

2024

HITO DEL 1ER TRIMESTRE DE MARIO DEL REY

CampusFP
Mario del Rey
20-11-2024

Tabla de contenido

PREGUNTAS TEORICAS.....	2
Explica los orígenes de los lenguajes de marcas.....	2
Explicar los diferentes estándares y organismos oficiales en el desarrollo y diseño de lenguajes de marcas.....	2
Explica los orígenes y las diferencias entre CSS y SCSS.	3
Principales etiquetas HTML5, así como una breve descripción de que son y cómo funcionan las etiquetas semánticas.....	4
Principales elementos CSS y su aplicación.....	5
Guía de creación de un sitio web con el paso a paso del desarrollo.....	6
Guía y uso de librerías de CSS para el desarrollo web.	6
Webgrafía en cada HTML de las páginas consultadas.....	¡Error! Marcador no definido.
BIBLIOGRAFIA.....	8

PREGUNTAS TEORICAS

Explica los orígenes de los lenguajes de marcas.

Los lenguajes de marcas tienen sus orígenes en la necesidad de estructurar y representar de manera más eficiente los documentos de texto, particularmente en el ámbito de la informática y la web. Su propósito principal es marcar (es decir, identificar) ciertos elementos dentro de un texto para indicar cómo deben ser presentados o procesados por un sistema informático, sin que la estructura del contenido se vea alterada.

Explicar los diferentes estándares y organismos oficiales en el desarrollo y diseño de lenguajes de marcas.

1. World Wide Web Consortium (W3C)

El **W3C** es, sin duda, el organismo más influyente y ampliamente reconocido en el desarrollo de estándares para la web, incluidos los lenguajes de marcas.

HTML (HyperText Markup Language): El W3C supervisa la evolución de HTML, que es el lenguaje de marcado principal para la creación de páginas web.

XML (Extensible Markup Language): XML, aunque más orientado al almacenamiento y transporte de datos, fue desarrollado bajo el auspicio del W3C. Es un lenguaje de marcas flexible y extensible que permite definir datos de manera estructurada.

CSS (Cascading Style Sheets): Aunque no es un lenguaje de marcado, CSS se utiliza junto con HTML para definir el diseño y la presentación visual de los documentos web.

SVG (Scalable Vector Graphics): Es un lenguaje de marcas basado en XML para describir gráficos vectoriales en la web.

XHTML (Extensible Hypertext Markup Language): Es una versión más estricta y XML-compatible de HTML. Aunque no es tan popular como HTML5, fue una transición importante entre HTML y XML.

ARIA (Accessible Rich Internet Applications): Aunque no es un lenguaje de marcado per se, ARIA proporciona roles y propiedades para mejorar la accesibilidad de aplicaciones web dinámicas, especialmente aquellas basadas en tecnologías como JavaScript.

2. International Organization for Standardization (ISO)

La **ISO** es una organización no gubernamental que desarrolla y publica estándares internacionales en una variedad de áreas, incluidos los lenguajes de marcas. Su trabajo complementa a organizaciones como el W3C, y en algunos casos, contribuye con especificaciones más generales y abstractas que pueden ser aplicadas en distintas tecnologías.

ISO/IEC 8879:1986 (SGML - Standard Generalized Markup Language): Este es el estándar original de SGML, que sirvió como base para HTML y XML. La ISO desarrolló SGML para permitir la creación de documentos estructurados de manera jerárquica y flexible. Aunque SGML ha sido reemplazado por tecnologías más modernas como XML, sigue siendo un estándar importante en la historia de los lenguajes de marcas.

ISO/IEC 19503 (UML - Unified Modeling Language): Aunque UML no es un lenguaje de marcas en el sentido tradicional, se utiliza para especificar, visualizar y

documentar modelos de sistemas de software, y a menudo involucra un enfoque similar al de los lenguajes de marcas en cuanto a la estructura y la representación de información.

3. Ecma International

Ecma International es una organización que desarrolla estándares para tecnologías de la información y la comunicación. Aunque es más conocida por sus trabajos en lenguajes de programación y plataformas como **ECMAScript** (el estándar subyacente a JavaScript), también participa en la creación de estándares relacionados con los lenguajes de marcas, especialmente en lo que respecta a la interoperabilidad y la integración entre diferentes plataformas tecnológicas.

ECMA-357 (XML Document Object Model - DOM): Este estándar describe una interfaz de programación para representar documentos XML como estructuras de objetos, lo que facilita su manipulación mediante lenguajes de programación.

4. Internet Engineering Task Force (IETF)

La **IETF** es una organización de estándares abierta, que trabaja principalmente en estándares relacionados con la infraestructura de Internet, pero también tiene un impacto en los lenguajes de marcas, especialmente aquellos utilizados para la transmisión de datos.

RFC 3023 (XML Media Types): Este documento describe cómo los documentos XML deben ser identificados y transportados en la web, especificando el tipo MIME que debe usarse para XML y otros tipos de documentos relacionados con lenguajes de marcas.

RFC 7230-7235 (HTTP/1.1): Si bien no es un estándar de lenguaje de marcas en sí, estos documentos regulan el protocolo HTTP, que es el medio principal de transmisión de documentos HTML y XML en la web. En este contexto, el trabajo de la IETF en HTTP afecta directamente a la interoperabilidad de los lenguajes de marcas utilizados en la web.

5. W3C Web Accessibility Initiative (WAI)

El **WAI** es una subcomisión del W3C dedicada a mejorar la accesibilidad de la web para personas con discapacidades. A través de sus directrices y recursos, el WAI ayuda a garantizar que los lenguajes de marcas sean inclusivos y accesibles para todos los usuarios.

WCAG (Web Content Accessibility Guidelines): Estas pautas proporcionan recomendaciones para hacer que el contenido web sea accesible a una gama más amplia de personas con discapacidades. Las pautas de accesibilidad están estrechamente vinculadas a la forma en que se diseñan y estructuran los lenguajes de marcas, especialmente en HTML y ARIA

Explica los orígenes y las diferencias entre CSS y SCSS.

Orígenes css:

El concepto de CSS (Cascading Style Sheets) nació en 1994 gracias al trabajo de Håkon Wium Lie y Bert Bos. Håkon Wium Lie, quien trabajaba en el CERN con Tim Berners-Lee (el creador de la World Wide Web), propuso un sistema para separar la estructura del contenido (HTML) de su presentación (estilos visuales). Este sistema fue presentado en la conferencia World Wide Web Conference en 1994, y fue adoptado y estandarizado por el W3C (World Wide Web Consortium) a partir de 1996.

Orígenes scss:

SCSS es una sintaxis de Sass, un preprocesador de CSS que fue creado por Hampton Catlin en 2006. El objetivo original de Sass era proporcionar características avanzadas como variables, mixins y anidamiento para mejorar la escritura de estilos en CSS. En 2010, se introdujo SCSS como una nueva sintaxis de Sass, compatible con el CSS tradicional, para que fuera más accesible para los desarrolladores acostumbrados a la sintaxis estándar de CSS.

Diferencias entre css y scss

scss soporta variables de forma natural mientras que css solo las soporta con CSS Custom Properties, Css no soporta anidamiento y scss si, en Css no existe mixins mientras que scss soporta mixins reutilizable, en Css no soporta herencias mientras que scss solo lo soporta mediante @extend, css no soporta operaciones matemáticas y scss si y css tiene compatibilidad nativa con el nativo mientras scss requiere de un css

Principales etiquetas HTML5, así como una breve descripción de que son y cómo funcionan las etiquetas semánticas.

1. **Html:** es la etiqueta raíz que envuelve todo el contenido de un documento HTML
2. **Hear:** contiene metadatos sobre el documento HTML
3. **Title:** define el título del documento que se muestra en la barra de título del navegador o en la pestaña
4. **Meta:** Define metadatos del documento como la codificación de caracteres, autor, descripción, entre otros
5. **Link:** Se utiliza para vincular archivos externos como hojas de estilo CSS, iconos o fuentes web.
6. **Script:** Permite incluir o hacer referencia a un archivo JavaScript en el documento HTML.
7. **Body:** Contiene todo el contenido visible del documento HTML, como texto, imágenes, videos, etc.
8. **Header:** Representa la sección introductoria de una página o una parte de la misma. Usualmente contiene un logo, título o menú de navegación.
9. **Footer:** Define el pie de página de un documento o sección. Usualmente contiene información sobre derechos de autor, enlaces de contacto, etc.
10. **Nav:** Representa una sección de navegación que contiene enlaces a otras partes del sitio web.
11. **Section:** Representa una sección genérica de contenido, con un tema o propósito específico. Puede contener un encabezado.
12. **Article:** Define contenido autónomo y completo, como artículos, publicaciones de blogs, etc., que puede ser distribuido de forma independiente.
13. **Aside:** Representa contenido relacionado pero separado del contenido principal, como barras laterales, citas, anuncios, etc.
14. **Main:** Define el contenido principal de un documento, que debe ser único y relevante para el tema de la página.
15. **H1 a h6:** Representan encabezados de diferentes niveles, donde `<h1>` es el más importante y `<h6>` el menos importante.
16. **P:** Define un párrafo de texto.
17. **A:** Representa un enlace que puede dirigir a otro documento o una parte del mismo.
18. **Img:** Permite insertar una imagen en la página. Usa el atributo `src` para definir la fuente de la imagen.
19. **Ul:** Define una lista no ordenada. Los elementos de la lista se especifican con ``

20. Ol: Define una lista ordenada, en la que los elementos tienen un orden numérico o alfabético.
21. Li: Define un ítem dentro de una lista ordenada () o no ordenada ().
22. Table: Define una tabla. Los elementos internos incluyen <tr>, <td>, y <th>.
23. Tr: Define una fila dentro de una tabla (<table>).
24. Td: Representa una celda dentro de una fila de tabla (<tr>).
25. Th: Define una celda de encabezado en una tabla, generalmente en negrita y centrada.
26. Form: Define un formulario que puede contener campos interactivos como cuadros de texto, botones de envío, etc.
27. Input: Define un control interactivo dentro de un formulario, como un cuadro de texto, una casilla de verificación, etc.
28. Button: Define un botón, generalmente usado para enviar un formulario o realizar alguna acción JavaScript.
29. Video: Permite incrustar un video en una página web. Soporta diferentes fuentes de video y controles.
30. Audio: Permite incrustar contenido de audio en una página web. Al igual que <video>, puede incluir controles como reproducir, pausar, volumen, etc.

Principales elementos CSS y su aplicación.

Color: Define el color del texto

background-color: Define el color de fondo de un elemento.

Font-family: Especifica la fuente del texto. Puede usar fuentes estándar o fuentes web externas.

Font-size: Establece el tamaño de la fuente.

Font-weight: Define el grosor del texto (normal, negrita, etc.).

Line-height: Controla la altura de la línea de texto, es decir, el espacio vertical entre las líneas.

Text-align: Alinea el texto dentro de un elemento (izquierda, derecha, centrado, justificado)

Text-transform: Controla la transformación del texto (mayúsculas, minúsculas, capitalización de cada palabra).

Padding: Controla el espacio interno (relleno) entre el borde de un elemento y su contenido.

Margin: Controla el espacio externo (margen) alrededor de un elemento, separándolo de otros elementos

Border: Define un borde alrededor de un elemento.

Width y height: Establecen el ancho y la altura de un elemento.

Display: Controla cómo se comporta un elemento en el flujo del documento. Por ejemplo, puede ser block, inline, flex, grid, none, etc.

Position: Controla el tipo de posicionamiento de un elemento. Puede ser static, relative, absolute, fixed, sticky.

top, right, bottom, left: Usados junto con position para mover un elemento desde su posición inicial

z-index: Define la profundidad de apilamiento de los elementos posicionados. A mayor valor, más "arriba" en la pila.

Overflow: Controla cómo se manejan los contenidos que exceden el tamaño de un contenedor (puede ser visible, hidden, scroll, o auto).

flex y flex-direction: Usados en un contenedor con display: flex para alinear y distribuir los elementos dentro de él.

Justify-content: Controla la distribución de los elementos en el eje principal de un contenedor flex (por ejemplo, centrado, espaciado, alineado al inicio o al final).

Align-items: Alinea los elementos a lo largo del eje transversal en un contenedor flex.

Grid y grid-template-columns: Se utiliza para crear un diseño de cuadrícula (grid) con múltiples columnas y filas.

Box-shadow: Añade una sombra alrededor de un elemento.

Text-shadow: Añade una sombra al texto.

Opacity: Controla la opacidad de un elemento, permitiendo hacer que sea más transparente.

Transition: Define cómo los cambios de estilo de un elemento deben ocurrir durante un tiempo (animación de transición).

Animation: Define animaciones complejas mediante un conjunto de reglas clave (keyframes).

Cursor: Cambia el estilo del cursor cuando se pasa por encima de un elemento

Visibility: Controla si un elemento es visible o no, pero conserva su espacio en el diseño

Border-RADIUS: redondea las esquinas de un elemento, como botones o contenedores

White-space: Controla cómo se maneja el espacio en blanco dentro de un elemento, como si se permite el ajuste de línea o no.

Guía de creación de un sitio web con el paso a paso del desarrollo.

Creación de un proyecto el cual tendrá una cantidad de archivos tanto de html como de css también tendremos que organizar todos los documentos del proyecto para que este bien ordenado dotas los archivos multimedia guardados en su respectiva carpeta y los css también guardar en su respectiva carpeta. Estructuraríamos los HTML para que tengan la forma que queramos las páginas y añadimos en los css las clases necesarias para que nuestra página acabe teniendo forma.

Guía y uso de librerías de CSS para el desarrollo web.

CSS-Tricks:

- ☐ Ve a la sección de **Snippets** (en el menú principal).

- ☐ Usa la barra de búsqueda para encontrar efectos específicos, como "hover effects" o "responsive design".
- ☐ Explora los artículos para entender cómo funcionan y copia los fragmentos de código CSS.
- ☐ Pega y adapta el código en tu archivo CSS según tus necesidades.

CodePen:

- ☐ Usa la barra de búsqueda para buscar términos como "CSS animations" o "responsive cards".
- ☐ Filtra por **Pens** (proyectos individuales) para encontrar ejemplos rápidos.
- ☐ Haz clic en cualquier ejemplo para ver el código en la pestaña de **HTML**, **CSS**, y **JS**.
- ☐ Puedes copiar el código o hacer un "Fork" para editar tu propia versión.

Hover.css:

Poniendo el siguiente enlace podremos usar la librería de hover

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/hover.css/2.3.1/css/hover-min.css">
```

Animate.css:

Para usar animate hay que poner en el html el siguiente enlace

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css">
```

CSS Loaders:

- ☐ Navega por la galería de efectos y selecciona el loader que te guste.
- ☐ Copia tanto el HTML como el CSS asociado.
- ☐ Ajusta el código para integrarlo en tu proyecto.

BIBLIOGRAFIA

ChatGPT. (n.d.). Chatgpt.com. Retrieved November 12, 2024, from <https://chatgpt.com/>

CSS: Cascading Style Sheets. (n.d.). MDN Web Docs. Retrieved November 12, 2024, from <https://developer.mozilla.org/en-US/docs/Web/CSS>

CSS tutorial. (n.d.). W3schools.com. Retrieved November 12, 2024, from <https://www.w3schools.com/css/default.asp>

Fernández-Valmayor, C. A., Navarro, A., Fernández-Manjón, B., & Sierra, J. L. (n.d.). Lenguajes de programación, lenguajes de marcado y modelos hipermedia: una visión interesada de la evolución de los lenguajes informáticos. Rediris.Es. Retrieved November 12, 2024, from http://elies.rediris.es/elies24/fernandezvalmayor_cap3.htm

HTML: Lenguaje de etiquetas de hipertexto. (n.d.). MDN Web Docs. Retrieved November 12, 2024, from <https://developer.mozilla.org/es/docs/Web/HTML>

HTML tutorial. (n.d.). W3schools.com. Retrieved November 12, 2024, from <https://www.w3schools.com/html/default.asp>

(N.d.). Html.com. Retrieved November 12, 2024, from <https://html.com/document/>