene 600 px o menos, el color de fondo será azul claro:

¿Qué Tipo de medio usar?

- all
- screen
- *tt*1
- print
- fx
- projected
- braille

¿Que tipo de consulta realizar?

- width | min-width | max-width
- height | min-height | max-height
- orientation (landscape / portrait)
- aspect-ratio | min-aspect-ratio | max-aspect-ratio
- color | min-color | max-colo

¿Puedo usar condiciones?

- and
- not
- all
- only

2.1 EJEMPLOS @ media

Las consultas de medios también se pueden usar para cambiar el diseño de una página según la <u>orientación</u> del navegador.

Otro uso común de las consultas de medios es ocultar elementos en diferentes tamaños de pantalla:

También puede usar consultas de medios para <u>cambiar el tamaño de fuente</u> de un elemento en diferentes tamaños de pantalla:

2.2 ¿Qué significa Mobile First?

Mobile First significa diseñar para dispositivos móviles antes de diseñar para computadoras de escritorio o cualquier otro dispositivo (esto hará que la página se muestre más rápido en dispositivos más pequeños).

Es decir.... En lugar de cambiar los estilos cuando el ancho esmenora 768 px, debemos cambiar el diseño cuando el ancho es mayor a 768 px. Esto hará que nuestro diseño Mobile First:

```
[class*="col-"]{
    width:100%;
}

@media only screen and (min-width: 768px){
    /* Para Escritorio */
    .col-1{width:8.33%;}
}
```

2.3 ¿Que puntos de ruptura utilizar? - Breakpoint

Hay toneladas de pantallas y dispositivos con diferentes alturas y anchos, por lo que es dificil crear un punto de interrupción exacto para cada dispositivo. Para simplificar las cosas, podría dirigirse a cinco grupos:

```
• Pantallas extra-pequeñas \rightarrow Teléfonos, 600px y inferior @media only screen and (max-width: 600px) \{...\}
```

- Pantallas pequeñas → Tablets y Teléfonos grandes, 600px and superior @media only screen and (min-width: 600px) {...}
- Pantallas medias \rightarrow landscape tablets, 768px and superior @media only screen and (min-width: 768px) $\{...\}$
- Pantallas grandes → portátiles/Escritorio, 992px and superior @media only screen and (min-width: 992px) {...}
- Pantallas extra grandes → Portátiles y escritorio grandes, 1200px and up @media only screen and (min-width: 1200px) {...}

Es una decisión muy importante. No hace falta que hagamos el estudio. Ya otros lo han hecho Twitter:

https://getbootstrap.com/

2.4 Hojas de estilo diferentes

Otra alternativa es utilizar hojas de estilo diferentes para cada tipo de tamaño (punto de ruptura)

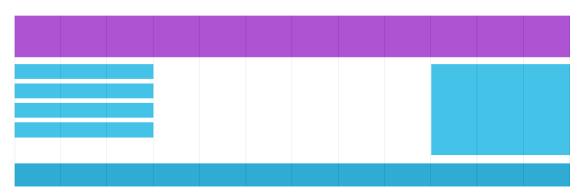
```
k rel="stylesheet" media="(max-width: 576px)" href="small.css" />
k rel="stylesheet" media="(max-width: 768px)" href="medium.css" />
k rel="stylesheet" media="(max-width: 992px)" href="large.css" />
```

3. GRID VIEW

Muchas páginas web se basan en una vista de cuadrícula, lo que significa que la página está dividida en columnas:



El uso de una vista de cuadrícula es muy útil al diseñar páginas web. Facilita la colocación de elementos en la página.



Una vista de cuadrícula a menudo tiene 12 columnas y tiene un ancho total del 100%, y se reducirá y expandirá a medida que cambia el tamaño de la ventana del navegador.

3.1 ¿Cómo construir un Grid View?

Comencemos a construir una vista de cuadrícula receptiva....

Primero asegúrese de que todos los elementos HTML tengan la propiedad **box-sizing** establecida en **border-box**. Esto asegura que el relleno y el borde estén incluidos en el ancho y alto total de los elementos.

```
* {
    box-sizing: border-box;
}
```

Creamos la vista con 12 columnas para tener más control sobre la página web. Primero debemos calcular el porcentaje para una columna: 100% / 12 columnas = 8,33%. Luego creamos una clase para cada una de las 12 columnas:

```
class="col-XX"
```

Todas estas columnas deben estar flotando a la izquierda y tener un relleno de 15px:

```
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
                                          [class*="col-"] {
.col-3 {width: 25%;}
                                             float: left;
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
                                             padding: 15px;
.col-6 {width: 50%;}
                                             border: 1px solid red;
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
```

Cada fila debe estar envuelta en una etiqueta <div clas="row">. El número de columnas dentro de una fila siempre debe sumar 12:

```
<div class="row">
    <div class="col-3">...</div> <!-- 25% -->
    <div class="col-9">...</div> <!-- 75% -->
</div>
```

Las columnas dentro de una fila están todas flotando hacia la izquierda y, por lo tanto, se eliminan del flujo de la página, y otros elementos se colocarán como si las columnas no existieran. Para evitar esto, agregaremos un estilo que borre el flujo:

```
.row::after {
  content: "";
  clear: both;
  display: table;
}
```

Ahora te toca practicar con los ejemplos de la siguientes webs:

- https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_cols
- https://www.w3schools.com/css/tryit.asp?filename=tryresponsive styles

Estudia, analiza y modifica la web para adaptarla a tí:

- ¿Puedes aplicar sass?, para reducir las líneas de css
- Añade más elementos a la web para que sea más completa
 footer, aside, artículos, etc...
- Modifica el ancho de las columnas para observar su comportamiento
- Aplica @media a un Grid-View
- etc...

