

- 1. Diseño Adaptativo (RWD)
 - 1.1. CSS3 Media Queries
 - 1.2. CSS2 Media Types
 - 1.3. CSS3 Media Queries
 - 1.3.1. Sintaxis de una Media Query
 - 1.3.2. Ejemplos
 - 1.4. Ejemplo layout CSS y RWD
 - 1.5. Enlaces de ampliación

1. Diseño Adaptativo (RWD)

RWD: Responsive Web Design

Tiene disponible ejemplos de este apartado en la Unidad Compartida.

Objetivo: adaptar nuestros contenidos y presentación al dispositivo que utilice en cada momento el usuario. Se suele fijar en el ancho de la pantalla, pero también podría considerar la orientación, etc.

Hay tres aproximaciones posibles:

- Mejora progresiva: iniciar el diseño desde el layout de un móvil e ir "mejorándolo" hasta llegar a layouts tipo escritorio. La anchura disponible va creciendo. Llamada *"First Mobile"*, es la que usaremos.
- Degradación progresiva: justo al revés.
- Crear layouts totalmente independientes para cada escenario.

Para usar RWD necesitamos

- Usar un elemento nuevo: las conocidas como *CSS Media Query*.
- Introducir en la cabecera la etiqueta `meta viewport` para que los contenidos funcionen correctamente en móviles, tablets, etc.

En el siguiente apartado tiene un ejercicio sencillo de layout con float que se adapta según RWD. Las "media query" permiten incluir **puntos de ruptura (breakpoints)** en el diseño del layout para que se vaya adaptando al nuevo espacio disponible.

Nota: para ver cómo se adapta el diseño al ancho puede:

- Reducir el ancho de la ventana del navegador
- o quizás más cómodo, abrir el inspector de elementos y jugar con el ancho con la barra que separa la ventana de visualización de la de desarrollo.

1.1. CSS3 Media Queries

La precursora de las *Media Queries* son las *Media Types*

1.2. CSS2 Media Types

La regla `@media` de CSS2 permite aplicar diferentes estilos para diferentes medios, principalmente `screen` y `print`. Si tenemos el fichero `impresora.css`, al añadir las reglas CSS dentro de la directiva `@media print` hacemos que esas dos reglas CSS sólo se apliquen si la página se va a imprimir:

```
/* impresora.css */
@media print {
  body { font-size: 10pt;}
  aside { display: none;}
}
```

Otra opción es indicar el media al enlazar el fichero css, con lo que no sería necesario el `@media print` en el fichero `impresora.css`, ya que todo el fichero se vería afectado por esa restricción:

```
<link rel="stylesheet" media="print" href="impresora.css" />
```

Con lo que el fichero `impresora.css` contendría simplemente las reglas CSS sin la media query:

```
/* impresora_mod.css */
/* No hace falta la directiva @media print:
   se enlaza especificando media="print" */
body {
  font-size: 10pt;
}
aside {
  display: none;
}
```

1.3. CSS3 Media Queries

Las *Media Queries* en CSS3 extienden la idea de las CSS2 media types: en vez de mirar sólo el tipo de dispositivo pueden consultar las características del dispositivo, por ejemplo la anchura y altura del dispositivo, la orientación, la anchura mínima o máxima, etc.

1.3.1. Sintaxis de una Media Query

Añade a la sintaxis de una CSS2 media type una o más expresiones, las cuales dan como resultado un valor verdadero o falso. El resultado global es verdadero si el tipo coincide y la expresiones son verdaderas. Si es verdadera, las reglas de estilo incluidas se aplican.

En el siguiente ejemplo las reglas CSS se aplican si el destino de la página es la pantalla y la anchura disponible es de al menos 480px.

```
@media screen and (min-width: 480px) {
```

```
/* Reglas CSS */  
}
```

El tipo del medio (**screen** en el ejemplo previo) es opcional: si no se incluye vale **all**. Por lo tanto estas directivas son equivalentes:

```
@media (orientation: portrait) {  
  ...  
}  
/*es equivalente a */  
@media all and (orientation: portrait) {  
  ...  
}
```

1.3.2. Ejemplos

1. Modificar el color de fondo al alcanzar una anchura mínima (**min-width**):

https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_mediaquery. El código CSS necesario sería:

```
#contenedor {  
  background-color: lightblue;  
}  
  
@media screen and (min-width: 480px) {  
  #contenedor {  
    background-color: lightgreen;  
  }  
}
```

2. Si el ancho es suficiente el menú de navegación flota a la izquierda del contenido

Ejemplo 2 en https://www.w3schools.com/css/tryit.asp?filename=trycss3_media_queries2

```
@media screen and (min-width: 480px) {  
  #leftsidebar {  
    width: 200px;  
    float: left;  
  }  
}
```

3. En función de la orientación del dispositivo (**orientation: landscape**):

https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_mediaquery_orientation

```
@media only screen and (orientation: landscape) {  
  body {  
    background-color: lightblue;  
  }  
}
```

1.4. Ejemplo layout CSS y RWD

En este ejemplo inicial, disponible en los ficheros `grid-rwd.html` y `grid-rwd.css` vemos como tenemos 2 puntos de ruptura, a 580px y 992px. Esto provoca 3 layouts distintos:

- hasta 580px.
- mayor que 580px
- mayor que 992px

En este caso el layout se ha realizado con grid:

```
/* grid-rwd.css */  
@media (min-width: 580px) {  
  #contenedor {  
    display: grid;  
    grid-template-columns: 1fr 2fr;  
  }  
  header {  
    grid-column: 1 / span 2;  
  }  
  aside, footer {  
    grid-column: 1 / span 2;  
  }  
}  
  
@media (min-width: 992px) {  
  #contenedor {  
    grid-template-columns: 1fr 3fr 2fr;  
  }  
  header, footer {  
    grid-column: 1 / span 3;  
  }  
  aside {  
    grid-column: 3;  
  }  
}
```

En este ejemplo se parte de un diseño apilado (***stacked**) para móvil, y se va modificando a medida que el ancho aumenta usando media queries con **min-width**. En cada salto o punto de ruptura se añaden o modifican las propiedades necesarias a las del layout previo.

Puede acceder al ejemplo en los ficheros:

- [ejemplo_layout_rwd.html](#) y
- [ejemplo_layout_rwd-float.css](#) o [ejemplo_layout_rwd-grid.css](#).

Este es el diseño CSS RWD usando float:

```
/* Si no se especifica nada, los bloques ocupan el 100%, luego
   aparecen apilados (stacked) */

/* primer punto de ruptura : si el ancho es 600 o más..*/
@media screen and (min-width: 600px) {
  aside,
  section {
    float: left;
    width: 50%;
  }
}

/* si el ancho pasa 900px: se aplican las de >600 y estas de >900) */
@media screen and (min-width: 900px) {
  aside,
  nav {
    float: left;
    width: 25%;
  }
  section {
    float: left;
    width: 50%;
  }
}

/* si pasamos de 1200: todas las anteriores más.. */
@media screen and (min-width: 1200px) {
  aside,
  nav {
    float: left;
    width: 20%;
  }
  section {
    float: left;
    width: 40%;
  }
  header {
    float: left;
    width: 20%;
  }
}
```

Tiene otro ejemplo para el mismo HTML con un layout creado con grid en el fichero [ejemplo_layout_rwd-grid.css](#). En este ejemplo hemos modificado la disposición de algunos elementos para que observe que grid es mas flexible y las posibilidades son mayores que con float.

```
/* Si no se especifica nada, los bloques ocupan el 100%, luego
aparecen apilados (stacked) */

/* primer punto de ruptura : si el ancho es 600 o más..*/
@media screen and (min-width: 600px) {
  body {
    display: grid;
    grid-template-columns: 1fr 1fr;
  }
  footer, header, nav {
    grid-column: 1 / 3;;
  }
}

/* si el ancho pasa de 900:
(nota: se aplican las de >600 y estas de >900) */
@media screen and (min-width: 900px) {
  body {
    grid-template-columns: 1fr 2fr 2fr ;
  }
  nav {
    grid-column: 1;
  }
  footer, header {
    grid-column: 1 / 4;
  }
}

/* si pasamos de 1200: todas las anteriores más.. */
@media screen and (min-width: 1200px) {
  body {
    grid-template-columns: 1fr 1fr 1fr ;
  }
  nav {
    grid-column: 3;
    grid-row-start: 1 ;
  }
  header {
    grid-column: 1 / 3;
    grid-row-start: 2;
  }
  footer {
    grid-column: 3;
    grid-row: 2 ;
  }
}
```

También es posible, además del layout, modificar otras características de la presentación de los datos. Por ejemplo puedo modificar una lista en una barra de navegación para que se muestre en horizontal o vertical. Tiene este código en los dos ficheros anteriores.

```
/* convierte botonera en horizontal */
@media screen and (max-width: 900px) and (min-width:420px) {
  nav ul {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
  }
}
```

En este caso aparece el uso de `repeat(3, 1fr)` que es equivalente a `1fr 1fr 1fr`.

1.5. Enlaces de ampliación

- Ejemplo de uso de `grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));` para automatizar la presentación de una galería de tarjetas. El código está disponible en [otros/grid-rwd-galeria.html](#)
- Un ejemplo de RWD con Bootstrap. El código está disponible en [otros/layoutHTML5-BS-RWD.html](#)
- Un ejemplo de cómo ir mejorando una lista de enlaces con direcciones de correo electrónico a medida que aumenta el tamaño disponible: <https://css-tricks.com/css-media-queries/>
- Qué dicen desde Google sobre RWD: <https://web.dev/responsive-web-design-basics/>
- Explicando meta viewport: <https://dev.opera.com/articles/an-introduction-to-meta-viewport-and-viewport/>