**Curso de Frontend Developer**

**2. HTML y CSS definición y usos:**

Internet: interconnected y network.

Red de computadoras conectadas en todo el mundo.

Tim Berners-Lee invento Word Wide Web (WWW)

Fundo el consorcio W3C para estandarizar el desarrollo de las tecnologías base internet.

HTTP: Hyper Text Tranasfer Protocol :// (permite la comunicación de datos entre dispositivos).

URL: Uniform Resource Locator: (Dirección de un sitio web).

HTML: Hyper Text Markup Language (lenguaje para un describir la estructura de un sitio web).

En 1994 apareció CSS Casacade Style Sheet.

**3. ¿Que son y para que nos sirven HTML y CSS?:**

HTML: es un lenguaje de marcado usado para decirle a tu navegador como estructurar la página que visita.

CSS: Es un lenguaje que permite crea páginas web con un diseño agradable para los usuarios.

**4.DOM, CSSOM, Render Tree y el proceso de renderizado de la web.**

DOM: Document Object Model. Es una transformación del código HTML escrito por nosotros a objetos entendibles para el navegador.

CSSOM: así como el DOM para el HTML, el CSSOM es una representación de objeto de nuestros estilos en CSS.

Render Tree: es la unión entre el DOM y el CSSOM para renderizar todo el código de nuestra página web.

1. Procesa el HTML para construir el DOM.
2. Procesa el CSS para construir el CSSOM
3. El DOM se una con el CSSOM para crear el Render Tree.
4. Se aplican estilos CSS en el Render Tree.
5. Se “Pintan” los nodos en la pantalla para que los usuarios vean el contenido de la página web.

**6.Anatomia de un elemento HTML: atributos, andamiento y elementos vacíos.**

Nuestro HTML se componen de:

* Etiqueta de apertura; el nombre de nuestra etiqueta encerrando entre símbolos de mayor o menor. Por ejemplo: <H1>
* Contenido: Dentro de nuestras etiquetas podemos añadir texto u otros elementos HTML, lo que conocemos como anidamiento.
* Etiqueta de cierre: Son casi iguales que las etiquetas de apertura, pero también necesitan un Slash (/) antes del nombre de la etiqueta. Por ejemplo </H1>.

**7. Anatomia de un Documento HTML: DOCTYPE, HTML, head y body.**

DOCTYPE: Aquí definimos el documento esta bajo el estándar HTML5.

<meta>: Define los metados que no pueden ser definidos usando otro elemento HTML, como por ejemplo el tipo de codificación; UTF-8.

<input>: es un campo que en donde el usuario puede ingresar datos. Es usada normalmente en formularios como el de inicio de sesión, registro.

<nav> </nav>: esta etiqueta que usamos para colocar enlaces de navegación.

<article> </article>: la usamos normalmente para definir contenido que puede vivir independiente al resto del contenido.

<section> </section> : en esta etiqueta colocamos el contenido de las secciones de nuestra página.

<aside> </aside> el contenido de esta etiqueta no le agrega sentido al resto del contenido de la página, es decir que, si se elimina, no afecta. Como por ejemplo la publicidad de nuestro sitio.

<ol> </ol>: describe una lista ordenada. Los ítems se escriben con la etiqueta <li>.

<ul></ul>: describe una lista desordenada.

<li></li>: utilizado para escribir los ítems de una lista.

<form></form> usado para crear formularios.

**9. La importancia del código semántico.**

Es importante que como desarrolladores tengamos claro el significado de escribir código. Debes ser consciente de que la manera en la que codeas tenga sentido.

La semántica HTML no es más que darle sentido y estructura a lo que estas escribiendo. Muy importante para el navegador. No todos los elementos deberían ser un div.

**10. Tipos de errores en HTML, debugging y servicio de validación de etiquetas.**

**Errores sintácticos:** Son errores de escrituras en el código evitan que el programa funcione. Pueden ser errores de tipado

**Errores lógicos:** En estos la sintaxis es correcta, pero el código no hace lo que debería. El programa funciona, pero de forma incorrecta.

**12. Anatomía de una declaración CSS: Selectores, Propiedades y Valores:**

**Selector:** Son la referencia a los elementos HTML que queremos estilizar. Los nombre de estas etiquetas van seguidas de una llave de apertura y otra de cierre ({}).

**Propiedades:** Son el tipo de estilos que queremos darle a nuestros elementos. Van seguidas de dos puntos.

**Valores:** Son el estilo que queremos que tomen nuestro elemento HTML con respecto a una propiedad.

**13**. **Tipos de selectores, pseudo-clases y de pseudo elementos:**

\* (asterisco): es el selector universal. Las propiedades se aplicarán a todos los elementos de nuestro HTML.

Tipo: Son selectores que se aplicaran a ciertos elementos HTML en específico.

Clase: Si nuestras etiquetas de HTML tienen un atributo class podemos usar ese valor o identificador para que los cambios en el CSS afecten únicamente a ese elemento.

ID: Es similar al anterior, si la etiqueta HTML tiene un ID podemos afectar solo ese elemento.

Las pseudo-clases y pseudo-elementos nos permiten ser aún más específicos con que elementos o partes de nuestros elementos deben recibir estilos.

Ejemplo:

* P: first-child.
* P: last-child.
* P: nth-child(2n). para los pares
* P: nth-child(2n+1). Para los impares.

**15. Valores relativos y absolutos:**

Los valores absolutos son, por ejemplo, centímetro, milímetros, pixeles y pulgadas. Se llaman de esta forma porque no tienen en cuenta a nadie más, no depende dela medida de otra unidad.

Los valores relativos, dependen de otra unidad de medida o elemento. Por ejemplo, porcentanjes, vmx, em, entre otros.

**18. Posicionamiento en CSS:**

Position:relative; el elemento se coloca en relación a su posición normal.

Position:absolute: los elementos con esta posición se ubican en relación al elemento relativo mas cercano.

Position:fixe; el elemento se ubica en relación con la ventana del navegador.

Position:sticky; el elemento se posiciona en función de la posición de desplazamiento del usuario.

Position:static; los elementos HTML son estáticos por defecto, siguen el flujo normal de la página.

Position:initial; ubica el elemento en la posición que tiene de forma predeterminada.

Position:inherital; como su nombre lo indica, este valor hereda esta propiedad de su elemento padre.

**19. ¿Qué son y para que nos sirven las arquitecturas CSS?**

Objetivos

* Predecible
* Realizable
* Mantenible
* Escalable

Buenas practicas

* Establecer reglas
* Explicar la estructura base
* Establecer estándares de codificación
* Evitar largas hojas de estilo
* Documentación

**20. OOCSS, BEM, SMACSS, ITCSS y Atomic Design:**

**Buscar en google documentación.**

**Object-fit: cover; (La imagen vuelve a recuperar su tamaño)**