CICLO ESCOLAR 2020-2021

COLEGIO DE ESTUDIOS CIENTÍFICOS Y TECNOLÓGICOS DEL ESTADO DE CAMPECHE

PLANTEL:

SEMESTRE: FEBRERO - JULIO 2021

CECyTE Campeche

CUADERNILLO DE TRABAJO DEL MÓDULO I SUBMÓDULO 1

NOMBRE DEL ALUMNO:

_____

SEDUC

SEGUNDO PARCIAL

APRENDIZAJE ESPERADO 1

Competencia Profesional:
- Emplea html para construir páginas web

Situación:
- Con html

Conceptos fundamentales:
1.4 Identificar conceptos básicos
1.5 Aplicar edición y formato de página

Conceptos Subsidiarios:
1.4.1. Características y tipos de web´s estáticas
1.4.2. HTML
1.5.1.1 Formato básico de una página web

Producto Esperado:
- Resumen

Instrumento de evaluación:
- Rúbrica 1

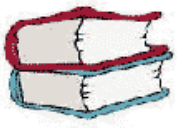
Ponderación:
- 40%

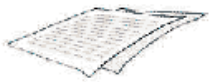
Producto: Resumen

¿Qué es un resumen?

El resumen es la exposición breve de lo esencial de un tema o materia, tanto efectuada de manera oral como escrita. Consiste en reducir o sintetizar el contenido de una lectura, texto, documento. El resumen es un texto que contiene la información más importante de un texto más largo. Su función principal es informar y para esto se sintetizan las ideas principales y se quita todo aquello que sobra

texto base → resumen





Características y función del resumen

- *El resumen contiene todas las ideas básicas del texto original*
- *Presenta ideas relacionadas por enlaces como conjunciones, preposiciones o signos*
- *Se escribe con estilo propio sin influencias del estilo del autor original*
- *Es un texto más breve que el original*
- *Respeta las normas básicas de ortografía y redacción*

Pasos para hacer un resumen

1.- Lee con atención el texto, de preferencia dos o más veces para que detectes las ideas principales. Busca el significado de los términos que no conozcas. Recuerda que la comprensión completa del texto es fundamental para hacer un buen resumen.

2.- Identifica las ideas principales del texto. Puedes subrayarlas, hacer una esquema o tomar notas en una libreta.

3.- Redacta el resumen construyendo una secuencia de ideas generales y omite todo lo que no sea importante.

4.- Adecúa el lenguaje usado para expresar fielmente las ideas del autor original

5.- Corrige el texto para que tenga coherencia. Asegúrate que contenga todas las ideas importantes y que se conserve el estilo.

- **REALIZA UN RESUMEN DE HTML PARA CONSTRUIR PÁGINAS WEB**

1. EMPLEA HTML PARA CONSTRUIR PÁGINAS WEB
1.4.- IDENTIFICAR CONCEPTOS BÁSICOS
1.4.1. CARACTERÍSTICAS Y TIPOS DE WEB´S ESTÁTICAS

Los sitios **Web estáticos** son aquellos sitios enfocados principalmente a mostrar una información permanente, donde el navegante se limita a obtener dicha información, sin que pueda interactuar con la página Web visitada, las Web estáticas están construidas principalmente con hipervínculos o enlaces (links) entre las páginas Web que conforman el sitio, este tipo de Web son incapaces de soportar aplicaciones Web como gestores de bases de datos, foros, consultas on line, e-mails inteligentes...

Esta es una opción más que suficiente para aquellos sitios Web que simplemente ofrecen una descripción de su empresa, quiénes somos, donde estamos, servicios, etc... ideal para empresas que no quieren muchas pretensiones con su sitio Web, simplemente informar a sus clientes de sus productos y su perfil de empresa.

La principal ventaja de este tipo de páginas es lo económico que resulta crearlas, con un diseño vistoso e incluyendo las imágenes y el texto con el cual queremos informar a los navegantes, se puede crear fácilmente sin necesidad de ningún tipo de programación especial (php, asp, ...).

La gran desventaja de los sitios Web estáticos reside en lo laborioso que resulta su actualización así como la pérdida de potentes herramientas soportadas con bases de datos, como pueden ser la creación de registros históricos de los clientes, pedidos on-line,...

Características de las páginas web estática

Una página web estática es un **documento adaptado a la web que proporciona un contenido fijo a todos los usuarios**, es decir, es un sitio de internet que no modifica su contenido para personalizarlo y adaptarlo a cada interacción. Por tanto, son páginas que **no se actualizan constantemente**. También se conocen como páginas planas.

Funciones de las páginas web estáticas

La principal función de una página web estática es **presentar una información estable** que no es necesario modificar con frecuencia. Por poner un ejemplo claro, se pueden crear páginas web estáticas para informar sobre la dirección de un negocio, para **detallar los servicios fijos** que ofrece, para contar la historia y la trayectoria de la empresa, etc.

Tipos de web estáticas

CURRICULUM VITAE.

Cada día nos encontramos con más personas que utilizan la Web para presentar su curriculum vitae, es una buena manera de presentarse hacia las empresas a la hora de encontrar trabajo o establecer relaciones comerciales.

WEB QUEST.

Son sitios Web desarrollados principalmente por profesores y universidades, con un claro objetivo educativo, proponiendo a un grupo de alumnos un tema en particular (biología, geometría, física, literatura ....) con un serie de cuestiones formuladas acerca del tema elegido, el grupo tiene disponible, en la misma Web, direcciones (links) a otras páginas Web donde pueden indagar e investigar acerca del tema, de tal forma que puedan responder a las preguntas planteadas. El principal objetivo de las Web Quest reside en aprender los contenidos del tema elegido de una manera interactiva.

BLOG O BITÁCORA.

Es un sitio Web donde se recopilan cronológicamente mensajes de uno o varios autores, sobre una temática en particular siempre conservando el autor la libertad de dejar publicado lo que crea pertinente, también llamado blog o bitácora.

WEB INFORMATIVA.

Aquellos sitios Web que simplemente ofrecen información acerca de un tema específico (Fiestas en Valencia: Las fallas, Gastronomía andaluza.....), por lo general estos sitios no se actualizan, de tal forma que los contenidos permanecen invariables a lo largo de su vida.

Ventajas de las páginas web estáticas

Algunas de las **ventajas** que ofrecen las aplicaciones web estáticas son:

- Funcionan en cualquier servidor.
- Tardan poco en cargar.
- Gran funcionalidad.
- No requieren instalación.
- Mejoran el posicionamiento.
- Se gasta menos en alojamiento.
- No requieren mucho mantenimiento.

Desventajas de las páginas web estáticas

Por otro lado, las **desventajas** de estas aplicaciones web son:

- No tienen muchas funcionalidades.
- El proceso de actualización es complejo.
- El usuario no puede seleccionar o modificar el contenido o el diseño.
- El programador tiene que acceder al servidos donde se aloja la página para realizar cambios.

Extensiones de páginas web

Como hemos mencionado las páginas web son documentos y, como tales, suelen tener una extensión que nos da una idea sobre del tipo de contenido que posee ese documento. Las extensiones más habituales de las páginas web con las que nos podemos encontrar a la hora de navegar son:

- **html, y htm**: Suele ser la extensión estándar para páginas web estáticas, formadas por contenido html.

1.4.2 HTML

El **HTML** es el principal lenguaje de marcado utilizado por programadores y diseñadores web de cualquier nivel de experiencia, siendo la abreviatura de (HyperText Markup Language o **Lenguaje de Marcado de Hipertexto**). Muchas personas al leer la definición técnica piensan en todo el conjunto de elementos que conlleva el diseño de una página web cuando éste solo funciona para indicar el orden del contenido dentro de una página web a través de "tags" de hipertexto.

¿Para qué sirve html?

Básicamente el lenguaje HTML sirve para describir la estructura básica de una página y organizar la forma en que se mostrará su contenido, además de que HTML permite incluir enlaces (links) hacia otras páginas o documentos.

HTML es un lenguaje de marcado descriptivo que se escribe en forma de etiquetas para definir la estructura de una página web y su contenido como texto, imágenes, entre otros, de modo que HTML es el encargado de describir (hasta cierto punto) la apariencia que tendrá la página web.

¿Cómo funciona html?

Utilizar HTML es bastante más sencillo de lo que muchas personas consideran. Comienza abriendo un documento HTML a través de cualquier programa de edición de textos, y al momento de guardar el archivo como una página web es necesario usar la extensión ".html"; así mismo es probable que el programa te realice una advertencia de que se convertirá en un "texto sin formato".
Es importante que tomes en cuenta que no existe una diferencia entre ".html" y ".htm".
El siguiente paso es observar tu creación en un navegador de internet. Al guardar el documento en blanco, lo abres con un navegador –haciendo doble click en el archivo- y debe ingresar como una página en blanco, a medida que edites el documento puedes ver el progreso en la página.
No es necesario tener internet para esto ya que nadie puede ingresar al archivo, es solamente para leer el documento HTML.

Elementos del lenguaje HTML

Se distinguen varios componentes: elementos y atributos.

- **Elementos:** Desarrollan la estructura de las páginas webs. Los elementos h1, h2, h3, h4, h5 y h6 se utilizan para crear encabezados. Por orden jerárquico h1 representa el primer nivel y, normalmente, el texto más grande. h2 es el segundo nivel con un texto algo más pequeño, así hasta llegar al h6 que es el texto más pequeño de todos.

- **Atributos:** Los atributos de las etiquetas HTML definen detalles de comportamiento o presentación de la etiqueta donde van colocados. Se escriben dentro de la propia etiqueta con su nombre de atributo y el valor del atributo entre comillas.

Ventajas y desventajas del lenguaje html

Siguen habiendo versiones nuevas cada cierto tiempo con nuevas funcionalidades. Esto, ya que es un lenguaje muy utilizado por programadores, especialmente, para desarrollar proyectos web. Se pueden encontrar algunas ventajas y desventajas de su uso.

1. **Ventajas:** Permite describir hipertexto, tiene un despliegue rápido, lo reconoce y admite cualquier tipo de explorador y permite archivos pequeños.

2. **Desventajas:** El diseño es más lento, tiene un lenguaje estático y las etiquetas son limitadas.

• REGLAS DE HTML:

Obligatorio:

- Delimitar el ámbito de las etiquetas (salvo excepciones)
- Etiqueta interna se cierra antes que externa
- Etiquetas a nivel de bloque pueden contener etiquetas a nivel de línea, pero no al revés
- Etiquetas no sensitivas a mayúsculas
- En vez de espacios agregar raya al piso

Pero no está de más cumplir además con XHTML:

- Cerrar todos los tags, ie: ... , <li> ...
- Incluir marca de cierre para tags vacíos, ie: <hr />,
- Poner entre comillas todos los atributos, ie:
- Todos los tags en minúsculas
- Terminar con </html>

ETIQUETAS HTML

Las etiquetas HTML son fragmentos de texto rodeados por corchetes angulares cuyo uso es el de escribir código HTML. Estas no aparecerán en la página web como texto, en cambio, son las que le indican al navegador como exponer el contenido y la página.

Se diferencian dos tipos de etiquetas:

1. Etiquetas de apertura. <etiqueta>
2. Etiquetas de cierre. </etiqueta>

Se deben añadir diversas etiquetas que determinan el encabezado, el cuerpo y el contenido de la página web. En el caso del texto, el mismo puede ser agregado en varios estilos.

Escritura de etiquetas

Las normas que debemos seguir para escribir las etiquetas en HTML son:

Apertura y cierre

La mayoría de las instrucciones tienen etiqueta de apertura y etiqueta de cierre. Es necesario poner las dos, y entre éstas está el elemento afectado. Sin embargo en algunos casos la etiqueta de cierre puede no ponerse, aunque es conveniente ponerla para evitar confusiones. Tal es el caso de la etiqueta <p> (parrafo) cuando después de la etiqueta de cierre le sigue otra etiqueta <p>

Podemos escribir: `<p> ... </p><p>... </p>`

o también: `<p> ... <p>... </p>`

Aunque es más que recomendable escribir siempre la primera forma, y poner siempre la etiqueta de cierre. El lenguaje XHTML indica que siempre hay que poner la etiqueta de cierre.

Las instrucciones de etiqueta única (etiquetas que no tienen otra de cierre) se escriben normalmente, si bien en XHTML se pone una barra inclinada al final para indicar que la misma etiqueta tiene la apertura y el cierre. Por ejemplo la etiqueta `<br>` (salto de línea), por su misma naturaleza no tiene etiqueta de cierre, pero en XHTML se escribe `<br/>`, con la barra inclinada al final.

En HTML es indiferente escribir las palabras clave de las etiquetas en mayúsculas, minúsculas, o una combinación de ambas. Podemos escribir por ejemplo `<br>` o `<BR>` o incluso `<bR>`, Si bien en XHTML sólo se admiten las minúsculas.

Los atributos

Como vimos, las etiquetas pueden llevar atributos, en algunas de ellas es obligatorio. El atributo lleva casi siempre un valor, el cual se indica mediante el signo igual (=) seguido de un número, un porcentaje o una palabra clave. Este valor se puede poner o no entre comillas.

Podemos escribir : `<p align="center"> ... </p>`

O podemos escribir : `<p align=center> ... </p>`

sin embargo en XHTML deben ponerse siempre las comillas.

Jerarquía de etiquetas

Lo normal es que entre la etiqueta de apertura y la de cierre, no se incluya otra etiqueta sin que tenga también su etiqueta de cierre. Es decir, que unas etiquetas engloben a otras, no dejando el cierre de una etiqueta secundaria para después de la etiqueta que la engloba. Aunque HTML admite que eso pueda hacerse, es algo que debe evitarse para no dar lugar a confusiones. Por ejemplo:

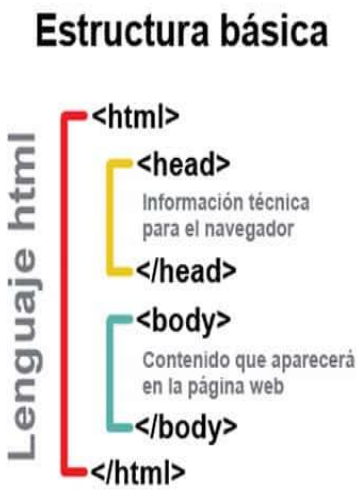
Está bien: `<p><b> ... </b></p>`

Está mal: `<p><b> ... </p></b>`

Aunque HTML admite la segunda apción, en algunos navegadores esto puede dar lugar a errores, y en todo caso XHTML no admite esta opción.

Salida	Etiquetas requeridas
El **texto** en negritas.	El <B>texto</B> en negritas.
El *texto* en itálicas.	El <I>texto</I> en itálicas.
El `texto` en formato teletipo.	El <TT>texto</TT> en formato teletipo.
El texto en 2 tamaños más grandes y en rojo. El tamaño puede ser un valor absoluto en el rango 1..7.	El <FONT SIZE = +2 COLOR = "RED"> texto </FONT> en 2 tamaños más grandes y en rojo. El tamaño puede ser ...
Utilizar la dirección de e-mail *claudio@acm.org* para contactarme.	Utilizar la dirección de e-mail <ADDRESS> claudio@acm.org </ADDRESS> para contactarme.

- ## ESTRUCTURA BÁSICA DE UNA PÁGINA WEB



Para crear un verdadero documento HTML comenzará con tres elementos contenedores:

<html>

<head>

<body>

Estos tres se combinan para describir la estructura básica de la página:

- **<html>**: Este elemento envuelve todo el contenido de la página (excepto la **DTD**)
- **<head>**: Este elemento designa la parte de encabezado del documento. Puede incluir información opcional sobre la página Web, como puede ser el título (el navegador lo muestra en la barra de título), palabras clave de búsqueda opcionales y una hoja de estilo opcional.
- **<body>**: Este elemento alberga el contenido de su página Web, es decir, aquello que queremos que aparezca en el área de navegación del navegador..

Sólo hay una manera correcta de combinar estos tres elementos. He aquí su colocación exacta, con el *doctype* al comienzo de la página:

```
<!DOCTYPE  html>
<html>
    <head>
    ...
    </head>
    <body>
        ...
    </body>
</html>
```

Toda página Web utiliza esta estructura básica. Los puntos suspensivos (...) muestran dónde insertaría la información adicional.

Una vez colocado el esqueleto XHTML, debe añadir dos conectores más a la mezcla.

Toda página Web requiere un elemento **<title>** en la sección del encabezado. A continuación, deberá crear un contenedor para el texto en la sección del cuerpo de texto (<body>).

Un elemento contenedor de texto multiuso es **<p>**, que representa un párrafo. Veamos con más detalle los elementos que hay que agregar:

<title>: Establece el título de la página Web, el cual tiene varias funciones. Primero, los navegadores lo muestran en la parte superior de la ventana. Segundo, cuando un visitante crea un marcador para la página, el navegador emplea el título para etiquetarlo en el menú Marcadores (o favoritos). tercero, cuando la página aparece en una búsqueda Web, el **motor de búsqueda** suele enseñar este título como primera línea en los resultados, seguido de un fragmento del contenido de la página.

<p>: Indica un párrafo. Los navegadores no los sangran pero añaden un pequeño espacio entre varios consecutivos para mantenerlos separados.

He aquí la página con estos dos nuevos ingredientes:

```
<!DOCTYPE  html>
<html>
    <head>
        <title>Bienvenido  a mi sitio Web</title>
    </head>
    <body>
        <p></p>
    </body>
</html>
```

Si abre este documento en un navegador Web, verá que la página está vacía, pero ahora aparece el título.

Cuando un navegador muestra una página Web, el título se presenta en la parte superior de la ventana, con el texto al final. Si el suyo utiliza la navegación por pestañas, el título también aparece en ellas.

Tal y como está ahora, este documento HTML es una buena plantilla para futuras páginas. La estructura básica está en su lugar; simplemente necesita cambiar el título y añadir algo de texto.

Lo primero que tenemos que saber es que en toda página web existen dos partes claramente diferenciadas: la cabecera, o **head**, y el cuerpo, o **body**.

Vamos hacer es crear una carpeta, dentro de "Mis documentos", para almacenar los ficheros de prueba que vayamos a usar.

A esta carpeta la llamaremos **pruebas-html** en el resto de los ejercicios. Con la carpeta abierta, haga doble clic sobre el ícono que representa al bloc de notas.

Se le abrirá un documento en blanco.

```
<!DOCTYPE  html>
<html>
    <head>
        <title>Título de la página</title>
    </head>
    <body>
    </body>
</html>
```

Cuando lo tenga escrito, guárdelo en la carpeta con el nombre **plantilla.html**.

Es vital que la extensión sea **.html**, ya que sólo por la extensión el sistema operativo reconoce este archivo como un documento web, y no como un simple y archivo de texto.

El nombre del archivo deberá escribirse como está, en minúsculas y sin espacios ni caracteres especiales.

Los únicos signos de puntuación admitidos son el punto (sólo uno), que lo usaremos para separar el nombre de la extensión y el guión bajo.

El nombre podrá contener letras y números, pero deberá empezar por una letra. Así mismo, nos abstendremos de meter en el nombre de un archivo letras acentuadas, eñes, cedillas, o cualquier otro carácter de algún alfabeto local.

Únicamente se admiten los caracteres del alfabeto internacional (ingles).

Observe los siguientes ejemplos:

página1.htm es **incorrecto**, pues tiene una letra acentuada.

Pagina 1.htm es **incorrecto**, pues tiene un espacio en blanco y una mayúscula

cañas.htm es **incorrecto**, pues tiene una eñe.

pagina/1.htm es **incorrecto**, pues tiene un carácter no permitido (la barra)

pagina1-htm es **incorrecto**, porque la extensión no está separada del nombre por un punto y porque emplea un carácter no permitido. (el guión normal).

pagina.1.htm es **incorrecto**, porque hay más de un punto. Sólo debe usarse uno, para separar el nombre de la extensión.

pagina1.htm es **correcto**

pagina_1.html es **correcto**

pagina-1.html es **correcto**

Puede que todas estas precauciones le parezcan excesivas ahora. La mayoría de ellas son realmente innecesarias en Windows, pero no olvide que está trabajando en la Red.

Los servidores web son mucho más sensibles a determinados aspectos del nombre de los ficheros que su máquina local.

Otra norma importante es que los archivos llevan una extensión, pero las carpetas en las que las guardamos no. Así pues, en el nombre de una carpeta nunca incluiremos puntos.

Una vez que haya guardado su código en su carpeta con el nombre adecuado, cierre el bloc de notas. En su carpeta, además del bloc de notas, verá el ícono que representa al archivo que acaba de grabar.

Observe que el ícono recuerda al de Explorer o Google Chrome (suponiendo que alguno de ellos sea su navegador por defecto).

Esto es así porque al haber guardado el fichero con la extensión .htm el sistema operativo reconoce este fichero como documento web (también llamado documento html).

Si graba incorrectamente la extensión, el ícono será diferente y no podrá usar su archivo como documento web.

Para ejecutar la página que acaba de crear, haga doble clic sobre el ícono. Automáticamente, se abrirá el navegador y se cargará la página. La ruta completa de acceso y el nombre del fichero aparecerán en la barra de dirección.

Ahora veamos en detalle cuál es el código de esta página que hemos creado y qué es lo que hace.

En primer lugar, encontramos el tag **<html>**. Este tag se emplea siempre para iniciar el código html.

Es la forma de decirle al navegador que en este punto empieza el código. Por esta razón se pone esta línea siempre como la primera del código.

No incluiremos ningún tag antes de este, con excepción del **DOCTYPE**. Por el contrario, al final del código tenemos el tag **</html>** , que le indica al navegador que el código finaliza en ese punto.

No deberemos incluir ningún tag ni nada más después de esta línea. Observe que el tag de finalización es igual al de inicio, pero incluyendo la barra inclinada al principio**.**

Dentro del código html encontraremos claramente diferenciados los dos bloques que mencionamos antes: el **head** (cabecera) y el **body** (cuerpo de la página).

Veamos el código de la cabecera:

```
<head>
   <title>
      Título de la página
   </title>
</head>
```

Al igual que el código general, la cabecera tiene un principio indicado por **<head>** y un final indicado por **</head>**.

Todo lo que queramos incluir en la cabecera irá colocado entre estas dos líneas.

Como decíamos antes, en la cabecera incluiremos algunos datos generales de configuración de la página. En esta plantilla sólo hemos incluido el título de la página.

El título de la página se incluye entre los tags **<title>** y **</title>**, que indican, respectivamente, el comienzo y final de dicho título.

El título es un literal (una cadena de texto) que aparecerá en la barra de título del navegador en tiempo de ejecución (cuando se ejecute la página).

Quiero recalcar que este título es solamente eso: un texto libre para que aparezca en la página y no tiene nada que ver con el nombre que le dimos al fichero.

Una vez completado el head, pasamos a la segunda sección de toda la página web: el body o cuerpo.

El **body** incluye, tal como lo mencionaba anteriormente, todo lo que queramos visualizar en el área de navegación.

Se inicia con el tag **<body>** y se finaliza con el tag **</body>**. Como vemos en el archivo que hemos creado, de momento no hemos incluido nada.

Por esta razón, al ejecutar la página, se ve el área de navegación vacía. Dentro del body iremos metiendo código e iremos visualizando los resultados.

Lo primero que vamos hacer es incluir texto en la página para que se muestre en el área de navegación.

Para ello, cierre el navegador y, dentro de la carpeta en la que está trabajando, arrastre el ícono de **plantilla.html** sobre el bloc de notas. De esta forma, se abrirá el bloc de notas y cargará el código fuente que había tecleado anteriormente.

En la zona del código reservada para body de la página entre las líneas **<body>** y **</body>** teclee lo siguiente:

Esto es una línea de texto.

Por lo tanto, su código quedará como sigue:

```
<!DOCTYPE html>
<html><
    head>
        <title>
            Título de la página
        </title>
    </head>
    <body>
        Esto es una línea de texto
    </body>
</html>
```

A continuación haga clic en el menú ARCHIVO y, dentro de éste, elija la opción GUARDAR COMO … asegúrese de guardarlo en su carpeta.

Como nombre de fichero ponga **texto-simple.html**. Una vez guardado, cierre el bloc de notas y ejecute el fichero haciendo doble clic sobre su ícono.

El resultado en la pantalla será como el que se muestra:

.Como ve para mostrar un texto en el área de navegación, basta con incluir dicho texto dentro del cuerpo de la página.

Evidencia:

INSTRUMENTO DE EVALUACIÓN

COLEGIO DE ESTUDIOS CIENTÍFICOS Y TECNOLÓGICOS DEL ESTADO DE CAMPECHE
INSTRUMENTO DE EVALUACIÓN PARA LA ASIGNATURA DE
MODULO 3 SUBMODULO 2
REFORMA INTEGRAL DE LA EDUCACIÓN MEDIA SUPERIOR

ALUMNO (A):	
PLANTEL:	EVIDENCIA: RESUMEN DE CONCEPTO
SEMESTRE:	ENTE EVALUADOR: HETEROEVALUACION
CARRERA:	TIPO DE EVALUACIÓN: FORMATIVA

SEGUNDO PARCIAL

COMPETENCIA DISCIPLINAR: Interpreta tablas, gráficas, mapas, diagramas y textos con símbolos matemáticos y científicos.
COMPETENCIA GENÉRICA: 4.1.- Expresa ideas y conceptos mediante representaciones lingüísticas, matemáticas o gráficas.

RUBRICA 1

ELEMENTOS	DESEMPEÑO				
	EXCELENTE 4 PUNTOS	BUENO 3 PUNTOS	SATISFACTORIO 2 PUNTOS	DEFICIENTE 1 PUNTO	PUNTUACIÓN
CONTENIDO	El resumen contiene todos los conceptos fundamentales del segundo parcial y su definición es concisa y completa.	El resumen contiene el 80 % de los conceptos del segundo parcial y su definición es bastante clara	El resumen contiene más del 60% de los conceptos del segundo parcial. Su definición es un poco confusa o limitada.	El resumen contiene menos del 50% de los conceptos del segundo parcial y, además, carece de claridad y concisión.	
CAPACIDAD DE SINTESIS	Las definiciones de los conceptos son breves y sustanciosas. No hay excesos de palabras o circunlocuciones ociosas.	Las definiciones o explicaciones de los conceptos expresan lo sustancial de estos, pero, se puede ser aún más sintético.	Las definiciones son más bien explicaciones que se extienden sin acabar de dar con la importancia de los conceptos.	Las definiciones dejan de serlo y se pierden los datos sustanciales o esenciales de los conceptos. Hay muchas palabras y pocas ideas.	
CORRECCIÓN GRAMATICAL Y CALIGRAFIA	No tiene errores ortográficos, de acentuación o de conjugación de verbos. Su letra es legible y bien cuidada.	Tienen muy pocos errores ortográficos, o de acentuación o conjugación de verbos. Su letra es legible casi en su totalidad.	Tiene notables errores ortográficos, de acentuación o conjugación de verbos. Escribe de modo irregular, pero se entiende.	Tiene tantos errores ortográficos y gramaticales que distraen considerablemente al lector y prácticamente no se entiende o es ilegible la letra con que escribe.	
TOTALES					

RESULTADO DE LA EVALUACIÓN:	**OBSERVACIONES:**

EVALUÓ (NOMBRE Y FIRMA)	**LUGAR Y FECHA DE APLICACIÓN**

DOCENTE

_____

FIRMA

APRENDIZAJE ESPERADO 2	
	Competencia Profesional: • Desarrollo de una aplicación web
	Situación: • Usando html
	Conceptos fundamentales: **1.5** Aplicar edición y formato de página **Conceptos Subsidiarios:** 1.5.1.1 Formato básico de una página web
	Producto Esperado: • Página web en html
	Instrumento de evaluación: • Rúbrica
	Ponderación: • 50%

Aplicar edición y formato de página
1.5.1. Formato básico de una página web
Formato básico

Un documento HTML posee una estructura básica que contiene una variedad de etiquetas que definen la estructura de la página resultante, dichas etiquetas deben estar adecuadamente estructuradas para que la página funcione sin problemas.
¡Fantástico! recordemos que las etiquetas son quienes le dirán al navegador, qué y cómo será la estructura de una página web.
Ahora que ya sabemos que es una etiqueta y para que sirve, es momento de construir nuestra estructura básica de HTML utilizando las etiquetas destinadas para este fin.

Etiqueta <html>

La etiqueta <html> sirve para indicar que el documento es un documento HTML y el navegador lo interpretara como tal. Todo el contenido de nuestra página irá ordenadamente dentro de la etiqueta <html> y de hecho, es lo primero que debemos escribir en nuestro documento:

```
<html>
  ...contenido...
</html>
```

Etiqueta <head>

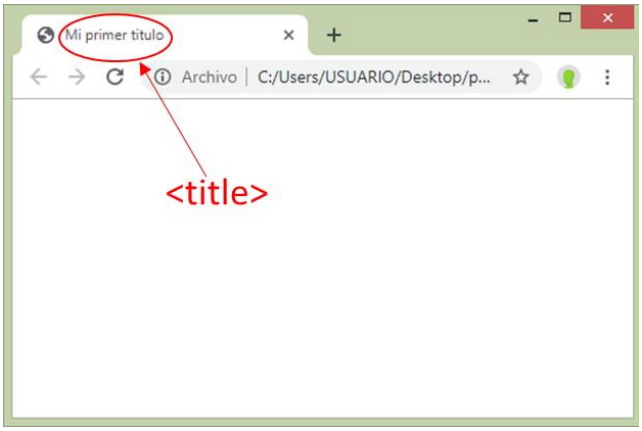
<head> es la etiqueta en la que ira la cabecera de nuestra página y va justo después de la etiqueta de apertura <html>, esta etiqueta contiene elementos que ayudarán a que nuestra página funcione correctamente y generalmente dentro de esta etiqueta van elementos que no se visualizan en el navegador.

```
<html>
  <head>
    ...elementos de la cabecera...
  </head>
</html>
```

Etiqueta <title>

La etiqueta <title> va dentro de la etiqueta <head> en la cabecera, **<title> contiene el título de la página** y esta se muestra en la pestaña de los navegadores, más no se ve en el cuerpo de la página.

```
<html>
<head>
<title> Mi primer titulo </title>
</head>
</html>
```



Etiqueta <body>

Después de la cabecera sigue el cuerpo de la página y se indica mediante la etiqueta<body>, el cuerpo o body contiene prácticamente los elementos visuales de nuestra página, aquí irán los textos de encabezado o títulos, subtítulos, párrafos, listas, tablas, formularios, multimedia, etc. Es decir el contenido de la página, todo lo que queremos que se vea en el navegador.

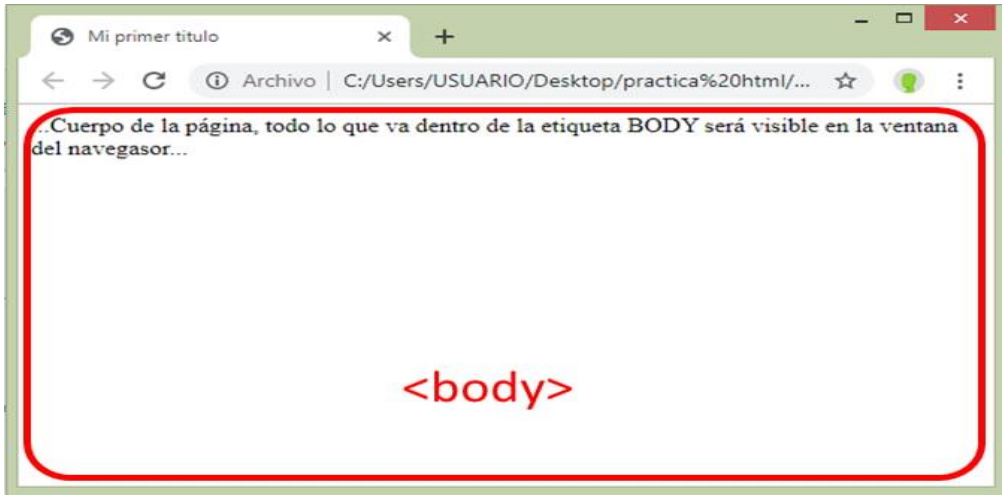
```
        <body>
        ...cuerpo de la página...
</body>
```

Estructura básica final del documento HTML

Entonces finalmente la estructura básica HTML quedará de la siguiente manera.

```
<html>

<head>
<title> Mi primer titulo </title>
</head>

<body>
...Cuerpo de la página, todo lo que va dentro de la etiqueta BODY será visible en la ventana del navegasor...
</body>

</html>
```

Resultado del código anterior:



Como podemos observar, tenemos que indicar al navegador que nuestro documento es un documento HTML <html>, también tenemos que indicarle cuál es la cabecera<head>, dentro de la cabecera va el titulo <title> y también debemos indicar cuál es el cuerpo de nuestra página <body>.

-Formato de texto

Etiquetas de párrafo y encabezados HTML

Las etiquetas que siempre indican texto y son:

- <p> Indica que el elemento es un párrafo.
- <h1> Señala que se trata de un titulo principal.
- <h2>, <h3>, <h4>, <h5>, <h6>: indica niveles de encabezado, subtítulos.

EJEMPLO

```
<h1> Este es un título de primer nivel</h1>
<h2>Título de segundo nivel</h2>
<h3>Título de tercer nivel</h3>
<p>Elemento de parrafo en html</p>
```

Lista de formatos y etiquetas de texto HTML

- Etiqueta HTML <b>: Para texto en negrita.
- Etiqueta HTML : Para enfatizar texto.
- Etiqueta HTML <i>: Para texto en cursiva.
- Etiqueta HTML <u>: Para texto subrayado.
- Etiqueta HTML : Para textos importantes.
- Etiqueta HTML <big>: Texto grande.
- Etiqueta HTML <small>: Para texto pequeño.
- Etiqueta HTML <sub>: Para texto subíndice.
- Etiqueta HTML <sup>: Para texto superíndice.

- Etiqueta HTML <ins>: Para texto insertado.
- Etiqueta HTML <del>: Para texto borrado.
- Etiqueta HTML <mark>: Para texto marcado.

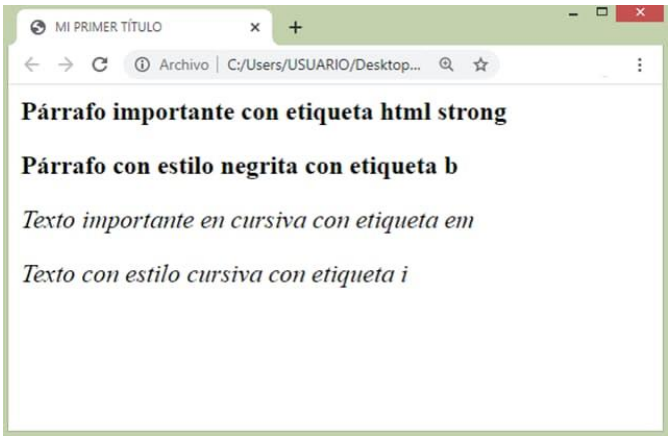
Diferencia entre /<b> y /<i>

Las **etiquetas y <b>** se muestran visualmente iguales en los navegadores, lo mismo ocurre con **las etiquetas y <i>** pero hay diferencias.

- <b> define texto en negrita, sin ninguna importancia adicional.
- define texto fuerte, con importancia semántica aumentada.
- <i> define el texto en cursiva, sin ninguna importancia adicional.
- define el texto enfatizado, con una importancia semántica aumentada.

Ejemplos de formatos de texto HTML

```
<strong><p> Párrafo importante con etiqueta html strong</p></strong>
<b><p>Párrafo con estilo negrita con etiqueta b</p></b>
<em><p>Texto importante en cursiva con etiqueta em</p></em>
<i><p>Texto con estilo cursiva con etiqueta i</p></i>
```

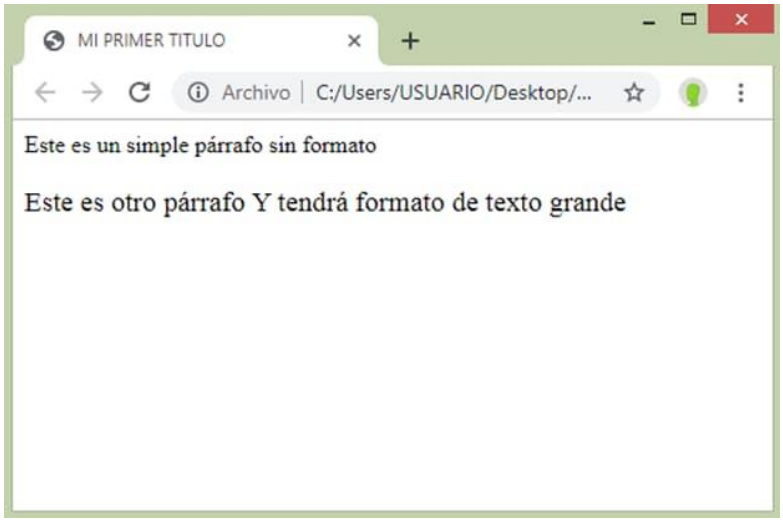


Obviamente estas etiquetas y formatos las podemos aplicar a otras etiquetas de texto, como párrafos, títulos, enlaces, etc.

Ejemplos de formatos y etiquetas de texto en html

Empecemos con un párrafo sin ningún formato adicional, posteriormente aplicaremos el formato de texto grande al siguiente párrafo, visualizar las diferencias entre el primer párrafo y el segundo, nos ayuda a entender mejor cómo funcionan los formatos de texto en html, veamos:

```
<p> Este es un simple párrafo sin formato </p>

<big><p> Este es otro párrafo Y tendrá formato de texto grande</p></big>
```



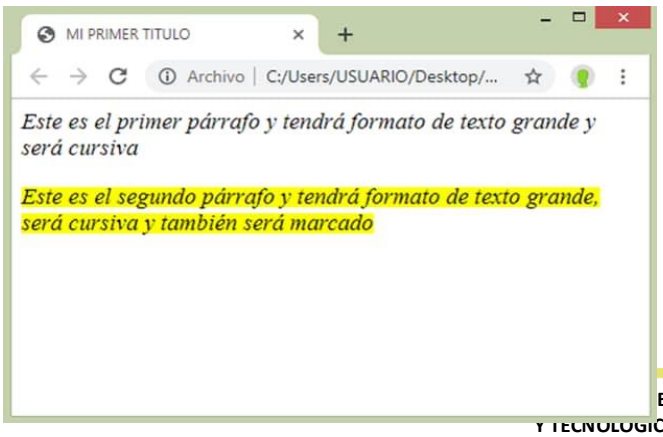

```
<p> Este es un simple párrafo sin formato </p>

<big><p> Este es otro párrafo Y tendrá formato de texto grande</p></big>
```


Es posible seguir agregando más formatos a nuestras páginas en html, para ello simplemente debemos abrir y cerrar las respectivas etiquetas con el contenido dentro de ellas.

Agregando más formatos html

A continuación aplicaremos los formatos de texto grande y cursiva a nuestro primer párrafo y además aplicaremos un marcado para el segundo párrafo.

```
<i><big><p> Este es el primer párrafo y tendrá formato de texto grande y será cursiva</p></big></i>

<p><mark><i><big> Este es el segundo párrafo y tendrá formato de texto grande, será cursiva y también será marcado</big></i></mark></p>
```



- INSERTAR IMÁGENES

Insertar una imagen en HTML es muy simple, simplemente debemos indicar que el elemento es una imagen y **cuál es el archivo o donde esta ubicado** para que sea mostrado en el navegador, veamos.

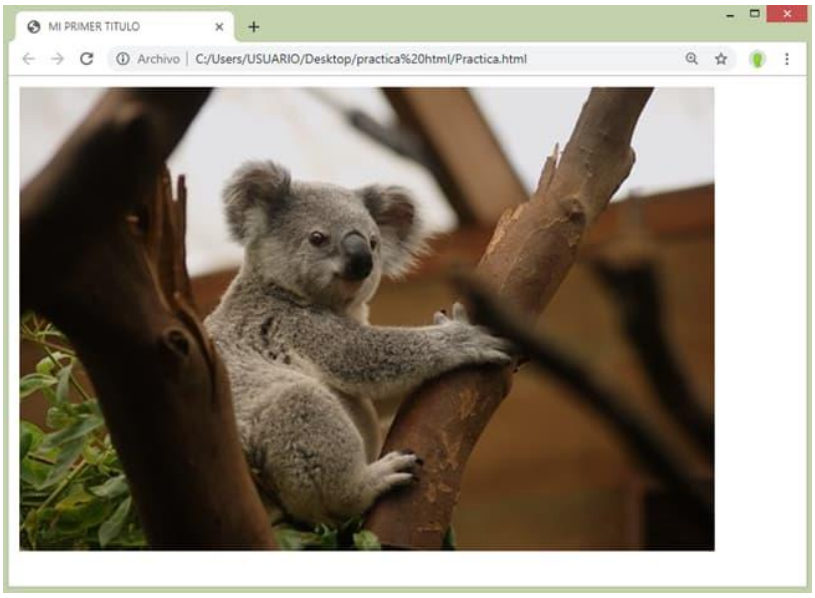
Etiqueta <img/>

Utilizaremos la etiqueta `<img/>` para insertar imágenes, a diferencia de otras etiquetas, `<img/>` no tiene una etiqueta de cierre, pero si atributos.

Atributo src

Para indicar la ubicación o dirección de la imagen, utilizaremos el atributo `src`, la imagen puede estar almacenada en la misma carpeta que el documento HTML o también podemos agregar la imagen desde una dirección URL, de la siguiente manera:

```
<img src="URL o nombre del archivo"/>
```

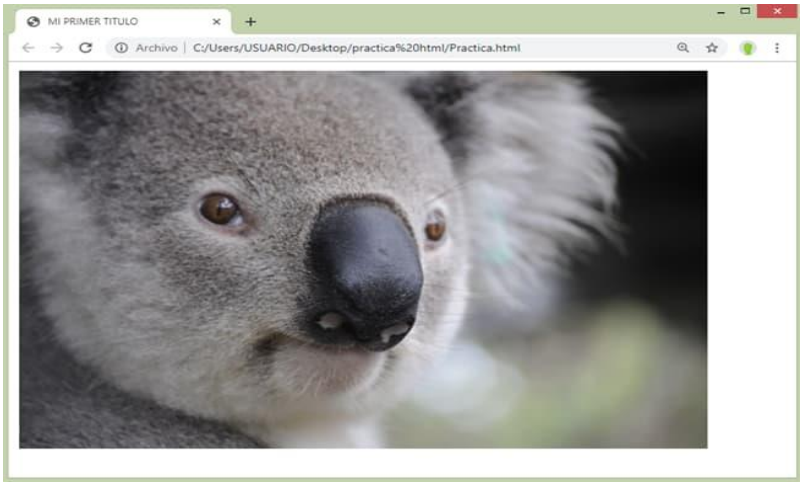


Insertar imagen HTML desde URL

En el caso de que la imagen está ubicada en una dirección URL externa, entonces el valor de **src** será la URL de la imagen.

Ejemplo de código para insertar imagen desde URL:

```
<img src="https://cdn.pixabay.com/photo/2015/02/09/20/03/koala-630117__340.jpg"/>
```



Atributo alt

El atributo **alt** contiene el texto alternativo, este texto se mostrará si por alguna razón no carga la imagen, por lo tanto tiene que ser un texto que describa a dicha imagen.

Ejemplo de código:

```
<img src="gatito.jpg" alt="foto de gatito"/>
```

Tamaño de imagen

Para definir el tamaño de una imagen podemos utilizar los atributos **width** (ancho) y **height** (alto) y su valor estará expresado en px o % (pixeles o porcentajes)

Ejemplo de código para dimensionar imagen en HTML:

```
<img src="koala.jpg" height="50px" width="150px"/>
<img src="koala.jpg" height="50%" width="150%"/>
```

Atributo border

El atributo `border` crea un borde para la imagen el valor de este atributo se puede expresar en píxeles o en otra unidad de medida compatible. Solo por mencionar, con CSS podemos personalizar las imágenes con mayor libertad, ahora lo que nos importa es conocer su estructura.

Ejemplo de borde:

```
<img border="2px" src="koala.jpg" />
```

Atributo vspce

Este atributo **coloca un espacio en la parte superior e inferior de la imagen** y su valor se puede expresar en píxeles u otra unidad como em, %, pt, etc.

Cómo poner **enlaces en imágenes** HTML

Poner un enlace a una imagen es tan sencillo como colocar la etiqueta `<img/>` dentro de una etiqueta `<a>` de la siguiente manera.

```
<a hrf="URL"> <img src="imagen.jpg"/> </a>
```

Recordemos que la etiqueta <a> sirve para crear enlaces o hipervínculos, por tanto, los elementos dentro de ella, pueden apuntar a una dirección URL.

Hipervínculos

Colocar un enlace en HTML a un texto es mucho más fácil de lo que puedas imaginar, es tan simple como abrir y cerrar la etiqueta para enlace son sus respectivos atributos y valores.

Etiqueta **<a>**

Para definir enlaces o **hipervínculos utilizaremos la etiqueta <a>** de la siguiente manera:

```
<a> texto del enlace</a>
```

Sin embargo aún no hemos indicado a donde debe apuntar dicho enlace y eso lo haremos con el atributo href.

Atributos para enlaces HTML

Existen varios atributos para los enlaces, en esta lección los desarrollamos detalladamente cada uno de ellos, pero a continuación los nombramos:

href

target

rel

Existen muchos más, pero estos te son los principales. Ahora vemos como se usan y los valores que pueden tomar estas.

Atributo href para enlaces HTML

Para indicar la dirección destino del enlace utilizaremos el atributo href, En todo caso quedaría de la siguiente manera.

```
<a href="URL"> texto ancla </a>
```

El atributo href contiene el valor de la url de destino

El texto ancla es el texto visible, o el texto que contiene el enlace, el texto donde se hará clic.

<a> puede ir dentro de un párrafo para un texto ancla especifico.

Se puede colocar enlace tanto a texto como a imágenes.

Ahora nos toca crear nuestro primer enlace de texto, veamos el ejemplo:

Ejemplos de Hipervínculos en HTML

A continuación veamos ejemplos concretos en las que se coloca un en lace a un pedazo de texto dentro de párrafos:

HTML

```
1 Haga clic <a href="https://desarrolladoresweb.org/">en este enlace </a> para ir a
  desarrolladoresweb.org
2
3 <p> Este es un párrafo bastante bueno este es un párrafo   bastante bueno este es un
  párrafo bastante seguidamente colocaremos un enlace html para que veas como se
  inserta un enlace en el siguiente texto: <a href="https://www.deingenieriaswq.com/">
  haga clic en este enlace</a>, bueno este es un párrafo bastante bueno este es un
  párrafo bastante bueno este es un párrafo bastante bueno</p>
```

Ahora ver;as que si clicamos sobre los respectivos enlaces, estos nos dirigen hacia las respectivas direcciones asignadas.

Atributo target para enlaces HTML

Por defecto, los enlaces se abren en una misma ventana del navegador web, **para que el enlace se habrá en una nueva ventana el atributo target toma el valor «_blank»** de la siguiente manera:

Ejemplo:

```
<a href="https://www.deingenierias.org/" target="_blank">Este enlace se abrirá en una nueva ventana</a>
```

Atributo rel para enlaces HTML

El atributo rel de HTML sirve para indicar la relación que existe entre la pagina del cual sale el enlace y la pagina que recibe el enlace, muy importante para indicar a los motores de búsqueda dicha relación y también a los usuarios.

Valores para el atributo rel

sponsored.- para indicar que el enlace es un enlace de pago.

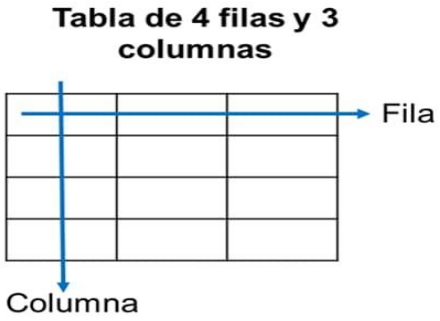
ugc.- indica que el enlace es un enlace generado por el usuario.

nofollow.- indica a los motores de búsqueda que no siga el enlace como si fuera natural.

```
<a href="https://www.deundominid.com/"  rel="nofollow">Este enlace, es un enlace nofollow de ejemplo</a>
```

Tablas

Como sabemos una tabla es una estructura formada por filas y columnas, las filas son las distribuciones horizontales y las columnas son las distribuciones verticales



En HTML la estructura de una tabla tiene que ver con etiquetas, una etiqueta <table> indica que es un elemento de tabla, dentro de ella va otra etiqueta <tr> y esta señala las filas de una tabla y finalmente, dentro de cada fila expresada con etiqueta <tr> va otra etiqueta <td> y es básicamente un cuadrito de la tabla contenida en una fila y dentro de ella ira el contenido deseado; si no lograste captar la idea, no hay problema ¡Lo harás con los ejemplos! sigamos.

Etiqueta <table>

Para definir una tabla en HTML empleamos la etiqueta <table>

```
<table>
</table>
```

Etiqueta <tr>

Para crear una fila dentro de una tabla empleamos la etiqueta <tr> si deseamos varias filas, entonces crearemos varias etiquetas <tr> con su respectiva etiqueta de cierre. Veamos un ejemplo de una tabla con 3 filas.

```
<table>
  <tr>
  </tr>
  <tr>
  </tr>
  <tr>
  </tr>
</table>
```

Etiqueta <td>

Esta etiqueta definirá una columna y está dentro de <tr>.

<td> Prácticamente serán los contenedores, dentro irán los contenidos de tabla, puede ir texto, imágenes, e incluso más tablas y otros elementos HTML, ¡pruébalos!

Veamos el ejemplo:

```
<table>
 <tr>
  <td> Primero </td>
  <td> Segundo </td>
  <td> Tercero </td>
 </tr>
</table>
```

Para que entiendas, capta esto: dentro de las tablas <table> van las filas <tr> y dentro de las filas van las columnas <td>, además, dentro de las tablas pueden varias filas y dentro de las filas pueden ir varias columnas como veremos más adelante.

Como podemos observar en la imagen, el resultado es una tabla de una fila y tres columnas, sin embargo no parece y para mejorar el aspecto emplearemos a continuación el atributo border.

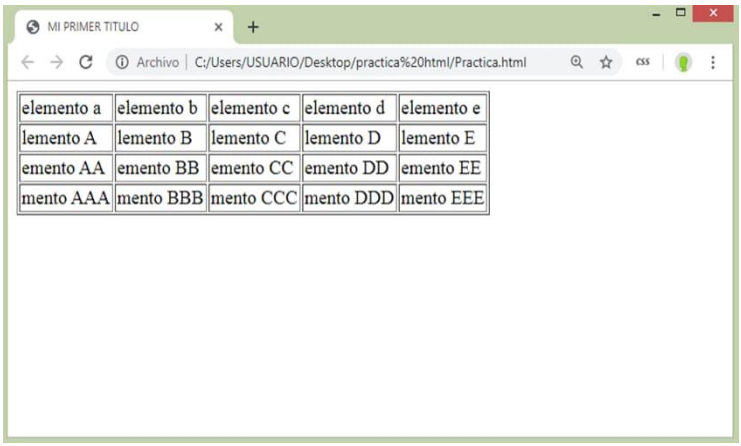
Atributo border

Cuando creamos una tabla, por defecto no tiene ningún borde, más parece una lista sin estilo.

Para ponerle borde a la tabla utilizamos el atributo border y en su valor la dimensión de ese borde.

Como ejemplo veamos la misma tabla anterior pero en este caso le ponemos borde de 1px:

```
<table border="1px">
 <tr>
  <td> Primero </td>
  <td> Segundo </td>
  <td> Tercero </td>
 </tr>
</table>
```



Ejemplo de tablas en HTML

El ejemplo a continuación es una tabla de 4 filas y 5 columnas con un borde de 1px:

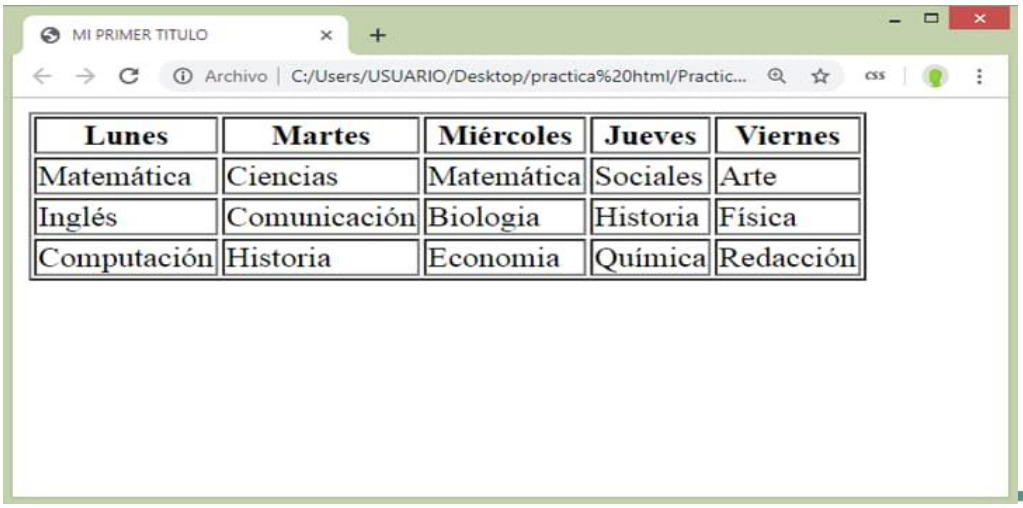
```
<table border="1px">
<tr>
<td> elemento a </td>
<td> elemento b </td>
<td> elemento c </td>
<td> elemento d </td>
<td> elemento e </td>
</tr>
<tr>
<td> lemento A </td>
<td> lemento B </td>
<td> lemento C </td>
<td> lemento D </td>
<td> lemento E </td>
</tr>
<tr>
<td> emento AA </td>
<td> emento BB </td>
<td> emento CC </td>
<td> emento DD </td>
<td> emento EE </td>
</tr>
<tr>
<td> mento AAA </td>
<td> mento BBB </td>
<td> mento CCC </td>
<td> mento DDD </td>
<td> mento EEE </td>
</tr>
</table>
```

Etiqueta <th>

Utilizaremos `<th>` para poner encabezado a la tabla en su primera fila.

Cada elemento `<th>` será el encabezado de una columna, veamos un ejemplo y como se ve:

```
<table border="1px">
<tr>
<th> Lunes </th>
<th> Martes </th>
<th> Miércoles </th>
<th> Jueves </th>
<th> Viernes </th>
</tr>

<tr>
<td> Matemática  </td>
<td> Ciencias </td>
<td> Matemática </td>
<td> Sociales </td>
<td> Arte </td>
</tr>

<tr>
<td> Inglés </td>
<td> Comunicación </td>
<td> Biologia </td>
<td> Historia </td>
<td> Física </td>
</tr>

<tr>
<td> Computación </td>
<td> Historia </td>
<td> Economia </td>
<td> Química </td>
<td> Redacción </td>
</tr>
</table>
```



Lunes	Martes	Miércoles	Jueves	Viernes
Matemática	Ciencias	Matemática	Sociales	Arte
Inglés	Comunicación	Biologia	Historia	Física
Computación	Historia	Economia	Química	Redacción

Como podemos notar, los elementos dentro de las etiquetas de encabezado de tabla `<th>` se resaltan, de tal manera que poseen un significado semántico.

A continuación aprenderemos a combinar filas y columnas, algo que será de mucha utilidad ¡Ya lo verás!

Atributo colspan y rowspan

En muchas ocasiones es necesario combinar celdas de una tabla, como verás, algunas veces es necesario combinar celdas de una misma fila y en otras celdas de una misma columna y todo ello aprenderemos a continuación, es muy sencillo:

Combinar filas en tablas HTML

Para combinar celdas de distintas filas en una tabla HTML emplearemos el atributo `rowspan` dentro del elemento `<td>` desde donde se extenderá la combinación y en su valor colocaremos el número de filas que serán combinadas, veamos un ejemplo:

```
<table border="1">
<tr>
<td rowspan="2"> Oso </td>
<td> León </td>
<td> Tigre </td>
</tr>
<tr>
<td> Cebra </td>
<td rowspan="3"> Panda </td>
</tr>
<tr>
<td> Loro </td>
<td> Pato </td>
</tr>
<tr>
<td> Buo </td>
<td> Águila </td>
</tr>
</table>
```

Observa como la celda oso desplaza una fila más, es decir ocupa dos celdas en la misma columna, esto ocurre por el valor de 2 colocada para la propiedad ROWSPAN. Por otro lado, la celda Panda ocupa 3 celdas por su valor 3.

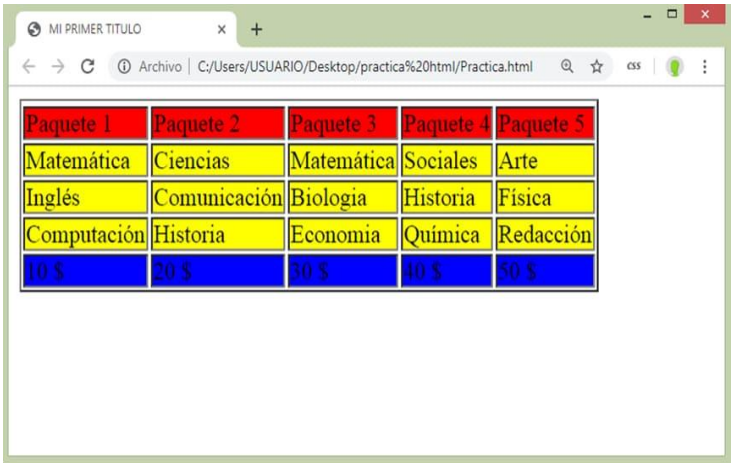
En fin, ROWSPAN sirve para combinar celdas en filas distintas y en la misma columna.

Combinar columnas de tablas en HTML

Ahora aprenderemos a combinar elementos de la misma fila y de diferentes columnas, es bastante sencillo, simplemente emplearemos el atributo COLSPAN en el elemento desde donde se hará la combinación, a partir de ahí se combinará el número de celdas especificada en el valor del atributo colspan; veamos el ejemplo:

```html
<table border="1">
<tr>
<td > Rojo </td>
<td colspan="2"> Verde </td>
</tr>

<tr>
<td> Café </td>
<td> Marrón </td>
<td> Tinto </td>
</tr>

<tr>
<td colspan="3"> Naranja </td>
</tr>

<tr>
<td colspan="2"> Blanco </td>
<td> Negro </td>
</tr>

</table>
```

```
<table border="1px">
  <thead bgcolor="red">
<tr>
<td> PAQUETES </td>
<td> Paquete 1  </td>
<td> Paquete 2 </td>
<td> Paquete 3 </td>
<td> Paquete 4 </td>
<td> Paquete 5 </td>
</tr>
  </thead>
  <tbody bgcolor="yellow">
<tr>
<td rowspan="3"> CURSOS  </td>
<td> Matemática  </td>
<td> Ciencias </td>
<td> Matemática </td>
<td> Sociales </td>
<td> Arte </td>
</tr>
<tr>
<td> Inglés </td>
<td> Comunicación </td>
<td> Biologia </td>
<td> Historia </td>
<td> Física </td>
</tr>
<tr>
<td> Computación </td>
<td> Historia </td>
<td> Economia </td>
<td> Química </td>
<td> Redacción </td>
</tr>
  </tbody>
  <tfoot bgcolor="blue">
    <tr>
<td> PRECIOS  </td>
<td> 10 $ </td>
<td> 20 $ </td>
<td> 30 $ </td>
<td> 40 $ </td>
<td> 50 $ </td>
</tr>
  </tfoot>
</table>
```



Formularios

Crear un formulario en html implica construir los campos de información, estos pueden ser: campos de texto, de número, selectores, desplegables, casillas de verificación y entre otras más.

como ya es sabido en html se emplean etiquetas para crear los diversos elementos existentes, y el caso de los formularios no es la excepción, para crear los componentes del formulario HTML también haremos uso de una variedad de etiquetas y atributos, los mismos que conoceremos a continuación.

Etiqueta <form>

Utilizaremos la etiqueta `<form>` para indicar al navegador que el elemento será un formulario.

`<form>` se emplea para cualquier tipo de formulario HTML, ya sea uno de contacto, suscripción, registro, login, encuesta, etc.

Cómo crear Formularios en HTML

Crear un formulario en html implica construir los campos de información, estos pueden ser: campos de texto, de número, selectores, desplegables, casillas de verificación y entre otras más.

como ya es sabido en html se emplean etiquetas para crear los diversos elementos existentes, y el caso de los formularios no es la excepción, para crear los componentes del formulario HTML también haremos uso de una variedad de etiquetas y atributos, los mismos que conoceremos a continuación.

Etiqueta <form>

Utilizaremos la etiqueta `<form>` para indicar al navegador que el elemento será un formulario.

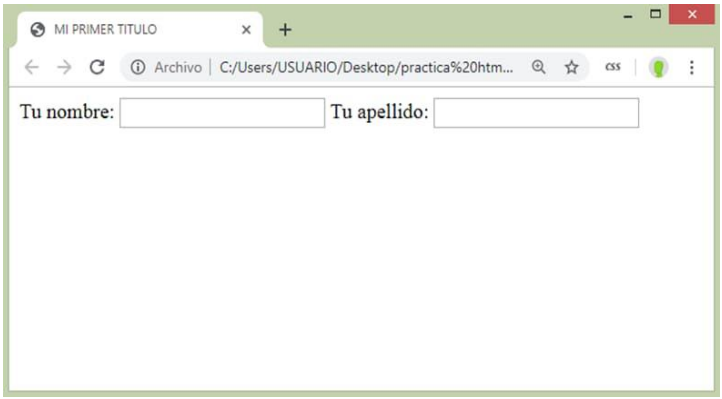
`<form>` se emplea para cualquier tipo de formulario HTML, ya sea uno de contacto, suscripción, registro, login, encuesta, etc.

```
<form>
…Aquí irán los campos del formulario...
</form>
```

Etiqueta <input>

Crear los campos en el formulario es tan sencillo como utilizar la etiqueta <input> antecedida de lo que deseas obtener. Por ejemplo:

```
<form>
  Tu nombre: <input/>
  Tu apellido: <input/>
</form>
```



Existen varios tipos campos para formularios HTML y también debemos especificar que tipo de campo es un elemento del formulario, para ello aprenderemos a utilizar el atributo **TYPE** a continuación.

Atributo type

El atributo type va dentro de la etiqueta <input> y su valor indicará que tipo de campo se está creando. Veamos algunos valores que puede tomar el atributo type en un formulario HTML:

Valores de atributo type en formularios HTML

Text: para campos de texto

Password: para campos de contraseña

Checkbox: para casillas de verificación

Radio: para casillas de selección

Submit: para botones de envío

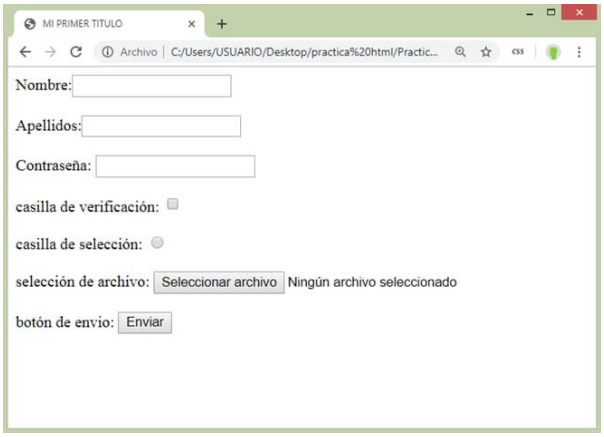
Reset: para botones de resetear

File: para campos de selección de archivo

Ejemplo de formulario html con atributo type

En el siguiente ejemplo crearemos un formulario empleando los diversos tipos de campo que aprendimos anteriormente y para organizar y diferenciar mejor cada campo, emplearemos la etiqueta de salto de linea <br/>, veamos:

```
<form>
Nombre:<input type="text"/>  <br/><br/>
Apellidos:<input type="text"/>   <br/><br/>
Contraseña: <input type="password"/>  <br/><br/>
casilla de verificación: <input type="checkbox"/> <br/><br/>
casilla de selección: <input type="radio"/>  <br/><br/>
selección de archivo:  <input type="file">  <br/><br/>
botón de envio: <input type="submit">
</form>
```



Si observas el anterior ejemplo te darás cuenta de que todos los campos están construidos con la etiqueta html <input>. Pero existen otros campos que no utilizan <input>, sino etiquetas específicas como la etiqueta <select> y <textarea>, conozcamos de que se tratan.

Crear una desplegable con opciones HTML

Ahora aprenderemos a **realizar desplegables en formularios**, seguramente viste esos campos en las que existe un desplegable con varias opciones dentro para seleccionar una de ellas, pues ahora aprenderás a hacerlo.

Etiqueta <select>

Utilizaremos la etiqueta <select> para crear un campo desplegable, este tipo de campos en HTML permiten seleccionar una opción del desplegable y además son muy útiles en esas ocasiones en las que buscamos buena presentación de espacio en el formulario.

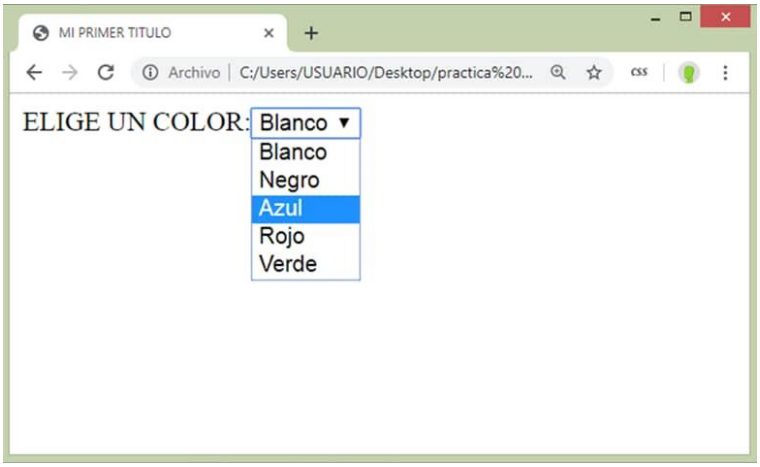
Etiqueta <option>

Utilizaremos la etiqueta <option> dentro de la etiqueta <select>. <option> contiene una opción del desplegable y podemos implementar la cantidad de opciones que deseemos.

Ejemplo de un campo desplegable en formularios HTML

El ejemplo a continuación trata de un desplegable con 5 opciones de colores para elegir, veamos:

```
<form>
ELIGE UN COLOR:<select>
<option>Blanco</option>
<option>Negro</option>
<option>Azul</option>
<option>Rojo</option>
<option>Verde</option>
</select>
</form>
```



Por defecto estará seleccionada la primera opción, además es necesario pasar el cursor y clic para desplegar las opciones, una vez seleccionada la opción, esta regresa a su estado inicial pero con la opción seleccionada.

Atributos para etiqueta <select> en formularios html

Atributo size.- Utiliza el atributo size para determinar la cantidad de opciones a mostrar y su valor en números.

Atributo multiple.- Utiliza el atributo **multiple** con valor también **multiple** para permitir que el desplegable sea de múltiple selección de tal manera que permita seleccionar varias opciones.

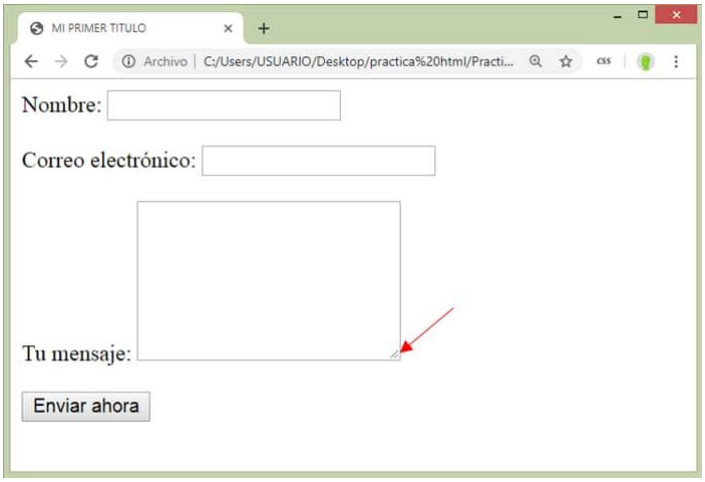
Crear un campo de texto grande en formularios HTML

Un campo de texto con varias lineas disponible en el formulario puede ser necesario para permitir a nuestros usuarios enviar extensos textos y para ello haremos uso de la ***etiqueta*** ***<textarea>***.

Etiqueta <textarea>

La etiqueta <textarea> se utilizara para crear un área de texto de varias líneas, puede servirnos por ejemplo para un capo de mensaje, veamos un ejemplo concreto:

```
<form>
Nombre:
  <input type="text"/>  <br/><br/>
Correo electrónico:
  <input type="email"/>  <br/><br/>
Tu mensaje:
  <textarea name="mensaje"> </textarea> <br/><br/>
  <input type="submit" value="Enviar ahora"/>
  </form>
```



Si observamos, el campo de texto multilinea posee en la esquina inferior derecha un asistente que permite ampliar el campo con el cursor.

Atributo action

El valor del atributo action será la URL de la página que queremos que cargue después de que se envíe la información de un formulario. Este atributo va dentro de la etiqueta <form> y lo podemos utilizar si deseamos.

```
<form action="URL">
```

Atributo method

El atributo method puede tomar como valores a GET o POST y esto indica el método http que será utilizado cuando se envíe la información.

Ejemplo:

```
<form action="URL" method="get">
<form action="URL" method="post">
```

Cuando se usa el **valor get**, los datos del formulario podrán ser vistos en la dirección de la página.

Cuando se usa el **valor post** los datos enviados no serán visibles en la dirección de la página.

Por seguridad se recomienda utilizar usar el valor post, para datos sensibles.

Atributo name

Utiliza el atributo name para ponerle un nombre al formulario

```
<form name="miformulario" action="URL" method="post">
```

Para que el formulario funcione correctamente utilizaremos el atributo name para cada campo del formulario, así será posible recibir correctamente los datos enviados.

```
<form>
Nombre: <input name="nombre"  type="text"/>  <br/><br/>
Correo electrónico:   <input name="correo" type="email"/>  <br/><br/>
Tu mensaje:  <textarea name="mensaje"> </textarea> <br/><br/>
  <input type="submit" value="Enviar ahora"/>

</form>
```

El atributo name será necesario para procesar la información de cada campo y que estas lleguen a su destino correctamente.

EVIDENCIA

INSTRUMENTO DE EVALUACIÓN

COLEGIO DE ESTUDIOS CIENTÍFICOS Y TECNOLÓGICOS
DEL ESTADO DE CAMPECHE
INSTRUMENTO DE EVALUACIÓN PARA LA ASIGNATURA DE
MODULO 3 SUBMODULO 2
REFORMA INTEGRAL DE LA EDUCACIÓN MEDIA SUPERIOR

ALUMNO (A):	
PLANTEL:	EVIDENCIA: DESARROLLO DE UNA PAGINA WEB EN HTML
SEMESTRE:	ENTE EVALUADOR: HETEROEVALUACION
CARRERA:	TIPO DE EVALUACIÓN: FORMATIVA
SEGUNDO PARCIAL	
COMPETENCIA DISCIPLINAR: Interpreta tablas, gráficas, mapas, diagramas y textos con símbolos matemáticos y científicos.	
COMPETENCIA GENÉRICA: 4.1.- Expresa ideas y conceptos mediante representaciones lingüísticas, matemáticas o gráficas.	

RUBRICA 2

CATEGORY	4 Sobresaliente	3 Notable	2 Aprobado	1 Insuficiente
Título	El título se ajusta muy bien a los contenidos de la página web. El título es sugerente y muy creativo.	El título se ajusta bien a los contenidos de la página web. El título es atractivo.	El título se ajusta suficientemente al contenido de la página web.	El título no se ajusta a los contenidos de la página web.
La información	Aparece muy ordenada, es coherente. Existe una gran relación entre texto e imagen.	Aparece ordenada y, en su mayoría es coherente. Casi siempre existe relación entre el texto y la imagen.	Es suficientemente ordenada y coherente. Algunas veces, no existe relación entre el texto y la imagen.	En muchos casos es desordenada e incoherente. y no hay relación entre imagen y texto.
El texto	Resume muy claramente la información esencial.	Resume bien la información esencial.	Resume suficientemente la información esencial.	No resume la información esencial.
Otros recursos	A lo largo de la página web, aparecen imágenes, direcciones de Internet o hipervínculos (3-4)y vídeos relacionados con el tema (3)	En la mayoría de la página web, aparecen imágenes, direcciones de Internet o hipervínculos (2-1) y vídeos relacionados con el tema (2)	En parte de la página web, aparecen imágenes, direcciones de Internet o hipervínculo (1) y vídeos relacionados con el tema (1)	Página web pobre en imágenes, y sin direcciones de Internet ni vídeos. Si aparecen, no tienen que ver con el tema.
La ortografía	No existen errores ortográficos.	La ortografía es buena. Falta algún acento.	La ortografía es suficiente pero existen dos faltas de ortografía.	Existen importantes fallos ortográficos.
Diseño	Muy original y creativo tanto en la estructura de la página web como en el fondo y paleta de colores.	Página web original y creativa tanto en su estructura como en el fondo y paleta de colores.	Página web que, unas veces, mantiene una estructura coherente y otras no. El fondo y la paleta de colores es suficiente	Página Web que pocas veces mantiene la coherencia en su estructura. El fondo y la paleta cromática son inadecuados.

Estimado alumno(a), ahora TÚ ERES:

¡TODOS SOMOS CECyTEC!
¡NUNCA LO OLVIDES!