**Ejercitación 1 - Responder el siguiente cuestionario**

**1. Qué es HTML, cuando fue creado, cuáles fueron las distintas versiones y cuál es la última?**

El HTML es un lenguaje entendido universalmente que sirve para publicar y distribuir información globalmente. Es el código que se utiliza para estructurar y desplegar una página web y sus contenidos. Consiste en una serie de elementos que se usan para “encerrar” diferentes partes del contenido con el fin de que se vean o comporten de una manera determinada. Permite:

* Publicar documentos en línea con encabezados, textos, tablas, listas, fotos, etc.
* Obtener información en línea a través de vínculos de hipertexto, haciendo clic con el botón de un ratón.
* Diseñar formularios para realizar transacciones con servicios remotos, para buscar información, hacer reservas, pedir productos, etc.
* Incluir hojas de cálculo, videoclips, sonidos, y otras aplicaciones directamente en sus documentos.

Fue originalmente desarrollado por Tim Berners-Lee mientras estaba en el CERN, y fue popularizado por el navegador Mosaic desarrollado en el NCSA. Con el paso del tiempo, el HTML se desarrolló de diferentes maneras:

El HTML 2.0 se desarrolló bajo los auspicios de la Internet Engineering Task Force para codificar lo que era la práctica común a finales de 1994. HTML+ y HTML 3.0 propusieron versiones mucho más ricas de HTML y, a pesar de no haber logrado nunca el consenso en las discusiones sobre estándares, llevaron a la adopción de nuevas características. Los esfuerzos del grupo de trabajo HTML del World Wide Web Consortium para codificar la práctica común en 1996 condujeron a HTML 3.2.

Cada versión del HTML ha intentado reflejar un consenso cada vez mayor entre los interlocutores de la industria.

El HTML 4 desarrolla el lenguaje HTML con mecanismos para hojas de estilo, ejecución de scripts, marcos, objetos incluidos, soporte mejorado para texto de derecha a izquierda y direcciones mezcladas, tablas más ricas y mejoras en formularios, ofreciendo mejoras de accesibilidad para personas con discapacidades.

El HTML 4.01 es una revisión de HTML 4.0 que corrige errores e introduce algunos cambios.

La última versión del HTML es el HTML 5, que mejora varios aspectos de HTML 4.01 que estaban obsoletos y eran difíciles de trabajar. Por ejemplo, el HTML 5 puede realizar cosas de forma nativa, siendo independiente de sistemas que hace tiempo están en desuso.

**2. ¿Cuáles son los principios básicos que el W3C recomienda seguir para la creación de documentos con HTML?**

* Principio 1: Perceptible. la información y los componentes de la interfaz de usuario deben ser mostrados a los usuarios en formas que ellos puedan entender.
* Principio 2: Operable. Los componentes de la interfaz de usuario y la navegación debe ser manejable.
* Principio 3: Comprensible. La información y las operaciones de los usuarios deben ser comprensibles.
* Principio 4: Robustez. El contenido debe ser suficientemente robusto para que pueda ser bien interpretado por una gran variedad de agentes de usuario, incluyendo tecnologías de asistencia.

**3. En las Especificaciones de HTML, ¿cuándo un elemento o atributo se considera desaprobado? ¿y obsoleto?**

Desaprobado: Un elemento o atributo desaprobado es aquel que ha quedado anticuado por la presencia de estructuras nuevas. Los elementos desaprobados se definen en el manual de referencia en los lugares apropiados, pero claramente marcados como desaprobados. Los elementos desaprobados pueden declararse obsoletos en versiones futuras de HTML.

Obsoleto: Un elemento o atributo obsoleto es aquél para el cual no hay garantía de soporte por parte de un agente de usuario. Los elementos obsoletos han dejado de estar definidos en la especificación, pero se enumeran por motivos históricos en la [sección de cambios](http://html.conclase.net/w3c/html401-es/appendix/changes.html) del manual de referencia.

**4. ¿Qué es el DTD y cuáles son los posibles DTDs contemplados en la especificación de HTML 4.01?**

Un documento HTML válido declara qué versión de HTML se utiliza en el documento. La *declaración del tipo de documento* especifica la definición del tipo de documento (DTD) que se usa en el documento (ver [[ISO8879]](http://html.conclase.net/w3c/html401-es/references.html#ref-ISO8879)).

HTML 4.01 especifica tres DTDs, de modo que los autores deben incluir una de las siguientes declaraciones del tipo de documento en sus documentos. Los DTDs varían en cuanto a los elementos que soportan.

**El** [**DTD HTML 4.01 Estricto**](http://html.conclase.net/w3c/html401-es/sgml/dtd.html) **(Strict DTD)** incluye todos los elementos y atributos que no han sido [desaprobados](http://html.conclase.net/w3c/html401-es/conform.html#deprecated) o que no aparecen en documentos con marcos. Para los documentos que usen este DTD, utilice esta declaración del tipo de documento:



**El** [**DTD HTML 4.01 Transicional**](http://html.conclase.net/w3c/html401-es/sgml/loosedtd.html) **(Transitional DTD)** incluye todo lo que incluye el DTD estricto más los elementos y atributos desaprobados (la mayoría de los cuales están relacionados con la presentación visual). Para los documentos que usen este DTD, utilice esta declaración del tipo de documento:



**El** [**DTD HTML 4.01 para Documentos con Marcos**](http://html.conclase.net/w3c/html401-es/sgml/framesetdtd.html) **(Frameset DTD)** incluye todo lo que incluye el DTD Transicional más los marcos. Para los documentos que usen este DTD, utilice esta declaración del tipo de documento:



**5. ¿Qué son los metadatos y cómo se especifican en HTML?**

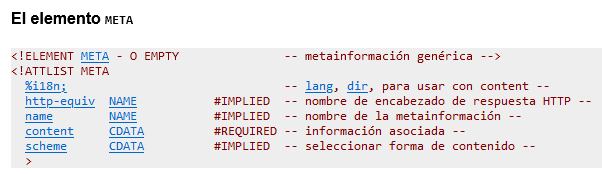
Los metadatos suelen describirse como "datos sobre datos". En otras palabras, si el HTML de su página web son datos, los metadatos son información adicional que se utilizan para explicar varias cosas sobre ese HTML, como:

* La versión de HTML que utiliza una página web.
* El título de una página web.
* Una imagen o un texto que resuma el contenido de la página web.

En general, la especificación de metadatos implica dos pasos:

1. Declaración de una propiedad y de un valor para esta propiedad. Esto puede hacerse de dos maneras:
   1. Desde dentro de un documento, por medio del elemento [META](http://html.conclase.net/w3c/html401-es/struct/global.html#edef-META).
   2. Desde fuera de un documento, vinculando los metadatos por medio del elemento [LINK](http://html.conclase.net/w3c/html401-es/struct/links.html#edef-LINK) (véase la sección sobre [tipos de vínculos](http://html.conclase.net/w3c/html401-es/types.html#type-links)).
2. Referencia a un [perfil](http://html.conclase.net/w3c/html401-es/struct/global.html#profiles) en el que se definen la propiedad y sus valores legales. Para designar un perfil, se usa el atributo [profile](http://html.conclase.net/w3c/html401-es/struct/global.html#adef-profile) del elemento [HEAD](http://html.conclase.net/w3c/html401-es/struct/global.html#edef-HEAD).

Obsérvese que al estar definido un perfil por el elemento [HEAD](http://html.conclase.net/w3c/html401-es/struct/global.html#edef-HEAD), se aplica el mismo perfil a todos los elementos [META](http://html.conclase.net/w3c/html401-es/struct/global.html#edef-META) y [LINK](http://html.conclase.net/w3c/html401-es/struct/links.html#edef-LINK) de la cabecera del documento.



**Ejercitación 2**

**Analizar los siguientes segmentos de código indicando en qué sección del documento HTML se colocan, cuál es el efecto que producen y señalar cada uno de los elementos, etiquetas, y atributos (nombre y valor), aclarando si es obligatorio.**

2.a)

<!-- Código controlado el día 12/08/2009 →

No se coloca en ninguna parte del documento. Sirve para comentar el código. No es obligatorio, pero puede facilitar el entendimiento de las razones y significados de distintas partes del código. No es obligatorio.

2.b)

<div id="bloque1">Contenido del bloque1</div>

Se puede colocar en cualquier lado, lo que hace es hacer una división lógica del contenido de la página para agrupar dentro de ella elementos. Se pueden usar para centrar bloques de contenidos, crear efectos de columna, áreas de color, etc.

En este caso el id del mismo es “bloque1” y muestra el mensaje “Contenido del bloque1”. Son casi obligatorios porque son muy comunes y se utilizan en la mayoría de casos para mostrar o agrupar cosas.

2.c)

<img src="" alt="lugar imagen" id="im1" name="im1" width="32" height="32"

longdesc="detalles.htm" />

Este elemento (img) representa una imagen que puede ser ubicada en el cuerpo de un documento. Para que se despliegue correctamente, en el atributo “src” se especifica la ubicación de la imagen. El atributo “alt”, por otra parte, sirve para que se muestre una versión de texto alternativa de la imagen en caso de que no pueda desplegarse; en este caso, si hay algún problema, se mostrará “lugar imagen”.

El tercer atributo es el id, que es simplemente un identificador para el elemento y debe ser único. Puede utilizarse para referirse al elemento con vínculos, scripts, definiciones de estilo, etc.

El atributo que sigue es el name, que sirve como nombre para el elemento, pero se ha vuelto inválido en HTML5 por tener propósitos únicamente visuales.

Los atributos de width y height sirven para especificar el tamaño de la imagen. El width especifica el número de píxeles que indica el ancho de la imagen, y el height los píxeles que indican su altura.

El último atributo presente es el longdesc, que es un URI de un recurso con una explicación más extensa del propósito de la imagen que la que se provee en el alt. Actualmente es obsoleto.

2.d)

<meta name="keywords" lang="es" content="casa, compra, venta, alquiler " />

<meta http-equiv="expires" content="16-Sep-2019 7:49 PM" />

Son etiquetas que se incorporan en los encabezados de la página web, las personas que entran al sitio no las pueden ver, pero son de gran utilidad para navegadores u otros programas que puedan aprovechar o necesitar dicha información.

Se encuentran dentro de la cabecera que se declara como <head>

**Elementos, etiquetas y atributos:**

name: Especifica el nombre del metadato. En este caso, se establece como "keywords".

lang: Indica el idioma del contenido del metadato. Está establecido como "es" para español.

content: Define el valor del metadato. En este caso, contiene una lista de palabras clave: "casa", "compra", "venta", "alquiler".

http-equiv: Especifica que el elemento debe actuar como un encabezado HTTP. En este caso, se establece como "expires".

content: Define el valor del encabezado HTTP especificado por http-equiv. En este caso, se establece una fecha y hora de vencimiento: "16-Sep-2019 7:49 PM".

Son imprescindibles para que los motores de búsqueda o navegadores interpreten experimenten el contenido web de la manera en que los creadores de ese contenido pretenden

2.e)

<a href="http://www.e-style.com.ar/resumen.html" type="text/html" hreflang="es" charset="utf-8"

rel="help">Resumen HTML </a>

Esta etiqueta es una etiqueta de “anclaje”, incluye una etiqueta de apertura, cualquier atributo de etiqueta, el texto de anclaje u objeto una etiqueta de cierre. Se usa principalmente para crear hipervínculos

href: Especifica la URL a la que el hipervínculo apunta. En este caso, la URL es "http://www.e-style.com.ar/resumen.html".

type: Indica el tipo de contenido del recurso enlazado. Se establece como "text/html".

hreflang: Indica el idioma del recurso vinculado. Está establecido como "es" para español.

charset: Especifica el conjunto de caracteres del recurso enlazado. Se establece como "utf-8".

rel: Define la relación entre el documento actual y el recurso enlazado. En este caso, se establece como "help", lo que sugiere que el enlace proporciona ayuda o información adicional.

En este caso el texto visible del enlace es “Resumen HTML” que si la gente le da click a este enlace van a ser redirigidos a la URL que está contenida en el href.

2.f)

<table width="200" summary="Datos correspondientes al ejercicio vencido">

<caption align="top"> Título </caption>

<tr>

<th scope="col">&nbsp;</th>

<th scope="col">A</th>

<th scope="col">B</th>

<th scope="col">C</th>

</tr>

<tr>

<th scope="row">1º</th>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<th scope="row">2º</th>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

</table>

**Table:**

El elemento “table” representa una tabla. Es decir, un grupo de filas que contiene, y sus respectivas columnas. El atributo width especifica el ancho deseado para toda la tabla, y puede ser en pixeles o en un porcentaje relativo al espacio horizontal disponible. Es obsoleto en HTML5 debido a su naturaleza visual.

El atributo summary es un resumen del propósito de la tabla, pero también se volvió obsoleto debido al elemento <caption>.

**caption:**

Este elemento es el título de la tabla en la que es declarado. Se muestra arriba de la tabla.

El atributo “align” es para ubicar el título en relación a la tabla. En este caso está encima de la tabla, pero también puede estar debajo (bottom), a la izquierda (left) o a la derecha (right). Sin embargo, este atributo es obsoleto en HTML5.

**tr:**

Representa una fila de la tabla. Todo lo que contenga será parte de esta fila.

Es importante para representar la información de la tabla.

**th:**

Representa el encabezado de una celda en una tabla. El atributo scope es para aclarar el grupo de celdas para las cuales se brinda información de encabezado. Puede ser para todas las celdas de la misma fila (row), todas las celdas en la misma columna (col), las celdas en el mismo grupo de filas (rowgroup), las celdas en el mismo grupo de columnas (colgroup), o los navegadores pueden decidir las celdas afectadas (auto).

Si bien no es obligatorio, ayuda a comprender el significado de la tabla.

**td:**

Representa una celda de datos de una tabla. Se encuentra dentro de un elemento tr. Es importante para representar la información de la tabla.

**Ejercitación 3**

**En cada caso, explicar las diferencias entre los segmentos de código y sus visualizaciones:**

3.a)

<a href="http://www.google.com.ar">Click aquí para ir a Google</a>

En este segmento, simplemente se abre google chrome sobre la misma página. Es decir, el navegador redirecciona a la página de google en la misma pestaña.

<a href="http://www.google.com.ar" target="\_blank">Click aquí para ir a Google</a>

En este caso, al tener el target= blank se abre google en una pestaña nueva.

<a href="http://www. google.com.ar" type="text/html" hreflang="es" charset="utf-8" rel="help">

El tercer caso, al tener un espacio, rompe el link y no sirve. Además, los atributos de type, hreflang, charset, y rel, no corresponden a un elemento del tipo “a”.

<a href="#">Click aquí para ir a Google</a>

En esta ocasión, se recarga la página sobre la misma pestaña.

<a href="#arriba">Click aquí para volver arriba</a>

<a name="arriba" id="arriba"></a>

Por último, esto hará que la página vaya al elemento cuyo id es igual a “arriba” (dentro de esta misma página).

3.b)

<p><img src="im1.jpg" alt="imagen1" /><a href="http://www.google.com.ar">Click aquí</a></p>

En el primer caso, tenemos un párrafo compuesto por una imagen seguida de un enlace. Al hacer click en “Click aquí”, nos redirigirá a google sobre la misma pestaña.

<p><a href="http://www.google.com.ar"><img src="im1.jpg" alt="imagen1" /></a> Click aquí</p>

En este otro caso, el párrafo también tiene la imagen seguida del “Click aquí”. Sin embargo, al ser el elemento img el contenido en el elemento a, y al no estar contenido el “Click aquí”, es la imagen la que redirigirá a google si la presionamos. El click aquí es, de este modo, simplemente visual, y no redirige a ningún lado si es clickeado.

<p><a href="http://www.google.com.ar"><img src="im1.jpg" alt="ima

gen1" />Click aquí</a></p>

Esta situación mantiene el mismo formato visual que las dos anteriores, con la diferencia que el alt de la imagen y el “Click aquí forman parte del mismo elemento y no están separados. De este modo, la imagen y el texto juntos constituirán el enlace.

<p><a href="http://www.google.com.ar"><img src="im1.jpg" alt="imagen1" /></a> <a

href="http://www.google.com.ar">Click aquí</a></p>

En este último caso, la imagen y el texto son elementos “a” distintos, por lo que constituirán diferentes enlaces (aunque ambos redirigen a google). Cualquiera de los dos que presionemos redirige a la pestaña deseada.

3.c)

| <ul>  <li>xxx</li>  <li>yyy</li>  <li>zzz</li>  </ul> | <ol>  <li>xxx</li>  <li>yyy</li>  <li>zzz</li>  </ol> | <ol>  <li>xxx</li>  </ol>  <ol>  <li value="2">yyy</li>  </ol>  <ol>  <li  value="3">zzz</li>  </ol> | <blockquote>  <p>1. xxx<br />  2. yyy<br />  3. zzz</p>  </blockquote> |
| --- | --- | --- | --- |
| Se trata de una lista no ordenada con los textos “xxx”, “yyy”, y “zzz”. Al ser una lista no ordenada, los elementos se presentan con puntos negros | En este caso, la lista es ordenada, razón por la cual se presentan los elementos con números que los ordenan. | Este otro caso no es una sóla lista, sino que crea una lista ordenada para cada elemento que quiere mostrar. Esto genera una mayor separación entre cada elemento. Además, usa el atributo value para inicializar la lista ordenada, pues si no estuviera, todos los elementos tendrian el “1”. | En este último caso, si bien no es una lista en sí, se utiliza una cita de texto que contiene un párrafo. El efecto visual de la lista lo generan con elementos de párrafo (br /). |

3.d)

| <table border="1" width="300">  <tr>  <th>Columna 1</th>  <th>Columna 2</th>  </tr>  <tr>  <td>Celda 1</td>  <td>Celda 2</td>  </tr>  <tr>  <td>Celda 3</td>  <td>Celda 4</td>  </tr>  </table> | <table border="1" width="300">  <tr>  <td><div align="center"><strong>Colum  na1</strong></div></td>  <td><div align="center"><strong>Columna  2</strong></div></td>  </tr>  <tr>  <td>Celda 1</td>  <td>Celda 2</td>  </tr>  <tr>  <td>Celda 3</td>  <td>Celda 4</td>  </tr>  </table> |
| --- | --- |
| En este caso, se presenta una tabla con bordes y un ancho de 300. La primera fila consiste en encabezados de tabla que muestran “Columna 1” y “Columna 2”, dándoles enfasis. La segunda fila ya no es un encabezado, sino que son celdas de datos, que muestran la información. Esto mismo pasa para la segunda fila. | Si bien, visualmente, este otro caso es igual al anterior, la diferencia recae en que la primera fila no utiliza los elementos th para los encabezados de la tabla, sino que recurre a utilizar un elemento td para la celda de datos, y genera un div (que alinea en el centro de la columna) y un strong para darle el mismo formato que a un encabezado. En cuanto a legibilidad y conveniencia, es preferible el otro caso. |

3.e)

| <table width="200">  <caption>  Título  </caption>  <tr>  <td bgcolor="#dddddd">&nbsp;</td>  <td bgcolor="#dddddd">&nbsp;</td>  <td bgcolor="#dddddd">&nbsp;</td>  </tr>  <tr>  <td bgcolor="#dddddd">&nbsp;</td>  <td bgcolor="#dddddd">  &nbsp;</td>  <td bgcolor="#dddddd">&nbsp;</td>  </tr>  </table> | <table width="200">  <tr>  <td colspan="3">  <div align="center">Título</div>  </td>  </tr>  <tr>  <td bgcolor="#dddddd">&nbsp;</td>  <td bgcolor="#dddddd">&nbsp;</td>  <td bgcolor="#dddddd">&nbsp;</td>  </tr>  <tr>  <td bgcolor="#dddddd">&nbsp;</td>  <td bgcolor="#dddddd">&nbsp;</td>  <td bgcolor="#dddddd">&nbsp;</td>  </tr>  </table> |
| --- | --- |

Al igual que en el ejercicio 3.d, no hay diferencia visual entre los dos casos, y la diferencia, que se presenta solo en el título de la tabla, está en la legibilidad y conveniencia. En el primer caso, se hace uso del elemento caption para representar el título, y luego se recurre a la declaración de filas y de las celdas de datos; en el segundo, se simula y “fuerza” el formato del título mediante la declaración de una fila que combina 3 columnas y un div que sirve para centrarlo. Es más conveniente el uso del caption, cuya función es específicamente cumplir con el formato del título de la tabla.

3.f)

| <table width="200">  <tr>  <td colspan="3"><div  align="center">Título</div></td>  </tr>  <tr>  <td rowspan="2" bgcolor="#dddddd">&nbsp;</td>  <td bgcolor="#dddddd">&nbsp;</td>  <td bgcolor="#dddddd">&nbsp;</td>  </tr>  <tr>  <td bgcolor="#dddddd">&nbsp;</td>  <td bgcolor="#dddddd">&nbsp;</td>  </tr>  </table> | <table width="200">  <tr>  <td colspan="3"><div  align="center">Título</div></td>  </tr>  <tr>  <td colspan="2"  bgcolor="#dddddd">&nbsp;</td>  <td bgcolor="#dddddd">&nbsp;</td>  </tr>  <tr>  <td bgcolor="#dddddd">&nbsp;</td>  <td bgcolor="#dddddd">&nbsp;</td>  <td bgcolor="#dddddd">&nbsp;</td>  </tr>  </table> |
| --- | --- |
| En este caso, se presenta una tabla de un ancho de 200px con un título que se conforma con la combinación de tres columnas (colspan=3) de la primera fila y se centra.  Seguido de esto, aparece combinada la primera columna de la tabla, conformada por dos filas debido al atributo rowspan=2.  A la derecha de esta celda combinada, aparecen visibles y explícitas las dos filas con dos columnas más cada una. | Esta tabla es similar a la del caso anterior, con un título centrado conformado por 3 celdas, y un ancho de 200.  La diferencia principal recae en la segunda fila, cuya primera celda de datos presenta un atributo colspan=2, en vez de rowspan=2. Debido a esto, en vez de tener una columna formada por dos filas, tenemos una fila formada por dos columnas. A su derecha está la celda restante (hay tres celdas de datos por fila), y abajo aparece la última fila con sus tres celdas de datos explícitas. |

3.g)

| <table width="200" border="1">  <tr>  <td colspan="3"><div  align="center">Título</div></td>  </tr>  <tr>  <td colspan="2"rowspan="2">&nbsp;</td>  <td>&nbsp;</td>  </tr>  <tr>  <td width="50%">&nbsp;</td>  </tr>  </table> | <table width="200" border="1" cellpadding="0"  cellspacing="0">  <tr>  <td colspan="2"><div  align="center">Título</div></td>  </tr>  <tr>  <td rowspan="2">&nbsp;</td>  <td>&nbsp;</td>  </tr>  <tr>  <td width="50%">&nbsp;</td>  </tr>  </table> |
| --- | --- |
| El primer td del título tiene colspan 3, pero no cambia nada porque tenés 2 columnas, así que no se va a extender más  Tiene el título centrado.  Luego contiene un segundo td el cual se extiende sobre dos columnas y 2 filas.  Hay una celda que ocupa el 50% del ancho total de tabla, pero es parte de la columna que se divide en 2 partes. | Los atributos de “cellpadding” y de “cellspacing” indican que no hay espacios adicionales dentro de cada cuadrado de la tabla ni entre cuadrados.  El primer td del título tiene un colspan de 2 el cual se extiende sobre las 2 columnas.  Tiene el título centrado.  El segundo tr contiene un td que que se extiende con un rowspan de 2 sobre las 2 filas.  Hay una celda que ocupa el 50% del ancho total de tabla, pero esta es independiente. |

3.h)

| <form id="form1" name="form1" action="procesar.php" method="post" target="\_blank">  <fieldset>  <legend>LOGIN</legend>  Usuario: <input type="text" id="usu1" name="usu1" value="xxx" /><br />  Clave: <input type="password" id="clave1" name="clave1" value="xxx" />  </fieldset>  <input type="submit" id="boton1" name="boton1" value="Enviar" />  </form> | <form id="form2" name="form2" action="" method="get" target="\_blank">  LOGIN<br />  <label>Usuario: <input type="text" id="usu2" name="usu2" /></label><br />  <label>Clave: <input type="text" id="clave2" name="clave2" /></label><br />  <input type="submit" id="boton2" name="boton2" value="Enviar" />  </form> | <form id="form3" name="form3" action="mailto:xx@xx.com” enctype=text/plain method="p  ost" target="\_blank">  <fieldset>  <legend>LOGIN</legend>  Usuario: <input type="text" id="usu3" name="usu3" /><br />  Clave: <input type="password" id="clave3" name="clave3" />  </fieldset>  <input type="reset" id="boton3" name="boton3" value=  "Enviar" />  </form> |
| --- | --- | --- |
| En este primer formulario, los datos se envían a “procesar.php”mediante un post. Contiene los campos dentro de un recuadro debido al elemento fieldset. A su vez, los valores de los campos del usuario y la clave ya aparecen predeterminados debido al atributo value. Una vez presionado el botón enviar, el resultado se muestra en otra página debido al target: \_blank | La primera diferencia con el formulario anterior es que los datos se envían a la página actual debido a que el atributo action está vacío y que el método ahora no será un post, sino un get.  Este formulario no tiene un fieldset, así que los campos no están recuadrados. Al tampoco asignar un value, al cargar la página los campos están vacíos. Una vez presionado el botón, el target:\_blank muestra el resultado en otra página. | Este último formulario envía los datos al mail especificado con un método post. Al igual que el primer caso, posee un fieldset que hace que los campos estén recuadrados, pero, como en el segundo caso, no hay ningún value asignado, por lo que al cargar la página, los campos están vacíos.  Como en todos los casos, el target: \_blank hace que al presionar el botón de envíar, los resultados se muestren en una nueva página. |

3.i)

| <label>Botón 1  <button type="button" name="boton1" id="boton1">  <img src="logo.jpg" alt="Botón con imagen " width="30" height="20" /><br />  <b>CLICK AQUÍ</b></button></label> | <label>Botón 2  <input type="button" name="boton2" id="boton2" value="CLICK AQUÍ" />  </label> |
| --- | --- |
| En este caso, hay un texto de “Botón 1”, que tiene, al lado, un botón. Este botón está conformado por una imagen de un ancho de 30px y una altura de 20px, y un texto en negrita (elemento b) con un salto de línea (br /). | A diferencia del primer caso, el botón ahora no está conformado por una imagen y un texto en negrita, sino por sólo un texto titulado CLICK AQUI.  Ambos botones tienen etiquetas de texto asociadas, y ninguno realiza ninguna acción. |

3.j)

| <p><label><input type="radio" name="opcion" id="X" value="X" />X</label><br />  <label><input type="radio" name="opcion" id="Y" value="Y" />Y</label></p> | <p><label><input type="radio" name="opcion1" id="X" value="X" />X</label><br />  <label><input type="radio" name="opcion2" id="Y" value="Y" />Y</label></p> |
| --- | --- |
| El name del id=’X’ es igual al del id=’Y’ llamandose ambos opcion. Si bien se obtiene el mismo resultado visual esto puede llegar a ser un problema si se quiere utilizar dichas variables en un futuro. | Los nombres son distintos y el correspondiente a X se llama opcion1 y el de y opcion2. |

3.k)

| <select name="lista">  <optgroup label="Caso 1">  <option>Mayo</option>  <option>Junio</option>  </optgroup>  <optgroup label="Caso 2">  <option>Mayo</option>  <option>Junio</option>  </optgroup>  </select> | <select name="lista[]" multiple="multiple">  <optgroup label=" Caso 1">  <option>Mayo</option>  <option>Junio</option>  </optgroup>  <optgroup label=" Caso 2">  <option>Mayo</option>  <option>Junio</option>  </optgroup>  </select> |
| --- | --- |
| En este caso, se presenta una lista similar a una dropdown bar que permite seleccionar una única opción. Contiene dos grupos de opciones titulados “Caso 1” y “Caso 2”, ambos con dos opciones iguales de “Mayo” y “Junio”. | En este otro se presenta una lista la cual te permite navegar a través de ella desplazandote con una barra lateral, la cual te permite gracias al atributo “múltiple” si se mantiene apretado el botón “ctrl” seleccionar múltiples objetos dentro de nuestra lista, la cual está declarada como una lista |