M4.252 - HTML y CSS aula 1

PEC3

Estudiante: Ivan Lenin Schartun Brito

Primera parte

Pregunta 1. Sobre técnicas de posicionamiento y modos de diseño, responded a las siguientes preguntas:

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

Respuesta. –

* *Posicionamiento estático*: Es el valor por defecto que poseen cada uno de los elementos HTML, básicamente indica que el ítem no se va a mover a ninguna posición distinta a la del flujo normal del documento:

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* *Posicionamiento relativo*: Cuando se asigna este valor a un elemento, este cambiara su posición con respecto a su ubicación inicial en el flujo del documento. Dicha traslación dependerá de los valores que se coloquen en las propiedades adicionales y relacionadas a *position:relative;* las cuales son: *top, left, right bottom.* Los contenidos adyacentes al ítem con la propiedad relativa no se verán afectados.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  | Relativa a la imagen de arriba |

* *Posicionamiento absoluto*: Cuando se asigna este valor a un elemento, este cambiara su posición con respecto al elemento ancestro del mismo. En el caso de que un elemento no tenga un ancestro se utilizara el bloque contenedor inicial. Al igual que en el caso anterior para que la propiedad tenga efecto se debe incluir *top, left, right bottom.*

Una forma de asignar un ancestro al elemento es indicar que el contenedor tiene una posición relativa.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  | La división interna se mueve dentro de la división externa que es su ancestro |

* *Posicionamiento fijo*: Cuando se asigna este valor a un elemento este se ubicará en relación con el *viewport.* Esto quiere decir que siempre permanece en el mismo lugar con respecto a la ventana incluso si se desplaza la página. De igual forma las propiedades *top, left, right bottom* son necesarias para lograr el posicionamiento.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

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

Respuesta. –

* *grid-template-columns*: Esta propiedad permite definir tanto el número, como ancho de las columnas que contendrá la cuadrícula. Los valores que se usan en la propiedad van separados por espacios; cada uno de ellos especifica el tamaño de la columna respectiva.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* *grid-template-rows :* Muy similar a la propiedad anterior, este elemento permite definir tanto el número, como el ancho de las filas que contendrá la cuadrícula. Los valores que se usan en la propiedad van separados por espacios; cada uno de ellos especifica el alto de la fila respectiva.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* *grid-gap:* Esta propiedad permite definir el tamaño del espacio entre cada fila y columna dentro de la cuadrícula. Es una abreviación de las propiedades hijas grid-row-gap y grid-column-gap, la primera referida a filas y la segunda a columna.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* *place-content:* Es una propiedad abreviada, del contenedor padre, que permite posicionar los elementos de la cuadricula según su alineación y justificación de contenido. El primer valor establece align-content y el segundo valor establece justify-content. Si solo se proporciona un valor, establece ambas propiedades.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* 1. **Explicad para qué sirven las siguientes propiedades del módulo CSS Flexbox: flex-direction, flex-wrap, order y align-self.**
* *flex-direction*: Es una propiedad que permite definir la dirección en la que se dispondrán los elementos dentro de un contender flexible (elementos flexibles), indicando el eje principal del layout. Dicha propiedad puede tomar valores como: row, row-reverse, column, columna reverse
  + row: organiza los elementos del contenedor horizontalmente de izquierda a derecha.
  + row-reverse: organiza los elementos del contenedor horizontalmente de derecha a izquierda.
  + column: organiza los elementos del contenedor verticalmente de izquierda a derecha.
  + column-reverse: organiza los elementos del contenedor verticalmente de derecha a izquierda.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* *flex-wrap*: Es una propiedad que indica si los elementos flexibles deben ajustarse o no al tamaño de su contenedor. Por defecto el valor es *nowrap*, sin embargo, puede tomar valores como *wrap y wrap-reverse*

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* *flex-order*: Es una propiedad que especifica el orden de un elemento flexible en relación con el resto de los elementos flexibles dentro del mismo contenedor, sobre el eje en que están dispuestos.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* *align-self*: Es una propiedad que permita dar alineación a un elemento dentro del contenedor flexible. La misma puede manejar los siguientes valores:
  + flex-start: el elemento flexible se alineará verticalmente en la parte superior del contenedor.
  + flex-end: el elemento flexible se alineará verticalmente en la parte inferior del contenedor.
  + flex-center: el elemento flexible se alineará verticalmente en el centro del contenedor.
  + Stretch: el elemento flexible se alineará verticalmente de manera que ocupe todo el espacio vertical del contenedor.
  + baseline: el elemento flexible se alineará en la línea de base del eje transversal.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

Pregunta 2. Sobre diseño responsivo, responded a las siguientes preguntas:

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

Respuesta. –

* El diseño web responsivo (RWD) es un enfoque implementado en el desarrollo de sitios web que considera los diferentes tipos de dispositivos que un usuario puede usar para acceder al sitio. La idea de este diseño es ajustar la forma en que se muestra el contenido de una página de acuerdo con las dimensiones de la pantalla del dispositivo, ofreciendo de esta forma una experiencia usuario adecuada al portal de presentación.
  1. **Enumerad al menos cuatro características y/o técnicas que ayudan a realizar un diseño responsivo.**

Respuesta. –

* *Fluid layout (diseño fluido):* Es una técnica en la que el diseño de la página cambia de tamaño a medida que cambia el tamaño de la ventana. Esto se logra mediante la definición de áreas de la página utilizando porcentajes en lugar de anchos de píxeles fijos.
* *Imágenes flexibles*: básicamente es una técnica que busca adaptar este tipo de contenido a la pantalla donde se presentan, ajustando sus dimensiones sin perder la relación de aspecto.
* *Meadia Queries*: es una técnica de CSS que permite que las propiedades de CSS dentro de la regla estén activas si se cumplen ciertas condiciones. Las condiciones pueden ser varias cosas, como la resolución de la pantalla, la profundidad del color, la relación de aspecto y mucho más.
* *Tipografía responsiva:* es una técnica de css en la cual mediante el uso de *media queries* o funciones de css como clamp(), cal() se puede hacer una adaptación del tamaño de una fuente según el dimensionamiento que sea necesario en el momento de presentación.
  1. **Explicad brevemente qué se entiende por "mobile first".**

Respuesta. –

* Es una estrategia de diseño que nos indica que al momento de crear un sitio web, se debe comenzar con un prototipado o esquematización enfocada en pantallas pequeñas inicialmente para luego dar paso al trabajo con *viewports* más grandes. Esencialmente, se trata de ofrecer la experiencia de usuario correcta al dispositivo correcto, tomando en consideración el gran auge que tiene los dispositivos portátiles de pequeñas dimensiones.

Primera parte

Informe documental de sitio web.

* 1. **Descripción de sitio:**

Sitio web que recopila información sobre el planeta Marte desde diversos puntos de vista: los proyectos de la NASA, la literatura y el cine.

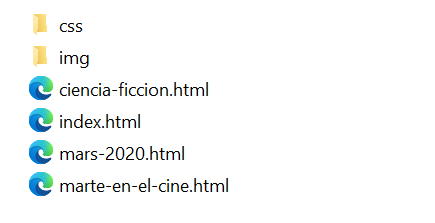
* 1. **Organización de archivos del sitio:**

El sitio se encuentra conformado por 4 páginas webs independientes que a su vez se interconectan entre sí por el menú de navegación e hipervínculos internos:

* Index.html
  + Es la pagina principal, donde se presenta de una manera compacta y grafica las diversas misiones hechas por la nasa al planeta marte
* Mars-2020.html
  + Es una página que presta información sobre una de las misiones listadas en la pagina principal del sitio. Es la página ejemplo de todas las demás misiones listada, se debe usar como esquema para el desarrollo de las demás. Se enlaza con index.html mediante un enlace desde esta,
* Ciencia-ficcion.html
  + Es una página secundaria listada en el menú de navegación que presenta información sobre diversos escritores que han desarrollado obras literales del tema marte (ciencia ficción).
* Marte-en-el-cine.html
  + Es una página secundaria listada en el menú de navegación que presenta información sobre diversas obras de cine que tratan sobre el tema marte.

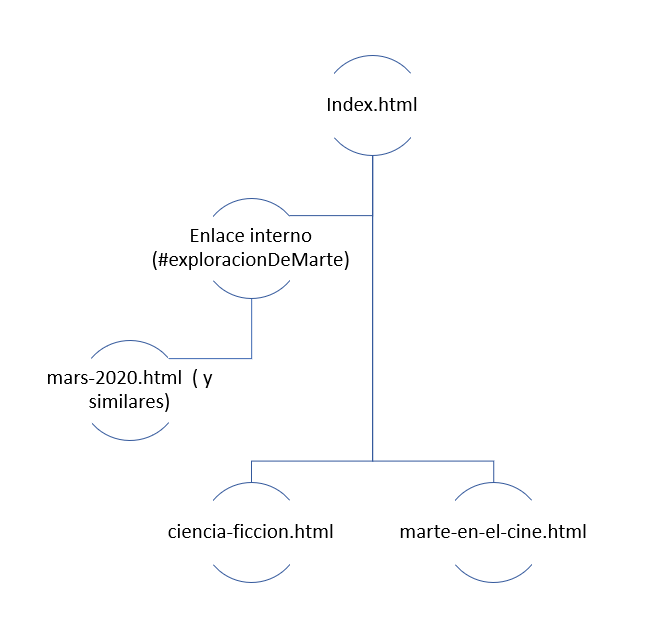
Las cuatro paginas se encuentran en un mismo nivel, las cuales se ven nutridas por imágenes dispuesta en una carpeta, llamada ***img*** al mismo nivel, pero que contiene los archivos correspondientes.

A su vez todas ellas se les aplican reglas CSS en el archivo ***estilos.css*** que se encuentra dentro de una carpera a nivel llamada ***css.*** Mediante las reglas de CSS se logra desarrollar un diseño responsivo, para pantallas menores a 480px y superiores a esa dimensión.



*Organización de archivos del sitio web*

* 1. **Arquitectura de información de sitio:**

**

*Esquema de sitio por paginas*

* 1. **Explicación de los elementos HTML y de las declaraciones CSS:**

*Entidades HMTL en común*

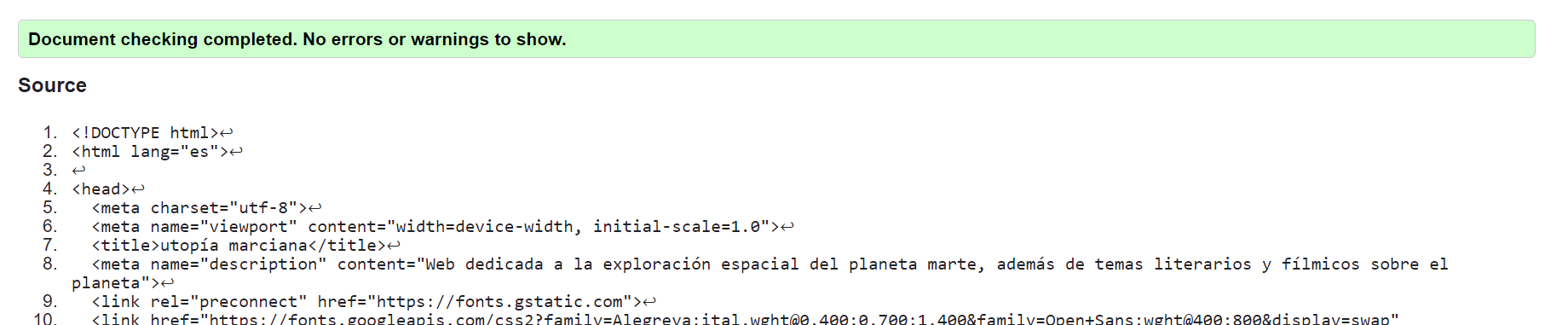
|  |  |
| --- | --- |
| **Etiqueta** | **Explicación** |
| <!DOCTYPE html> | Permite definir el tipo de documento y se agregó a cada página HTML. |
| <html lang="es"> | Define la raíz del documento HTML. En este se usa el atributo “lang” para definir en qué tipo de idioma (español) está escrita la página española. |
| <head> | Contiene la meta data y la información característica del documento. Enlaces a otros recursos, títulos de la página. |
| <meta charset="utf-8"> | En el tag se definen los meta datos del documento. En este caso especificamos la codificación del HTML (utf-8). |
| <title>utopía marciana</title> | Define el título del documento. En este ejercicio se usaron 4 títulos diferentes. |
| <meta name="viewport" content="width=device-width, initial-scale=1.0"> | Esta etiqueta indica al navegador que establezca el ancho del viewport igual al ancho de la pantalla del dispositivo (width=device-width), además que exista una escala inicial del 100%. |
| <meta name="description" content="Web dedicada a la exploración espacial del planeta marte, además de temas literarios y fílmicos sobre el planeta"> | Esta mata información provee información extra sobre cada página. |
| <link rel="preconnect" href="https://fonts.gstatic.com"> | La etiqueta *Link* se usó para establecer una relación entre el HTML y otro documento externo  En este caso indica al navegador que la página tiene la intención de establecer una conexión con otro dominio, en este caso *https://fonts.gstatic.com,* y que desea que el proceso comience lo antes posible. |
| <link href="https://fonts.googleapis.com/css2?family=Alegreya:ital,wght@0,400;0,700;1,400&family=Open+Sans:wght@400;800&display=swap" rel="stylesheet"> | Establece un enlace con las el portal de fuentes de Google. Especifica que fuentes vamos a importar desde Google fonts. |
| <link rel="stylesheet" href="css/estilos.css"> | Estable la relación de la pagina con el archivo css *estilos.css* |
| *<!-- las tres siguientes líneas son un truco para obtener elementos semánticos de HTML5 que funcionan en versiones de Internet Explorer antiguas -->*  *<!--[if lt IE 9]>*  *<script src="https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv.js"></script>*  *<![endif]-->* | Permite ingresar un comentario, en este ejercicio para comentar un truco de CSS en conjunto con un Script que ayuda a que las etiquetas semánticas funcionen en versiones antiguas de IE. |
| <header> | Definimos la cabecera del documento. |
| <div class="container"> | Una etiqueta sin semántica con fines de estilo al ala cual le adiciona una clase (container) para lograr la disposición visual deseada el menú de navegación de todas las paginas. |
| <a href="index.html"><img src="img/logo-black-es.png" alt="logotipo marte ciencia arte literatura"></a> | Nos permite incluir el logo de la página, y además proveer un enlace a la pagina principal desde cualquiera de las paginas que conforman el sitio. |
| <nav>       <ul>            <li><a href="#exploracionDeMarte">Exploración de Marte</a></li>            <li><a href="ciencia-ficcion.html">Ciencia ficción</a></li>            <li><a href="marte-en-el-cine.html">Marte en el cine</a></li>      </ul>  </nav> | Etiqueta semántica *nav* que encapsula el menú de navegación del sitio.  El menú de navegación fue diseñado mediante una lista no ordenada, donde cada item contiene un enlace que con *href* a cada página del sitio.  Es de notar que en el caso de “exploración de marte” se trata de un enlace interno en index.html. |
| <footer> | Se usó para especificar el pie de página. |
| <div class="container"> | Al igual que en el header, se uso la misma entidad con la clase para lograr la disposición del contenido del footer según el diseño solicitado. |
| <img src="img/logo-white-es.png" alt="logotipo blanco marte ciencia arte literatura"> | Nos permite incluir el logo de la página distinto al de la cabecera, en el pie de página. |
| <a href="mailto:hola@utopiamarciana.net">hola@utopiamarciana.net</a> | Hipervínculo que apunta a correo electrónico de la pagina. |
| <form action="#" method="post">  <p>Siempre al día con nuestra <span>newsletter</span></p>          <div>  *<!-- La etiqueta no es necesaria -->*            <input type="email" id="email" name="email" placeholder="Escribe tu e-mail" required>          </div>          <div>            <p>              <label for="acuerdo">                <input type="checkbox" id="acuerdo" name="acuerdo" value="OK">                Acepto la <a href="">política de protección de datos</a>              </label>            </p>            <p>              <button type="submit">Suscríbeme</button>            </p>          </div>   </form> | Esqueleto de una forma pequeña usada para recolectar los correos electrónicos de los interesados en el newsletter .  Su esquema es muy sencillo. Contiene un *input* para correo electrónico, con un texto descriptivo como *placeholder.* Para proceder con el envío este campo de define como *required.*  Al final de la forma se encuentra *un input* tipo *checkbox* con enlace a política de datos de la página.  Además, se encuentra la etiqueta *button* para creación de botón de envío. |
| <div class="bottomFooter">          <ul>            <li><a href="">Aviso Legal</a></li>            <li><a href="">Politica de privacidad</a></li>            <li><a href="">Politica de cookies</a></li>          </ul>          <p><time datetime="2021">2021</time> &#169; Utopía marciana</p>          <p>Actividad realizada por Ivan Schartun, para la asignatura HTML y CSS. Segundo semestre del curso <time datetime="2021">2020</time>-<time datetime="2021">2021</time></p>        </div> | División sin semántica que contiene información de aviso legal, política de privacidad y política de cookies.  A estas informaciones se puede acceder mediante los enlaces dispuestos en una lista no ordenada.  Además, se incluye al final información del creador del sitio.  De notar el uso de etiquetas *time* para proporcionar a la maquina información de tiempo. |

*Entidades HMTL de index.hmtl*

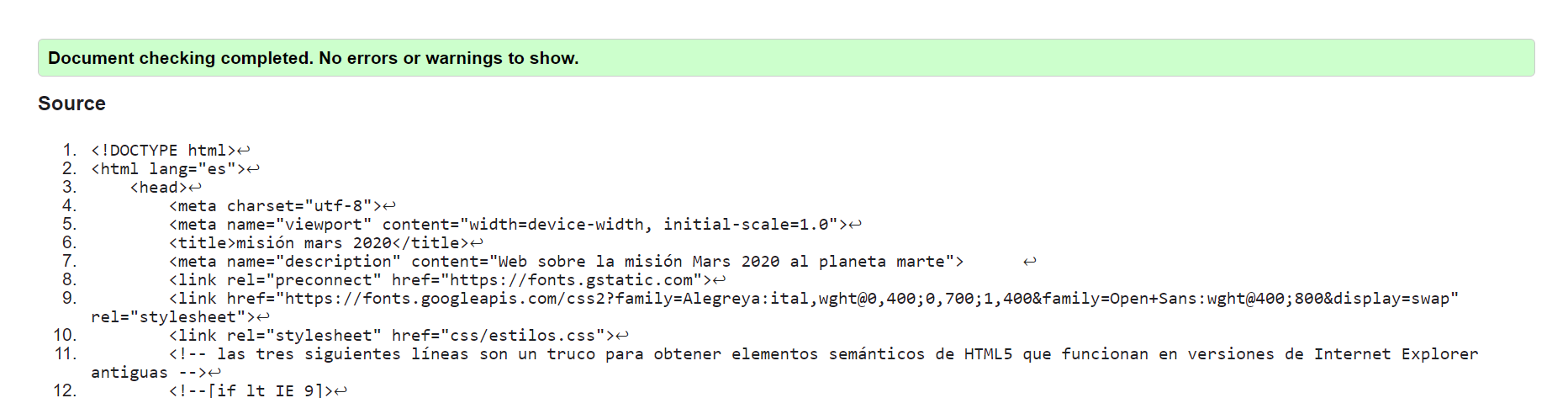
|  |  |
| --- | --- |
| **Etiqueta** | **Explicación** |
| <main> | Especifica el contenido principal del documento. |
| <section class="hero"> | Permite encapsular un contenido genérico dentro del documento, no semántico. En este caso el encabezado con imagen de la página principal, contiene una clase para aplicarle css. |
| <div class="container"> | División no semántica que se usa para encapsular información dentro del encabezado con la imagen, no es header. Con clase *container.* |
| <h1> | Con esta etiqueta definí el encabezado de la sección de index.html. |
| <p> | Define un párrafo. |
| <section class="container"> | División no semántica que se usa para encapsular información principal de la página inicial del sitio. Con clase container para aplicar el respectivo css. |
| <h2 id="exploracionDeMarte">Exploración de Marte</h2> | Encabezado de segundo nivel que agrega un sub titilo al inicio de información de página. Hay que destacar que se le agrega un *id* que sirve como un ancla para un salto desde el menú de navegación principal, según lo requerido en el ejercicio. |
| <abbr> | Es una etiqueta que permitió dar información sobre abreviaciones dentro de los párrafos. En ella se usó el atributo “title” para ingresar lo que significaba la abreviación. |
| <ul class="grid"> | Etiqueta que permite crear una lista no ordenada. En esta pagina se uso dicha referencia para crear un listado de las misiones expuestas por el portal. Se agrega una clase para aplicar css |
| <li> | Etiqueta que permite crear un elemento de lista. Cada misión fue contenida en este elemento. |
| <img src="img/mars-odyssey.jpg" alt="Sonda Mars Odyssey"> | Etiqueta que permite insertar una imagen en el documento. En el caso de index.html. Cada item de lista contenía esta identidad, que mostraba una imagen de la misión. |
| <h3> | Etiqueta para encabezado de 3 nivel. Cada *li* contenía un subtitulo con el nombre de la misión. |
| <p>Mars | Etiqueta de párrafo. Cada *li* contenía un párrafo explicativo de cada misión. |
| </time> | Define un tiempo específico de una forma que es entendible para la máquina. Se usó el atributo “datetime” con las fechas que se mostraban en dentro del contenido. |

Debéis elaborar un El objetivo del informe es conseguir que cualquiera que debiera continuar construyendo el sitio tuviera claro el procedimiento a seguir (organización de archivos, arquitectura de la información, diseño gráfico…). Este informe, como mínimo, incluirá una explicación de los elementos HTML y de las declaraciones CSS que se han utilizado, explicando lo que es común a todas las páginas y también las estructuras particulares de cada una de ellas. Este informe deberá entregarse junto con las preguntas teóricas de la primera parte, en un documento con cualquiera de las extensiones siguientes: .doc, .odt o .pdf.

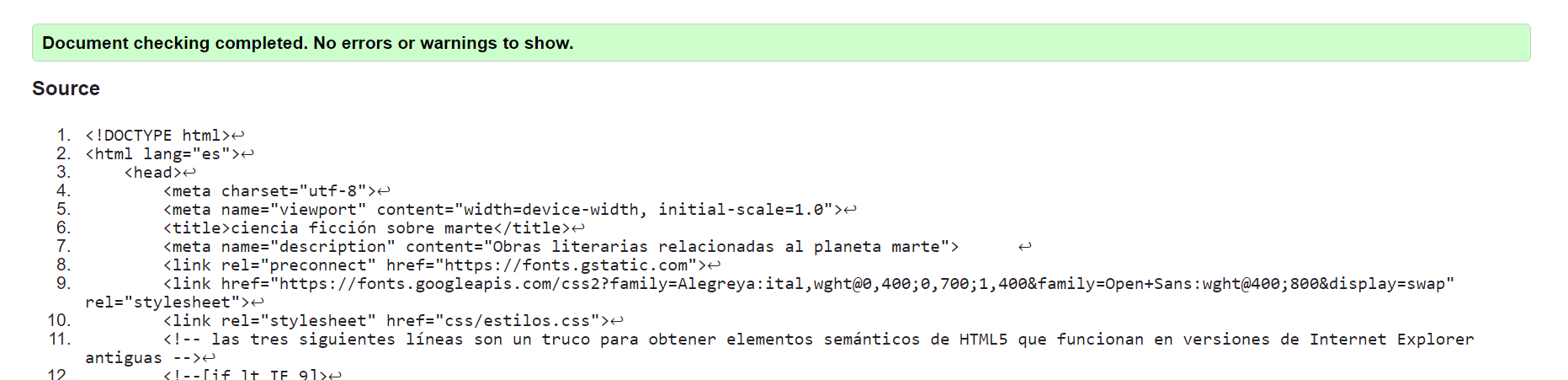
* 1. **Validaciones de archivos en 3WC:**
* *Validación index.html*



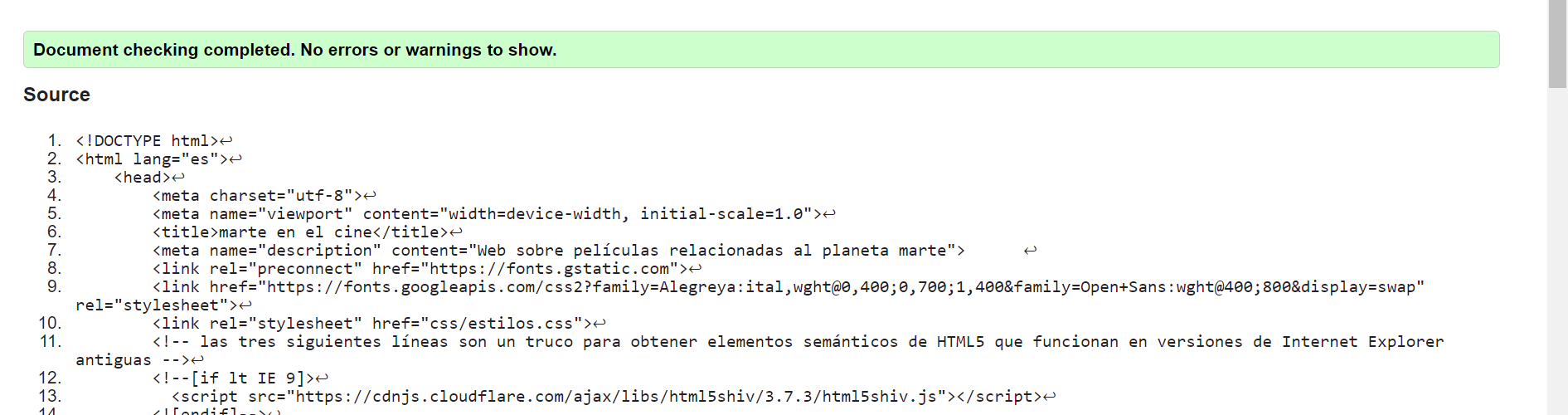
* *Validación mars-2020.hmtl*



* Validación ciencia-ficcion.html



* Validación marte-en-el-cine.html



* Validación de archivo de estilos.css

