UT 4 – CSS Avanzado. Diseño adaptativo.

Contenido

1. Hojas de estilo alternativas
2. Tipos de medios
3. Media queries
4. Estilos para impresión
5. Imágenes
6. <meta name="viewport"…>
7. Mobile first
8. Menú de navegación

1. Hojas de estilo enlazadas

- **Hojas persistentes**: Se aplican siempre.

```
<link rel="stylesheet" href="css/todos.css">
```

- **Hojas alternativas:**
  - Tienen el atributo **title**.
  - El usuario puede elegir qué hoja o grupo de hojas (en caso de que varias tengan el mismo valor de title) aplicar.
  - Por defecto se aplican las que aparecen primero.
  - Sin embargo, si unas tienen **rel="stylesheet"** y otras tienen **rel="alternate stylesheet"**, se entiende que las primeras son las preferidas.

1. Hojas de estilo enlazadas

- **Hojas persistentes**: Se aplican siempre.

```
<link rel="stylesheet" href="css/todos.css">
```

- **Hojas alternativas:**
  - **Preferida**:

```
<link rel="stylesheet" href="css212/212.css" title="CSS Zen Garden 212">
```

  - **Alternativa**:

```
<link rel="alternate stylesheet" href="css214/214.css" title="CSS Zen Garden 214">
```

Nota: En Firefox, mostrar el menú superior con la tecla Alt y seleccionar Ver➔Estilo de página.

2. Tipos de medios

- **Tipo de medio**:
  - Describe la categoría general de un dispositivo.
  - Es opcional y será interpretado como all en caso de no especificarse.
    - Excepto si se utilizan los operadores lógicos only o not.
  - Los más comunes son:
    - **all**: Apto para todos los dispositivos.
    - **print**: Destinado a material impreso y visualización de documentos en una pantalla en el modo de vista previa de impresión.
    - **screen**: Destinado principalmente a las pantallas.
    - **speech:** Destinado a sintetizadores de voz.
  - CSS2.1 y Media Queries 3 definieron varios tipos de medios adicionales (tty, tv, projection, handheld, braille, embossed y aural), pero fueron desaprobados mediante Media Queries 4 y no deberían ser usados.

3. Media queries

- **¿Qué son?** Consisten en un **tipo de medio** opcional y una o más expresiones de **características de medios**.
  - Son útiles cuando se desea modificar la página web en función del tipo de dispositivo (ej. impresora o pantalla) o de características y parámetros específicos (ej. resolución de la pantalla o el ancho del viewport del navegador).
- Se pueden combinar varias consultas utilizando **operadores lógicos**.
  - El resultado de la consulta es “verdadero” cuando el tipo de medio (si se especifica) coincide con el dispositivo en el que se está mostrando el documento y todas las expresiones en la *media query* son verdaderas.
  - Las consultas sobre tipos de medio desconocidos son siempre falsas.

```
@media tv and (min-width: 700px) and (orientation: landscape) { ... }
```

3. Media queries

- **Usos**:
  - Aplicar estilos condicionales con las reglas de CSS:
    - @media
    - @import
  - Indicar medios específicos en los elementos <link>, <source> y otros elementos HTML.
  - Testear y monitorizar los estados de los medios usando los métodos de JavaScript:
    - Window.matchMedia()
    - MediaQueryList.addListener().

3. Media queries

- **Ejemplos**:
  - Aplicar estilos condicionales con las reglas de CSS:
    - @media

```
<!-- CSS media query within a style sheet -->
<style>
@media (max-width: 600px) {
  .facet_sidebar {
    display: none;
  }
}
</style>
```

    - @import

```
<style>
    @import url("fancyfonts.css") screen;
</style>
```

3. Media queries

- **Ejemplos**:
  - Indicar medios específicos en los elementos <link>, <source> y otros elementos HTML.
  - Atributo **media**.

```
<!-- CSS media query on a link element -->
<link rel="stylesheet" media="(max-width: 800px)" href="example.css" />
```

3. Media queries

- **Ejemplos**:
  - Testear y monitorizar los estados de los medios usando los métodos de JavaScript:
    - Window.matchMedia()

```
if (window.matchMedia("(max-width: 700px)").matches) {
  /* The viewport is less than, or equal to, 700 pixels wide */
} else {
  /* The viewport is greater than 700 pixels wide */
}
```

3. Media queries

- **Ejemplos**:
  - Testear y monitorizar los estados de los medios usando los métodos de JavaScript:
    - MediaQueryList.addListener().

```
<body>
    <p id="para"></p>
    <script>
        var para = document.getElementById("para");
        var mql = window.matchMedia('(max-width: 600px)');

        function screenTest(e) {
            if (e.matches) {
                /* ancho de viewport es igual a 600 píxeles o menos */
                para.textContent = 'Ventana estrecha – Ancho menor que 600px.';
                document.body.style.backgroundColor = 'red';
            } else {
                /* ancho de viewport mayor que 600 píxeles */
                para.textContent = 'Ventana ancha – Ancho mayor que 600px.';
                document.body.style.backgroundColor = 'blue';
            }
        }

        mql.addListener(screenTest);
    </script>
</body>
```

3. Media queries

- **Operadores lógicos**:
  - Se pueden redactar *media queries* utilizando operadores lógicos, incluyendo **not**, **and**, y **only**.
  - Además se puede combinar múltiples queries en una **lista separada por comas** múltiples.
    - Si cualquiera de las *queries* en la lista es verdadera, la hoja de estilo asociada es aplicada.
    - Esto es equivalente a una operación lógica "or".

3. Media queries

- **Operadores lógicos**:
  - **Operador and**: La *query* devuelve verdadero si se cumplen todas las condiciones establecidas.
  - Ejemplos:

```
@media (min-width: 700px) and (orientation: landscape) { ... }
```

```
@media tv and (min-width: 700px) and (orientation: landscape) { ... }
```

3. Media queries

- **Operadores lógicos**:
  - **Lista separada por comas**: Se comportan como el operador **or**. Si una de las *queries* del listado se cumple devuelve verdadero.
  - Ejemplos:

```
@media (min-width: 700px), handheld and (orientation: landscape) { ... }
```

3. Media queries

- **Operadores lógicos**:
  - **Operador not**: Negara una *query* si es posible. El not es evaluado al final y niega la *query* completa.
  - Ejemplos:
    - La siguiente *query*:

```
@media not all and (monochrome) { ... }
```

    - Es evaluada de esta forma:

```
@media not (all and (monochrome)) { ... }
```

3. Media queries

- **Operadores lógicos**:
  - **Operador only**: Previene que navegadores antiguos que no soportan *media queries* con determinadas características aplican los estilos especificados. No tiene efecto en navegadores actuales.
  - Ejemplos:

```
<link rel="stylesheet" media="only screen and (color)" href="Ejemplo.css" />
```

3. Media queries

- **Funciones multimedia**: La mayoría de las funciones multimedia pueden ser precedidas por “min-” o “max-” para expresar “mayor o igual” y “menor o igual” respectivamente.
  - **color**: Indica el número de bits por componente de color del dispositivo de salida. Si no soporta colores, este valor es 0.

```
@media all and (min-color: 4) { ... }
```

  - **color-index**: Indica el número de entradas en la tabla de colores para el dispositivo de salida.

```
<link rel="stylesheet" media="all and (min-color-index: 256)" href="http://foo.bar.com/stylesheet.css" />
```

  - **aspect-ratio**: Representa la relación de aspecto de los píxeles horizontales a los verticales.

```
@media screen and (min-aspect-ratio: 1/1) { ... }
```

  - **device-aspect-ratio**: Describe la relación de aspecto del dispositivo de salida.

```
@media screen and (device-aspect-ratio: 16/9), screen and (device-aspect-ratio: 16/10) { ... }
```

3. Media queries

- **Funciones multimedia**: La mayoría de las funciones multimedia pueden ser precedidas por “min-” o “max-” para expresar “mayor o igual” y “menor o igual” respectivamente.
  - **device-height y device-width**: Altura y anchura del dispositivo de salida. Ej. monitor.

```
<link rel="stylesheet" media="screen and (max-device-width: 799px)" />
```

  - **height y width**: Altura y ancho de la superficie a renderizar. Ej. ancho de la ventana.

```
@media screen and (min-width: 500px) and (max-width: 800px) { ... }
```

  - **grid**: Representa cuándo el dispositivo de salida es un dispositivo de cuadrícula o mapa de bits (ej. pantalla de teléfono de solo texto).

```
@media handheld and (grid) and (max-width: 15em) { ... }
```

  - **monochrome**: Indica el número de bits por píxel en un dispositivo monocromático (escala de grises). Si no es monocromático devuelve 0.

```
@media all and (min-monochrome: 8) { ... }
```

3. Media queries

- **Funciones multimedia**: La mayoría de las funciones multimedia pueden ser precedidas por “min-” o “max-” para expresar “mayor o igual” y “menor o igual” respectivamente.
  - **orientation**: Indica cuándo el dispositivo está en modo landscape (ancho de pantalla es mayor que el alto) o modo portrait (el alto de la pantalla es mayor al ancho).

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

  - **resolution**: Indica la resolución (densidad de píxeles) del dispositivo de salida. La resolución puede ser especificada en puntos por pulgada (dpi) o en puntos por centímetros (dpcm).

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

  - **scan**: Describe el proceso de exploración de televisión de los dispositivos de salida, progresivo o entrelazado (progressive o interlace).

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

4. Estilos para impresión

- **@media print**:
  - **Ocultar los elementos que no se van a imprimir**: utilizando el atributo display:none.

```
#cabecera, #menu, #lateral, #comentarios {
    display: none !important;
}
```

  - **Corregir la estructura de la página:** Haciendo que todas las capas ocupen todo el ancho y que no haya capas flotantes.

```
body, #contenido, #principal, #pie {
    float: none !important;
    width: auto !important;
    margin: 0 !important;
    padding: 0 !important;
}
```

4. Estilos para impresión

- **@media print**:
  - **Modificar los colores y tipos de letra:**

```
body {
    color: #000; font: 100%/150% Georgia, "Times New Roman",
Times, serif;
}
```

  - **Imprimir los enlaces con las pseudoclases after y before:**
    - Si queremos que se imprima después del enlace:

```
a:after {
    content: " (" attr(href) ") ";
}
```

    - Si queremos que se imprima antes del enlace:

```
a:before {
    content: " (" attr(href) ") ";
}
```

5. Imágenes

- **Imágenes flexibles**: Utilizar width, min-width y max-width con porcentajes del contenedor. Ej. 100%.

```
img {
  width: 100%;
  height: auto;
}
```

```
img {
  max-width: 100%;
  height: auto;
}
```

- **Imágenes de fondo flexible**: Utilizar la propiedad background-size para las imágenes del fondo.
  - background-size:contain
  - background-size:100% 100%
  - background-size: cover
- Elemento picture: Cargar diferente imagen en función del medio.

```
<picture>
  <source srcset="img_smallflower.jpg" media="(max-width: 400px)">
  <source srcset="img_flowers.jpg">
  <img src="img_flowers.jpg" alt="Flowers">
</picture>
```

- https://www.w3schools.com/css/css_rwd_images.asp

6. <meta name="viewport"…>

- El **viewport** del navegador es el área de la ventana en donde el contenido web está visible.
- Generalmente no es del mismo tamaño que la página renderizada, en donde se brindan barras de desplazamiento para que el usuario pueda acceder a todo el contenido.
- Dispositivos con pantallas estrechas (ej. smartphone) muestran la página en una ventana virtual o viewport, que es usualmente más ancho que la pantalla y la comprimen de manera que pueda verse completa.
  - Ejemplo: Si una pantalla móvil tiene un ancho 640px, las páginas pueden ser procesadas con un viewport de 980px, y después comprimidas para que entren en 640px.
- El usuario podrá recorrerla y hacer zoom para ver diferentes áreas de la página.

6. <meta name="viewport"…>

- **Viewport básico**:

```
<meta name="viewport" content="width=device-width, user-scalable=no">
```

- **width**: Controla el tamaño del viewport. Puede definirse con un número en píxeles o con un valor especial, device-width, que es el equivalente al ancho de la pantalla en píxeles CSS (1in=96px) en una escala de 100%.
- **initial-scale**: Controla el nivel de zoom cuando la página se carga por primera vez. Un valor de 1 indica que la página se muestra al 100% de su tamaño.
- **minimum-scale** y **maximum-scale**: Valores mínimo y máximo de zoom permitido.
- **user-scalable**: Define si el usuario puede hacer zoom para escalar el contenido (yes, no).

6. <meta name="viewport"…>

Sin viewport:

Con viewport:

- Otro ejemplo: https://alistapart.github.io/code-samples/responsive-web-design/ex/ex-site-flexible.html

7. Mobile first

- Empezar desarrollando el diseño más sencillo, más pequeño.
- Usar *media queries* para definir los estilos de los diseños para dispositivos con ancho de pantalla mayor, de forma que estos últimos sobrescriban a los anteriores.

```
/* Estilos de la version móvil */
...

/* Estilos para pantallas pequeñas */
@media screen and (min-width: 600px) { ... }

/* Estilos para pantallas grandes */
@media screen and (min-width: 800px) { ... }
```

8. Menú de navegación

- **HTML**:

```
<!-- Load an icon library to show a hamburger menu (bars) on small screens -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<div class="topnav" id="myTopnav">
  <a href="#home" class="active">Home</a>
  <a href="#news">News</a>
  <a href="#contact">Contact</a>
  <a href="#about">About</a>
  <a href="javascript:void(0);" class="icon" onclick="myFunction()">
    <i class="fa fa-bars"></i>
  </a>
</div>
```

https://www.w3schools.com/howto/howto_js_topnav_responsive.asp

8. Menú de navegación

- **CSS**:

```css
/* Add a black background color to the top navigation */
.topnav {
  background-color: #333;
  overflow: hidden;
}

/* Style the links inside the navigation bar */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}
```

https://www.w3schools.com/howto/howto_js_topnav_responsive.asp

8. Menú de navegación

- **CSS**:

```css
/* Change the color of links on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}

/* Add an active class to highlight the current page */
.topnav a.active {
  background-color: #4CAF50;
  color: white;
}

/* Hide the link that should open and close the topnav on small screens */
.topnav .icon {
  display: none;
}
```

https://www.w3schools.com/howto/howto_js_topnav_responsive.asp

8. Menú de navegación

- **CSS**: Media queries.

```css
/* When the screen is less than 600 pixels wide, hide all links, except for the first one ("Home").
Show the link that contains should open and close the topnav (.icon) */
@media screen and (max-width: 600px) {
  .topnav a:not(:first-child) {display: none;}
  .topnav a.icon {
    float: right;
    display: block;
  }
}

/* The "responsive" class is added to the topnav with JavaScript when the user clicks on the icon.
This class makes the topnav look good on small screens (display the links vertically instead of
horizontally) */
@media screen and (max-width: 600px) {
  .topnav.responsive {position: relative;}
  .topnav.responsive a.icon {
    position: absolute;
    right: 0;
    top: 0;
  }
  .topnav.responsive a {
    float: none;
    display: block;
    text-align: left;
  }
}
```

https://www.w3schools.com/howto/howto_js_topnav_responsive.asp

8. Menú de navegación

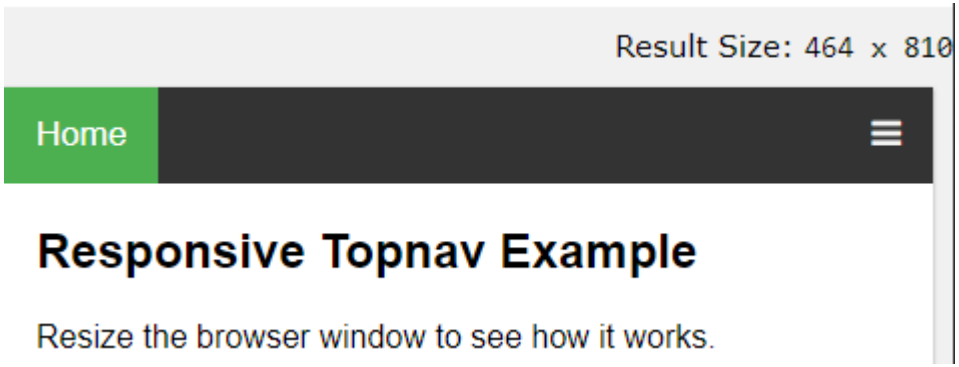
- **Javascript**.

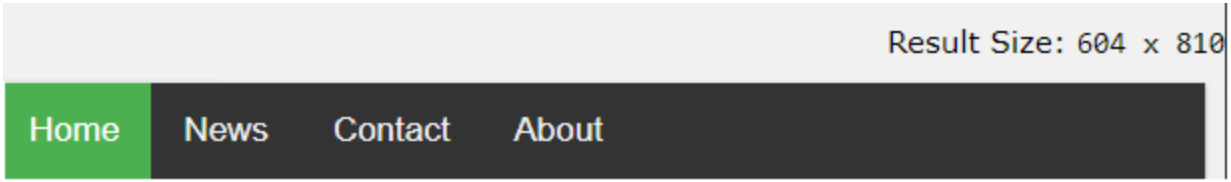
```
/* Toggle between adding and removing the "responsive" class to topnav when the user clicks on the
icon */
function myFunction() {
  var x = document.getElementById("myTopnav");
  if (x.className === "topnav") {
    x.className += " responsive";
  } else {
    x.className = "topnav";
  }
}
```

https://www.w3schools.com/howto/howto_js_topnav_responsive.asp

8. Menú de navegación

- **Resultado:**





https://www.w3schools.com/howto/howto_js_topnav_responsive.asp

Referencias

- MDN. *CSS media queries*.
https://developer.mozilla.org/es/docs/CSS/Media_queries
- MDN. Usando la etiqueta meta viewport.
https://developer.mozilla.org/es/docs/M%C3%B3vil/Viewport_meta_tag
- W3C. CSS Snapshot 2018.
https://www.w3.org/TR/CSS/#css
- W3C. *Descriptions of all CSS specifications*.
https://www.w3.org/Style/CSS/specs.en.html
- W3Schools
https://www.w3schools.com/css/css_rwd_intro.asp