

# FUNDAMENTOS DEL DESARROLLO DE PÁGINAS WEB



## Módulo 1





# CLASE 4

Introducción

Contenedores

Características de los formularios

Estructura de los formularios

Tipos de inputs

Envío de información

Iframes

Navegar en la página

---

# Introducción



En la clase de hoy comenzaremos a trabajar con los Formularios.

Más de una vez se ha cruzado con páginas web donde la sección de “Contacto” está compuesta de un formulario y los datos para comunicarse con esa empresa o persona.

Pero para los programadores, un formulario es más que realizar casillas y botones de “enviar”.

Así que daremos paso a los conceptos fundamentales para poder realizar un formulario que pueda enviar información a nuestra cuenta de email.

# Los contenedores `<div>` y `<span>`

Las **etiquetas** `<div>` y `<span>` son contenedores vacíos que nos sirven para agrupar el contenido de una página. Serían como cajas de vidrio que lo rodean para, luego, poder modificarlo con CSS.

Los `<div>` **son tags de tipo block**. Es decir, ocupan todo el ancho de la pantalla y desplazan a los elementos externos hacia arriba y hacia abajo. Generalmente se usan para agrupar grandes secciones del documento.

Los `<span>`, en cambio, **son tags de tipo inline**. Es decir, afectan únicamente el espacio que ocupa su contenido. Generalmente se usan para seleccionar partes específicas de un documento, como una frase de un párrafo, una palabra o una imagen.

Source Code:	Submit Code »	Result:	<a href="https://www.w3schools.com">W3Schools.com</a> - Try it yourself
<pre>&lt;div&gt;This is a div&lt;/div&gt; &lt;div&gt;This is a div&lt;/div&gt; &lt;div&gt;This is a div&lt;/div&gt;  &lt;span&gt;This is a span&lt;/span&gt; &lt;span&gt;This is a span&lt;/span&gt; &lt;span&gt;This is a span&lt;/span&gt;</pre>		<pre>This is a div This is a div This is a div This is a span This is a span This is a span</pre>	

# Características de los formularios

Los formularios que existen en la web son modulares, es decir, se construyen a partir de distintos tipos de módulos o inputs.

Un input es un campo de datos que permite que el usuario ingrese o elija un valor. Es decir, es una de las maneras que tenemos de recolectar su información.

Hay muchos tipos de input, desde espacios vacíos para que el usuario complete con letras o números, listas desplegables, campos para escribir una contraseña, etc. En esta unidad, veremos sólo los más comunes.

Contáctanos

Nombre

Apellido

Email \*

Escribe un mensaje

¡Gracias por tu mensaje!

# Estructura de los formularios

## Etiqueta `<form>`

La etiqueta `<form>` define el comienzo y el fin de un formulario HTML. Este tag se abre y se cierra normalmente.

Tiene dos atributos principales: `action` y `method`:

- `action`: Define a dónde debe mandar la información del formulario. Puede ser una URL o carpeta dentro de un servidor.
- `method`: Define qué método HTTP usará. Puede ser `get` si pide información o `post` si la envía.

# Estructura de los formularios

## Etiqueta `<input>`

Dentro de `<form>` se escriben los tags `<input>` que crean los campos donde el usuario ingresa la información.

Tienen el **atributo `type` que define el tipo de información** que recibe. Por ejemplo:

- `type="text"`: Define un campo donde el usuario puede escribir cualquier texto.
- `type="password"`: Oculta la información al momento de escribirla. Generalmente, reemplaza los caracteres ingresados por asteriscos o puntos.
- `type="email"`: Define un campo donde el usuario debe escribir una dirección de correo electrónico.
- `type="submit"`: Genera un botón que envía la información del formulario.

**Importante:** Los tags `<input>` **son autocontenidos**, es decir, no se cierran.

Veamos cómo se ve en el código:

```
<form method="POST" action="/login" >
  <input type="text">
  <input type="password">
  <input type="email">
  <input type="submit">
</form>
```

Cuando apretamos **submit**, el formulario envía la información ingresada en forma de key-value pairs. Es decir, genera una variable para guardar el contenido de cada `<input>`. Por lo tanto, cada `<input>` deberá tener un atributo `name` que indique a qué campo corresponde la información ingresada.

Por ejemplo, si queremos enviar la edad de una persona, el servidor recibirá un campo `edad` y el valor numérico que ingresó el usuario. El siguiente código enviará un campo con el nombre del usuario, otro con su edad y otro con su contraseña:

```
<form>
  <input type="text" name="nombre">
  <input type="number" name="edad">
  <input type="password"
name="contraseña">
  <input type="submit">
</form>
```

Si prueba el siguiente código y se fija en la URL verá que ahora contiene un signo de pregunta (?) seguido por la información de cada campo, separado por un signo et (&). A esta parte de la URL se la llama **Query String**.

Los **Query String** son bloques de texto sin espacios que se usan para enviar información. El espacio se reemplaza por **%20** y otros símbolos (como las tildes) que tienen sus propias formas de escribirse en este formato.





## *Input Password*

El `<input type="password">` oculta la información que ingresa el usuario usando **asteriscos o puntos**.

Si bien cuando escribimos en el campo con `type="password"` el texto aparece oculto, al apretar submit, la información se envía sin encriptar. Esto, por ahora, nos sirve para generar una mejor experiencia del usuario; sin embargo, no es la forma más segura.

**Importante:** Para mantener segura la información y enviarla encriptada, se usan sistemas complejos del lado del servidor que estaremos estudiando dentro del Módulo 8 de nuestro Bootcamp.

# El atributo *Placeholder*

El atributo **placeholder** se agrega dentro del tag `<input>` y sirve como **referencia visual** para el usuario.

Es decir, es el texto que se muestra en el `<input>` y que indicará al usuario qué debe escribir ahí.

Así se ve en el código:

```
<form>
  <input type="text" name="nombre" placeholder="Ingrese su nombre" >
  <input type="number" name="edad" placeholder="Ingrese su edad" >
  <input type="password" name="contraseña" placeholder="Ingrese su
contraseña" >
  <input type="submit">
</form>
```

Y esto es lo que ve el usuario:

El placeholder no genera un valor dentro del campo; es decir, **la información nunca se envía porque es solo una referencia visual** para el usuario.



# El atributo *Label*

El tag `<label>`, en cambio, es la etiqueta que **lleva el nombre del** `<input>`.

Nombre:  Edad:  Contraseña:

Hay dos formas de usarlo:

1. Escribiendo el nombre del campo y el tag `<input>` dentro de `<label>`:  
`<label>Nombre: <input type="text" name="username" placeholder="Ingrese su nombre"> </label>`
2. Escribiendo dentro del tag `<label>` un atributo `for` que, a su vez, tome el valor del `id` que le daremos al `input`.

```
<label for="nombre_usuario">Nombre:</label> <input type="text"
id="nombre_usuario" name="username" placeholder="Ingrese su nombre">
```



Hasta aquí tenemos mucha información... Así que pondremos manos al código...

Realizaremos un InstaPet. Para eso debe seguir el siguiente paso a paso:

1. Cree un documento HTML.
2. Dele la bienvenida a los usuarios a su "InstaPet". Recuerde de titularlo con un `<h1>` y de agregarle un párrafo breve que describa de qué se trata la página.
3. Haga un formulario que le pida al usuario:
  - a. Nombre de la mascota.
  - b. Edad de la mascota.
  - c. Correo electrónico.
  - d. Contraseña.
4. Utilice un tag `<label>` a cada campo, alternando entre las dos formas vistas en la sección anterior.
5. Agregue los placeholder a cada `<input>`.
  - Googlee qué input type le permite ingresar el color de su mascota y agregalo a su formulario.
  - Haga que todos los campos sean obligatorios.

# Dropdown

Un **dropdown** es una **lista desplegable**. Es decir, es un tipo de input que le muestra las opciones disponibles dentro de un menú.

Para realizarlo seguiremos los siguientes pasos:

1. Escribimos un **tag** `<select>` con su opening y closing tag. Ellos contendrán las opciones disponibles.
2. Luego, escribimos un **atributo name** que indique en tu Query String a qué campo corresponde la opción elegida.
3. Ingresamos los valores disponibles usando un **tag** `<option>` para cada una de ellos.
4. Agregamos un **atributo value** que indique cómo queremos que aparezca en la Query String la opción elegida por el usuario.
  - *Importante:* Puede tener el mismo valor que la *option*, una abreviación o cualquier otra nomenclatura. En caso de que no agregue ese atributo, aparecerá en la Query String el valor dado en la option.

# Radio Buttons

Los **radio buttons son botones** que nos permiten seleccionar una opción entre dos o más dadas. La idea de este tipo de input es que no se puedan seleccionar ambos en simultáneo.

Para crearlos deberá escribir:

- Un tag `<label>` que indique el nombre que aparecerá antes del radio button.
- Un tag `<input>` que tenga estos 3 atributos:
  - **name:** Indicará a qué variable refiere el valor del input al momento de enviar la información. Es decir, en la Query String, el campo "name" tendrá el valor de la opción que haya elegido el usuario.
  - **type:** Indicará qué tipo de botón es, en este caso, radio.
  - **value:** Indicará cómo quiere que aparezca en la Query String la opción elegida por el usuario (puede tener el mismo valor que la label, una abreviación o cualquier otra nomenclatura).

¿Cómo hacer para que los radio button sean excluyentes entre sí?

Si el nombre dado al **atributo name es el mismo, los radio buttons serán excluyentes entre sí.**

Es decir, los usuarios solo podrán elegir una de las opciones dadas.

# Checkbox



Los **checkbox** son similares a los radio button, excepto porque permiten **seleccionar y deseleccionar la opción**.  
Por ejemplo, el checkbox de "Acepto Bases y Condiciones".

El checkbox tiene un tag `<label>` que indica el nombre que aparecerá antes. Además, tiene un tag `<input>` con 3 atributos:

- **name:** Indicará a qué variable refiere el valor del input al momento de enviar la información. Es decir, en la Query String, el campo "name" tendrá el valor de la opción que haya elegido el usuario.
- **type:** Indicará qué tipo de botón es, en este caso, checkbox.
- **value:** Indicará cómo aparecerá en la Query String la opción elegida por el usuario.

Importante: Puede tener el mismo valor que la label, una abreviación o cualquier otra nomenclatura.

Siguiendo el ejemplo anterior, quedaría así el código del checkbox:

```
<label for="checkbox">Acepto recibir información sobre el curso: </label>  
  
    <input id="checkbox" name="mailing" type="checkbox" value="consent">
```

Pruébalo en su editor de texto y vea cómo se ve en su browser.

# Etiqueta `<textarea>`

Un **tag** `<textarea>` es un campo para **ingresar texto**. Es similar a un `<input type="text">` solo que el espacio dado al usuario es mayor.

**Importante:** No es un tag `<input>` con un `type="textarea"`, sino **una etiqueta diferente**, que se abre y se cierra.

La etiqueta `<textarea>` tiene un opening y closing tag y los siguientes atributos:

- **rows (filas):** Es un valor numérico que indica el alto de la caja de texto. Por ejemplo, "20".
- **cols (columnas):** Es un valor numérico que indica el ancho de la caja de texto. Por ejemplo, "40".

**Importante:** Si quiere, puede agregarle un atributo placeholder para que sea más claro para el usuario.

Así queda el código:

```
<label for="consultas">¿Tiene alguna otra duda?</label>
```

```
<br>
```

```
<textarea id="consultas" rows="20" cols="40" placeholder="Dejanos tu consulta"></textarea>
```

Vealo en su browser, copie y pegue el código anterior en su editor de texto.



# Envío de información

[Formspree](#) es un servicio en línea que nos permite dar funcionalidad a formularios de páginas web estáticas, es decir, páginas web con HTML y CSS.

Para integrar formspree a nuestra página web no es necesario conocer o dominar ningún lenguaje de programación ni programar absolutamente nada del lado del backend.

Lo primero que haremos será dar de alta nuestra cuenta en el en el sitio web de formspree. Es completamente gratis.

Una vez hayamos verificado nuestra cuenta ya seremos capaces de hacer uso de formspree.

La forma más sencilla de integrar formspree es haciendo uso de nuestro user\_id. Para ello click sobre *new form* e *integration*. Obtendremos el siguiente ejemplo:

```
<form action="https://formspree.io/<USER_ID>" method="POST">
  <input type="text" name="name">
  <input type="email" name="_replyto">
  <input type="submit" value="Send">
</form>
```



El formulario anterior es básico, sin embargo podemos cambiar los campos si ningún problema. Además, observamos que la url de action hace referencia a nuestra cuenta.

Es importante colocar nuestro *user\_id* en la dirección de envío, además que todos los campos deben poseer el atributo *\_name*.

Al hacer el submit seremos redirigidos a completar un pequeño captcha, y listo, recibiremos en nuestro correo (el que utilizamos para crear la cuenta) la información enviada a través del formulario.

# *Iframes*

Un **Iframe** (por *inline frame* en inglés o *marco incorporado*) nos permite mostrar **contenidos de otros HTML** en nuestra página. Pueden ser videos, mapas o, incluso, webs enteras.

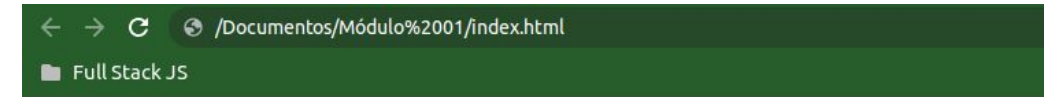
Por ejemplo para integrar un mapa de Google Maps realizaremos los siguientes pasos:

1. Ingresamos un tag **<iframe>**.
2. Agregamos un **atributo src="..."** con la dirección del contenido externo que queremos mostrar.
3. Definiremos un **atributo width="..."** con el **ancho** que le queremos dar y un **atributo height="..."** con el **alto**.

Por ejemplo, si definimos el siguiente código:

```
<iframe
src="https://www.google.com/maps/embed?pb=!1m14!
1m8!1m3!1d13316.982127283809!2d-70.6538699!3d-33
.4429091!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x0%3
A0x5d0fa12b4d88ae82!2sPalacio%20de%20La%20Moneda
!5e0!3m2!1ses-419!2sar!4v1648185087803!5m2!1ses-
419!2sar" width="600" height="450"
style="border:0;" allowfullscreen=""
loading="lazy" referrerpolicy="no-referrer-when-d
owngrade"></iframe>
```

Estaremos obteniendo el siguiente mapa en  
nuestro HTML.



# ¿Cómo direccionar a secciones dentro de la misma página?

1. Primero ponemos un **ID** a la sección a la que queremos direccionar al usuario.
2. Luego creamos un link en otra parte de la misma página poniendo el nombre del ID con un **#** dentro del **atributo href**. Por ejemplo:

```
<a href="#seccion1"> Link a la sección1 <a>
```

3. Ese link lo llevará a la sección1 dentro de la misma página.

Si quiere redireccionar a esta sección desde otra página, deberá agregar el ID al final de la dirección.  
Por ejemplo:

```
<a href="http://www.ejemplo.com/#seccion1"> Link a la sección1 desde una  
página externa<a>
```

## ¿Cómo abrir un mail en blanco?

Es muy común que, al hacer click sobre "Contacto", se abra un **mail en blanco** en nuestra casilla de correo. Para configurarlo, utilice como valor del atributo href:

```
href="mailto:ejemplo@tumail.com"
```



¿Cómo hacer que una página se abra en una pestaña nueva?

Para poder descubrir esto, siga ejercitando esta habilidad clave de los programadores y googlee cómo hacer para que la página se abra en una pestaña nueva.

MÓDULO 1

+

o



.

# GRACIAS

Aquí finaliza la clase n°4 del módulo 1

## OTEC PLATAFORMA 5 CHILE