



TALLER DE PROGRAMACIÓN WEB



SEPARATA I: DISEÑO Y CONSTRUCCIONES DE PÁGINAS WEB

Eric Gustavo Coronel Castillo
gcoronel@uni.edu.pe
gcoronelc.github.io

ÍNDICE

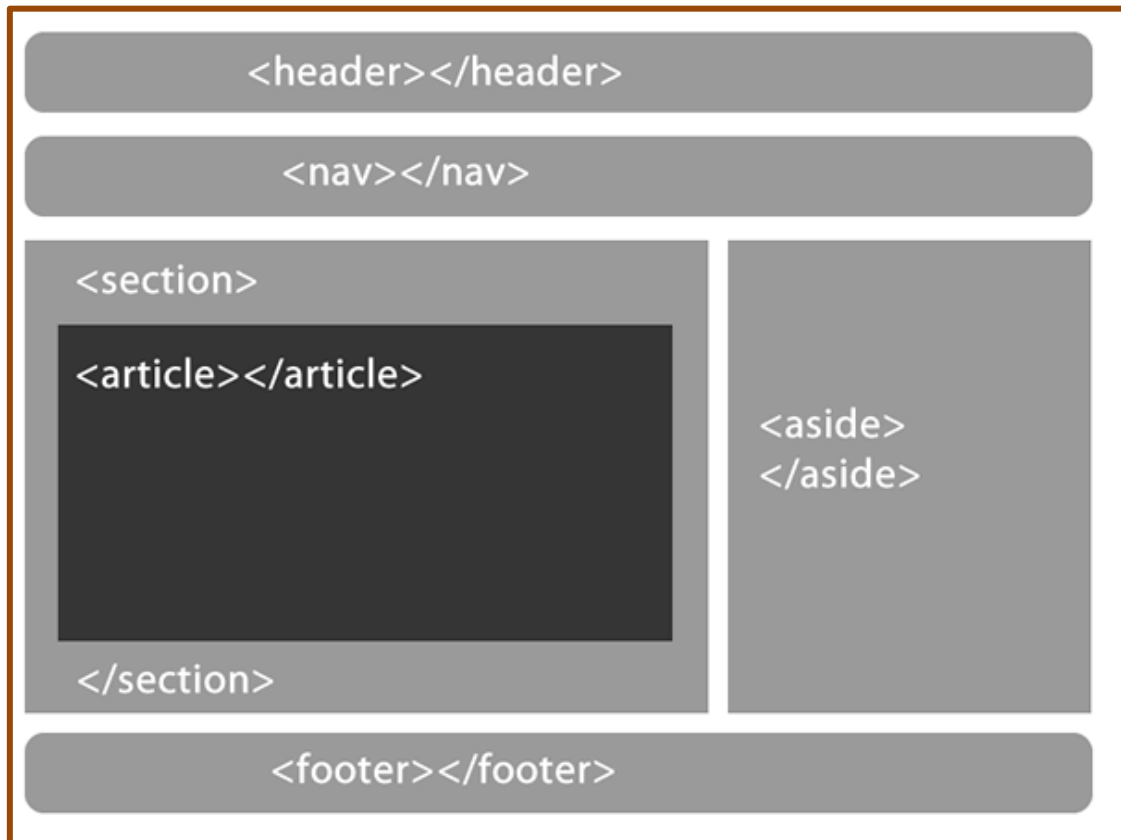
CAPÍTULO 1 FUNDAMENTOS DE HTML.....	5
OBJETIVO	5
INTRODUCCIÓN.....	6
ESTRUCTURA.....	7
<i>Etiqueta DOCTYPE.....</i>	<i>7</i>
<i>Etiqueta HTML.....</i>	<i>8</i>
<i>Etiqueta HEAD</i>	<i>8</i>
<i>Etiqueta BODY</i>	<i>8</i>
ETIQUETAS HTML5	9
<i>Modelo 1</i>	<i>9</i>
<i>Modelo 2</i>	<i>9</i>
<i>Etiqueta HEADER</i>	<i>10</i>
<i>Etiqueta NAV</i>	<i>10</i>
<i>Etiqueta MAIN</i>	<i>10</i>
<i>Etiqueta FOOTER</i>	<i>10</i>
<i>Etiqueta SECTION.....</i>	<i>10</i>
<i>Etiqueta ARTICLE</i>	<i>10</i>
<i>Etiqueta ASIDE.....</i>	<i>11</i>
<i>Ejemplo.....</i>	<i>11</i>
ETIQUETAS BÁSICAS.....	13
<i>Encabezados.....</i>	<i>13</i>
<i>Párrafos.....</i>	<i>13</i>
<i>Vínculos.....</i>	<i>14</i>
<i>Ejemplo.....</i>	<i>16</i>
TABLAS.....	18
<i>Etiquetas Básica.....</i>	<i>18</i>
<i>Ejemplo.....</i>	<i>19</i>
FORMULARIOS.....	21
<i>Fundamentos.....</i>	<i>21</i>

<i>Estructura</i>	21
<i>Entrada Básica de Datos</i>	22
<i>Ejemplo</i>	24
EJERCICIOS	26
<i>Ejercicio 1</i>	26
<i>Ejercicio 2</i>	26
<i>Ejercicio 3</i>	26
CAPÍTULO 2 FUNDAMENTOS DE CSS3	27
OBJETIVO	27
INTRODUCCIÓN	28
CONCEPTOS BÁSICOS	29
<i>Estilos en línea</i>	29
<i>Estilos embebidos</i>	30
<i>Archivos externos</i>	31
REFERENCIAS	33
<i>Selector universal</i>	33
<i>Referencia con palabra clave</i>	33
<i>Referenciando con el atributo id</i>	34
<i>Referenciando con el atributo class</i>	34
<i>Referenciando con cualquier atributo</i>	37
<i>Referenciando con pseudo clases</i>	39
NUEVOS SELECTORES	43
<i>Selector ></i>	43
<i>Selector +</i>	45
<i>Selector ~</i>	47
EJERCICIOS	49
<i>Ejercicio 4</i>	49
<i>Ejercicio 5</i>	51
CAPÍTULO 3 RETO	54

Capítulo 1

FUNDAMENTOS DE HTML

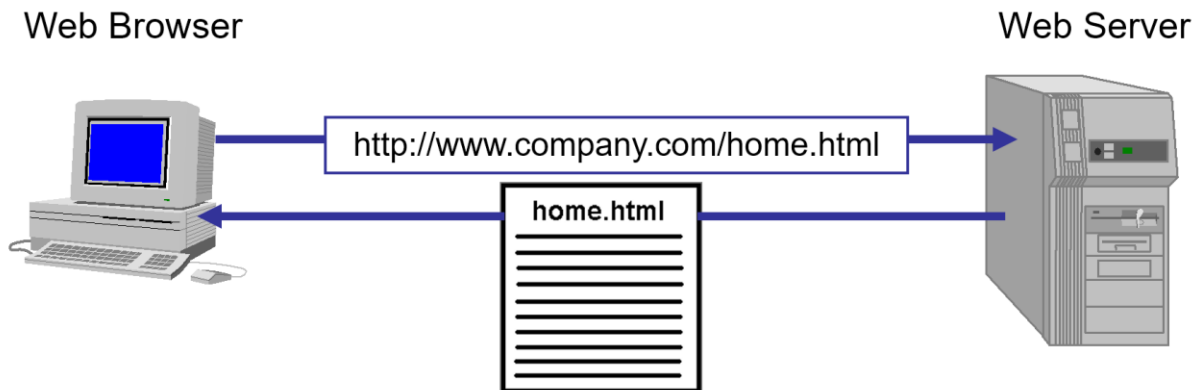
Objetivo



El objetivo es que entiendas los fundamentos de HTML, que es la base de las páginas y sistemas web.

Es muy importante que tengas claro cuáles son las partes de un documento HTML para que puedas crear aplicaciones consistentes y que los browser y motores de búsqueda lo interpreten correctamente.

Introducción



Cuando realizas el requerimiento de una página Web, el servidor la reporta tal como está guarda, es el browser el que interpreta el código HTML, CSS y JavaScript.

HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas web. Se trata de la sigla que corresponde a **HyperText Markup Language**, es decir, **Lenguaje de Marcas de Hipertexto**, que podría ser traducido como **Lenguaje de Formato de Documentos para Hipertexto**.

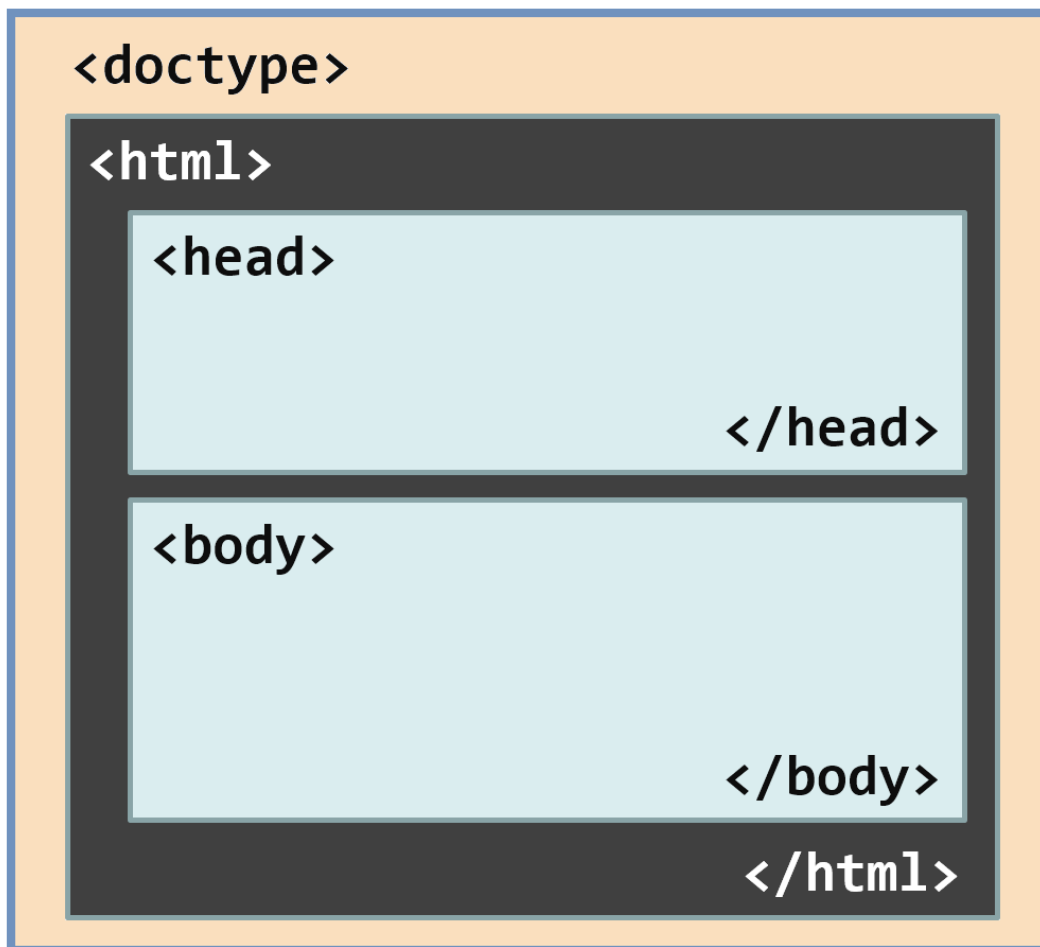
HTML, está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. HTML dispone de etiquetas para imágenes, hipervínculos que nos permiten dirigirnos a otras páginas, saltos de línea, listas, tablas, formularios, etc.

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos web definidos con HTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas Web que tengan una apariencia que sea de fácil interpretación por el usuario final.

JavaScript (abreviado comúnmente como JS) es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

JS es implementado como parte de los navegadores Web permitiendo mejoras en la interfaz de usuario y páginas Web dinámicas.

Estructura



Etiqueta DOCTYPE

La etiqueta DOCTYPE o "Declaración del tipo de documento" es una instrucción especial que va al inicio del documento HTML y que permite al navegador entender qué versión de HTML estás utilizando. Esta información determinará la manera en la que el navegador procesará el documento, un DOCTYPE distinto podría implicar hasta una visualización diferente del documento web dentro del mismo navegador.

Para HTML5 se utiliza de la siguiente forma:

```
<!DOCTYPE HTML>
```

Etiqueta HTML

Representa la raíz de un documento HTML o XHTML. Todos los demás elementos deben ser descendientes de este elemento.

Etiqueta HEAD

Representa una colección de metadatos acerca del documento, incluyendo enlaces a, o definiciones de, scripts y hojas de estilo.

Etiqueta BODY

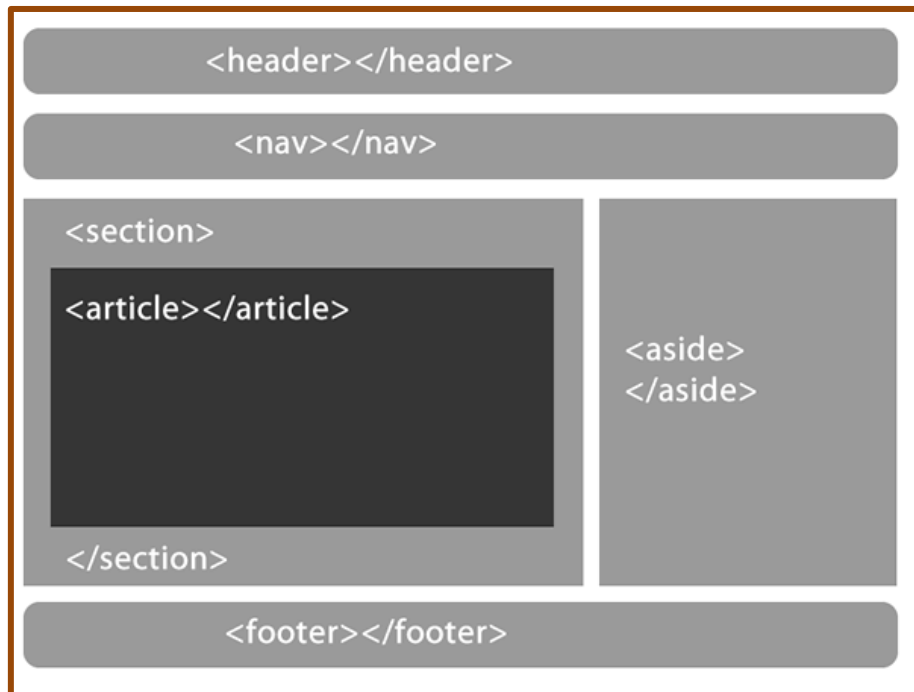
Representa el contenido principal de un documento HTML. Solo hay un elemento <BODY> en un documento Web.

A continuación, se tiene un documento básico:

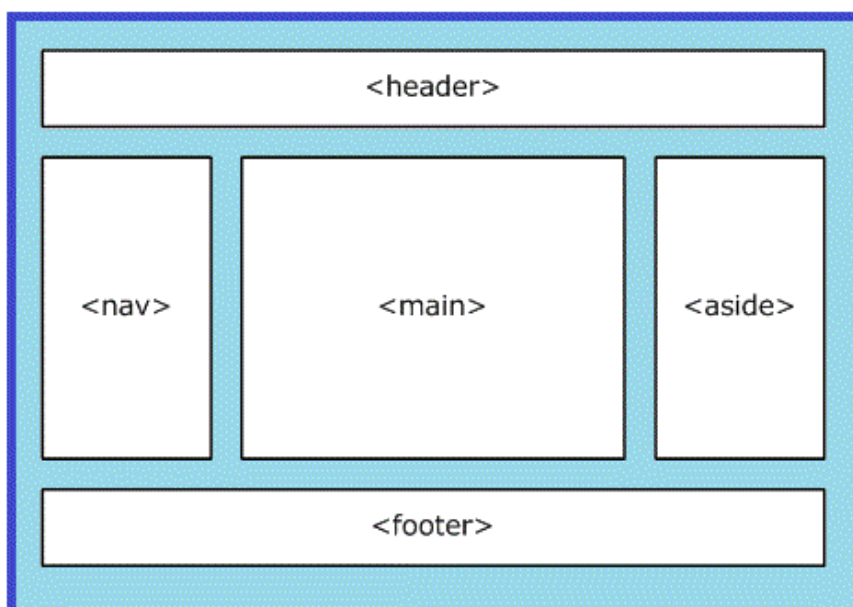
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>DEMO</title>
</head>
<body>
  <h1>DEMO</h1>
  <p>Este es un documento HTML básico</p>
</body>
</html>
```

Etiquetas HTML5

Modelo 1



Modelo 2



Etiqueta HEADER

Define la cabecera de una página o sección. Usualmente contiene un logotipo, el título del sitio Web y una tabla de navegación de contenidos.

Etiqueta NAV

Define una sección que solamente contiene enlaces de navegación o un menú.

Etiqueta MAIN

El elemento HTML MAIN representa el contenido principal del BODY de un documento o aplicación.

El área principal del contenido consiste en el contenido que está directamente relacionado, o se expande sobre el tema central de un documento o la funcionalidad central de una aplicación.

Este contenido debe ser único al documento, excluyendo cualquier contenido que se repita a través de un conjunto de documentos como barras laterales, enlaces de navegación, información de derechos de autor, logos del sitio y formularios de búsqueda.

Etiqueta FOOTER

Define el pie de una página o sección. Usualmente contiene un mensaje de derechos de autoría, algunos enlaces a información legal o direcciones para dar información de retroalimentación.

Etiqueta SECTION

El elemento de HTML SECTION representa una sección genérica de un documento. Sirve para determinar qué contenido corresponde a qué parte de un esquema. Piensa en el esquema como en el índice de contenido de un libro; un tema común y subsecciones relacionadas.

Es, por lo tanto, una etiqueta semántica. Su funcionalidad principal es estructurar semánticamente un documento web.

Etiqueta ARTICLE

El Elemento ARTICLE de HTML representa una composición auto-contenida en un documento, página, una aplicación o en el sitio, que se destina a distribuir de forma independiente o reutilizable, por ejemplo, en la sindicación. Podría ser un mensaje en un

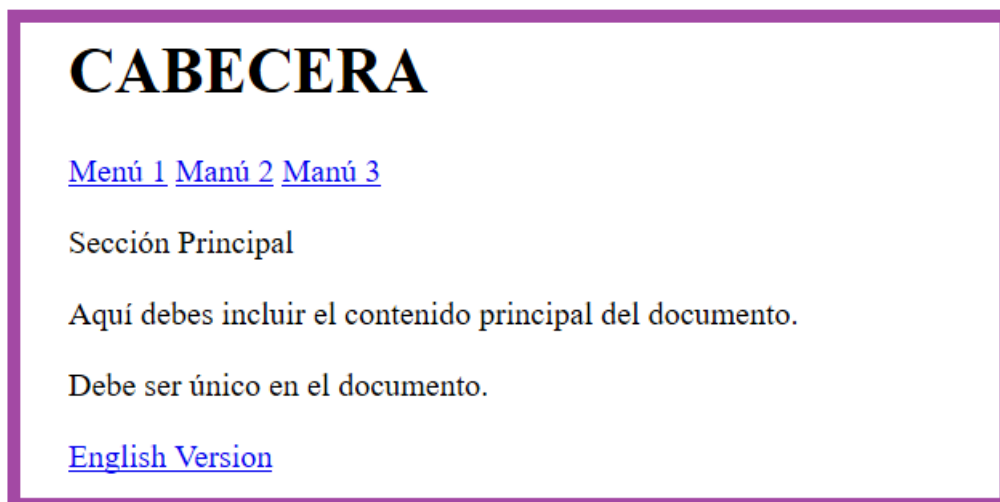
foro, un artículo de una revista o un periódico, una entrada de blog, un comentario de un usuario, un widget interactivo o gadget, o cualquier otro elemento independiente del contenido.

Etiqueta ASIDE

El Elemento HTML ASIDE representa una sección de una página que consiste en contenido que está tangencialmente relacionado con el contenido que le rodea, que podría ser considerado independiente de ese contenido.

Estas secciones son a menudo representadas como barras laterales y contienen una explicación al margen como una definición de glosario, elementos relacionados indirectamente como publicidad, la biografía del autor, o en aplicaciones web, la información de perfil o enlaces a blogs relacionados.

Ejemplo



```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5">
  <title>DEMO HTML5</title>
</head>
<body>
  <header>
    <h1>CABECERA</h1>
  </header>
  <nav>
    <a href="#">Menú 1</a>
    <a href="#">Manú 2</a>
    <a href="#">Manú 3</a>
  </nav>
  <main>
    <p>Sección Principal</p>
    <p>Aquí debes incluir el contenido principal del documento.</p>
    <p>Debe ser único en el documento.</p>
  </main>
  <aside >
    <a href="#">English Version</a>
  </aside>
</body>
</html>
```

Etiquetas Básicas

Encabezados

Etiqueta	Ejemplo
<H1>	Título 1: HTML
<H2>	Título 2: HTML
<H3>	Título 3: HTML
<H4>	Título 4: HTML
<H5>	Título 5: HTML
<H6>	Título 6: HTML

Párrafos

Los párrafos inician con la etiqueta <P> y terminan </P>. No importa como escribamos el texto, este se adaptará a los márgenes del visualizador del Browser.

Vínculos

La etiqueta que permite abrir y cerrar vínculos es: <A> ...

La sintaxis es:

```
<A HREF="xxx"> yyy </A>
```

Donde:

xxx es el destino del enlace.

yyy es el texto indicativo en la pantalla y generalmente esta subrayado.

ENLACE A UNA PÁGINA EXTERNA

Si quieres enlazar a una página o archivo que está en otro servidor:

```
<a href ="http://gcoronelc.blogspot.pe/p/java.html" title ="Programador Java">  
PROGRAMADOR JAVA  
</a>
```

- El atributo **href** contiene la URL a la que queremos enlazar.
- Es muy importante que no se nos olvide el protocolo (en este caso http://) o no funcionará.
- En **title** escribimos una descripción de la página que enlazamos.

ENLACE A UNA PAGINA LOCAL

Para enlazar a una página que esté en nuestro servidor, necesitamos saber la ruta (path) desde donde estemos hasta la ubicación del archivo. Si el origen (página donde está el link) está en el mismo directorio que el destino (página a la que apunta el link), entonces sólo tenemos que escribir su nombre:

```
<a href ="datosPersonales.html" title="Información sobre mí">  
Ficha personal  
</a>
```

Si el destino estuviera en un subdirectorio, se debe utilizar una barra / para indicar el camino:

```
<a href="galeria/fotos.html" title="Galería de fotos">  
    Ver galería de fotos  
</a>
```

Si el destino estuviera un directorio por encima, pondríamos dos puntitos y una barra ../ de esta manera:

```
<a href = "../index.html" title="Página principal">  
    Volver al inicio  
</a>
```

ENLACE A UNA DIRECCION DE EMAIL

Podemos crear un enlace que, al pulsar sobre él, se abra automáticamente una ventana del cliente de correo que tenga el usuario para que escriba un mensaje a esa dirección.

Para ello, sólo tenemos que usar **mailto:** en el atributo **href**, seguido de una dirección de correo electrónico:

```
<a href ="mailto:gcoronelc@gmail.com" title="Escribe tu consulta.">  
    Escribir consulta  
</a>
```

Ejemplo

LISTADO DE CURSOS

[Desarrollo de Software](#) [Gestión de Proyectos](#) [Base de Datos](#)

HTML es ...

Las principales etiquetas son ...

Sirve para ...

[English Version](#)

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5">
  <title>LISTADO DE CURSOS</title>
</head>
<body>
  <header>
    <h1>LISTADO DE CUSAROS</h1>
  </header>
  <nav>
    <a href="#">Desarrollo de Software</a>
    <a href="#">Gestión de Proyectos</a>
    <a href="#">Base de</a>
  </nav>
  <main>
    <p>HTML es ...</p>
    <p>Las principales etiquetas son ...</p>
    <p>Sirve para ...</p>
```

```
</main>  
<aside >  
  English Version  
</aside>  
</body>  
</html>
```


Tablas

Etiquetas Básica

Elemento	Descripción
<table>	Es la etiqueta principal para crear una tabla.
<caption>	Representa el título de una tabla.
<colgroup>	Representa un conjunto de una o más columnas de una tabla.
<col>	Representa una columna de una tabla.
<tbody>	Representa el bloque de filas que describen los datos concretos de una tabla.
<thead>	Representa el bloque de filas que describen las etiquetas de columna de una tabla.
<tfoot>	Representa los bloques de filas que describen los resúmenes de columna de una tabla.
<tr>	Representa una fila de celdas en una tabla.
<td>	Representa una celda de datos en una tabla.
<th>	Representa una celda de encabezado en una tabla.

Ejemplo

LISTADO DE CUSAROS

[Desarrollo de Software](#) [Gestión de Proyectos](#) [Base de Datos](#)

DESARROLLO DE SOFTWARE

Curso Disponibles

Curso	Horas	Horario
Java Developer	90	16:00 - 20:00
HTML5	30	16:00 - 20:00
Spring Framework	60	16:00 - 20:00

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5">
  <title>LISTADO DE CURSOS</title>
</head>
<body>
  <header>
    <h1>LISTADO DE CUSAROS</h1>
  </header>
  <nav>
    <a href="#">Desarrollo de Software</a>
    <a href="#">Gestión de Proyectos</a>
    <a href="#">Base de Datos</a>
  </nav>
  <main>
    <h2>DESARROLLO DE SOFTWARE</h2>
    <table>
```

```
<caption>Curso Disponibles</caption>
<head>
  <tr>
    <th>Curso</th>
    <th>Horas</th>
    <th>Horario</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td>Java Developer</td>
    <td>90</td>
    <td>16:00 - 20:00</td>
  </tr>
  <tr>
    <td>HTML5</td>
    <td>30</td>
    <td>16:00 - 20:00</td>
  </tr>
  <tr>
    <td>Spring Framework</td>
    <td>60</td>
    <td>16:00 - 20:00</td>
  </tr>
</tbody>
</table>
</main>
</body>
</html>
```

Formularios

Fundamentos

HTML permite usar formularios en el documento, como una manera básica de proveer al usuario de un método para enviar información hacia el servidor.

Las aplicaciones de los formularios son muy variadas, entre las que podemos mencionar:

- Ordenar un producto
- Enviar un comentario
- Hacer una operación bancaria
- Inscribirse en una base de datos
- Etc.

Estructura

La etiqueta HTML que nos va a permitir la creación de formularios es `<FORM>`. Su formato es el siguiente:

```
<FORM action="URL del programa que trata el formulario"
      method="post/get">

      Elementos del formulario

</FORM>
```

Estructura de un formulario:

1. Etiqueta de inicio: `<FORM METHOD="get/post" ACTION="...">`
2. Cuerpo del formulario, con los distintos elementos para poder introducir datos.
3. Botones de envió y de borrado.
4. Etiqueta de cierre: `</FORM>`

Entrada Básica de Datos

Estructura

```
<INPUT type= text/password/checkbox/radio/hidden/submit/reset  
      name="Variable"  
      value="Valor inicial">
```

Ingreso de Texto

```
<INPUT      type="text"  
            name="Variable"  
            value="Valor inicial"  
            size="Tamaño"  
            maxlength="Longitud máxima">
```

Ingreso de Password

```
<INPUT      type="password"  
            name="Variable"  
            value="Valor inicial"  
            size="Tamaño"  
            maxlength="Longitud máxima">
```

CheckBox

```
<INPUT      type="checkbox"  
            name="Variable"  
            value="Valor"  
            checked>
```

Botón de Envío de Datos

```
<INPUT type= submit value="Mensaje a mostrar">
```

Botón de Borrado de Datos

```
<INPUT type= reset value="Texto del botón">
```

Entrada de Datos en Múltiples Líneas

```
<TEXTAREA name="Variable" rows=Filas cols=Columnas>
```

Contenido por defecto.

```
</TEXTAREA>
```

Entrada a través de listas de selección

```
<SELECT name="Variable" size=n multiple>  
  <OPTION selected value="Valor_1"> Primera Opción</OPTION>  
  
  <OPTION value="Valor_2"> Segunda Opción</OPTION>  
  
  <OPTION value="Valor_n"> Enésima Opción</OPTION>  
  
</SELECT>
```

Ejemplo

FORMULARIO BASICO

Datos de Contacto

Escribe tu nombre:

Escriba su correo electrónico:

Enviar

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5">
  <title>DEMO HTML5</title>
</head>
<body>
  <header>
    <h1>FORMULARIO BASICO</h1>
  </header>
  <main>
    <h2>Datos de Contacto</h2>
    <form action="datos.php" method="post">
      Escribe tu nombre: <br/>
      <input type="text" name="nombre" size="50" /><br/><br/>
      Escriba su correo electrónico:<br/>
```

```
<input type="text" name="correo" size="50"/><br/><br/>  
<input type="submit" value="Enviar"/>  
</form>  
</main>  
</body>  
</html>
```


Ejercicios

Ejercicio 1

Crear una página con los datos de un curso, los estudiantes matriculados y sus respectivas notas.

Ejercicio 2

Crear una página para registrar las preferencias de los estudiantes de la UNI con respecto a los cursos y horarios para el siguiente ciclo, será a modo de encuesta.

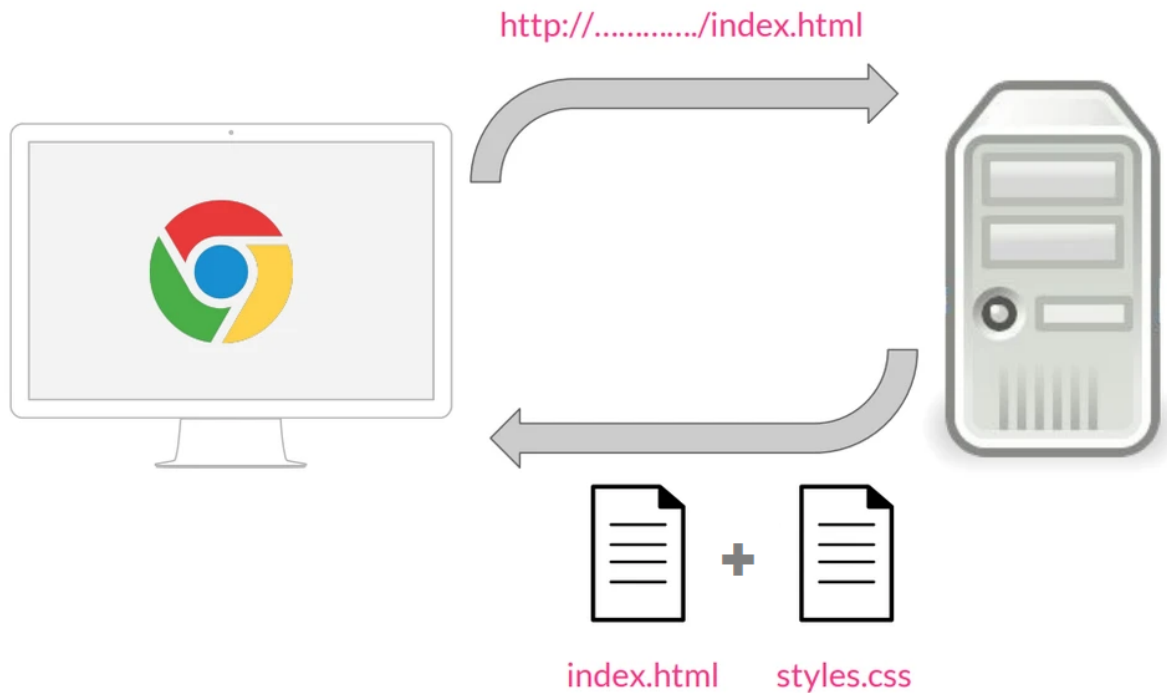
Ejercicio 3

Crear una página para registrar un reclamo por una nota de un estudiante de la UNI.

Capítulo 2

Fundamentos de CSS3

Objetivo



CSS significa hojas de estilo de cascada y es un lenguaje de hojas de estilo. CSS te permiten definir la apariencia de cada elemento dentro de los documentos en HTML.

La nueva especificación de HTML5 define nuevos elementos HTML. Pero la Web demanda diseño y funcionalidad, no solo organización estructural o definición de secciones. En este nuevo paradigma, HTML se presenta junto con CSS y JavaScript como un único instrumento integrado de desarrollo Web.

En este contexto, CSS es un lenguaje de estilos con el que crearas el diseño y presentación de las páginas Web, es decir, cómo lucen cuando un usuario las visita. Funciona junto con el lenguaje HTML que se encarga del contenido de las páginas.

Introducción

Los nuevos elementos HTML responsables de la estructura del documento ya fueron explicados en el capítulo anterior. Ahora es momento de analizar CSS, su relevancia dentro de esta unión estratégica y su influencia sobre la presentación de documentos HTML.

Oficialmente CSS nada tiene que ver con HTML5. CSS no es parte de la especificación y nunca lo fue. Este lenguaje es, de hecho, un complemento desarrollado para superar las limitaciones y reducir la complejidad de HTML. Al comienzo, atributos dentro de las etiquetas HTML proveían estilos esenciales para cada elemento, pero a medida que el lenguaje evolucionó, la escritura de códigos se volvió más compleja y HTML por sí mismo no pudo más satisfacer las demandas de los diseñadores. En consecuencia, CSS pronto fue adoptado como la forma de separar la estructura de la presentación. Desde entonces, CSS ha crecido y ganado importancia, pero siempre desarrollado en paralelo, enfocado en las necesidades de los diseñadores y apartado del proceso de evolución de HTML.

La versión 3 de CSS sigue el mismo camino, pero esta vez con un mayor compromiso. La especificación de HTML5 fue desarrollada considerando CSS a cargo del diseño. Debido a esta consideración, la integración entre HTML y CSS es ahora vital para el desarrollo Web y esta es la razón por la que cada vez que mencionamos HTML5 también estamos haciendo referencia a CSS3, aunque oficialmente se trate de dos tecnologías completamente separadas.

En este capítulo, vamos a estudiar conceptos básicos de CSS y las nuevas técnicas de CSS3 ya disponibles para presentación y estructuración. También aprenderemos cómo utilizar los nuevos selectores y pseudo clases que hacen más fácil la selección e identificación de elementos HTML.

Conceptos Básicos

Aplicar estilos a los elementos HTML cambia la forma en que estos son presentados en pantalla. Los navegadores proveen estilos por defecto que en la mayoría de los casos no son suficientes para satisfacer las necesidades de los diseñadores. Para cambiar esto, debes sobrescribir estos estilos usando diferentes técnicas.

Estilos en línea

Una de las técnicas más simples para incorporar estilos CSS a un documento HTML es la de asignar los estilos dentro de las etiquetas por medio del atributo **style**.

El siguiente ejemplo muestra un documento HTML simple que contiene el elemento `<p>` modificado por el atributo **style** con el valor **font-size: 20px**. Este estilo cambia el tamaño por defecto del texto dentro del elemento `<p>` a un nuevo tamaño de 20 pixeles.

Archivo: pagina0201.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8"/>
  <title>PAGINA PERSONAL</title>
</head>
<body>
  <p style="font-size: 20px;">Hola, mi nombre es Gustavo Coronel.</p>
</body>
</html>
```

Usar esta técnica es una buena manera de probar estilos y obtener una vista rápida de sus efectos, pero no es recomendado para aplicar estilos a todo el documento. La razón es simple: cuando usamos esta técnica, debemos escribir y repetir cada estilo en cada uno de los elementos que queremos modificar, incrementando el tamaño del documento a proporciones inaceptables y haciéndolo imposible de mantener y actualizar. Solo imagine lo que ocurriría si decide que en lugar de 20 pixeles el tamaño de cada uno de los elementos `<p>` debería ser de 24 pixeles. Tendría que modificar cada estilo en cada etiqueta `<p>` en el documento completo.

Estilos embebidos

Una mejor alternativa es insertar los estilos en la cabecera del documento y luego usar referencias para afectar los elementos HTML correspondientes.

Archivo: pagina0202.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8"/>
  <title>TALLER DE PROGRAMACIÓN WEB</title>
<style>
  p {font-size:20px;}
</style>
</head>
<body>
  <p>TALLER DE PROGRAMACIÓN WEB</p>
  <p>Profesor: Gustavo Coronel</p>
  <p>Correo: gcoronelc@gmail.com</p>
  <p>Página Web: gcoronelc.github.io</p>
</body>
</html>
```

El elemento `<style>` te permite agrupar estilos CSS dentro del documento. En versiones previas de HTML era necesario especificar qué tipo de estilos serían insertados. En HTML5 los estilos por defecto son CSS, por lo tanto, no necesitamos agregar ningún atributo en la etiqueta de apertura `<style>`.

En el ejemplo mostrado no se tuvo que escribir el estilo dentro de cada etiqueta `<p>` porque todos los elementos `<p>` ya fueron afectados. Con este método, reduces tu código y asignas los estilos que quieras a elementos específicos utilizando referencias.

Archivos externos

Declarar los estilos en la cabecera del documento ahorra espacio y hace el código más consistente y actualizable, pero requiere que realices una copia de cada grupo de estilos en todos los documentos de tu sitio web.

La solución es mover todos los estilos a un archivo externo y luego debes utilizar la etiqueta `<link>` para insertar este archivo dentro de cada documento donde lo necesites. Este método te permite cambiar los estilos por completo simplemente incluyendo un archivo diferente. También te permite modificar o adaptar tus documentos a cada circunstancia o dispositivo.

Utilizando la línea:

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

Le indicas al navegador que cargue el archivo `estilos.css` que contiene todos los estilos que necesitas para presentar el documento en pantalla. Esta práctica ha sido ampliamente adoptada por diseñadores que están trabajando con HTML5. La etiqueta `<link>` referenciando el archivo CSS tienes que insertarla en cada uno de tus documentos que requieren de esos estilos.

Archivo: estilos.css

```
h1 {  
  font-size: 28px;  
  color: #098ee3;  
  border-bottom: 2px solid #098ee3;  
}  
div {  
  margin: 10px 0px;  
  padding: 20px;  
  background-color: #dffbf7;  
}  
p {  
  font-family: "Lucida Sans", Geneva, Verdana, sans-serif;  
  color: #272c30;
```

```
}
```

Archivo: pagina0203.html

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <link rel="StyleSheet" type="text/css" href="estilos.css" />
    <title>TALLER DE PROGRAMACIÓN WEB</title>
  </head>
  <body>
    <h1>TALLER DE PROGRAMACIÓN WEB</h1>
    <div>
      <p>Profesor: Gustavo Coronel</p>
      <p>Correo: gcoronelc@gmail.com</p>
      <p>Página Web: gcoronelc.github.io</p>
    </div>
  </body>
</html>
```

A modo de practica modifica el archivo [pagina0203.html](#) con tus datos y también el archivo de estilos [estilos.css](#) según tu criterio.

Referencias

Almacenar todos tus estilos en un archivo externo e insertar este archivo dentro de cada uno de tus documentos que lo necesite es muy conveniente, sin embargo, debes usar buenos mecanismos que te ayuden a establecer una específica relación entre estos estilos y los elementos del documento que van a ser afectados.

Existen varios métodos para seleccionar cuáles elementos HTML serán afectados por las reglas CSS:

- Referencia por la palabra clave del elemento
- Referencia por el atributo `id`
- Referencia por el atributo `class`

CSS3 es bastante flexible e incorpora nuevas y más específicas técnicas para referenciar elementos, pero por ahora aplicaremos solo estas tres.

Selector universal

El selector universal CSS (*) se aplica a todos los elementos de cualquier tipo.

Sintaxis

```
* { style properties }
```

Referencia con palabra clave

Cuando declaras las reglas CSS utilizando la palabra clave del elemento afectas a todos los elementos de la misma clave en el documento. Por ejemplo, la siguiente regla cambiará los estilos de todos los elementos `<p>`:

```
p { font-size: 20px; }
```

Esta es la técnica que has utilizado en el archivo `estilos.css`. Utilizando la palabra clave `p` le estas diciendo al navegador que esta regla la debe aplicar a cada elemento `<p>` que se encuentre en el documento HTML. Todos los textos que se encuentren en las etiquetas `<p>` tendrán el tamaño de 20 píxeles.

Referenciando con el atributo id

El atributo **id** es como un nombre que identifica al elemento. Esto significa que no puedes tener elementos con este atributo duplicado. Este nombre debe ser único en todo el documento.

Para referenciar un elemento en particular usando el atributo **id** en tu archivo CSS debes declararlo con el símbolo **#** como se ilustra a continuación:

```
#titulo { font-size: 24px; }
```

La regla anterior es aplicada al elemento HTML identificado con el atributo **id="titulo"** como se ilustra a continuación:

```
<h1 id="titulo">TALLER DE PROGRAMACIÓN WEB</h1>
```

Referenciando con el atributo class

La mejor manera de aplicar CSS es mediante el atributo **class**. Este atributo es más flexible y puede ser asignado a cada elemento HTML en el documento que comparte un diseño similar.

Para que puedas trabajar con el atributo **class**, debes primero declarar la regla CSS con un punto antes del nombre. A continuación, tienes un ejemplo de la definición de clases en un archivo CSS.

Archivo: estilos.css

```
.titulo {  
  font-size: 28px;  
  color: #098ee3;  
  border-bottom: 2px solid #098ee3;  
}  
  
.region {  
  margin: 10px 0px;
```

```
padding: 20px;
background-color: #dffbf7;
}

.parrafo {
font-family: "Lucida Sans", Geneva, Verdana, sans-serif;
color: #272c30;
}
```

La ventaja de este método es que solo debes insertar el atributo **class** con el valor de la clase respectiva y es suficiente para que asignes los estilos a cualquier elemento del documento.

A continuación, tienes un ejemplo de como debes utilizar en el atributo **class**:

Archivo: pagina0204.html

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <link rel="StyleSheet" type="text/css" href="estilos.css" />
    <title>TALLER DE PROGRAMACIÓN WEB</title>
  </head>
  <body>
    <h1 class="titulo">TALLER DE PROGRAMACIÓN WEB</h1>
    <div class="region">
      <p class="parrafo">Profesor: Gustavo Coronel</p>
      <p class="parrafo">Correo: gcoronelc@gmail.com</p>
      <p class="parrafo">Página Web: gcoronelc.github.io</p>
    </div>
  </body>
</html>
```

La razón por la que debes utilizar un punto delante del nombre de la regla es que es posible construir referencias más complejas. Por ejemplo, puedes utilizar el mismo valor para el atributo **class** en diferentes tipos de elementos, pero asignando diferentes estilos para cada tipo de elemento.

A continuación, tienes un ejemplo de la hoja de estilo.

Archivo: estilos.css

```
h1.formato {
  font-size: 28px;
  color: #098ee3;
  border-bottom: 2px solid #098ee3;
}

.region {
  margin: 10px 0px;
  padding: 20px;
  background-color: #dffbf7;
}

p.formato {
  font-family: "Lucida Sans", Geneva, Verdana, sans-serif;
  color: #272c30;
}
```

A continuación, tienes un documento Web donde se ilustra su aplicación.

Archivo: pagina0205.html

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <link rel="StyleSheet" type="text/css" href="estilos.css" />
    <title>TALLER DE PROGRAMACIÓN WEB</title>
  </head>
  <body>
```

```
<h1 class="formato">TALLER DE PROGRAMACIÓN WEB</h1>
<div class="region">
  <p class="formato">Profesor: Gustavo Coronel</p>
  <p class="formato">Correo: gcoronelc@gmail.com</p>
  <p class="formato">Página Web: gcoronelc.github.io</p>
</div>
</body>
</html>
```

Referenciando con cualquier atributo

Los métodos de referencia que has aprendido cubren un variado espectro de situaciones, pero en algunos casos te serán insuficientes para encontrar el elemento exacto. La última versión de CSS ha incorporado nuevas formas de referenciar elementos HTML. Uno de ellos es el selector de atributo. Ahora puedes referenciar un elemento no solo por los atributos **id** y **class**, también puedes hacerlo a través de cualquier otro atributo.

A continuación, tienes la siguiente regla CSS:

```
p[name] { font-size: 20px; }
```

Esta regla se aplica solo elementos **<p>** que tienen un atributo llamado **name**.

Si quieres hacer algo similar a lo que se realizó con los atributos **id** y **class**, puedes también especificar el valor del atributo.

A continuación, tienes la siguiente regla CSS:

```
p[name="parrafo"] { font-size: 20px; }
```

CSS3 te permite combinar "=" con otros caracteres para que puedas hacer una selección más específica, como se ilustra a continuación:

```
p[name^="parrafo"] { font-size: 20px; }  
p[name$="párrafo"] { font-size: 20px; }  
p[name*="parrafo"] { font-size: 20px; }
```

El significado de estos selectores en CSS3 tienen similar significado a las expresiones regulares de otros lenguajes, como Java, JavaScript, etc.:

- La regla con el selector ^= será asignada a todo elemento <p> que contiene un atributo name con un valor que comienza con "párrafo", por ejemplo "parrafoPrincipal", "parrafoResaltado", etc.
- La regla con el selector \$= será asignada a todo elemento <p> que contiene un atributo name con un valor que finaliza en "párrafo", por ejemplo "miparrafo", "elparrafo", etc.
- La regla con el selector *= será asignada a todo elemento <p> que contiene un atributo name con un valor que incluye el texto "párrafo", en este caso, el texto podría también encontrarse en el medio, como en "elparrafoprincipal".

Referenciando con pseudo clases

CSS3 también incorpora nuevas pseudo clases que hacen la selección aún más específica.

Para ilustrarte el uso de pseudo clases utilizare el código del archivo [pagina0206.html](#).

Archivo: pagina0206.html

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>PSEUDO CLASES</title>
    <link rel="stylesheet" href="estilos.css" />
  </head>
  <body>
    <h1>APLICANDO PSEUDI CLASES</h1>
    <div id="principal">
      <p class="parrafo">Mi texto1</p>
      <p class="parrafo">Mi texto2</p>
      <p class="parrafo">Mi texto3</p>
      <p class="parrafo">Mi texto4</p>
    </div>
  </body>
</html>
```

Este código HTML contiene cuatro elementos `<p>`, considerando la estructura HTML puedes ver que se encuentran al mismo nivel, entonces, son hermanos entre sí e hijos del mismo elemento `<div>`.

Usando pseudo clases puedes aprovechar esta organización y referenciar un elemento específico sin importar cuánto conocemos sobre sus atributos y el valor de los mismos.

Caso 1

```
p:nth-child(2) {  
  font-family: "Lucida Sans", Geneva, Verdana, sans-serif;  
  color: blue;  
  font-weight: bold;  
}
```

La pseudo clase las agregadas usando dos puntos luego de la referencia y antes del su nombre. En el ejemplo de la regla del **Caso 1** estás haciendo referencia solo elementos `<p>`. Esta regla puede incluir otras referencias. Por ejemplo, podrías escribirla como `.miclase:nth-child(2)` para referenciar todo elemento que es hijo de otro elemento y tiene el valor de su atributo `class` igual a `miclase`. La pseudo clase puede ser aplicada a cualquier tipo de referencia estudiada previamente.

La pseudo clase `nth-child()` te permite encontrar un hijo específico. Como ya explicamos, el ejemplo HTML del **Caso 1** tiene cuatro elementos `<p>` que son hermanos. Esto significa que todos ellos tienen el mismo padre que es el elemento `<div>`. Lo que esta pseudo clase está realmente indicando es algo como: “**el hijo en la posición...**” por lo que el número entre paréntesis será el número de la posición del hijo, o índice. La regla del **Caso 1** está referenciando cada segundo elemento `<p>` encontrado en el documento.

El resultado es siguiente:

APLICANDO PSEUDO CLASES

Mi texto1
Mi texto2
Mi texto3
Mi texto4

Usando este método de referencia puedes seleccionar cualquier hijo, solo necesitas cambiar el número de índice.

Caso 2

```
p {  
  font-family: "Lucida Sans", Geneva, Verdana, sans-serif;  
  font-weight: bold;  
}  
  
p:nth-child(odd) {  
  color: blue;  
}  
  
p:nth-child(even) {  
  color: red;  
}
```

En el caso de las reglas del **Caso 2**, estas definiendo un color para los elementos `<p>` con índice impar y otro color para los de índice par.

El resultado es el siguiente:

APLICANDO PSEUDO CLASES

Mi texto1

Mi texto2

Mi texto3

Mi texto4

Caso 3

```
p {  
  font-family: "Lucida Sans", Geneva, Verdana, sans-serif;  
  font-weight: bold;  
}  
  
p:last-child {  
  color: blue;  
}  
  
p:first-child {  
  color: red;  
}
```

En el caso de las reglas del **Caso 3**, estas definiendo un color para el primer elemento y otro para el último elemento.

El resultado es el siguiente:

APLICANDO PSEUDO CLASES

Mi texto1

Mi texto2

Mi texto3

Mi texto4

Nuevos selectores

Ahora con CSS3 cuentas con nuevos selectores que te pueden resultar útiles para tus diseños. Estos selectores usan los símbolos **>**, **+** y **~** para especificar la relación entre elementos.

Selector **>**

El selector **>** indica que el elemento a ser afectado por la regla es el elemento de la derecha cuando tiene al de la izquierda como su padre.

A continuación, tienes el siguiente ejemplo:

```
div > p.parrafo {  
  color: blue;  
}
```

Esta regla modifica los elementos **<p>** que son hijos de un elemento **<div>**. Este caso, es bien específico y se hace referencia solamente los elemento **<p>** con el valor **parrafo** en su atributo **class**.

A continuación, tienes su aplicación:

```
<!DOCTYPE html>  
<html lang="es">  
  <head>  
    <title>PSEUDO CLASES</title>  
    <link rel="stylesheet" href="estilos.css" />  
  </head>  
  <body>  
    <h1>APLICANDO PSEUDO CLASES</h1>  
    <div id="principal">  
      <p>Mi texto1</p>  
      <p class="parrafo">Mi texto2</p>  
      <p class="parrafo">Mi texto3</p>  
      <p>Mi texto4</p>  
    </div>
```

```
<p class="parrafo">Mi texto5</p>  
</body>  
</html>
```

El resultado es el siguiente:

APLICANDO PSEUDO CLASES

Mi texto1

Mi texto2

Mi texto3

Mi texto4

Mi texto5

Como puedes observar, la clase `parrafo` solo se aplica a los elementos `<p>` que se encuentran dentro del elemento `<div>`.

Selector +

Sintaxis

```
Elemento1 + elemento2 { style properties }
```

El selector **+** indica que será afectado el primer **elemento2** que se ubica inmediatamente después del **elemento1**.

Aquí tienes un ejemplo:

```
div + p.parrafo {  
    color: blue;  
}
```

Esta regla modifica el primer elemento **<p>** que se encuentra inmediatamente después de un elemento **<div>**, además, en el atributo **class** debe tener el valor **parrafo**.

A continuación, tienes su aplicación:

```
<!DOCTYPE html>  
<html lang="es">  
  <head>  
    <title>PSEUDO CLASES</title>  
    <link rel="stylesheet" href="estilos.css" />  
  </head>  
  <body>  
    <h1>APLICANDO PSEUDO CLASES</h1>  
    <div id="principal">  
      <p>Mi texto1</p>  
      <p class="parrafo">Mi texto2</p>  
      <p class="parrafo">Mi texto3</p>  
    </div>  
    <p class="parrafo">Mi texto4</p>  
    <p class="parrafo">Mi texto5</p>  
    <p class="parrafo">Mi texto6</p>
```

```
</body>  
</html>
```

El resultado es el siguiente:

APLICANDO PSEUDO CLASES

Mi texto1

Mi texto2

Mi texto3

Mi texto4

Mi texto5

Mi texto6

Selector ~

Sintaxis

```
Elemento1 ~ elemento2 { style properties }
```

El selector ~ indica que serán afectados todos los elementos de tipo **elemento2** que se ubican después del **elemento1**.

Aquí tienes un ejemplo:

```
div ~ p.parrafo {  
    color: blue;  
}
```

Esta regla modifica los elementos **<p>** que se encuentra después de un elemento **<div>**, además, en el atributo **class** debe tener el valor **parrafo**.

A continuación, tienes su aplicación:

```
<!DOCTYPE html>  
<html lang="es">  
  <head>  
    <title>PSEUDO CLASES</title>  
    <link rel="stylesheet" href="estilos.css" />  
  </head>  
  <body>  
    <h1>APLICANDO PSEUDO CLASES</h1>  
    <div id="principal">  
      <p>Mi texto1</p>  
      <p class="parrafo">Mi texto2</p>  
      <p class="parrafo">Mi texto3</p>  
    </div>  
    <p>Mi texto4</p>  
    <p class="parrafo">Mi texto5</p>  
    <p class="parrafo">Mi texto6</p>
```

```
</body>  
</html>
```

El resultado es el siguiente:

APLICANDO PSEUDO CLASES

Mi texto1

Mi texto2

Mi texto3

Mi texto4

Mi texto5

Mi texto6

Ejercicios

Ejercicio 4

A partir del código HTML y CSS que se muestra, añadir los selectores CSS que faltan para aplicar los estilos deseados.

Cada regla CSS incluye un comentario en el que se explica los elementos a los que debe aplicarse:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8" />
<title>EJERCICIO DE SELECTORES</title>
<style type="text/css">
/* Todos los elementos de la pagina */
{ font: 1em/1.3 Arial, Helvetica, sans-serif; }

/* Todos los parrafos de la pagina */
{ color: #555; }

/* Todos los párrafos contenidos en #primero */
{ color: #336699; }

/* Todos los enlaces la pagina */
{ color: #CC3300; }

/* Los elementos "em" contenidos en #primero */
{ background: #FFFFCC; padding: .1em; }

/* Todos los elementos "em" de clase "especial" en toda la pagina */
{ background: #FFCC99; border: 1px solid #FF9900; padding: .1em; }

/* Elementos "span" contenidos en .normal */
{ font-weight: bold; }
</style>
```



```
</head>

<body>

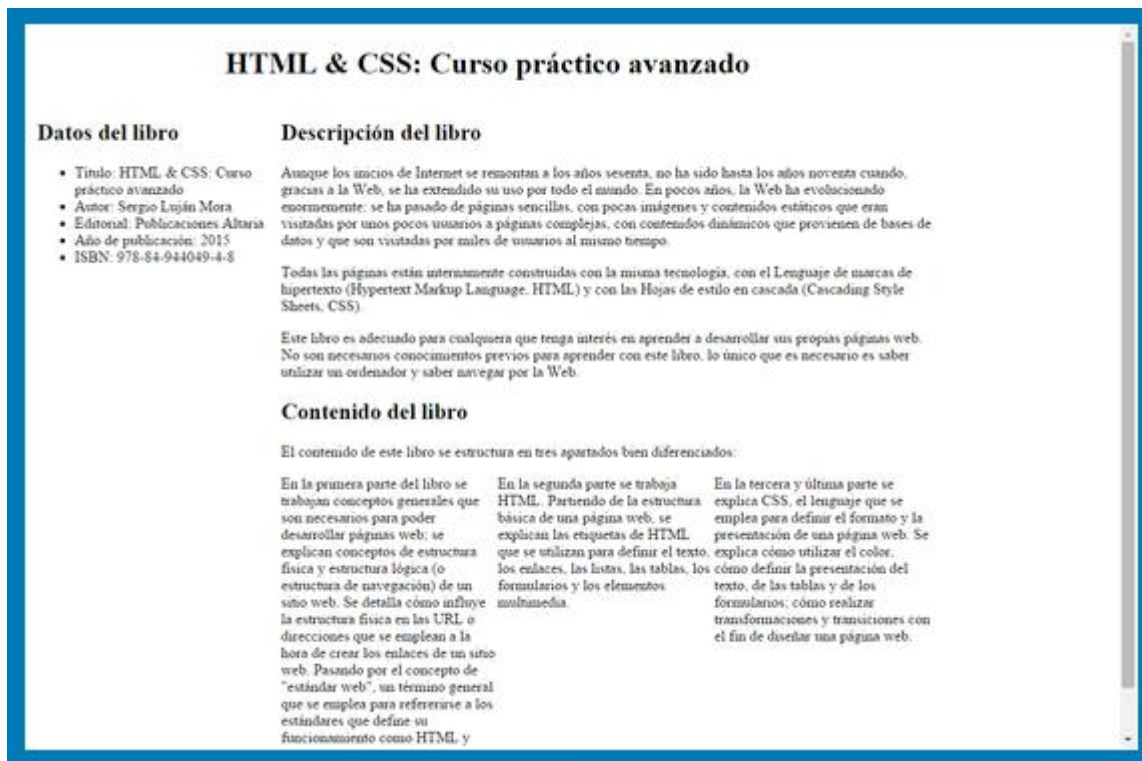
<div id="primero">
<p>Lorem ipsum dolor sit amet, <a href="#">consectetuer adipiscing elit</a>.
Praesent blandit nibh at felis. Sed nec diam in dolor vestibulum aliquet. Duis
ullamcorper, nisi non facilisis molestie, <em>lorem sem aliquam nulla</em>, id
lacinia velit mi vestibulum enim.</p>
</div>

<div class="normal">
<p>Phasellus eu velit sed lorem sodales egestas. Ut feugiat. <span><a
href="#">Donec porttitor</a>, magna eu varius luctus,</span> metus massa tristique
massa, in imperdiet est velit vel magna. Phasellus erat. Duis risus. <a
href="#">Maecenas dictum</a>, nibh vitae pellentesque auctor, tellus velit
consectetuer tellus, tempor pretium felis tellus at metus.</p>
<p>Cum sociis natoque <em class="especial">penatibus et magnis</em> dis parturient
montes, nascetur ridiculus mus. Proin aliquam convallis ante. Pellentesque habitant
morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nunc
aliquet. Sed eu metus. Duis justo.</p>
<p>Donec facilisis blandit velit. Vestibulum nisi. Proin volutpat, <em
class="especial">enim id iaculis congue</em>, orci justo ultrices tortor, <a
href="#">quis lacinia eros libero in eros</a>. Sed malesuada dui vel quam. Integer
at eros.</p>
</div>

</body>
</html>
```

Ejercicio 5

A partir de la página web que se te proporciona, debes escribir las reglas CSS necesarias para lograr una página web que tenga el mismo aspecto que la siguiente imagen:



Las características del diseño de esta página son:

- Un diseño fijo centrado con un ancho de 960px.
- El encabezado de nivel 1 centrado.
- El contenido principal de la página se organiza en dos columnas, una de 260px de ancho y la otra el espacio restante de 700px.
- La lista que explica el contenido del libro se muestra como un texto a tres columnas del mismo ancho.

Puedes modificar el código HTML proporcionado para añadir los identificadores y clases que necesites. También puedes añadir etiquetas `<div>` para definir elementos contenedores en la página.

Código base

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
```

```
<title>HTML & CSS: Curso práctico avanzado</title>
</head>
<body>
<h1>HTML & CSS: Curso práctico avanzado</h1>

<h2>Datos del libro</h2>
<ul>
<li>Título: HTML & CSS: Curso práctico avanzado</li>
<li>Autor: Sergio Luján Mora</li>
<li>Editorial: Publicaciones Altaria</li>
<li>Año de publicación: 2015</li>
<li>ISBN: 978-84-944049-4-8</li>
</ul>

<h2>Descripción del libro</h2>
<p>
Aunque los inicios de Internet se remontan a los años sesenta, no ha sido hasta los
años noventa cuando, gracias a la Web, se ha extendido su uso por todo el mundo. En
pocos años, la Web ha evolucionado enormemente: se ha pasado de páginas sencillas,
con pocas imágenes y contenidos estáticos que eran visitadas por unos pocos
usuarios a páginas complejas, con contenidos dinámicos que provienen de bases de
datos y que son visitadas por miles de usuarios al mismo tiempo.
</p>
<p>
Todas las páginas están internamente construidas con la misma tecnología, con el
Lenguaje de marcas de hipertexto (Hypertext Markup Language, HTML) y con las Hojas
de estilo en cascada (Cascading Style Sheets, CSS).
</p>
<p>
Este libro es adecuado para cualquiera que tenga interés en aprender a desarrollar
sus propias páginas web. No son necesarios conocimientos previos para aprender con
este libro, lo único que es necesario es saber utilizar un ordenador y saber
navegar por la Web.
</p>

<h2>Contenido del libro</h2>
```

<p>

El contenido de este libro se estructura en tres apartados bien diferenciados:

</p>

En la primera parte del libro se trabajan conceptos generales que son necesarios para poder desarrollar páginas web; se explican conceptos de estructura física y estructura lógica (o estructura de navegación) de un sitio web. Se detalla cómo influye la estructura física en las URL o direcciones que se emplean a la hora de crear los enlaces de un sitio web. Pasando por el concepto de "estándar web", un término general que se emplea para referirse a los estándares que define su funcionamiento como HTML y CSS, empleados para el desarrollo de las páginas web en el lado del cliente.

En la segunda parte se trabaja HTML. Partiendo de la estructura básica de una página web, se explican las etiquetas de HTML que se utilizan para definir el texto, los enlaces, las listas, las tablas, los formularios y los elementos multimedia.

En la tercera y última parte se explica CSS, el lenguaje que se emplea para definir el formato y la presentación de una página web. Se explica cómo utilizar el color, cómo definir la presentación del texto, de las tablas y de los formularios; cómo realizar transformaciones y transiciones con el fin de diseñar una página web.

</body>

</html>

Capítulo 3

Reto

En el portal miriadax.net matricularse en el curso:



Desarrollo Frontend con HTML, CSS y Javascript

0,00€

Este curso ha sido concebido como una introducción al diseño de aplicaciones Web en HTML5, CSS y JavaScript, tanto de tipo Web genérica, como para Android e iOS. Sus contenidos tienen como objetivo capacitar para la programación y el diseño de aplicaciones, utilizando HTML5, CSS y JavaScript en estos entornos.

Impartido por:



UNIVERSIDAD
POLITÉCNICA
DE MADRID