

Week 2_4(es)

Formularios de HTML

Vamos a crear formularios en HTML donde los usuarios pueden llenar los datos!

De vez en cuando los usuarios tienen que llenar la información y enviarla al servidor. Para hacerlo, necesitamos entradas y un botón para enviar como se muestra abajo.

Create a New Account

It's free and always will be.

First name Last name

Mobile number or email

New password

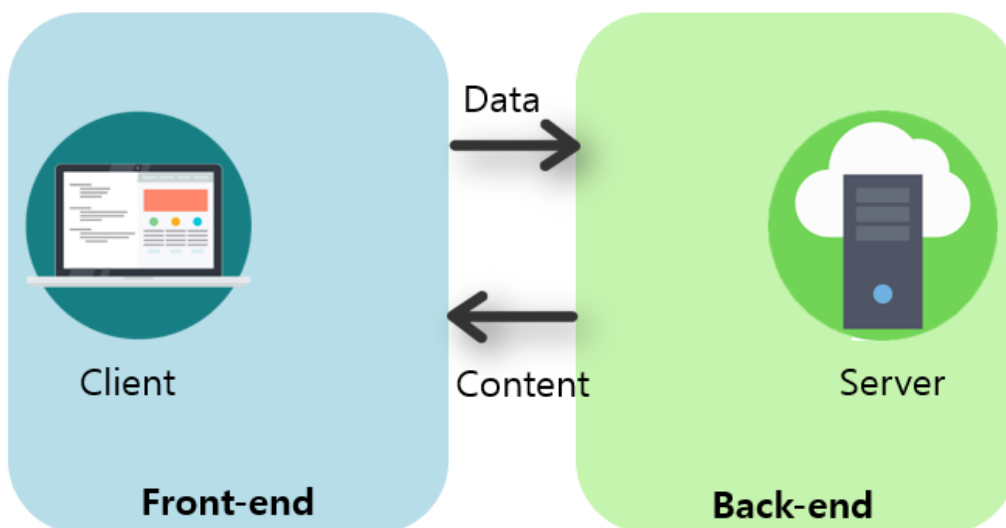
Birthday

Jul 14 1993 [Why do I need to provide my birthday?](#)

Female Male

By clicking Sign Up, you agree to our [Terms](#), [Data Policy](#) and [Cookies Policy](#). You may receive SMS Notifications from us and can opt out any time.

Mandar datos y guardarlos es el trabajo de back end, que no es el tema de nuestro curso. Pero, diseñar el formulario es el trabajo de front end.



Antes de empezar, vamos a crear un archivo nuevo nombrado `form.html`.

Formulario con Entrada Simple

Crear Entradas

Utilizamos la etiqueta `<form>` para indicar un formulario total con un botón de enviar.

La etiqueta `<input>` crea un entrada y la etiqueta `<label>` muestra la etiqueta de la entrada.

```
<form>
  <label>First Name:</label>
  <input type="text">
</form>
```

Como pueden ver, el elemento `<input>` es un elemento vacío (empty element).

Se ve bien porque la etiqueta y la entrada están situadas una al lado de la otra.

Pero necesitamos conectar la etiqueta de la entrada al elemento `<input>` en el código.

Queremos mover el cursor automáticamente a la entrada(field) cuando hacemos el clic en el nombre de la etiqueta.

Para hacerlo, utilizamos el atributo `for` en la etiqueta `<label>` como un valor y ponemos el mismo valor en el atributo `id` de la etiqueta `<input>`.

```
<form>
  <label for="firstName">First Name:</label>
  <input type="text" id="firstName">
</form>
```

Vamos a revisarlo en el navegador!

También es posible conectar la entrada y la etiqueta si hacemos nesting el elemento `<input>` dentro del elemento `<label>` como se muestra abajo.

```
<form>
  <label for="firstName">
    First Name:
    <input type="text" id="firstName">
  </label>
</form>
```

Los resultados de dos códigos parecen iguales, pero el segundo código *no funciona en algunos lectores de pantallas*.

Por eso les recomiendo utilizar el primer código.

Vamos agregar otra etiqueta y entrada para el apellido.

Crear Botón

Ahora, necesitamos un botón para enviar. Para hacerlo, utilizamos la etiqueta `<input>` con el atributo `type` con su valor `"submit"`.

Pueden poner la etiqueta del botón usando el atributo `value`.

```
<input type="submit" value="Save!">
```

Recuerden que el botón no va a funcionar en realidad!

Porque está relacionado a back end, y en este curso estamos construyendo la parte que los usuarios ven que es front end.

Pero, hay un atributo que necesitamos agregar a cada entrada para que el formulario sea accesible a los idiomas de programación (back end) para que pueda utilizar los datos.

Por este propósito existe el atributo `name`.

Utilizamos el atributo `name` para cada entrada(salvo una para el botón) para indicar el nombre de cada entrada. Y entonces el desarrollador de back end puede acceder a cada entrada.

```
<form>
  <label for="firstName">First Name:</label>
  <input type="text" id="firstName" name="firstName">

  <label for="lastName">Last Name:</label>
  <input type="text" id="lastName" name="lastName">

  <input type="submit" value="Save!">
</form>
```

Varios Tipos de Entradas

Email

Pueden usar la etiqueta `<input>` con el tipo(type) `"text"` para recibir correos de los usuarios. Pero existe un tipo de elemento especial solo para los correos, se llama tipo(type) `"email"`.

```
<input type="email">
```

Vamos a agregar las etiquetas y los atributos que hemos aprendido para completar el formulario.

```
<label for="email">Email:</label>
<input type="email" id="email" name="email">

<input type="submit" value="Save!">
```

Es útil usar el tipo `email` porque:

- El navegador automáticamente valida los datos.
- Si ponen un texto que no tenga el formulario de correo, lo va a avisar a los usuarios.
- Muestra el teclado adecuado en los aparatos táctiles.

En las pantallas táctiles, saldrán `@` y `.` automáticamente en el teclado.

No Service 9:13 PM 28%

Cancel accounts.google.com

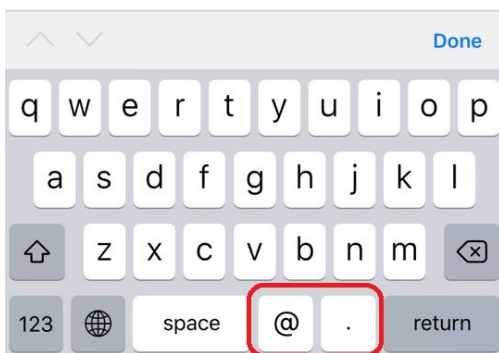
Sign in

Continue to YouTube. You'll also sign in to Google services in your apps & Safari.

Email or phone

Forgot email?

Create account **NEXT**



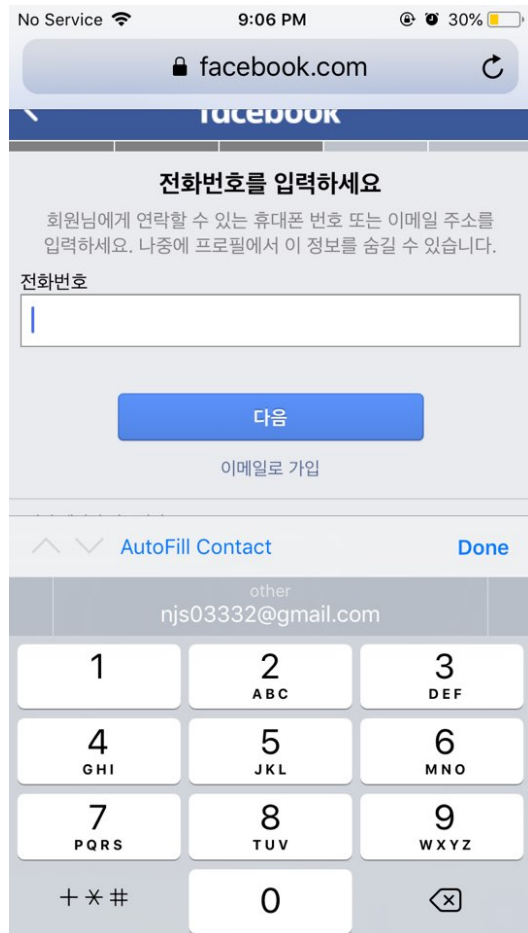
Número de Teléfono

Utilizamos el tipo(type) `"tel"` para poner los números de teléfono. Podemos escribir el código como se muestra abajo.

```
<label for="telephone">Telephone:</label>
<input type="tel" id="telephone" name = "telephone">

<input type="submit" value="Save!">
```

Si utilizamos la etiqueta `<input>` con el valor de `type` como `"tel"`, saldrán los números para marcar el número de teléfono.



[Otros tipos de entradas](#)

Recibir Varias Líneas de Texto

Cuando querremos recibir más que una línea de texto, utilizamos la etiqueta `<textarea>`.

- Al contrario al elemento `<input>`, la etiqueta `<textarea>` no es un elemento vacío(empty element).
- Podemos establecer un valor por defecto(default) colocando un contenido dentro del elemento `<textarea>`.
- El elemento `<textarea>` no requiere el atributo `type`.

```
<label for="yourMessage">Your Message:</label>
<textarea id="yourMessage" name="yourMessage">Thank you.</textarea>
```

Vamos a revisarlo en el navegador!

Los usuarios pueden escribir varias líneas o párrafos si pulsan la tecla ENTER en la entrada.

Elegir entre Opciones

Hay 3 tipos de formularios en que los usuarios pueden elegir opciones.

1. Select (dropdown)

What is your favorite color?

- Red
- Orange
- Yellow
- Blue
- Green
- Purple
- Black
- White
- Skyblue
- Lightgreen

El selectbox es útil cuando:

- Los usuarios ya saben que van a elegir
- Hay muchas opciones y es difícil mostrarlas todas en una pantalla

2. Radio

Which menu do you prefer? (choose one)

Fish Vegetables Salad

El radio es útil cuando:

- Es una decisión importante(tienen que ver todas las opciones)
- Solo hay pocas opciones y se pueden mostrar todas en una pantalla

3. Checkbox

Why do you like HTML? (check all that apply)

It's fun. It's easy. It's exciting. I don't like it. :(:(:(

El checkbox es útil cuando quieren que los usuarios elijan dos o más opciones simultáneamente.

Select Field

Utilizamos la etiqueta `<select>` para envolver la etiqueta `<option>` que significa cada opción.

```
<select>
  <option>Red</option>
  <option>Green</option>
  <option>Blue</option>
</select>
```

Necesitamos usar el atributo `value` en cada opción para que contenga el dato verdadero para transmitirlo al servidor. En este caso, necesitamos enviar el dato en la forma de `colorRed` para la opción Red, `colorGreen` para la opción Green y demás.

```
<select>
  <option value="colorRed">Red</option>
  <option value="colorGreen">Green</option>
  <option value="colorBlue">Blue</option>
</select>
```

Ahora, vamos a agregar el atributo `name` a la etiqueta `<select>`, para que el desarrollador de back end pueda acceder a la etiqueta.

```
<select name="favoriteColor">
```

Vamos a conectar la etiqueta con las opciones usando el atributo `for` y el atributo `id` que hemos aprendido.

```
<label for="favoriteColor">What is your favorite color?</label>
<select id="favoriteColor" name="favoriteColor">
  <option value="colorRed">Red</option>
  <option value="colorGreen">Green</option>
  <option value="colorBlue">Blue</option>
</select>
```

Radio Field

Con el código de abajo, pueden ver un botón de radio en el navegador.

```
<input type="radio">
```

Vamos a agregar las etiquetas.

```
<input type="radio" id="fish"><label for="fish">Fish</label>
<input type="radio" id="vegetables"><label for="vegetables">Vegetables</label>
<input type="radio" id="salad"><label for="salad">Salad</label>
```

Si ponemos el mismo valor en el atributo "name" en cada elemento `<input>`, el navegador sabrá que las tres opciones están **relacionadas** y que los usuarios tiene que elegir **solo una opción** entre las tres. También, necesitamos indicar el dato verdadero para enviar al servidor usando el atributo `value`.

```
<input type="radio" id="fish" name="favoriteMenu" value="menu1"><label for="fish">Fish</label>
<input type="radio" id="vegetables" name="favoriteMenu" value="menu2"><label
for="vegetables">Vegetables</label>
<input type="radio" id="salad" name="favoriteMenu" value="menu3"><label
for="salad">Salad</label>
```

Ahora vamos a agregar las preguntas para los grupos de opciones.

Utilizamos la etiqueta `<legend>` para escribir la pregunta y la etiqueta `<fieldset>` para envolver la pregunta y las opciones.

```
<fieldset>
  <legend>Which menu do you prefer? (choose one)</legend>
  <input type="radio" id="fish" name="favoriteMenu" value="menu1"><label for="fish">Fish</label>
  <input type="radio" id="vegetables" name="favoriteMenu" value="menu2"><label
for="vegetables">Vegetables</label>
  <input type="radio" id="salad" name="favoriteMenu" value="menu3"><label
for="salad">Salad</label>
</fieldset>
```

Checkbox Field

El checkbox field es casi igual que el radio field.

Solo utilizamos `checkbox` en vez de `radio`.

```
<fieldset>
  <legend>Why do you like HTML? (check all that apply)</legend>
  <input type="checkbox" id="fun" name="htmlReason" value="reason1"><label for="fun">It's fun.
</label>
  <input type="checkbox" id="easy" name="htmlReason" value="reason2"><label for="easy">It's
easy.</label>
  <input type="checkbox" id="exciting" name="htmlReason" value="reason3"><label
for="exciting">It's exciting.</label>
  <input type="checkbox" id="no" name="htmlReason" value="reason4"><label for="no">I don't like
it. :( :( :( </label>
</fieldset>
```