**Pregunta 1**

**Explicad los siguientes tipos de posicionamiento: estático, relativo, absoluto y fijo.**

* Estático: Representa el posicionamiento por defecto de los objetos, por lo que con esta opción los elementos se muestran tal y como se pintan en el flujo sin sufrir modificaciones.
* Relativo: Permite modificar la posición y moverlo con respecto a su posición en el flujo normal permitiendo la superposición sobre otros elementos.
* Absoluto: Coloca el elemento en una capa propia que permite colocarlo a partir del elemento HTML.
* Fijo: Se emplea de manera idéntica al posicionamiento absoluto, pero recoloca el elemento según la ventana gráfica y no según el elemento HTML.

**Explicad para qué sirven las siguientes propiedades del módulo CSS Grid: grid-template-columns, grid-template-rows, grid-gap y place-content.**

* Grid-template-columns: Especifica el tamaño y la cantidad de columnas de la cuadrícula de la página.
* Grid-template-rows: Especifica el tamaño y la cantidad de filas de la cuadrícula.
* Grid-gap: Especifica el espacio libre entre las diferentes celdas de la cuadrícula.
* Place-content: Atajo para las propiedades align-content y justify-content

**Explicad para qué sirven las siguientes propiedades del módulo CSS Flexbox: flex-direction, flex-wrap, order y align-self.**

* Flex-direction: Especifica la dirección(fila o columna) en las que se muestran los elementos hijo del elemento al que se le asigna la propiedad.
* Flex-wrap: Especifica si los elementos hijos pueden establecerse en varias líneas o solo pueden estar en una.
* Order: Permite establecer un orden o una prioridad a los elementos para modificar su posición sin necesidad de modificar el código fuente
* Align-self: Reemplaza la alineación de los elementos especificada mediante el atributo align-items.

**Pregunta 2**

**Explicad en breves palabras qué es el diseño responsivo (responsive desing).**

El diseño responsivo es un conjunto de buenas prácticas utilizado para crear un diseño que responda correctamente independientemente del dispositivo en el que se muestre la web, recolocando los elementos según las necesidades.

**Enumerad al menos cuatro características y/o técnicas que ayudan a realizar un diseño responsivo.**

* Cuadrículas flexibles
* Imágenes responsivas
* Tipografía responsiva
* La metaetiqueta viewport

**Explicad brevemente qué se entiende por “mobile first”.**

Mobile first es una estrategia de diseño basada en diseñar primero la web para pantallas pequeñas e irlas adaptando de menor a mayor.

Parte 2: Informe explicativo del sitio web.

La web creada para esta práctica debería desplegarse con los archivos html en una carpeta. Esta carpeta debe contener otras dos carpetas, una carpeta **img** que contenga todas las imágenes necesarias y una carpeta **css** que contiene el archivo estilos.css

Los archivos han sido subidos al sistema WebFTP del campus de la UOC y se puede acceder a la misma desde la web:

https://www.campus.uoc.edu/20202\_M4.252\_html/aula1/lrocamonde%40uoc.edu/index.html

HTML

Para estructurar las partes comunes de las páginas he usado, en primer lugar, la etiqueta **<head>** para definir la metainformación y el título de las páginas y la etiqueta **<body>** como contenedor del cuerpo de los elementos.

Dentro del **<body>** he utilizado las etiquetas **<header>,** **<main>** y **<footer>** para estructurar las distintas secciones, las cuales desglosaré, diferenciando página por página, a continuación.

**HEADER: Index.html, mars-2020.html, ciencia-ficcion.html, marte-en-el-cine.html**

El header del index contiene dentro la barra de navegación, la etiqueta **<nav>.** He decidido introducir la barra de navegación dentro del header ya que es el elemento que se encuentra en la parte superior. Como no existe un título o un encabezado que preceda al navegador me ha parecido más apropiado no diferenciar entre header y nav ya que considero que el header es la propia navegación.

Además, dentro de la etiqueta **<nav>** está la lista no ordenada, **<ul>**, que contiene los enlaces **<a>** a las distintas páginas de la web dentro de cada uno de los elementos de la lista **<li>**.

Como último detalle, en todas las páginas, excepto en el index.html, he introducido la clase **.paginaActual** para identificar la página en la que se encuentra el usuario y poder aplicarle un CSS distinto, además de eliminar el enlace a si misma.

**MAIN: Index.html**

El contenido del main de la página index está compuesto por una serie de divs.

Al principio hay un div que contiene el título de la página y un subtítulo. Además, a este div se le añade una clase para poder insertarle una imagen de fondo posteriormente mediante el CSS.

Tras este div hay otro div al mismo nivel que el anterior. Este div contiene un conjunto de divs que representan los diferentes apartados de la página. Cada uno de estos divs contiene una etiqueta figure la cual se utiliza para insertar una imagen y tras la figure se añade un párrafo que actúa como título de la sección, con un enlace vacío, y otro párrafo que describe dicha sección.

**MAIN: mars-2020.html**

La sección principal del artículo de la página mars-2020 contiene una serie de párrafos además de dos títulos que describen la misión mars-2020. Además de estos textos la página contiene una lista ordenada que contiene cuatro elementos.

Tras la lista ordenada también se puede ver otro párrafo y una figure que contiene un iframe, insertado dentro de un div para poder modificar su presentación mediante el css, y un figcaption con el subtítulo del vídeo.

**MAIN: ciencia-ficcion.html**

La página ciencia-ficcion está compuesta por una serie de divs anidados. El de nivel más alto se usa como contenedor para el resto de divs y en el siguiente nivel hay una serie de divs que representan los diferentes elementos de la página.

Cada div está compuesto de tres elementos:

* Un div, el cual contiene el título del elemento
* Un segundo div que contiene una cita en bloque del autor representado en el elemento, además del nombre del autor.
* Una figure que contiene la imagen del autor

**MAIN: marte-en-el-cine.html**

La página marte-en-el-cine también está compuesta por una serie de divs anidados.

El primer elemento es el título de la página y a continuación se añade el div de mayor nivel que se utiliza como contenedor para los diferentes divs que representan los elementos de la página.

Cada uno de estos divs contiene una figure la cual a su vez contiene dos divs. Estos divs representan:

* El primer div contiene un iframe que muestra un vídeo y está guardado con una clase para ajustar el vídeo con CSS.
* El segundo div contiene el figcaption con una clase específica para poder modificarlo mediante CSS

**ASIDE: mars-2020.html**

En este proyecto la página 2020 también contiene un apartado aside que muestra los distintos proyectos enviados a marte.

Este apartado aside está compuesto por una lista ordenada ul la cual está compuesta por nueve elementos li. Los ocho primeros son los títulos del resto de proyectos mostrados en el index y cada elemento contiene un enlace vacío. El último elemento no es un enlace.

**FOOTER: Index.html, mars-2020.html, ciencia-ficcion.html, marte-en-el-cine.html**

El footer es igual en todas las páginas y contiene varios elementos.

El primer elemento es un div vacío que se emplea para insertar una imagen de fondo mediante CSS. A continuación, se añade un párrafo con un correo y un formulario que permite a los usuarios suscribirse a la newsletter de la página.

Este formulario contiene un legend para describir el formulario, con una etiqueta span que permite aplicar negrita con el CSS, un input que permite indicar el correo del usuario. Tras este input se añade un div que contiene otro input y un label. Este div muestra un input dedicado a aceptar la política de privacidad. Finalmente, tras este div, aparece otro input, el submit del formulario.

Después del formulario se añade una lista ordenada la cual contiene tres elementos con enlaces vacíos.

Finalmente se añade un div el cual contiene tres párrafos que muestran los créditos de la página.

CSS

El CSS de estas webs se diferencia en dos partes. La primera parte está orientada hacia un diseño en pantallas pequeñas, la segunda utiliza el CSS creado para las pantallas pequeñas y le aplica las modificaciones necesarias para que se adapte a pantallas grandes.

En cuanto al CSS de la primera parte, especificaciones para pantallas pequeñas, cabe destacar:

* Uso de **@import** **url('https://fonts.googleapis.com/css2?family=Alegreya&family=Open+Sans:wght@300;400&display=swap');** las fuentes Alegreya y Open Sans
* Uso de **box-sizing: border-box;** para emplear el modelo de cajas alternativo
* Tamaño de fuente en 10px, **font-size: 10px;**, en el HTML para una mayor facilidad al usar unidades rem para los distintos tamaños de elementos.
* **text-align: center;, margin-left: auto;, margin-right: auto;** en el body para centrar el contenido.
* Se le añade un padding de 10vw a los elementos del header, la imagen y el menú de navegación.
* Se especifica **list-style-type: none;** para ocultar los puntos de la lista del menú de navegación.
* Se especifica la clase **.paginaActual** para aplicar un subrayado al elemento de la lista correspondiente.
* Se especifica el color **rgb(189, 55, 0);** para el estado hover de los enlaces del menú de navegación.
* Se utiliza la clase **.utopia** para especificar el color blanco del título de la página del index y que ocupe el 100% de la página.
* Se utiliza la clase **.utopiaImagen** para especificar la imagen de fondo. También se especifica que la imagen no se debe repetir y que debe posicionarse a la derecha.
* Se usa el selector **.mision p:first-of-type** para que el primer párrafo de cada elemento de la página index sea de un tamayo mayor
* Se utiliza la clase **.imagenFooter** para especificar la imagen de fondo y las mismas características que se han especificado para la imagen del titulo
* Se especifica para el footer el color de fondo y el color del texto y además se especifica que la fuente es Open Sans
* Mediante el id **#correo** se especifica que el input email del pie de página no debe tener bordes nada más que en el borde inferior.
* Se especifica que la lista ordenada del footer debe ter un display en línea y que los enlaces no deben tener decoración y que el color del estado :hover debe ser **rgb(189, 55, 0).**
* Se especifica **border:0;** para el selector fieldset para eliminar los bordes por defecto.
* Se especifican las clases **.cursiva** y **.negrita** para que los textos contenidos por un span con esa clase se muestren en cursiva y en negrita respectivamente.
* Las clases **.video-fluid, .video-fluid-item, .video-fluid::before** especifican el tamaño y el posicionamiento de los vídeos.
* Se especifica el color de fondo **background-color:rgba(0, 0, 0, 0.1);** para los elementos de la página ciencia-ficción.html
* El selector **cite::before** se utiliza para crear un elemento vacío antes del elemento cite que se subraya en rojo, mostrando visualmente una línea roja antes del nombre del autor.
* Mediante el selector **.videoMarteCine figcaption** se especifica que el texto se debe utilizar con el display table-cell para poder centrarlo.

Segunda parte: especificaciones para pantallas grandes:

* Se utiliza un **display: inline** para el navegador.
* En la página index se utiliza un **display: grid** con tres columnas, a las cuales se les especifica un tamaño de 1fr a cada una para que se muestren tres columnas del mismo tamaño repartiendo todo el espacio disponible.
* Se emplea el selector **.mision:nth-child(2n+1)** para aplicar un color de fondo gris a los elementos impares.
* Se usa un **display: inline** en los créditos del footer para que los textos se muestren alineados y posteriormente se le aplica un **display: block** al primer elemento de los créditos. De este modo el primer elemento se separa del resto de párrafos.
* A la página mars-2020 se le aplica un **display:grid** especificando dos columnas mediante la instrucción **grid-template-columns: 1fr 3fr;**. Esto se emplea para otorgarle 1 parte del espacio visible a la barra lateral(aside) y 3 partes al contenido del artículo.
* En los siguientes elementos se utiliza **grid-column** y **grid-row** para especificar, sección a sección, las posiciones de los distintos elementos de la página posicionando el aside a la izquierda del main y colocando al header y al footer ocupando la totalidad del width.
* Mediante la clase **.elementoCF** se utiliza un **display:flex** para que los elementos se sitúen de manera flexible.
* A las citas de la página se les especifica un tamaño mínimo de alto y ancho para evitar que se sobreponga el texto.
* Para al última página se emplea de nuevo un **display:grid** para crear dos columnas con tamaño especificado 1fr para cada una, es decir, cada columna debe ocupar la mitad del espacio disponible.