

HTML y CSS

PEC 3.

Parte teórica

Antonio Cerezo Garnateo

ancegar@uoc.edu

Primera parte.

Pregunta 1

Sobre técnicas de posicionamiento y modos de diseño, responde a las siguientes preguntas:

1. Explicar los siguientes tipos de posicionamiento: estático, relativo, absoluto, fijo y sticky.

El posicionamiento permite mover un elemento desde donde se colocaría cuando está en flujo normal a otra ubicación. El posicionamiento no es un método para crear diseños de página principal, se trata más bien de administrar y ajustar la posición de elementos específicos en la página.

- El posicionamiento estático es el valor por defecto que tiene establecido cada elemento, esto significa sencillamente «coloca el elemento en su posición normal en el flujo de diseño del documento, nada especial que ver aquí».
- El posicionamiento relativo permite modificar la posición de un elemento en la página y moverlo con respecto a su posición en el flujo normal, lo que incluye superposiciones con otros elementos de la página.
- El posicionamiento absoluto mueve un elemento completamente fuera del flujo de diseño normal de la página, como si estuviera asentado en una capa separada propia. A partir de aquí, puedes fijarlo en una posición relativa a los bordes del elemento `<html>` de la página (o su elemento padre más próximo). Esto es útil para crear efectos complejos de diseño de página, como cuadros con pestañas en que diferentes paneles de contenido se ubican uno encima del otro y se muestran y ocultan según se desee, o paneles de información que por defecto se asientan fuera de la pantalla, pero puedes hacer que se deslicen hacia la pantalla usando un control de tipo botón.
- El posicionamiento fijo es muy similar al posicionamiento absoluto, excepto que corrige un elemento relativo a la ventana gráfica del navegador, no otro elemento. Esto es útil para crear efectos, como un menú de navegación persistente que siempre permanece en el mismo lugar en la pantalla al contrario que el contenido, que se desplaza.
- El posicionamiento sticky (pegajoso) es un método de posicionamiento más reciente que hace que un elemento actúe como `position: static` hasta que alcanza un desplazamiento definido con respecto a la ventana gráfica, en cuyo punto actúa como un elemento con la propiedad `position: fixed`.

2. Explicar para qué sirven las siguientes propiedades del módulo CSS Grid: `grid-template-columns`, `grid-template-rows` i `gap`.

En Grid CSS, la forma principal de definir una cuadrícula es indicar el tamaño de sus filas y sus columnas de forma explícita. Para ello, sólo tenemos que usar las propiedades CSS `grid-template-columns` y `grid-template-rows`. `grid-template-columns` establece el tamaño definido de cada columna, mientras que `grid-template-rows` establece el size de cada fila.

Al utilizar la propiedad `gap` y darle un valor con cualquier unidad de medida CSS, estamos diciéndole al navegador que separe tanto las columnas como las filas de la cuadrícula CSS con ese tamaño.

3. Explicar para qué sirven las siguientes propiedades del módulo CSS Flexbox: `flex-direction`, `flex-wrap` i `order`.

La propiedad CSS `flex-direction` especifica cómo colocar los objetos flexibles en el contenedor flexible definiendo el eje principal y la dirección (normal o al revés).

La propiedad `flex-wrap` de CSS especifica si los elementos "hijos" son obligados a permanecer en una misma línea o pueden fluir en varias líneas. Si la cobertura (`wrap`) está permitida, esta propiedad también te permite controlar la dirección en la cual serán apilados los elementos.

La propiedad CSS `order` especifica el orden utilizado para disponer los elementos en su contenedor flexible. Los elementos estarán dispuestos en orden ascendente según el valor de `order`. Los elementos con el mismo valor de `order` se dispondrán en el orden en el cual aparecen en el código fuente.

Pregunta 2

Sobre diseño responsivo, responded a las siguientes preguntas:

1. Explicad qué se entiende por *mobile first*.

Mobile First es un concepto de diseño web optimizado para móviles, donde inicialmente se crea una página web para tablets y smartphones, para adaptarla luego a las necesidades de un navegador de escritorio. Hasta hace unos años, era común que los diseñadores y los programadores se hicieran cargo de la ejecución de páginas web pensadas únicamente para ser visualizadas en ordenadores. En principio, estas eran diseñadas con una gran variedad de características, adaptando sus gráficos y estructura para ser visualizados en grandes pantallas, usando conexiones de datos de gran velocidad. Así, se relegaría a un segundo plano la planificación de una web móvil, que hasta entonces era considerada una especie de apéndice. Con Mobile First, el flujo de trabajo se invierte y se establecen otras prioridades; impactando a la vez todo el desarrollo e infraestructura técnica detrás de una página web.

2. Explicar por qué es importante el uso de la etiqueta meta viewport.

Esta etiqueta nos permite elegir cómo se mostrará nuestra página en los navegadores de los dispositivos móviles, pudiendo especificar el ancho que tendrá la página dentro del navegador, el nivel de zoom que podemos hacer... lo que nos puede ayudar a crear diseños adecuados para móviles y perfeccionar nuestro responsive design.

3. Explicar a partir de un ejemplo de código cómo los atributos srcset y sizes del elemento img permiten mostrar imágenes diferentes según el dispositivo.

```

```

srcset define el conjunto de imágenes que el navegador podrá elegir, y el tamaño de cada imagen. Cada conjunto de información de imagen está separado del anterior por una coma. Para cada uno, escribimos:

1. Un nombre de archivo de imagen (elva-fairy-480w.jpg)
2. Un espacio.
3. El ancho intrínseco de la imagen en píxeles (480w): tenga en cuenta que esto usa la unidad w, no px como cabría esperar. Este es el tamaño real de la imagen, que se puede encontrar inspeccionando el archivo de imagen en su computadora (por ejemplo, en una Mac puede seleccionar la imagen en Finder y presionar Cmd + I para que aparezca la pantalla de información).

sizes define un conjunto de condiciones de medios (por ejemplo, anchos de pantalla) e indica qué tamaño de imagen sería mejor elegir cuando se cumplen ciertas condiciones de medios — estas son las sugerencias de las que hablamos anteriormente. En este caso, antes de cada coma escribimos:

1. Una condición de medios ((max-width: 600px)): aprenderá más sobre esto en el [tema CSS](#), pero por ahora digamos que una condición de medios describe un posible estado en el que puede estar la pantalla. En este caso, estamos diciendo "cuando el ancho de la ventana gráfica es de 600 píxeles o menos".
2. Un espacio.
3. El ancho de la ranura que la imagen llenará cuando la condición de medios sea verdadera (440px.)

Segunda parte.

Informe de las características del sitio web realizado.

El sitio web ha sido publicado en GitHub Pages y puede verse en el siguiente enlace:

<https://senseveu.github.io/PEC3/>

A continuación se detallan algunos aspectos de la elaboración del sitio.

El sitio web se compone de los siguientes cuatro archivos html:

- index.html
- referentes.html
- 11-de-febrero.html
- iniciativas.html

Además, existen dos carpetas en el directorio raíz. En la carpeta css se encuentra el fichero css con los estilos aplicados y en la carpeta img se encuentran las diferentes imágenes utilizadas en el diseño del sitio.

Las cuatro páginas html poseen una parte común a todas. Estas partes comunes se detallan a continuación:

Un primer bloque donde se recogen los aspectos relativos al header y donde se especifica la codificación de lenguaje, juego de caracteres, y enlace al archivo css utilizado.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/estilos.css">

  <title>Mujeres en la ciencia</title>
  <style></style>
</head>
```

En el segundo bloque, ya perteneciente al body, se detalla la estructura común al logo y el menú

```
<header>

  <div id="container_logo">

    <a href="index.html"></a>

    <a href="index.html">Ciencia y tecnología</a>

  </div>

  <nav>

    <ul class="nav_items">

      <li><a href="referentes.html">Referentes científicas</a></li>
      <li><a href="11-de-febrero.html">11 de febrero</a></li>
      <li><a href="iniciativas.html">Iniciativas</a></li>

    </ul>

  </nav>
</header>
```

En el último bloque común se recoge la estructura del pie de página:

```
<footer>

  
  <a class="enlace-logo"
href="hola@cienciaytecnologia.org">hola@cienciaytecnologia.org</a>

  <p>Siempre al día con nuestro boletín de noticias</p>
  <form action="submit">

    <input type="email" id="email" placeholder="Escribe tu e-mail">

  <hr>
  <input type="checkbox" id="politica" name="politica" value="politica">
  <label for="politica"> Acepto la política de protección de datos</label><br>
  <button type="submit" value="submit">SUSCRÍBEME</button>

</form>

  <div id="enlaces-footer">

    <a href="#">Aviso legal</a>
    <a href="#">Política de privacidad</a>
    <a href="#">Política de cookies</a>

  </div>

  <div id="pie-footer">

    <p>2022 © Ciencia y tecnología</p>
    <p>Actividad realizada por Toni Cerezo</p>
    <p>Información elaborada a partir de la biblioguia
```

```

        <a
href="https://biblioteca.uoc.edu/es/biblioguias/biblioguia/Mujeres-ciencia-y-tecnol
ogia/">
        Mujeres ciencia y tecnologia</a>
        de la biblioteca de la UOC</p>
        <p>Icono de la portada por <a
href="https://www.flaticon.es/iconos-gratis/atomo%5D">flatart_icons -
Flaticon</a></p>

    </div>
</footer>

```

El archivo css donde se codifican los estilos utilizados está estructurado de la siguiente manera.

Los estilos aplicados a cada página vienen precedidos del siguiente bloque, para marcar el comienzo:

```

/*****
/*Página iniciativas.html. MOBILE FIRST*/
*****/

```

Cuando se termina con el diseño de dispositivos pequeños, se podrán encontrar los media queries específicos para cada página, como por ejemplo, los aplicados en la página iniciativas.html:

```

/*Media Query para el ajuste de la lista en escritorio*/
@media (min-width: 1024px) {
    .lista{margin-left:20%;margin-right:20%}
}

```

La mayoría de media queries aplicados ajustan el bloque principal de contenidos, añadiendo márgenes a derecha e izquierda para ser lo más próximos al diseño propuesto en los mockups.

Otro de los aspectos a controlar por los media queries ha sido la distribución de los menús.

Por ejemplo:

```
/*Tratamiento del menú de enlaces en referentes.html.MOBILE FIRST*/

@media (min-width: 1024px) {
  #enlaces-referentes{width:20%;display: flex;flex-direction:column;margin:30px
17%;text-align: left;}
  #enlaces-referentes a{margin-bottom:5px}
  #enlaces-referentes a::after{content: "";}
}
```

En este media query puede verse como se aplican las reglas necesarias para mostrar el menú en vertical en lugar de en horizontal como en las versiones de dispositivos pequeños.