

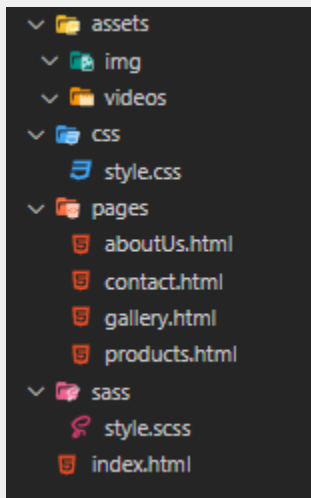
# Conceptos básicos de Diseño Web

## Nomenclatura

Al manejar código, siempre que no sea obligatorio (excluimos el texto visual como el texto que ve el usuario), usen:

- camelCase
- kebab-case
- snake\_case
- BEM

Eviten siempre caracteres especiales como tildes e inicien todo en minúscula ya que en programación las mayúsculas se reservan para casos especiales que no van a aplicar a este curso.



## Arquitectura del sitio

Siempre tengan el archivo index.html en la carpeta raíz al iniciar su proyecto, hay otras convenciones como las que se usa en angular o react pero como están programadas para ejecutarse en una página única (tienen un único html que es index.html) su estructura es distinta.

Para mantener la organización, el uso de carpetas es muy útil, yo recomiendo una carpeta para:

1. El resto de los HTML
2. Fotos y videos, este se puede subdividir por tipo de archivo o por página o uso, lo importante es que una vez que elijan un orden, se apeguen a él.
3. CSS, cómo ustedes van a usar un css único, pueden no hacer esta carpeta.
4. SASS, lo van a usar a futuro

## Anidación

En programación se dice que 2 elementos están anidados cuando uno se encuentra dentro de otro, se llama padre al elemento que contiene al hijo.

## Herencia

El elemento hijo puede heredar varias propiedades del padre, se ve caso a caso, pero el padre no hereda cosas del elemento hijo.

```

<main>
  <section>
    <h2>Noticias</h2>
    <article>
      <h3>Nieva en BS AS</h3>
      <h4>Suceso histórico</h4>
      
      <div>
        <p>Lorem ipsum dolor sit amet consectetur a
        <p>Lorem ipsum dolor sit amet consectetur a
      </div>
    </article>
  </section>
</main>

```

## Indentación

La imagen usa la extensión indent-rainbow de VS Code

Se llama indentación al espacio que se inserta antes del código. Al iniciar su proyecto decidan un número de espacios que equivalga a 1 nivel de indentación

y apeguense a el, por defecto en VS Code es 4 (abajo a la der Spaces: 4) y lo insertan al tabular (tecla tab). El elemento hijo siempre tiene 1 nivel de indentación más que su padre.

Los controles para indentación son:

- **tab**: agrega 1 nivel de indentación
- **Ctrl+[ o Ctrl+]** : modifican el nivel de indentación de las líneas seleccionadas bajando ( **Ctrl+[** ) o aumentando (**Ctrl+]**) la indentación. Se pueden seleccionar líneas múltiples.
- **Alt+↑ o Alt+↓**: mueve las líneas de código seleccionadas respetando indentación (si seleccionan solo parte del elemento lo van a romper)

## Comentarios

Un programador siempre comenta su código donde vea necesario para facilitar la lectura de otro programador, dependiendo del lenguaje de programación, los caracteres para comentar son distintos. Los atajos de VS Code son **Ctrl+k+c** , **Ctrl+k+u** y **Ctrl+/**

# HTML

## Etiquetas

```

<etiqueta atributo="valor">
  contenido
</etiqueta>

<h1>Rompi etiquetas</h1>
<etiqueta1 atributo="valor">
  <etiqueta2>
  </etiqueta1>
</etiqueta2>

```

HTML es un conjunto de etiquetas anidadas todas dentro de la etiqueta html. A estas etiquetas se les pueden agregar atributos y estos atributos pueden tener valores. Se va a ver con mayor claridad más adelante pero se puede pensar que cada etiqueta es una caja cerrada.

Las etiquetas poseen apertura <etiqueta> y cierre </etiqueta> o son únicas <etiqueta/>. La apertura de una etiqueta siempre tiene que tener el mismo nivel de indentación que su cierre, de lo contrario esta mal indentado.

Lo importante es no romper etiquetas, ya dijimos que son cajas cerradas, si un elemento que se abre adentro se cierra afuera, se te tuvo que romper la caja en algún lado para que salga.

Listado de etiquetas:

- [Mozilla](#)(Español)

- [W3Schools](#)(Ingles)

## Etiquetas únicas

- **DOCTYPE:** le dice al lector qué tipo de documento está viendo.
- **html:** es la etiqueta raíz de la página. Tiene un atributo lang cuyo valor indica el lenguaje del sitio, este valor es “en” para inglés y “es” para español.
- **head:** contiene la metadata de la página.
- **body:** contiene todas las etiquetas que se van a ver en la página.

## Etiquetas metadata:

```
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <link rel="stylesheet" type="text/css" href="./css/style.css">
9 </head>
```

- **meta:** dependiendo del atributo hay varias, en la imagen que se ve de arriba a abajo:
  - .1. Señala compatibilidad de caracteres, esto le permite mostrar tildes o ñ, hay distintos para caracteres japoneses, chinos, etc.
  - .2. Compatibilidad con internet explorer, por mucho tiempo internet explorer no se adhirió a las prácticas de w3schools y no soportaba muchas cosas que los otros navegadores si
  - .3. Permite a la página ver el ancho de pantalla
- **title:** pones el nombre que vas a ver en pantalla
- **link:** une el contenido de otros archivos al html

## Etiquetas semánticas

```
<body>
  <header>
    <a href="index.html">
      <img src="" alt="logo">
    </a>
    <h1>Nombre de la página</h1>
    <nav>
      <ul>
        <li>
          <a href="index.html">Home</a>
        </li>
        <li>
          <a href="pages/aboutUs.html">About US</a>
        </li>
        <li>
          <a href="pages/contact.html">Contact</a>
        </li>
        <li>
          <a href="pages/gallery.html">Gallery</a>
        </li>
        <li>
          <a href="pages/products.html">Products</a>
        </li>
      </ul>
    </nav>
  </header>
  <main>
    <!-- Contenido, puede tener section y article donde vean necesario -->
  </main>
  <footer>
    <div>
      <a href="" target="_blank"><img src="" alt="linkedin"></a>
      <a href="" target="_blank"><img src="" alt="github"></a>
    </div>
    <span>©Copyright</span>
  </footer>
</body>
```

Son etiquetas contenedoras que por convención se las especializó a partir de clases más utilizadas en divs para otorgar estructura y facilidad de lectura de código. Las más comunes:

- **header:** la etiqueta header es un contenedor que como dice el nombre, contiene el encabezado de la página. Lo normal es que contenga un logo y la barra de navegación. En este curso para conseguir puntuaciones óptimas en entregas intermedias se pide que usen un logo con un link al home (osea su index) en el encabezado de todas las páginas.
- **nav:** la etiqueta nav se utiliza para contener una agrupación de a (anchor/ancla, la que usan para links), es muy común usarla en el header con una lista desordenada con los a dentro de los li. Hay muchas librerías (compilaciones de código para el uso de otro) que dan efectos copados cuando el nav contiene un ul con li con los a adentro. A veces se usa también en el footer e incluso en otras secciones si hay muchos juntos.
- **footer:** el footer contiene como dice el nombre el pie de la página, siempre es el último elemento y suele contener links de multimedia (como links a facebook/linkedin/instagram), copyright y algunos ponen ahí el formulario de contacto.
- **main:** esta etiqueta suele contener el contenido principal de la página, algunos lo usan para lo más importante de la página, otros lo usan para contener todo lo que está entre el header y el footer.
- **aside:** o aparte, es un contenedor que se utiliza para todo lo que está separado del contenido principal de la página, como ads al costado.
- **section:** o sección, se utiliza para separar secciones de código temáticamente, en su caso un ejemplo es poner una sección llamada contacto y meter dentro el formulario de contacto o, un ejemplo que voy a expandir más abajo, una sección con comentarios de otras personas. Personalmente

considero al header y al footer una **section** especializada, así que no lo usaría dentro de él.

- **article:** o artículo, suele ser intercambiable con el **section** en función pero personalmente me hace más sentido una sección con varios artículos. Un ejemplo es un **section** que contiene una compilación de comentarios y cada artículo es un contenedor con una imagen, nombre y texto del comentario.

## Etiquetas con mucho uso

Busquenlas por su cuenta sus atributos y usos, pueden usar el link de etiquetas que proporcione antes.

- **div**
- **p**
- **h#**
- **a**
- **ul**
- **ol**
- **li**
- **form**
- **input**
- **label**
- **select**
- **option**
- **fieldset**
- **textarea**
- **img**
- **video**
- **iframe**
- **button**

## Comentarios

```
<body>
<!-- Inicio del header -->
<header>
  <a href="index.html">
```

Es normal en comentarios señalar cuando inician/terminan secciones importantes de código, nombre de la imagen en una galería grande y cualquier otra cosa que les facilite la lectura del código. No se comenta código que no se usa más, eso se borra directamente. Un comentario se abre con `<!--` y se cierra con `-->`.

## Estilar mediante atributos de CSS

Es posible hacerlo pero se considera mal en este curso y molesta mucho al momento de hacer un sitio responsivo.

# CSS

```
4 selector {
5   propiedad: valor;
6   propiedad: valor;
7 }
```

Es un lenguaje de estilo cascada. En este se llama un selector en el cual se le atribuyen varias propiedades de estilo con valores distintos.

## Linkear el CSS al HTML

```
<link rel="stylesheet" type="text/css" href="./css/style.css">
```

Esto se hace mediante una etiqueta link que vamos a colocar dentro del header, es imperativo que tenga la propiedad rel, que indica la relación al html y href, la ubicación relativa del CSS con respecto al HTML. Personalmente encuentro que agregar el atributo type con el valor "text/css" ayuda mucho con temas de compatibilidad.

```
<link rel="stylesheet" type="text/css" href="./css/style.css">
```

## Selectores

Más adelante junto animaciones se verán pseudoselectores en la clase de animación

Un selector llama al elemento al que le vamos agregar estilo. Hay 3 tipos:

- **Etiqueta:** `p{color: red;}` no lleva nada por delante y le da estilo a todas las etiquetas llamadas por igual, es la de menor importancia.
- **Clase:** `.descripcion{color: blue;}` lleva un "." por delante y da estilo a todas las etiquetas con esa clase. Para agregar una clase a una etiqueta html se lo hace mediante el atributo class `<p class="descripcion">`, una clase es una única palabra (podes poner cosas como descripcion-1 o imgLogo) y se pueden agregar varias clases separándolas con espacios `<p class="descripcion centrado">`. Si bien una clase descriptiva es buena, lo mejor es no hacerlas muy largas. Es de importancia mediana
- **ID:** `#desc1{color: green;}` lleva un "#" por delante y llama a la etiqueta con el id seleccionado (como sabemos el id es único siempre). Es de gran importancia.

Es posible llamar a varios selectores simultáneamente separándolos con ",":

```
p, .logo, span, #desc{
  color: cyan;
}
```

También se puede anidar los selectores lo que aumenta levemente su importancia:

```
div p{
  color: blueviolet;
}
```

(se pueden usar clases o id también al anidar)

## Efecto cascada

El efecto cascada de css se da cuando 2 selectores llaman a la misma propiedad de css de una etiqueta. Ahí css toma en cuenta el orden y la importancia. Si uno es más importante (estilo en atributo html>id>clase>etiqueta), el de mayor importancia es el que modifica al html. Por otro lado, si ambos son de igual importancia, el último (el de la última línea) es el que se toma.

Se pueden importar varios css a un mismo html y sería lo mismo que pegarle al primer css los otros al final, por lo que si a una etiqueta se lo llama con el mismo nivel de importancia en todos los css, por efecto cascada se toma el del último css linkeado ya que queda al final y “pisa” al resto

```
p{color: red;}
.descripcion{color: blue;}
/* resulta en azul */

p{color: red;}
.descripcion{color: blue;}
#desc1{color: green;}
/* resulta en verde */

p{color: red;}
div p{
  color: blueviolet;
}
/* resulta en violeta */
```

```
.descripcion{color: blue;}
div p{
  color: blueviolet;
}
/* resulta en azul */

.descripcion{color: blue;}
main section article div p{
  color: blueviolet;
}
/* resulta en azul */

#desc1{color: green;}
main section article div .descripcion{
  color: yellow;
}
/* resulta en verde */
```

## CSS innato de las etiquetas html

Esta en ingles pero la tabla que es lo único que les importa no varía con el lenguaje: [https://www.w3schools.com/cssref/css\\_default\\_values.asp](https://www.w3schools.com/cssref/css_default_values.asp)

## Lista de propiedades de css

- Español: <http://w3.unpocodetodo.info/css3/chuletacss3.php>
- Ingles: <https://www.w3schools.com/cssref/>
- El que más recomiendo y amo (ingles): <https://css-tricks.com/almanac/properties/>

## Propiedades que vieron por ahora que le van a dar más uso

- **color**
- **border**
- **border-radius**
- **background**
- **text-shadow**
- **box-shadow**
- **list-style**
- **font-type** (y google fonts)
- **font-style**
- **font-size**
- **overflow**

## Unidades

Las unidades relativas son más fáciles de hacer responsivas pero hay muchos casos en los que uno prefiere usar unidades fijas.

### Absolutas:

- **px** (pixeles/el más usado)
- **cm** (centímetros)
- **mm** (milímetros)
- **in** (pulgadas)
- **pt** (puntos)
- **pc** (picas)

### Relativas:

- **%** (porcentaje en relación al elemento padre/muy usado/si lo usan para fuentes hay tabla)
- **em** (relativo al tamaño de fuente del elemento/muy usado)
- **rem** (relativo al tamaño de fuente del elemento raíz/muy usado)
- **ex** (relativo a la altura de la letra)
- **ch** (relativo al ancho del caracter "0")



- **vw** (relativo al 1% del ancho de la pantalla/muy usado/si lo usan para fuentes hay tabla)
- **vh** (lo mismo para el alto)
- **vmin** (relativo a 1% de la dimensión más chica de pantalla)
- **vmax** (lo mismo para la más grande)