Clase 1:

Que es desarrollo web?

Todo aquello que puedes ver desde un navegador web tiene algo que ver con Desarrollo Web. Existen muchas divisiones o especialidades: desde Front-end hasta Backend, sin embargo, primeros nos concentraremos en aprender la base de cualquier sitio web.  
  
Que es HTML?  
HTML es la estructura de tu sitio. Si fuera un auto, sería describir sus elementos: “Mi auto tiene 4 ruedas, un volante, 4 puertas, 2 ventanas”. En los sitios web, esto puede traducirse a títulos, texto, imágenes, footers, headers, etc. Por ser un lenguaje de marcado, HTML cuenta con etiquetas que nos ayudan a estructurar nuestro sitio.   
  
Que es CSS?  
¿Cómo volvemos nuestro sitio más personalizable y único?, esto es gracias a CSS. “Mi auto tiene 4 llantas grandes, 4 puertas rojas, 2 ventanas polarizadas y un volante de piel”, quiere decir, describimos las características de los elementos.   
  
Con CSS podemos indicar: color, tamaño de fuente, tipo de letra, posición, de cualquier elemento dentro de nuestro HTML, por lo cual, siempre van a estar interactuando entre sí.

Clase 2

Es una buena practica hacer el mockup de la pagina web primero en Photoshop, AI o cualquier programa de edición e identificar las distintas partes del sitio.

Luego hay que armar la estructura de carpetas

Carpeta del Proyecto

Images (agrupa todas las imágenes)

Css (agrupa todas las hojas de estilos)

Idex.html (es el ejecutable del sitio web)

Etiquetas

Header <header>

Dentro del Header va la etiqueta <meta charset=”utf-8”>

Atributos de Header:

Id: no se puede repetir. Es la referencia dentro del código. Para invocarlo se utiliza <href=”#nombre del id”>

Section <section>: sirve para dividir sectores del sitio web

Para poner algo como comentario es con <!– comentario 🡪

<nav> se utiliza como etiqueta para los menúes.

Los menues tienen 2 tipos de etiquetas, Uno para listas ordenadas (<ol>) y otro para listas desordenadas (<ul>). Para que te tome los elementos que quieras meter en la lista tenes que incluirlos dentro de la etiqueta <li> (list ítem)

Listas ordenadas: elementos de la lista enumerados

Listas desordenadas: los elementos de la lista están organizados con bullets, por viñetas.

<a> = anchor o ancla, sirve para volver una palabra link a otra pagina. Si queremos que al hacer click en el link la pagina se abra en otra pestaña, tenemos que poner el atributo:

Target=”\_blank”

<figure> sirve para poner imágenes.

<img src=””> es la etiqueta para ubicar la imagen que se va a subir. IMG se cierra en si misma, sin </>

Las imágenes se suben a través de rutas. Las rutas pueden ser absolutas o relativas.   
rutas absolutas: desde un sitio web ([www.sitioweb.com/imagenes/nombredelaimagen](http://www.sitioweb.com/imagenes/nombredelaimagen))

Rutas relativas: parte desde donde esta index.html

Atributos de la etiqueta IMG:

Src: es la dirección de donde se va a cargar la imagen

Width: es el ancho (va antes que el alto)

Height: es el alto

Cuando no se sabe la relación ancho-alto, se pone solo el ancho y el alto se calcula automáticamente.

Alt: es el texto que va a cargar en caso que la imagen no pueda ser cargada. Se tiene que poner un texto corto y descriptivo de la imagen para saber cual es la que esta teniendo un error.

Los tamaños de las imágenes los va a dar el mockup que se tiene que hacer antes de empezar a codear la pagina.

<h1> es heading 1, tiene particularidades de que vuelve mas importante el texto como info de la pagina. Se usa 1 vez por pagina nomas. Sirve para actividades de SEO

Existen 6 niveles de h, de h1 (mas importante) a h6 (menos importante)

<article> se utiliza para poner pedazos de información específicos dentro de la pagina, por ejemplo, entradas a un blog.

ESTILOS

Existen 3 tipos de estilos.

* Estilos en línea:
  + Se hacen a través del atributo style=” ”
  + Es fácil de hacerlo cuando hay que poner un detalle
  + Si es un sitio web complejo se vuelve ilegible
* Estilos dentro de una etiquetas
  + La etiqueta es <style>
  + Se coloca dentro de la etiqueta head
  + La sintaxis es como la de una función en javascript, las variables se encierran entre {}
  + Como con los estilos en línea, se puede volver ilegible si son muchas las variables
  + Lo bueno es que no necesita de otro archivo extra para poner los estilos
* Estilos externos
  + Agregar una hoja de estilos externa se hace a través de la etiqueta “Link”
    - <link> para que funcione debe recibir 2 valores si o si.
      * Una relación, en este caso: rel=“stylesheet”
      * Una referencia, o sea: href=”ubicación del archivo”
      * En el archivo.css se ponen todos los atributos como si fuera un estilo dentro de etiqueta.

CSS

Primer concepto a aprender es el de SELECTORES.

Los selectores sirven para aplicar estilos a un objeto dentro de la web especifico y que no se aplique a todo el resto.

La sintaxis de los selectores son:

Tipo de selector {

propiedad: valor;

propiedad2: valor;

}

1er selector: Selector universa.

* + Se llama con el \* (o sea, en tipo de selector hay que poner un \*)
  + Le da estilos a TODO lo que hay en la pagina

2do selector: Selector de etiqueta.

* + Hace referencia a las etiquetas creadas (como header, figure, section, etc)
  + Se invoca escribiendo la etiqueta a la que se le quiere dar estilo

3er selector: selector descendente.

* + Es la base de CSS. Hace que la web funcione como una cascada, donde hay elementos dentro de otros elementos
  + Se escriben los elementos de afuera hacia adentro, por ejemplo
    - Article h3 {

}

* + - Article li {}

4to selector: selector de clase.

* + Necesitan un atributo dentro de la etiqueta de HTML denominada class=””
  + Se invocan usando un .

5to selector: Selector de ID

* + Al igual que el selector de clase, se debe poner el atributo id=”nombre” en el HTML
  + se invoca utilizando el #

Tipo de colores

Por ejemplo:

.portada {

Background: color

}

Background puede ser:

Normal: Background: nombre del color

Hexadecimal: background: #000000

RGB: background: rgb(12,6,53) (Esto setea en variables la cantidad de rojo, verde y azul que va a tener la imagen, permite crear otros tonos de colores que no sean los normales puros)

RGBA: background: rgba(12,6,64,.5) (esto agrega el canal “alfa” que se traducirá como la transparencia)

Para agregar una imagen al background a través de CSS es necesario hacerlo siguiente:

.portada {

Background-image: url(‘ruta de la imagen’);

}

Tipos de fuente

Para utilizar un tipo de fuente que los usuarios puede que no tengan, es recomendable usar googleFonts ([www.google.com/font](http://www.google.com/font) muy buena pagina, explica como poner los fonts)  
  
El cargarlo se llama CDN

Se carga a través de <link href=”dirección” debajo de meta charset y se integra al CSS con la variable:

Font-family: “Montserrat”, sans-serif;

Beneficios:

No es necesario descargarla, ni los usuarios tampoco

El CDN de google tiene la propiedad de poder cargar 1 o mas tipos de fuentes en una sola hoja de estilos y no tener que poner muchas líneas de código.

El código original para cargar una fuente es:

<link href='https://fonts.googleapis.com/css?family=Montserrat:400,700' rel='stylesheet' type='text/css'>

Si se le quiere cargar 2 o mas fuentes al mismo tiempo lo que hay que hacer:

<link href='https://fonts.googleapis.com/css?family=Montserrat:400,700' | Allerta rel='stylesheet' type='text/css'>

Disposicion en el entorno = DISPLAY

Hay distintos tipos de disposiciones:

En línea

En bloque

Mixta

Modelo de caja.

Las etiquetas de HTML vienen en formato de caja

Primero esta el contenido > padding > border > margen

El modelo de caja también asigna el tamaño de cada uno de los elementos.

“padding” es el espacio entre el contenido y el borde

El borde tiene 3 parametros, tamaño, forma (si es solid o transparente) y el color.

Border: 2px solid;   
(dashed son líneas punteadas)

Padding puede tener los 4 espacios en la misma línea:

El orden es el mismo que las agujas del reloj “arriba, derecha, abajo, izquierda”

Ejemplo: Padding: 10px, 15px, 5px, 20px;

Line hight es el espaciado entre líneas

Margin: auto; es una forma para centrar la pagina.

Clases

Un elemento HTML puede tener cualquier cantidad de clases

<div> sirve para crear otra caja en el html, se usa para separar contenido o para aplicar estilos determinados a una sección especifica

Float sirve para mover el contenido. Se aplica en CSS  
  
por ejemplo:

.guitarras img {

Float: right;

}  
En ese caso las imágenes se van a correr para el lado derecho.

Overflow: hidden;

Positions.

Existen varios tipos de positions, los mas usados son relative, absolute o fixed.

Luego de poner la posición relative se puede poner   
left: xx px;

Right: xx px;

Top: xx px;

Bottom: xx px;  
  
con esto se modifica la disposicion del elemento.

Con position fixed hace que el elemento continue bajando aunque el usuario scrollee hacia abajo o arriba.

Footer es el pie de pagina de la web… se puede poner con la etiqueta <footer>

Formularios

<form class=”formulario”> se cierra en si mismo.

Las subetiquetas mas comunes de un formulario son:

<label for=””></label>

<imput type=””>

Dentro de la etiqueta se pueden poner valores como required para volver el campo obligatiorio, o value para que tenga un texto por defecto.

Placeholder: sirve para que aparezca de fondo para que sirve la caja.

<label for=”nombre”>

<imput type=”text” id=”nombre” va a hacer que cuando se haga click en el label “nombre” el navegador haga focus en la caja de texto de nombre.

Name=”sexo” tanto en hombre como en mujer hace que el browser interprete que tiene que elegir 1 de los dos.

<textarea name=”comentarios” id=”comentarios” cols=”30” (cuantas columnas) rows=”10” (cuantas filas”) > </textarea>

<imput type=”submit” value=”enviar/> (submit pone un botón que envía el formulario al servidor)

Siempre poner value a todas las etiquetas del formuilario

Display flex

En CSS se puede utilizar una variable que se llama display: flex;   
luego justify-content: flex-start; para que los elementos se justifiquen en la pantalla hacia la izquierda. Es el valor por defecto cuando se pone display flex. La otra opción es flex-end para que los elementos se alinien a la derecha. O justify-content: center para que se alinien al centro.

Justify-content: space-around; distribuye los elementos dentro de su contenedor

Justify-content: space-between; es similar al de arriba

Flex-wrap: nowrap; hace que los elementos se redimensionen pero no se muevan de lugar cuando se achica la pantalla

Flex-wrap: wrap; hace que los elementos se vayan distribuyendo hacia abajo cuando se achica la pantalla

Flex-direction: row (por defecto, pone los elementos en horizontal) o column (para que vaya en vertical)

Flex le da propiedades SOLO a sus hijos DIRECTOS.

Align-items: center; es para que los subelementos internos de un elemento contenedor se alineen en el centro del mismo

Justify-content: center; también alinea los ítems dentro del elemento

TABLAS

Se hace en HTML a través de la etiqueta <table>

Subetiquetas de table:

* <thead> </thead>
  + <tr> </tr> (table row, crea una fila dentro de la tabla)
* <tbody> </tbody>