

---

---

# REPASO HTML

DWES UD0

---

---

# HTML - Hyper Text Markup Language

Lenguaje de programación de páginas web.



Sublenguaje de XML (**eX**tensible **M**arkup **L**anguage)  
XML define lenguajes de marcas.



HTML es un lenguaje de marcado.

# HTML - Etiquetas



\*excepciones: aquellas etiquetas en las que el programador no escribe información dentro de ellas, no tendrán etiqueta de cierre.

# HTML - Lenguaje interpretado

- El cliente (navegador) realiza una **petición** de una página web.
- El servidor **procesa** la petición y **envía** un archivo con contenido html.
- El cliente **interpreta** esa información y la coloca en la pantalla.

Las etiquetas no se muestran en el cliente, son las "marcas" que utiliza para poder interpretar el contenido del archivo con contenido html.

# HTML - doctype

Sea cual sea la versión de HTML que se use, siempre hay que indicar al principio de un documento con contenido HTML la versión, así el navegador sabrá qué reglas usar.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!doctype html>
```

# HTML - HTML5 Un conjunto separado

**HTML**



**JS**



**CSS**



# HTML

Las nuevas metodologías aconsejan separar de manera muy diferenciada los elementos que conforman una página web:

- Contenido → HTML
- Estilo → CSS
- Interacción → JavaScript

Se obtendrán aplicaciones web más modulares y reutilizables.

Se debe utilizar una estructura de directorios adecuada.

# HTML - Tildes y caracteres no ingleses

Se evitará usar tildes, ñe, caracteres especiales (@, #, \$...) y los espacios en blanco en los nombres de los archivos y directorios de nuestras aplicaciones web.

Hay que tener cuidado con mayúsculas y minúsculas. Si nuestro servidor web está en un sistema operativo **Case Sensitive** puede ocasionar problemas.



# HTML

1. Diseñaremos y crearemos con **HTML** la estructura de la aplicación web.
2. Se introducirá el contenido estático y/o dinámico (**php, Java...**).
3. Se aplicará estilo mediante **CSS**.
4. Se podrá añadir interacción mediante **JavaScript** y sus librerías (**jQuery...**).

# HTML 5 - Estructura básica html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Título de la página</title>
  </head>

  <body>
    Cuerpo de la página.
  </body>
</html>
```

# Tabulación y comentarios

**Un código tabulado ayuda a la identificación de sus partes.**

**Un código comentado ayuda a los desarrolladores presentes y futuros.**

El navegador agrupa todos los espacios en blanco y saltos de línea seguidos y los convierte en uno, así que no hay que tener miedo en tabular.

Existen soluciones software que eliminan caracteres innecesarios y comentarios comprimiendo el código para que pese menos.

# HTML 5 - Etiquetas

Las etiquetas sirven para marcar el contenido de la página web.

Dentro de una etiqueta puede haber otras etiquetas

Correcto:     `<h1>Hola <span>Pepe</span></h1>`

Erróneo:     `<h1>Hola <span>Pepe</h1></span>`

# HTML 5 - Etiquetas

Existen dos tipos de etiquetas:

**block**

**inline**

Aunque el estilo se aplicará mediante CSS, hay que conocer el comportamiento de las etiquetas para comprender cómo se colocan en una página web.

# HTML 5 - Etiquetas

**block**

inline

Una etiqueta de tipo **block** siempre va a ocupar todo el ancho posible.  
Ejemplo: el contenido de `<body>` ocupará todo el ancho de la ventana.

Se podría decir que una etiqueta **block** siempre tiene un salto de línea al principio y otra al final.

# HTML 5 - Etiquetas

block

**inline**

El contenido de una etiqueta **inline** ocupará el trozo de línea o líneas necesarias.

Cuando llegue al final de la línea y aún tenga contenido que mostrar se pasará a la línea siguiente.

# HTML 5 - Etiquetas

**block**

**inline**

Las etiquetas **block** pueden contener dentro otras etiquetas **block** y/o **inline**.

Las etiquetas **block** pueden estar dentro de otras etiquetas **block**.

Las etiquetas **inline** pueden contener dentro otras etiquetas **inline**.

Las etiquetas **inline** pueden estar dentro de otras etiquetas **block** y/o **inline**.



# HTML 5 - Etiquetas

Todas las etiquetas pueden tener atributos para ampliar su funcionalidad.

De hecho algunas etiquetas tienen atributos obligatorios.

Si revisamos la estructura básica de una página web HTML5:

`<meta charset="utf-8">`



**atributo**



**valor**

# Cabeceras: <h1></h1>...<h6></h6>

block

Son textos de cabecera de diferentes niveles. A número más pequeño el tamaño será más grande. Aunque es una etiqueta que indica estilo se puede utilizar ya que da información semántica.

```
<body>  
  <h1>Esto es una cabecera de primer nivel, tamaño grande.</h1>  
  <h2>Esto es una cabecera de segundo nivel.</h2>  
  <h6>Esto es una cabecera de sexto nivel, tamaño pequeño.</h6>  
</body>
```

**Esto es una cabecera de primer nivel, tamaño grande.**

**Esto es una cabecera de segundo nivel.**

Esto es una cabecera de sexto nivel, tamaño pequeño.

# Nueva línea: `<br>`

block

Se utiliza para pasar a la siguiente línea.

`<body>`

Como sabemos HTML agrupa en uno todos los

Por ello espacios, tabuladores

y

saltos de línea sirven para tener un código más limpio.

`<body>`

espacios  
`<br>`que estén juntos.  
`<br>`

**Como sabemos HTML agrupa en uno todos los espacios que estén juntos.**

**Por ello espacios, tabuladores y saltos de línea sirven para tener un código más limpio.**

# Párrafos: <p></p>

block

Se utiliza para separar en párrafos el contenido de la página web.

<p>Separar en párrafos el texto de una página nos da dos ventajas.</p>

<p>Primera: que cada párrafo se convierte en un bloque al ser etiqueta de bloque.</p>

<p>Segunda: No tenemos que incluir etiquetas br entre los párrafos.</p>

**Separar en párrafos el texto de una página nos da dos ventajas.**

**Primera: que cada párrafo se convierte en un bloque al ser etiqueta de bloque.**

**Segunda: No tenemos que incluir etiquetas br entre los párrafos.**

# Enlaces: <a href="URL\_destino"></a>

inline

Esta etiqueta es fundamental en las páginas web, ya que sin hipertexto internet no sería lo que es ni tendría la potencia que tiene. La URL destino puede ser a cualquiera de las siguientes opciones:

- Un lugar dentro de la propia página – Enlace interno.
- Un archivo de nuestro servidor/dominio – Enlace relativo o absoluto.
- Enlace absoluto a una página de un servidor externo.
- Enlace a otro protocolo. Por ejemplo email: mailto:nombre@dominio.com

<p>

Los enlaces son súper importantes en las páginas web ya que sin ellos internet no tendría la potencia que tiene. Veamos algún ejemplo de enlace:

<a href="http://google.com">Visita Google</a>

<a href="/index.html">Ir a la página principal del sitio</a>

</p>

**Los enlaces son súper importantes en las páginas web ya que sin ellos internet no tendría la potencia que tiene. Veamos algún ejemplo de enlace: [Visita Google](#) [Ir a la página principal del sitio](#)**

# Ejercicio

Crea una carpeta llamada HTML-ejercicios y dentro crea los siguientes archivos html, dentro de cada uno pon lo que se indica:

- principal.html  
Ha de tener una presentación tuya como desarrollador web y a continuación un enlace a cada una de las páginas siguientes. También tiene que tener un enlace para que se permita escribirte un correo electrónico directamente: `mailto:tu@mail.com`
- tecnologias.html  
Enumera las tecnologías web que usas en tus desarrollos  
A continuación pon un enlace a la página principal.
- rrss.html  
Pon dos enlaces uno a facebook y otro a twitter (a sus páginas principales) simula con el texto que se muestra que enlazan a tus perfiles en esas RRSS aunque realmente irán a `facebook.com` y `twitter.com`  
Añade un enlace a la página principal.

# Imágenes: ``

inline

Mediante esta etiqueta podemos insertar imágenes en nuestra página web. La ruta puede ser a nuestro servidor local o a una imagen en otro servidor de internet, en este segundo caso ojo con el **hotlinking**.

`<body>`

Es importante tener en cuenta que las imágenes son elementos de línea y que el tamaño de la imagen se pondrá con CSS al ser parte del estilo.  
`<br>`

Veamos una imagen: ``

`</body>`

**Es importante tener en cuenta que las imágenes son elementos de línea y que el tamaño de la imagen se pondrá con CSS al ser parte del estilo.**

**Veamos**



**imagen:**

# Ejercicio

En el archivo principal.html añade una foto tuya.

En el archivo rrss.html añade una imagen a cada enlace de las redes sociales.



# Listas: `<ol>` `<ul>` `<li>`

block

Permiten realizar listas de elementos ordenadas/desordenadas con posibilidad de anidar unas listas dentro de otras.

`<li></li>` - list item: sirve para definir un elemento dentro de una lista ya sea ordenada o desordenada. Cada elemento de la lista debe ir etiquetado con `<li></li>`.

`<ol></ol>` - ordenated list: define una lista ordenada, cada `<li></li>` que aparezca dentro será un elemento de la lista ordenada y tendrá un número.

`<ul></ul>` - unordenated list: define una lista desordenada, cada `<li></li>` que aparezca dentro será un elemento de la lista desordenada y un guion delante.

# Listas: <ol> <ul> <li>

block

```
<body>
  Los colores que me gustan:
  <ul>
    <li>Rojo</li>
    <li>Verde</li>
    <li>Amarillo</li>
    <li>Azul</li>
  </ul>
</body>
```

```
<body>
  Los colores que me gustan
  ordenados de más a menos:
  <ol>
    <li>Rojo</li>
    <li>Verde</li>
    <li>Amarillo</li>
    <li>Azul</li>
  </ol>
</body>
```

Los colores que me gustan:

- Rojo
- Verde
- Amarillo
- Azul

Los colores que me gustan ordenados de más a menos:

1. Rojo
2. Verde
3. Amarillo
4. Azul

# Ejercicio

Convierte a una lista desordenada el conjunto de tecnologías que pusiste en el archivo `tecnologias.html`

En la página principal añade un enlace a esta nueva página.

Agrupar: `<span></span>` **inline**

`<div></div>` **block**

Son dos etiquetas que visualmente no hacen nada, pero permiten etiquetar/encapsular elementos. Encapsular elementos nos permite aplicarles posteriormente un estilo mediante CSS.

```
<body>
```

```
  Ejemplo de encapsulación <span>para posteriormente</span> poder aplicar estilo.
```

```
  <div>
```

```
    <div>A esta parte podremos darle un estilo.</div>
```

```
    <div>Y a esta podremos darle un estilo diferente</div>
```

```
  </div>
```

```
</body>
```

**Ejemplo de encapsulación para posteriormente poder aplicar estilo.**

**A        esta        parte        podremos        darle        un        estilo.**

**Y        a        esta        podremos        darle        un        estilo        diferente**

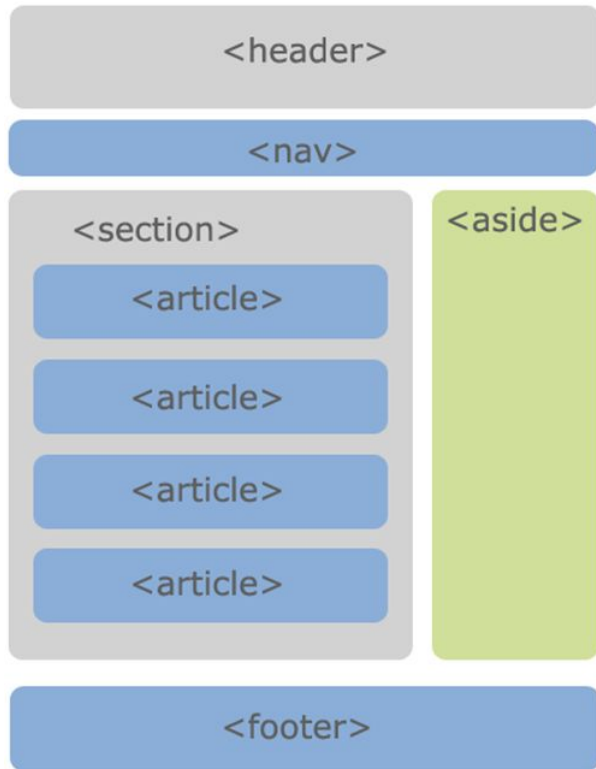
# Etiquetas semánticas

block

- <header>** Equivalente a la cabecera de la página web. Contiene el título, nombre...
- <nav>** Contiene enlaces de navegación, sería como una barra de menú.
- <section>** Es un gran contenedor dentro de la página web.
- <article>** Contenedor independiente de contenido que puede o no estar dentro un <section>.
- <aside>** Contenidos vagamente relacionados con el resto del contenido de la página. Si no es visualizado, el contenido restante seguirá teniendo sentido. Por ejemplo, anuncios, u otros contenidos.
- <footer>** Sería el pie de una sección o de toda la página web.
- <audio>** Permite insertar un sonido a reproducir
- <video>** Permite insertar un vídeo a reproducir. Antes de HTML5 se usaba un reproductor **Flash**.

# Etiquetas semánticas

block



```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ejemplo</title>
  </head>

  <body>
    <header>
      <h1>Ejemplos cabecera</h1>
    </header>
    <nav>
      <ul>
        <li><a href="index.html">Inicio</a></li>
        <li>Ejemplos HTML</li>
        <ul>
          <li><a href="a.html">Página a</a></li>
          <li><a href="b.html">Página b</a></li>
        </ul>
        </li>
      </ul>
    </nav>
    <section>
      <article><h1>1ª Sección</h1></article>
      <article><h1>2ª Sección</h1></article>
    </section>
    <footer>Pie de página</footer>
  </body>
</html>
```

# Ejercicio

Aplica las etiquetas semánticas a los archivos html que tienes en la práctica.

# Tablas: <table> </table>

block

Sólo hay que utilizar tablas cuando se desea mostrar información de forma tabulada, por ejemplo un horario, un calendario o la clasificación de una liga.

Mediante la etiqueta <table> se delimita el espacio que ocupará la tabla. A continuación hay que definir las filas de la tabla y dentro de cada fila habrá que definir cada una de las celdas. **Todas las filas han de tener el mismo número de celdas.**

# Tablas: <tr> </tr>

block

Delimita una fila de la tabla y contendrá al menos una celda.

# Tablas: <td> </td>

block

Delimita una celda dentro de una fila.



# Tablas:

block

```
<table>
  <tr>
    <td>Nombre</td>
    <td>1er Apellido</td>
    <td>2o Apellido</td>
    <td>email</td>
  </tr>
  <tr>
    <td>Ana</td>
    <td>Martínez</td>
    <td>García</td>
    <td>amargar@gmail.com</td>
  </tr>
  <tr>
    <td>Juan</td>
    <td>Fernández</td>
    <td>Fernández</td>
    <td>juferfer@gmail.com</td>
  </tr>
</table>
```

Nombre	1er Apellido	2o Apellido	email
Ana	Martínez	García	amargar@gmail.com
Juan	Fernández	Fernández	juferfer@gmail.com

\*Las celdas por defecto no tienen ningún estilo asociado por lo cual visualmente pueden no parecernos una tabla.

# Tablas:

block

Existe otra etiqueta para introducir celdas en una fila: `<th></th>`

La única diferencia con `<td>` es que con `<th>` el texto de la celda sale por defecto centrado y en negrita.

Aunque aporta estilo se puede utilizar por su carácter semántico, la **h** significa **header** (cabecera).

# Tablas:

block

Existen tres etiquetas que sirven para añadir valor semántico a la tabla, estas etiquetas solamente sirven para agrupar conjuntos de filas:

```
<thead></thead>
```

```
<tbody></tbody>
```

```
<tfoot></tfoot>
```

```
<table>
  <thead>
    <tr>
      <td>Nombre</td>
      <td>1er Apellido</td>
      <td>2o Apellido</td>
      <td>email</td>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Ana</td>
      <td>Martínez</td>
      <td>García</td>
      <td>amargar@gmail.com</td>
    </tr>
    <tr>
      <td>Juan</td>
      <td>Fernández</td>
      <td>Fernández</td>
      <td>juferfer@gmail.com</td>
    </tr>
  </tbody>
</table>
```

# Tablas:

block

Hemos visto antes que todas las filas tienen que tener el mismo número de celdas. Esto se puede variar si indicamos que una celda ocupe el espacio de varias. Se puede hacer tanto en horizontal como en vertical (más dificultad).

En horizontal: atributo **colspan="valor"**

```
<table>
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
  </tr>
  <tr>
    <td colspan="2">4</td>
    <td>5</td>
  </tr>
</table>
```

1	2	3
4		5

# Tablas:

block

En horizontal: atributo **rowspan="valor"**

```
<table>
  <tr>
    <td rowspan="2">1</td>
    <td>2</td>
    <td>3</td>
  </tr>
  <tr>
    <td>4</td>
    <td>5</td>
  </tr>
</table>
```

1	2	3
	4	5

# Tablas:

block

En vertical y horizontal:

```
<table>
  <tr>
    <td rowspan="2">1</td>
    <td>2</td>
    <td>3</td>
  </tr>
  <tr>
    <td colspan="2">4</td>
  </tr>
</table>
```

1	2	3
	4	

# Tablas:

```
<table>
  <tr>
    <td>1</td>
    <td rowspan="3">2</td>
    <td>3</td>
    <td>4</td>
  </tr>
  <tr>
    <td>5</td>
    <td>6</td>
    <td>7</td>
  </tr>
  <tr>
    <td>8</td>
    <td colspan="2">9</td>
  </tr>
  <tr>
    <td colspan="2" rowspan="2">10</td>
    <td>11</td>
    <td>12</td>
  </tr>
  <tr>
    <td>13</td>
    <td>14</td>
  </tr>
</table>
```

block

1	2	3	4
5		6	7
8		9	
10		11	12
		13	14

# Etiquetas semánticas

block

Existen atributos obligatorios específicos de una etiqueta

`<meta charset="...">`

`<a href="...">...</a>`

``

También existen una serie de atributos se pueden poner en cualquier etiqueta. Los más importantes son **id** y **class**, gracias a ellos se facilita mucho tanto aplicar estilos CSS como generar interacción con JavaScript.

- id: Permite poner un **nombre** o **identificador único**.

`<div id="presentación">Bienvenidos a este sitio web en el...</div>`

- class: Permite **agrupar diferentes etiquetas** (no tienen por qué ser la misma etiqueta), todas las etiquetas que tengan la misma clase pertenecerán al mismo grupo.

`<a class="enlace_interno" href="hobbies.html">Mis hobbies</a>`



# Formularios

block

Los formularios son la base de la web desde la web colaborativa (2.0), mediante éstos el usuario puede introducir datos para enviarlos al servidor (anteriormente solo se accedía mediante enlaces).

- Registro en la aplicación.
- Logueo en la aplicación.
- Búsqueda.
- Comentarios.
- Cambiar visualización de la página web..

```
<form action="URL_destino" method="metodo_envio">  
    Aquí irá todo el cuerpo del formulario  
</form>
```

# Formularios

block

```
<form action="URL_destino" method="metodo_envio">  
  Aquí irá todo el cuerpo del formulario  
</form>
```

**action:** indica qué archivo en el servidor recibirá los datos del formulario.

**post:** indica la manera de enviar los datos

- **get:** los datos se envían en la URL (típico en los formularios de búsqueda).
- **post:** los datos se envían en la petición de la página web.

Si los datos son sensibles (usuarios, contraseñas...) lo mejor es utilizar **post**.

# Formularios - Elementos

inline

**input** - puede tener diferentes tipos, cada uno tiene una función, algunos tipos son:

```
<input type="text" name="user">
```

- text - Para introducir texto.
- password - Para introducir contraseña.
- color - Para seleccionar un color.
- date - Para introducir una fecha.
- email - Para introducir una dirección de correo electrónico
- range - Barra horizontal para seleccionar un valor de un rango.
- search - Para introducir un texto que se ha de buscar.
- number - Para introducir números.
- checkbox - Casilla seleccionable
- radio - Para seleccionar entre varias opciones.
- button - Botón sin funcionalidad específica.
- submit - Botón para enviar la información a la URL del action.
- reset - Botón que borra todos los datos introducidos en el formulario.

# Formularios - Elementos

inline

```
<form action="#" method="get">
```

Campo de texto: `<input type="text" name="user">`

Campo de contraseña: `<input type="password" name="pass">`

Selección de color: `<input type="color" name="color">`

Selección de fecha: `<input type="date" name="fecha">`

Campo de email: `<input type="email" name="email">`

Selección de rango: `<input type="range" name="rango">`

Campo de búsqueda: `<input type="search" name="busqueda">`

Campo numérico: `<input type="number" name="numeros">`

Acepto las condiciones `<input type="checkbox">`

El elemento radio suele utilizarse en grupo:

`<input type="radio" name="gender" value="hombre">Hombre`

`<input type="radio" name="gender" value="mujer">Mujer`

`<input type="radio" name="gender" value="otro">Otro`

`<input type="button" value="Esto es un botón sin funcionalidad">`

`<input type="reset">` `<input type="submit">`

```
</form>
```

Campo de texto:

Campo de contraseña:

Selección de color:

Selección de fecha:

Campo de email:

Selección de rango:

Campo de búsqueda:

Campo numérico:

Acepto las condiciones ☐

El elemento radio suele utilizarse en grupo: ☐ Hombre ☐ Mujer ☐ Otro

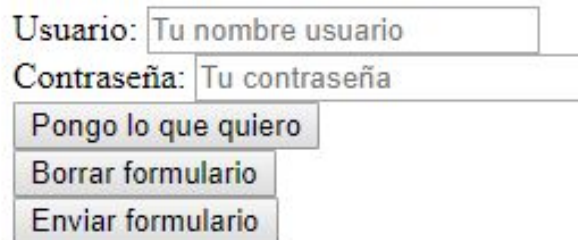
# Formularios - Elementos

inline

Los elementos **input** permiten una serie de atributos, entre ellos:

- name - necesario para que los datos se envíen al hacer **submit**, ya sea por get o por post.
- required - Campo obligatorio.
- value - Es el valor por defecto del campo, usado sobre todo en los botones.
- placeholder - Es un texto de ayuda.

```
<form action="#" method="get">  
  Campo de texto: <input type="text" name="user">  
  Campo de contraseña: <input type="password" name="pass">  
  
  <input type="button" value="Pongo lo que quiero">  
  <input type="reset" value="Borrar formulario">  
  <input type="submit" value="Enviar formulario">  
</form>
```



Usuario:

Contraseña:

# Formularios - Elementos

inline

**select** sirve para crear desplegables para seleccionar una opción

```
<form action="#" method="get">  
  Selecciona un Sistema Operativo:  
  <select name="so">  
    <option value="windows">Windows</option>  
    <option value="linux">Linux</option>  
    <option value="mac">Mac</option>  
    <option value="android">Android</option>  
    <option value="ios">iOS</option>  
  </select>  
</form>
```

Selecciona un Sistema Operativo:

Windows ▼

Selecciona un Sistema Operativo:

Windows ▼  
Windows  
Linux  
Mac  
Android  
iOS

# Formularios - Elementos

inline

**select** sirve para crear desplegables para seleccionar una opción

```
<form action="#" method="get">
  Selecciona un Sistema Operativo:
  <select name="so">
    <optgroup label="So Escritorio">
      <option value="windows">Windows</option>
      <option value="linux">Linux</option>
      <option value="mac">Mac</option>
    <optgroup label="SO Móvil">
      </optgroup>
      <option value="android">Android</option>
      <option value="ios">iOS</option>
    </optgroup>
  </select>
</form>
```

Selecciona un Sistema Operativo:

Windows ▼

Selecciona un Sistema Operativo:

Windows ▼  
SO Escritorio  
Windows  
Linux  
Mac  
SO Móvil  
Android  
iOS

# Formularios - Elementos

inline

**textarea** se utiliza para poder introducir texto sin límite

```
<form action="#" method="get">  
  Introduce un comentario:  
  <textarea name="comentario"></textarea>  
</form>
```

Introduce un comentario:



# Formularios - Elementos

inline

**fieldset** mediante esta etiqueta se pueden agrupar elementos del formulario y mediante la etiqueta **legend** se le pone un encabezado.

```
<form action="#" method="get">
  <fieldset>
    <legend>Datos personales</legend>
    Nombre <input type="text" name="nombre"><br>
    Apellidos <input type="text" name="apellidos">
  </fieldset>

  <fieldset>
    <legend>Gustos</legend>
    <input type="checkbox" name="f1">Formula 1<br>
    <input type="checkbox" name="moto">MotoGP<br>
    <input type="checkbox" name="futbol">Fútbol<br>
    <input type="checkbox" name="baloncesto">Baloncesto
  </fieldset>
</form>
```

Datos personales

Nombre

Apellidos

Gustos

☐ Formula 1

☐ MotoGP

☐ Fútbol

☐ Baloncesto

# Formularios - Elementos

inline

**label:** asocia el texto de la etiqueta label a un campo y cuando se hace clic sobre ese texto se lleva el foco al campo asociado.

Es necesario usar el atributo **id** y se recomienda que el valor de **id** sea el mismo que el atributo **name**.

```
<form action="#" method="get">
  <label for="nombre">Nombre</label>
  <input type="text" required name="nombre" id="nombre">
  <br>
  <label for="apellidos">Apellidos</label>
  <input type="text" required name="apellidos" id="apellidos">
</form>
```



Nombre

Apellidos

# Ejercicio

En el archivo principal.html añade un formulario en el que el usuario pondrá sus datos de contacto y podrá enviarte una consulta.

# HTML - ESTILO

Recordad:

El estilo siempre se aplicará con archivos .CSS, colocados en una carpeta llamada css.

# Referencia HTML

<https://www.w3schools.com>