-HTML (hyper text markup language) fue creado para **dar estructura y contenido a la página web**, programa descriptivo declarativo

-CSS (cascading style sheets) es un lenguaje de presentación creado **para dar estilo y apariencia al contenido web,** programa descriptivo declarativo

Selector Determina el estilo que se le va a aplicar a un elemento. Los selectores están antes de cualquier {...} =

Tipo: Son los más básicos. Funciona con los elementos sin ningún atributo especial. Hay que tratar de usarlos solo cuando sea posible.
html: ...
css: p{...}
Clase: Permite ponerle el mismo estilo a una lista de elementos, agregándole el atributo "class". Las clases se escriben en CSS poniendo un punto . delante del nombre de la clase. Está permitido usar la misma clase en múltiples elementos en la misma página Está permitido usar más de una clase por elemento
html:<p class: "hola">...
css: .hola{...}
ID: Son parecidos a los selectores de clases, pero son utilizados solo para UN ÚNICO elemento por página.
Usan un atributo id en el elemento HTML. Se escriben con un # delante del id en el css, y en Java Script también.
html: <´p id="identif"> …
css: #identif{...}

Propiedades El estilo que se aplica puede ser una o múltiples propiedades que están contenidas dentro de {...} y el texto seguido de :
Valor Define el comportamiento de una propiedad Es el valor que va después de los : y antes del ; Dependiendo de la propiedad el valor puede ser decimal / hexadecimal, incluir unidades o puede ser un texto.

¿Cómo incluir un estilo css?
La mejor práctica para usar CSS es usar un archivo externo, referenciado en la sección head con un <link>, incorporándose en el href:.

CSS resuelve las colisiones usando el mecanismo de cascada con el siguiente criterio: (las respuestas incorrectas restan puntaje) (PREGUNTA DE PARCIAL)

Falso, depende de lo que diga el CSS, Verdadero, siempre que el CSS no cambie el orden

---

-Java Script lenguaje que funciona para darle un comportamiento a la página
Puedo: Validar formularios, reaccionar a lo que hace el usuario, podemos ocultar y hacer visibles partes del html, hacer cálculos, cargar contenidos, hacer SPA (single page aplications), entre otros

(Son INDEPENDIENTES uno del otro. No hay que mezclarlos.)

Tags: <p> <a> <html> <head> <h1> <footer> <main> <header> <article>... (definen los objeto dentro de la pagina)

Atributos: <img> <input> <button> <script> <table> (para agregar instrucciones adicionales a los elementos)

Todos los documentos HTML tienen una estructura con los siguientes elementos

doctype: Es usado para decirle al navegador que versión de HTML va a usar el documento.
html: - Va seguido del doctype. - Indica el comienzo y el fin del documento. - Es el elemento raíz (root), todos los demás descienden de éste.
body: - Es utilizado para poner todo el contenido visible de la página web. - Dentro de este elemento, puede haber texto, hyperlinks, imágenes, etc. - Puede haber solo uno por archivo HTML.

Texto: text-align:Centrado (text-align: center)
Derecha (text-align: right)
Izquierda (text-align: left)
Justificado (text-align: justify)
text-transform: .mayuscula {text-transform: uppercase;}
.minuscula {text-transform: lowercase;}
.capitalizado {text-transform: capitalize;}
text-decoration: para agregar o eliminar decoración(no recomendable)

LAYOUTS

Disposición que tiene la página en cuanto a cómo se van a distribuir y organizar los elementos. Define la estructura básica de una página/interfaz del usuario en una aplicación
“Es el esqueleto general de la página”
-Header
-Nav
-Content
-Footer

Contenedores:

** **

Elemento que define un bloque para secciones o lugares del sitio. Puede incluir varios elementos y nos ayuda a construir el layout y el diseño.

<spam>

Elemento inline para agrupar texto, palabras o frases, por ejemplo partes de un párrafo u oración para resaltar una palabra

Son simples contenedores de HTML. Los div y span como todos los elementos, tienen las propiedades class o id. Es importante elegir un nombre de clase o id con sentido en el contexto. DIV Es un elemento que define un bloque. Generalmente para secciones largas del sitio. Puede incluir varios elementos. Nos ayuda a construir el layout y el diseño. SPAN Es un elemento “inline” usado para agrupar texto, palabras o frases. Por ejemplo dentro de un párrafo BOX MODEL

BOX MODEL

¿ Que tecnologías Front End necesita para implementar el modelo de cajas Box-Model?
¿En que consiste el modelo y que propiedades del estilo afectan las dimensiones del mismo? (PREGUNTA DE PARCIAL)

Tecnologias: Html + CSS
Componentes o propiedades del Box model: width, padding, border, margin
Dimensiones: Width Total del elemento = width + left padding + right padding + left border + right border + left margin + right margin

---

Modelo que usa HTML y CSS para diagramar la página. Este concepto plantea que todo elemento es una caja o pagina representada con una caja rectangular (contenedor)
CSS permite controlar el aspecto de las cajas, y lo hace con **4 partes:**

Content(contenido): Tiene un ancho (width), y un alto(height).
(Todos los elementos tienen un alto y ancho heredado.)

Padding(separacion entre el contenido y el borde desde dentro de la caja): Genera un espacio o margen interior transparente dentro del elemento.

Border(borde): Se utiliza para bordear con una línea alrededor del elemento.

Margin(margen): Genera un margen exterior transparente fuera de un elemento. Puede usarse para separar bloques .

Estos pueden modificar los diferente lados sin necesidad de relacionarse con los demás

Margin-top
Margin-bottom
Margin-left
Margin-right
Padding-top
Padding-bottom
Padding-left
Padding-right

Unidades de medida:

ABSOLUTAS: pixeles (px) (pt - mm - cm) Están completamente definidas, ya que su valor no depende de otro valor de referencia. Ajustan tamaños fijos en los navegadores y pantallas. Poca flexibilidad. Sirve cuando conocemos los tamaños de las **salidas.**
RELATIVAS: porcentaje (%) (em - rem - vw - vh) No están completamente definidas, ya que su valor siempre es dependiente respecto a otro valor de referencia padre. Permiten ajustes con cambios de tamaño de pantalla. Mayor flexibilidad.

Tipos de caja: CSS puede definir la manera en la que los elementos de una página "encajan" uno con otros. Estas maneras se pueden categorizar en:

BLOCK: Ocupan el ancho total del elemento contenedor. Las dimensiones son controlables. Insertar un salto de línea en el flujo al terminar.

INLINE: Ocupan el ancho total de su contenido. Las dimensiones no son fáciles de controlar. Se alinea al texto.

Se controlan con la propiedad display. elemento { display: block | inline ; }

Podemos controlarlos con diferentes propiedades:

Display: FLEX
INLINE
INLINE-BLOCK
BLOCK
GRID

FLEX BOX: Le da al contenedor la capacidad de alterar las dimensiones y el orden de sus items para manejar mejor el espacio disponible. Es un sistema unidimensional que define un eje y el otro actúa en función a este.

Funciona con un sistema bidimensional a través de dos ejes, Eje Principal en x (definido por flex-direction), y el Eje Transversal en y (perpendicular al principal)

Flex Direction: ROW
COLUMN
ROW-REVERSE
COLUMN-R

Propiedades del flex:

Justify-content: define la alineación de los componentes a lo largo del eje principal
Flex-start
Flex-end
Center
Space-betwen
Space-evenly
Space-around

Flex-wrap: Especifica si los elementos flexibles debe ajustarse o no al contenedor
Nowrap
Wrap
Wrap-reverse

Align-items: Define la alineación a lo largo del eje perpendicular
Flex-start
Flex-end
Center
Strech
Baseline (Alinea a la primer línea de texto)

Position

La propiedad position sirve para posicionar un elemento dentro de la página. Muy útil cuando queremos posicionar elementos fuera del flujo normal de la página. Es fundamental interpretar el funcionamiento del posicionamiento para poder dar la ubicación exacta a cada elemento dentro del Box Model.

● static: valor por default. Mantiene el elemento en el flujo normal.
● relative: permite usar propiedades como top, right, bottom y left para mover el elemento en la página.
● absolute: funciona con las mismas propiedades, pero rompen el flujo normal. Se corresponden con la posición de un ancestro (el primero que tiene position no static).
● fixed: funciona con las mismas propiedades, pero rompe el flujo normal. Al punto que establece una posición fija en la pantalla.
● sticky: el elemento es posicionado en base al scroll del usuario.

JAVA SCRIPT

Variable: Nombre simbólico que hace referencia a un valor u objeto. A lo largo de la ejecución del programa la variable puede cambiar de valor.

Función: Una función es un conjunto de líneas de código que realizan una tarea específica y puede retornar un valor. Las funciones pueden tomar parámetros que modifiquen su funcionamiento. Sirve principalmente para: dividir un problema grande en varios problemas chicos facilitar lectura del código

Constantes: Son el nombre que le damos a un valor, y nunca cambia con el tiempo. Se usan para aumentar la legibilidad del programa.

Eventos: Un evento es algo que ocurre en el sistema, originado por el usuario o otra parte del sistema y que se avisa al sistema. Ejemplos: El usuario hace click. Se terminó de cargar la página. Pasó un segundo desde que se terminó de procesar.

Tipado de Variables:
El tipado estático nos obliga a definir desde el principio el tipo de una variable. (c++, Java, c#, etc)

El tipado dinámico nos da la facilidad de no definir los tipos al declarar una variable. (PHP, JavaScript, Grooby, Phyton, etc)

Eventos: Se puede programar un evento, para ejecutar una función dentro de M milisegundos. Es algo que ocurre en un sistema, originado por el usuario para avisar al sistema. Las interfaces suelen programarse orientadas a eventos

ARREGLOS

Almacena varios elementos en una coleccion de datos. Es una variable que tiene varios posiciones.

MALA PRACTICA: Puedo usar el DOM para guardar informacion, pero no es recomendable
Siempre es mejor manejar variables en JS para el manejo de datos

.lenght; Permite saber la cantidad de elementos que tiene el arreglo
.push("..."); Permite agregar elementos al final
.pop(); Permite borrar el último elemento
... = []; Permite vaciar el arreglo

Objetos: Es la combinacion de Campos o Atributos (Almacenan datos, pueden ser de tipo primitivo y/o otro tipo de objetos), o Rutinas o Modelos (Llevan a cabo una determinada accion)

JSON

Es una forma de organizar variables y funciones. Encapsula datos y comportamientos. Formato ligero para el intercambio de datos.

MALA PRACTICA: Un arreglo no puede tener objetos diferente, ej: int, string, etc.

this: Encuentra un elemento involucrado en el evento

SPA(Simple page aplication)

Es un tipo de aplicacion que permite que nunca se recargue completamente el nav. Los recursos se cargan parcial y dinamicamente cuando lo requiera la pagina

AJAX (Asynchonous Java Script and XML)

Tecnica front end que se basa en las aplicaciones web modernas. Es una tecnologia (no es un lenguaje).
Es una tecnica que combina un set de tecnologias conocidas para hacer mejorar la experiencia de usuario en la pag web (mas amigable y rapida.)
Es una tecnica de carga asincrona, permite cambios dinamicos del lado del cliente, mejora la experiencia del usuario.

Ventajas: Mayor UX

Mayor Velocidad
El user no pierde tiempo
Guarda datos en el server

Desventajas: Puede fallar
Agrega complejidad
Puede confundir al usuario
Depende de la conexión
Necesita aprendizaje nuevo

Estilos

Partian Render: Carga un fragmento del HTTML de otro servidor y lo inserta en nuestro html. Permite asociar una url a lo que estamos viendo
Incorpora la interfaz fetch() para llamados ajax. El valor de retorno de la promesa de Fetch es un objeto Response con información del request realizado
REST: Consulta un objeto JSON y procesa del lado del cliente con Java Script

El procesamiento de la respuesta es una promesa
Promises ASYNC/AWAIT: Objeto que representa la terminacion o el fracaso de una operacion asincronica. Tiene 4 estados:

Cumplida(fulfilled)
Rechazada(rejected)
Pendiente(pending)
Finalizada(settled)

Async: Hace que una funcion devuelva una promesa. El return se encapsulara en la promesa automaticamente
Await: Desencapsula el contenido de una promesa. Se reescribe como el then de la promesa. Solo puede usarse dentro de la funcion async

API REST

Una API es una interfaz que nos da una aplicación para comunicarnos con ell

Interfaz que nos da una aplicación para comunicarnos con ella. La mayoría de las APIs REST usan JSON para comunicarse.
Tipo de arquitectura de desarrollo web que se apolla totalmente en el estandar HTTP

Se basa en acciones (llamadas verbos) que amplian los datos:

POST: Crea recursos
GET: trae el recurso del html que llamamos
PUT: Actualiza los recursos
DELETE: Borra un recurso

Tanto GET como POST son muy conocidos por su uso en formularios. Son dos metodos de protocolo HTTP que podemos ver como forma de envio de datos a traves de internet.

ASICRONISMO

Nos permite devolver el control al programa antes de haber terminado una tarea mientras sigue operando en otro plano. **Son llamadas no bloqueantes.** Dificil de seguir, aumentan la escalabilidad(mismo tiempo mas operaciones)

Ejecucion secuencial. Retorna cuando la operacion fue completada en su totalidad

SINCRONISMO

Es la forma principal y basica en la que uno aprende a programar. **Cada operacion en Bloqueante** (tiene que terminar para que empiece la siguiente)
Siempre respeta el orden, Facil de seguir.

La finalizacion de la operacion es notificada por el programa principl. El procesado de la respuesta se hace en algun momento futuro.

Describa para qué se usa JS. ¿ En qué orden se ejecuta un script de JS y cómo se vincula con HTML y CSS? (PREGUNTA DE PARCIAL)

Java Script se utiliza para darle un comportamiento y funcionalidades al html.
En el html se llama por medio de la etiqueta <script> (incorporandolo al final del código, dentro del body) en el cual con el src = "" llamó a la carpeta "js/..." y nombró al archivo main.js.
Se ejecuta de arriba hacia abajo a medida que son llamadas, dependiendo la cantidad de script que tenga en el html.
Y al el CSS lo añado desde el Java Script con el classList.

(faltaria explicar los event listener respecto del orden y como lo alteran)
//corrección del prof

Con el Dom (document.querrySelector("#archivoHTML")) incorporamos el elemento html al js para poder darle comportamiento o manipularlo. Se pueden obtener elementos del DOM consultando por un ID, nombre, clase o un selector.
El innerHTML es un atributo especial con el cual podemos modificar el elemento según lo que queramos hacer.

¿Se recomienda el uso de variables globales en Javascript? (PREGUNTA DE PARCIAL)

No, siempre pueden evitarse o hacerse semi-globales encapsularlas en una función

----

Una mezcla donde cada tecnología aporta algo:
-Presentación en estándares HTML y CSS.
-Display e interacciones dinámicas via DOM.
-Intercambio de datos mediante XML (o JSON)
-Lectura de datos asincrónica mediante fetch
-Y JavaScript para unir todo.

CSS

Herencia: La herencia en CSS es el mecanismo mediante el cual determinadas propiedades de un elemento padre se transmiten a sus hijos. No todas las propiedades CSS son heredadas, porque algunas de ellas no tendría sentido que lo fueran.
Para que nos sirve entonces la herencia? Escribir menos código.

Combinación de Selectores: Se pueden combinar selecciones para hacerlas más específicas. Elige los párrafos que contengan la clase “destacado”. p.destacado{ color: red; }

Selectores Anidados: Permite seleccionar elementos contenidos dentro de otros elementos. Así se puede aumentar el nivel de detalle. Selecciona los span que estén dentro de algún párrafo (incluye si está contenido indirectamente).EJ: p span { color: blue; }

Grupo de Selectores: Se pueden usar varios selectores juntos. Esto permite evitar duplicación de estilos. Además se pueden refinar las diferencias aparte, Se separan los selectores con ‘,’ creando un grupo de selectores con propiedades en común.

Cascada: CSS significa cascading style sheets (hojas de estilo en cascada). La cascada es el mecanismo que controla el resultado final cuando se aplican varias declaraciones CSS contrapuestas al mismo elemento. Hay tres conceptos principales que controlan el orden en el que se aplican las declaraciones de CSS:

- **Importancia.**
- **Especificidad.**
- **Orden en el código fuente**

El orden en que se aplica la cascada es:

1. Hoja de estilos default del navegador.
2. Declaraciones normales en hojas de estilo de usuario.
3. Declaraciones normales en hojas de estilo.
4. Declaraciones importantes en hojas de estilo.
5. Declaraciones importantes en hojas de estilo de usuario.

Pseudo Clases y PseudoElementos

Las pseudo-clases se utilizan para definir un estado o comportamiento especial de un elemento. Por ejemplo: se puede asignar un estilo cuando se pasa con el mouse encima de un texto
Sintaxis: selector:pseudo-clase { propiedad:valor; }.
Un pseudo-elemento de CSS es usado para dar un estilo a una parte de un elemento. Pueden usarse combinados.

El pseudo selector (pseudo elemento) cambia el “que elegimos” (ej: sólo el primer elemento)
Las pseudo clases cambia el “cuando” (ej: solo cuando paso el mouse).

Semántica Web

La Semántica se refiere a todo aquello que está vinculado o pertenece a la significación de las palabras. La misma está asociada al significado, interpretación y sentido de las palabras. En el contexto de la Web, la semántica, es la práctica de darle al contenido de una pagina Sentido y Estructura.

<header></header> = incluye el texto que se va a identificar como cabecera de página, artículo o sección

<footer></footer> = es usado para la parte más baja de la pagina, y funciona igual que el header

<nav></nav> = Se utiliza para identificar la seccion donde estan los links de navegación de la página

<section></section> = Se utiliza para presentar o agrupar una sección de un documento o aplicación y

<article></article> = Se utiliza para incluir contenido que independientemente puede ser distribuido y reutilizado

<time></time> = indica la fecha de publicación, esta hace referencia al article que lo contiene

<aside></aside> = Define al contenido relacionado al documento o sección que se encuentra alrededor

<figure></figure> = crea un contenedor para imágenes, videos, audios

<figcaption></figcaption> = crea un contenedor para imágenes, videos, audios

Diseño Responsive

Es una técnica de diseño web que permite ver al programa de igual manera en diferentes dispositivos. Implica cambios de dimensiones y la distribución dinámica de los elementos mejorando así la experiencia del usuario, y permitiendo un menor costo de desarrollo y mantenimiento.

- Media Queries

Es un recurso de Css que permite asignar diferentes estilos para diferentes tamaños y resoluciones de pantalla. Nos da la posibilidad de entregar distintas apariencias.
Para implementarlo se necesitan puntos de quiebre para indicar que ciertas partes del diseño se comportan diferente bajo ciertas condiciones.

MIN-WIDHT(establece un tamaño en el cual queremos hacer cambios)
@media only screen and (min-widht: 600 px){...}

MAX-WIDHT
@media only screen (max-widht 600 px){...}

GRID: Tiene 2 dimensiones (modelo bidimensional). Permite dividir un contenedor en varias secciones permitiendo posicionar y alinear items en columnas y filas.

columnas: grid-template-columns
filas: grid-template-row

Precisa el alto y el ancho de las columnas y filas, con un diseño de cuadricula. Los valores son una lista separada por espacios, donde cada valor especifica el tamaño de la columna respectiva.

grid-template-areas: Especifica nombres para cada una de las secciones del grid
grid-area: asocia el nombre del area a un item y específica en que area se posiciona el item.