

**TALENTO
DIGITAL**
INTELIGENCIA
HUMANA

Talento Digital para Chile:

Módulo 3 Fundamentos de Desarrollo Web

UN PROYECTO DE:

DESARROLLADO POR:



MÓDULO 3 – FUNDAMENTOS DE DESARROLLO WEB

3.1.- Contenido 1: Construcción de páginas Web

Objetivo de la jornada

- Conocer y utilizar herramientas que permitan crear sitios web.
 - Construir un documento HTML utilizando las principales etiquetas y atributos para resolver un problema planteado.
 - Detectar inconsistencias de los elementos de una página web usando las herramientas presentes en un navegador de uso masivo.
-

3.1.3.- Manejando Hojas de Estilo

3.1.3.1.- ¿Qué es una hoja de estilo y para qué sirve?

En tópicos anteriores se ha visto cómo crear un documento HTML, organizando su estructura, determinando qué elementos son más apropiados para representar su contenido; esto, por cierto, permite definir el documento, pero no determina de qué forma se mostrará en pantalla. Desde la introducción de HTML5 esa tarea es responsabilidad de CSS, que es un lenguaje que facilita instrucciones que se usan para asignar estilos a los elementos web, como colores, tipos de letra, tamaños, etc. Los estilos se definen con CSS y luego se asocian a los elementos hasta que se obtiene el diseño visual que se desea aplicar en una página.

Por motivos de compatibilidad, los navegadores web aplican estilos por defecto a algunos elementos HTML. Esta es la razón por la que algunos elementos en un sitio web puedan tener márgenes inesperados o saltos de línea, pero otros estando definidos de forma parecida muestran comportamientos dispares. Algunos de estos estilos son útiles, pero la mayoría deben ser reemplazados o complementados con estilos personalizados. En las hojas de estilo los diseños personalizados se declaran con propiedades; un estilo se define declarando el nombre de la propiedad y su valor separados por dos puntos.

`font-size: 24px;`

Ilustración 1: Combinación atributo-valor en CSS

En el ejemplo anterior se declara una propiedad que cambia el tamaño de la letra

a 24 píxeles, toda instrucción termina siempre con punto y coma “;”. Si esta propiedad se aplica a un elemento del sitio web, el texto contenido en él se mostrará en la pantalla con el tipo de letra definido por defecto, pero con un tamaño de 24 píxeles.

Respecto de los píxeles (px), aunque éstos son las unidades de medida que más se implementan, en algunas circunstancias, especialmente cuando se crean sitios web con diseño web adaptable a su contenedor, pueden ser más apropiadas otras unidades. En términos generales, las unidades más utilizadas son **px** (píxeles), **pt** (puntos), **in** (pulgadas), **cm** (centímetros), **mm** (milímetros), **em** (relativo al tamaño de letra del elemento), **rem** (relativo al tamaño de letra del documento), y **%** (relativo al tamaño del contenedor del elemento).

La mayoría de las propiedades se aplican a un solo aspecto del elemento. Si queremos cambiar varios conceptos al mismo tiempo, tenemos que declarar múltiples propiedades; el lenguaje CSS define una sintaxis que hace más sencillo el proceso de asignar múltiples propiedades a un elemento, característica que se conoce como “regla”. Una regla es una lista de propiedades declaradas entre llaves e identificadas por un selector; este último indica qué elementos se verán afectados por las propiedades. La siguiente regla se aplica a la etiqueta **p** y, por tanto, las propiedades que ahí se especifican se aplicarán a todos los elementos **<p>** del documento; esta regla incluye dos propiedades con sus valores agrupados por llaves. Si se aplica esta regla a un documento, el texto dentro de cada elemento **<p>** se mostrará en color rojo y con un tamaño de 24 píxeles.

```
p {
  color: #FF0000;
  font-size: 24px;
}
```

Ilustración 2: Estilo CSS aplicado sobre una etiqueta

3.1.3.2.- Sintaxis básica de una hoja de estilo

Las reglas y las propiedades definen los estilos que se asignarán a uno o más elementos, pero estos estilos no se aplican hasta que se agregan a un documento. Para ello existen tres alternativas:

- a) **estilos en línea:** usa atributo **style** para agregar estilos al elemento; toda etiqueta HTML cuenta con este atributo, y todos los estilos que ahí se apliquen se reflejarán posteriormente. ¿El problema? si se desea cambiar una característica gráfica del sitio, se debe hacer la modificación en todas las etiquetas, lo que incrementa el tamaño del archivo y complica su mantención posterior.

```

1  <!DOCTYPE html>
2  <html lang="es">
3      <head>
4          <title>Título del Documento</title>
5          <meta charset="utf-8">
6      </head>
7      <body>
8          <main>
9              <section>
10                 <p style="font-size: 20px;">
11                     Estilo aplicado directamente sobre un párrafo
12                 </p>
13             </section>
14         </main>
15     </body>
16 </html>
17

```

Ilustración 3: Estilo declarado en línea

- b) **estilos incrustados:** corresponde a agregar las reglas CSS dentro de la cabecera del documento usando atributos y valores que apliquen sobre las etiquetas a las que se desee aplicar un formato específico. Esto se debe hacer dentro de la etiqueta **<style>**. Lo que se declare dentro de la etiqueta es similar al caso del ejemplo que se indica antes; además, los estilos indicados aplican para todas las etiquetas correspondiente, no siendo necesario actualizarlas una por una. El inconveniente es que las mismas declaraciones se deberán repetir en cada documento si se desea mantener una gráfica uniforme.

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <title>Título del documento</title>
5      <meta charset="utf-8">
6      <style>
7          p{
8              font-size: 20px;
9          }
10     </style>
11 </head>
12 <body>
13 <main>
14     <section>
15         <p>El estilo se aplica en todos los párrafos</p>
16     </section>
17 </main>
18 </body>
19 </html>
20

```

Ilustración 4: Estilo incrustado en un documento

- c) **hojas de estilo:** esta acción corresponde a desplazar todos los estilos a un archivo con extensión CSS independiente y luego usar el elemento `<link>` para referenciarlo desde los documentos en los que estos estilos apliquen. Estos archivos se conocen como “hojas de estilo”, y solo contienen reglas provenientes del lenguaje aludido. Por cada elemento que se referencia, se pueden considerar parámetros dentro de la etiqueta `<link>`; en el caso de las hojas de estilo solo se usan **rel** y **href**, que corresponden al valor “**stylesheet**” y la ruta del archivo CSS que contiene los archivos, respectivamente.

```

1  <!DOCTYPE html>
2  <html lang="es">
3    <head>
4      <title>Título del documento</title>
5      <meta charset="utf-8">
6      <link rel="stylesheet" href="hojadeestilos.css">
7    </head>
8    <body>
9      <main>
10       <section>
11         <p>
12           El estilo de este párrafo está declarado
13           en un documento independiente.
14         </p>
15       </section>
16     </main>
17   </body>
18 </html>
19

```

Ilustración 5: Estilo declarado en un documento independiente

```

p {
  font-size: 20px;
}

```

Ilustración 6: Contenido del archivo hojadeestilos.css

Al igual que otros lenguajes, CSS cuenta con un conjunto de “buenas prácticas” que se deben considerar al momento de asignar estilos a un sitio. Dentro de ellas se puede destacar las siguientes:

- **Limpiar el documento CSS:** en la red existen varias aplicaciones que permiten optimizar el tamaño de un documento CSS eliminando los caracteres que no son aporte de nuevos estilos. Ejemplo de ello son aplicaciones tales como Proccsor, Clean CSS y Unused CSS.

- **Combinar hojas de estilo:** cada hoja de estilos declarada como archivo independiente es una petición http. Independiente del tamaño de cada documento referenciado, cada llamado es tiempo utilizado. Por tanto, combinar todos los CSS en uno solo es una práctica que puede beneficiar bastante en lo que respecta al desempeño del sitio.
- **Agregar una tabla de contenido:** esto consiste en agregar en modalidad de comentario, en las primeras líneas de un documento CSS, las secciones que contendrá el archivo y la línea en la que comienzan, esto claro está será pertinente en documentos CSS con muchas líneas de código.
- **Escribir el CSS de manera estándar:** cada declaración de estilos se puede hacer de la forma estándar (cómo aparece en los ejemplos anteriores) o bien en una única línea; se recomienda usar la primera opción.
- **Usar un editor de texto:** se recomienda hacer las modificaciones del texto de un CSS a través de un editor de texto. En todo sistema operativo existen diversas alternativas.

3.1.3.3.- Manejo de activos e imágenes

Un sitio web puede referenciar, para su correcto funcionamiento, archivos externos creados en otros lenguajes. Estos archivos tienen funciones específicas, y la distribución potencial (cantidad de archivos, largo de los archivos, tipo, etc.) lo define el programador.

Para agregar un recurso externo a un sitio se puede usar la etiqueta **<link>**, aplicando una sintaxis similar al siguiente:

```
<link rel="value">
```

Ilustración 7: Estructura etiqueta <link>

Dependiendo del tipo de recurso que se quiere importar, el atributo **rel** va tomando diversos valores:

- **icon:** importa un ícono a un documento.
- **stylesheet:** importa una hoja de estilos.
- **search:** entrega un enlace a un recurso que puede ser usado para buscar dentro de la página actual y en sus páginas relacionadas
- **author:** entrega un enlace al autor del documento
- **help:** provee un enlace para un sitio de ayuda

En secciones anteriores se indicó que la etiqueta **** permite insertar imágenes en un sitio. Una imagen se puede referenciar a través de CSS, haciendo referencia al método **url()**. Por ejemplo, si se desea agregar una imagen de fondo a un sitio web a través de CSS, el código que se muestra a continuación puede ser de mucha ayuda.

```

1  body {
2      /* Referencia relativa a la imagen */
3      background-image: url(imagenes/fondo.jpeg);
4
5      /* Imagen de fondo centrada, tanto vertical como horizontalmente */
6      background-position: center center;
7
8      /* Posición de la imagen fija en la ventana */
9      background-attachment: fixed;
10
11     /* Sin repetición */
12     background-repeat: no-repeat;
13
14     /* Imagen de fondo se reescala automáticamente con el cambio del ancho
15     de ventana del navegador */
16     background-size: cover;
17
18     /* Se muestra un color de fondo mientras se está cargando la imagen
19     de fondo o si hay problemas para cargarla */
20     background-color: #cccccc;
21 }
22

```

Ilustración 8: Estilos CSS para imágenes de fondo

Una imagen puede ser referenciada de dos maneras:

- **Referencia absoluta:** incluyen todas las partes de la URL (protocolo, servidor y ruta) por lo que no se necesita más información para obtener el recurso enlazado.
- **Referencia relativa:** no consideran algunas partes de las URL para hacerlas más cortas. Ya que son URL incompletas, es necesario disponer de información adicional para obtener el recurso enlazado; para que una URL relativa sea útil es imprescindible conocer la URL del origen del enlace.

3.1.3.4.- Reglas en CSS

A veces es recomendable declarar los estilos en un archivo independiente y luego cargar ese archivo desde un documento que lo necesite, pero obliga al desarrollador a implementar diferentes mecanismos para establecer la relación entre las reglas CSS y los elementos dentro del documento que se verán afectados por éstas.

Existen varios métodos para seleccionar los elementos que serán afectados por una regla CSS. En ejemplos anteriores se ha utilizado el nombre del elemento, pero también se pueden usar los valores de los atributos **id** y **class** para referenciar un solo elemento o un grupo de elementos, e incluso combinarlos para construir selectores más específicos.

<pre>span { font-size: 20px; }</pre>	Afecta a todos los elementos que estén contenidos dentro de la una etiqueta .
<pre>p, span { font-size: 20px; }</pre>	Asigna los mismos estilos a elementos con nombres diferentes; la regla afecta a todos los elementos <p> y encontrados en el documento.
<pre>main p { font-size: 20px; }</pre>	Afecta solo a los elementos <p> que se encuentran dentro de un elemento <main> , ya sea como contenido directo o insertados en otros elementos.
<pre>section > p { font-size: 20px; }</pre>	Usa el carácter > para referenciar un elemento que es hijo directo de otro. Indica que el elemento afectado es el elemento de la derecha cuando tiene como padre al elemento de la izquierda.
<pre>h1 + p { font-size: 20px; }</pre>	Este selector hace referencia a un elemento que está precedido por otro elemento; ambos deben compartir el mismo elemento padre. Afecta a todos los elementos <p> que se ubican inmediatamente después de un elemento <h1> .
<pre>p - p { font-size: 20px; }</pre>	Afecta a todos los elementos <p> que preceden a otro elemento <p> .
<pre>#mitexto { font-size: 20px; }</pre>	Se aplica al elemento identificado por el atributo id y el valor "mitexto" . La ventaja de este procedimiento es que cada vez que se crea una referencia usando el identificador mitexto en un archivo CSS, solo se modifica el elemento con esa identificación, pero el resto de los elementos no se ven afectados. Esta es una forma muy específica de referenciar a un elemento y se usa comúnmente con elementos estructurales, como <section> o <div> .
<pre>.mitexto { font-size: 20px; }</pre>	Este atributo es más flexible y se puede asignar a varios elementos dentro del mismo documento. Para referenciar un elemento usando su atributo class , el selector debe incluir el valor del atributo precedido por un punto.

Las propiedades son la pieza central de un documento CSS. Todos los estilos que es posible aplicar a un elemento se definen a través de ellas; hay cientos de propiedades disponibles, y se pueden clasificar en dos tipos: propiedades de formato y propiedades de diseño. Las primeras se encargan de dar forma a los elementos y su contenido, mientras que las de diseño están enfocadas a determinar el tamaño y la posición de los elementos en la pantalla. Las propiedades de formato se pueden clasificar según el tipo de modificación que producen. Por ejemplo, algunas propiedades cambian el tipo de letra que se usa para mostrar el texto, otras generan un borde alrededor del elemento, asignan un color de fondo, etc. A continuación se indicarán algunas propiedades fundamentales de los elementos, clasificadas según su tipo.

Texto

<code>font-family</code>	Declara el tipo de letra que se usa para mostrar en un texto. Se pueden declarar múltiples valores separados por coma para ofrecer al navegador varias alternativas en caso de que algunos tipos de letra no se encuentren disponibles en el equipo del usuario. Valores estándar son Georgia , "Times New Roman" , Arial , Helvetica , "Arial Black" , Gadget , Tahoma , Geneva , Helvetica , Verdana , Geneva , Impact y sans-serif .
<code>font-size</code>	Determina el tamaño de la letra; puede estar indicado en pixeles, puntos, porcentaje o cualquiera de las opciones que entrega el lenguaje.
<code>font-weight</code>	Indica si el texto se mostrará en negrita o no. Los valores posibles son normal y bold , pero también se puede indicar en múltiplos de 100 para grosores de letra.
<code>font-style</code>	Permite establecer el estilo de la letra. Valores posibles: normal , italic , y oblique .
<code>font</code>	Permite declarar varios atributos al mismo tiempo, separándolos por un espacio y en un orden adecuado. El estilo y el grosor se deben declarar antes que el tamaño, y el tipo de letra al final (por ejemplo, font: bold 24px Arial,sans-serif).

<code>text-align</code>	Alinea el texto de un elemento. Valores posibles: left , right , center , y justify .
<code>text-align-last</code>	Alinea la última línea de un párrafo. Valores posibles: left , right , center , y justify .
<code>text-indent</code>	Establecer el tamaño de sangría en un párrafo, que es el espacio vacío al comienzo de la línea.
<code>vertical-align</code>	Alinea elementos de forma vertical. Se usa para alinear texto con imágenes, y la propiedad se aplica a la imagen. Valores posibles: baseline , sub , super , text-top , text-bottom , middle , top , y bottom .
<code>line-height</code>	Define espacio entre líneas. Valor se declara en pixeles, porcentaje, puntos, o en cualquiera de las medidas establecidas en CSS.

Colores

<code>rgb</code> (rojo, verde, azul)	Define un color por medio de los valores especificados por los atributos (desde 0 a 255): nivel de rojo, nivel de verde y nivel de azul, en ese mismo orden.
<code>rgba</code> (rojo, verde, azul, alfa)	Similar a rgb() , pero con un parámetro adicional que define la opacidad; este valor va entre 0 (totalmente transparente) y 1 (totalmente opaco).
<code>hsl</code> (matiz, saturación, luminosidad)	Define un color desde valores especificados por los atributos. Los valores se declaran en números decimales y porcentajes.
<code>hsla</code> (matiz, saturación, luminosidad, alfa)	Cumple una función similar a hsl() , pero considera un parámetro adicional: alfa u opacidad.
<code>color</code>	declara el color del contenido del elemento.

A continuación se indica un ejemplo de cómo aplicar estilo de fuente a un elemento y un color.

```
#titulo {
  font: bold 26px Verdana, sans-serif;
  color: #CCCCCC;
}
```

Ilustración 9: Estilo para fuentes y colores

El color indicado anteriormente, #CCCCCC, es otra forma de representar un valor. Sigue la misma idea que el color RGB, y cada par de valores considera un color, y cada letra puede ir entre 0 - 9 y A - F. A este tipo de color se le denomina “color hexadecimal”.

Tamaño

width	Esta propiedad declara el ancho de un elemento. Se puede expresar en pixeles, porcentaje, puntos, etc.
height	Declara la altura de un elemento; se representa de igual forma en pixeles, puntos, porcentaje, centímetros, etc.
margin	Declara el margen de un elemento, y corresponde al espacio que hay alrededor de la caja. Puede recibir cuatro valores que representan el margen superior, derecho, inferior, e izquierdo, en ese orden y separados por un espacio (por ejemplo, margin: 10px 30px 10px 30px;).
padding	Declara el relleno de un elemento. Este último es el espacio entre el contenido del elemento y los límites de su caja. Los valores se declaran de la misma forma que lo hacemos para la propiedad margin .
overflow	Indica cómo se mostrará contenido que sale el elemento. Valores disponibles: visible (por defecto), hidden (esconde lo que no entra dentro de la caja), scroll (muestra barras para desplazar el contenido), auto (navegador decide).

Fondo

background-color	Asigna un color de fondo a un elemento.
background-image	Asigna una imagen de fondo a un elemento, referenciando la imagen con el método url() .
background-position	Declara el inicio de una imagen de fondo; los valores se pueden indicar en porcentaje, pixeles, o con las palabras clave center , left , right , top , y bottom .
background-size	Define el tamaño de la imagen de fondo; puede ser en pixeles o porcentaje, o bien usando las palabras clave cover o contain .
background-repeat	Determina si la imagen de fondo se repetirá, y en qué direcciones. Usa los valores repeat , repeat-x , repeat-y y no-repeat .

Bordes

border-width	Ancho del borde; acepta hasta cuatro valores separados por un espacio para especificar el ancho de cada lado del borde.
border-style	Indica estilos de bordes, cuatro valores separados por espacio. Valores posibles: none , hidden , dotted , dashed , solid , double , groove , ridge , inset , y outset .
border-color	Color del borde, cuatro valores separados.
border	Permite declarar todos los atributos del borde al mismo tiempo.
border-radius	Define el radio del círculo virtual que el navegador utilizará para dibujar las esquinas redondeadas.

Estas son solo algunas de las propiedades que se pueden definir a través de CSS. Desde la llegada de CSS3, se han agregado muchas opciones diversas para generar sitios web con diseños adecuados a las demandas actuales. Existen otras funciones más que es necesario al menos mencionar:

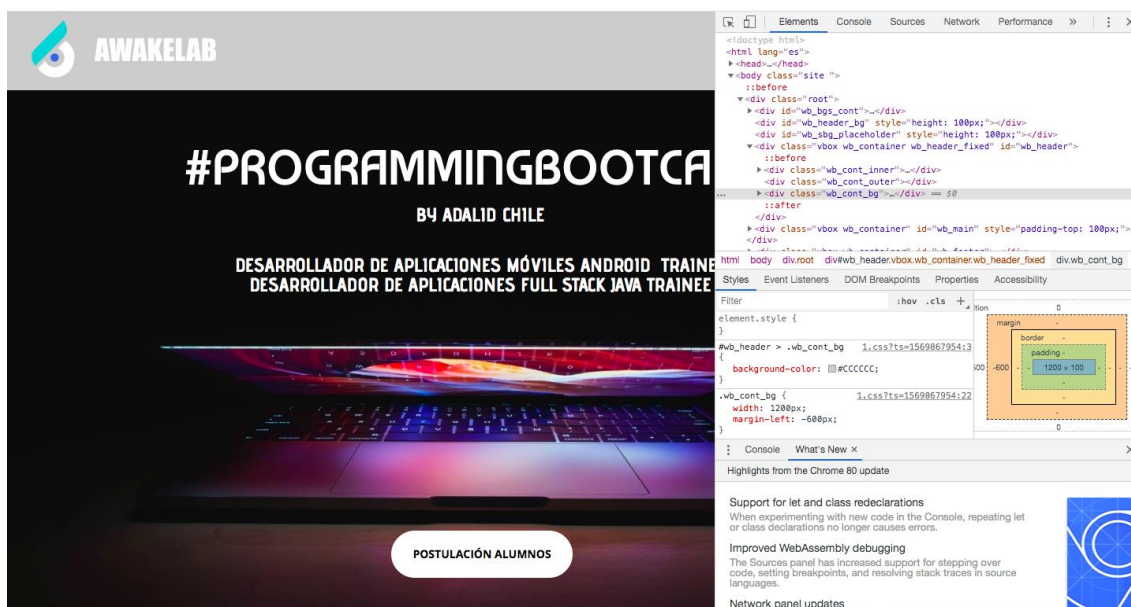
- **Sombras:** genera sombras para contenedores o textos; opciones posibles: **box-shadow**, **text-shadow**.
- **Gradientes:** se forma mediante una serie de colores que varían continuamente con una transición suave de un color a otro. Opciones posibles: **linear-gradient**, **radial-gradient**.
- **Filtros:** agregan efectos a un elemento y su contenido. CSS incluye la propiedad **filter** para asignar un filtro a un elemento y las siguientes funciones para crearlo. Opciones posibles: **blur()**, **grayscale()**, **drop-shadow()**, **sepia()**, **brightness()**, **contrast()**, **hue-rotate()**, **invert()**, **saturate()**, **opacity()**.
- **Transformaciones:** Esta propiedad realiza cuatro transformaciones básicas a un elemento: escalado, rotación, inclinación y traslación. Opciones posibles: **scale()**, **rotate()**, **skew()**, **translate()**; también existen en versión 3D.
- **Transiciones:** Con la seudoclase **:hover** podemos realizar transformaciones dinámicas. Sin embargo, una animación real requiere una transición entre los dos pasos del proceso. Para este propósito, CSS ofrece las siguientes propiedades: **transition-property**, **transition-duration**, **transition-timing-function**, **transition-delay**, **transition**.
- **Animaciones:** La propiedad **transition** crea una animación básica, pero solo se con dos estados en el proceso: el estado determinado por los valores actuales de las propiedades, y el estado determinado por los nuevos valores. CSS incluye las siguientes propiedades para componer animaciones más complejas: **animation-name**, **animation-duration**, **animation-timing-function**, **animation-delay**, **animation-iteration-count**, **animation-direction**, **animation-fill-mode**, **animation**.

3.1.3.5.- Inspeccionando elementos en el navegador

Cuando se desarrolla un sitio web, una herramienta muy altamente recomendable para el análisis de la composición del sitio es el “inspector de elementos”. Este tema se analizó en la sesión anterior, pero se hará el foco en una de sus características específicas.

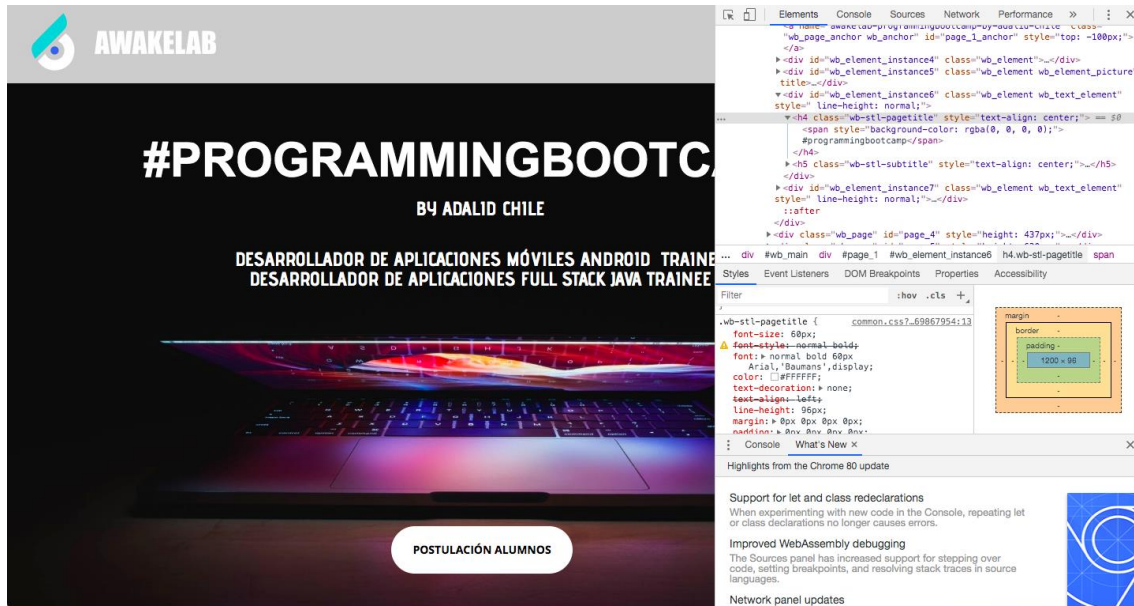
Se usará como ejemplo el sitio <http://www.awakelab.cl>; para ello se aplicarán cambios en los estilos del sitio. Es importante destacar que estos cambios son solo visibles localmente a nivel de equipo cliente, y no alterarán la estructura final del sitio bajo ningún aspecto. Se utilizará como referencia el navegador Google Chrome.

- Ingrese al sitio <http://www.awakelab.cl>
- En la cabecera del sitio, con el botón secundario del ratón, haga click y seleccione la opción “Inspeccionar”.
- En la pestaña “Elements” se desplegará el código fuente del sitio. Aparecerá seleccionada la etiqueta “<div class=“wb_cont_bg”>...</div>”.
- Se debe modificar el atributo “background-color: transparent;” a “background-color: CCCCCC;”. Esto modificará la cabecera desde un color negro a un gris.



- Se pueden modificar asimismo tipos de fuente. Para ello se debe situar el mouse sobre el texto “#programmingbootcamp”, y presionar el botón derecho, seleccionando la opción “Inspeccionar”.

- En los estilos declarados para la clase **“.wb-stl-pagetitle”**; en el atributo **“font”**, intercambie el orden de las fuentes de **“‘Baumans’,Arial,display”** a **“Arial, ‘Baumans’,display”**. Dado que las fuentes se aplican en orden, el título modificará su fuente a Arial.



- Siguiendo el mismo procedimiento, se pueden modificar todas las características del sitio. Es muy útil cuando se desee hacer pruebas sobre aplicación de estilos.

Anexo 1: Referencias

[1] Tutorial de CSS (en inglés)

Referencia: <https://www.w3schools.com/css/>

**[2] J.D Gauchat “El Gran libro de HTML5, CCS3 y JavaScript”
3ª edición**

[3] Curso de CSS3 (en inglés)

Referencia: <https://www.w3schools.in/category/css3/>

[4] Animaciones en CSS3

Referencia: https://www.w3schools.com/css/css3_animations.asp