Parte 2

Tratar con los datos

1. Tema cinco trabajando con formularios
2. Tema seis Mejorando tus formularios web con HTML5
3. Tema siete Validación de datos de formulario usando HTML5
4. Tema ocho usando microdatos
5. Tema nueve Asegurando la accesibilidad

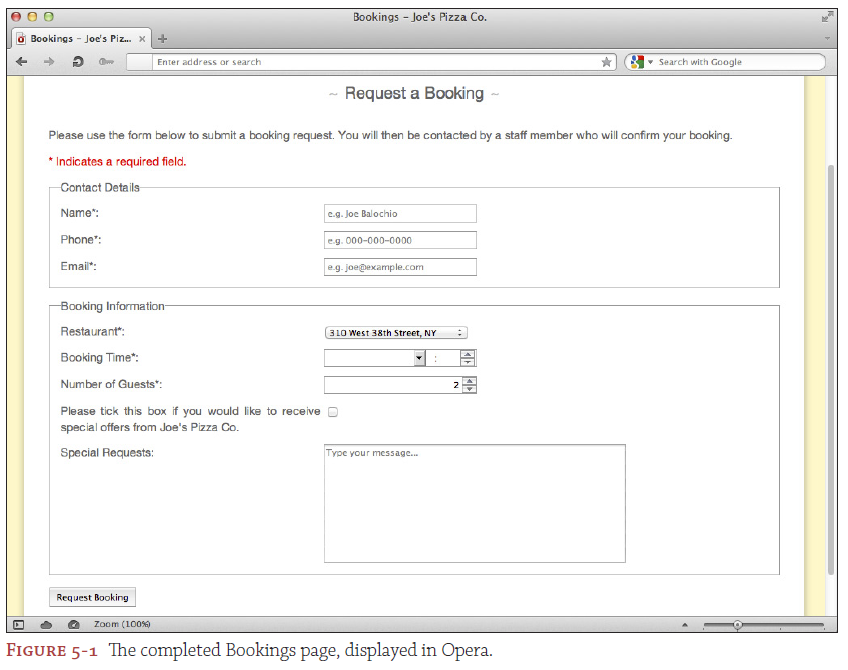
# Presentación

THE WORLD WIDE WEB es la herramienta más grande jamás inventada para recopilar información. Los formularios web son responsables de esto. Los usamos todos los días, ya sea que estemos comprando algo en línea, investigando con nuestro motor de búsqueda favorito o iniciando sesión en nuestro correo electrónico. Los formularios web se han convertido en una parte esencial de nuestra experiencia de navegación diaria; de hecho, sin ellos, Internet sería un lugar bastante aburrido (sí, realmente).

El sitio web de Joe necesita un formulario web para que los clientes puedan reservar una mesa directamente a través del sitio web. Este tema le enseña cómo crear un formulario web simple e insertarlo en sus páginas web. Primero, creará la nueva página de Reservas con un formulario barebones.

Una vez que haya configurado su página, comenzará a agregar una cantidad de campos de formulario diferentes utilizando un rango de elementos HTML, como <input>, <label>, <select> y <textarea>. Terminará actualizando la navegación del sitio y el mapa del sitio para incluir la nueva página de Reservas.

En este tema, solo se centrará en la creación de un formulario básico utilizando elementos que estaban presentes en HTML4. Una vez que haya configurado este formulario, las lecciones 6 y 7 le enseñarán cómo usar las nuevas funciones HTML5 para mejorar sus formularios web. Estas nuevas características incluyen tipos de entrada especializados que facilitan a los usuarios la entrada de datos y herramientas para validar los datos ingresados ​​por los usuarios. La Figura 5-1 muestra la página de Reservas completadas que construirá en estos tres temas.



# Configuración de la página de reservas con el elemento <form>

A lo largo de este tema, trabajará en la creación de un formulario de reservas para el sitio web de Joe´s Pizza Co. que permitirá a los usuarios solicitar reservas a través del sitio web. En esta sección, aprenderá sobre el elemento <form> y también creará la nueva página de Reservas.

Para crear un formulario web, utilice el elemento <form>. Este elemento contendrá todos sus campos de formulario, etiquetas y botones.

El elemento <form> requiere dos atributos, action y méthod, como en el siguiente ejemplo:

<form action=”search.php” method=”GET”>

<div class=”field”>

<label for=”query”>Query:</label>

<input type=”text” name=”query” id=”query”>

</div>

<div class=”field”>

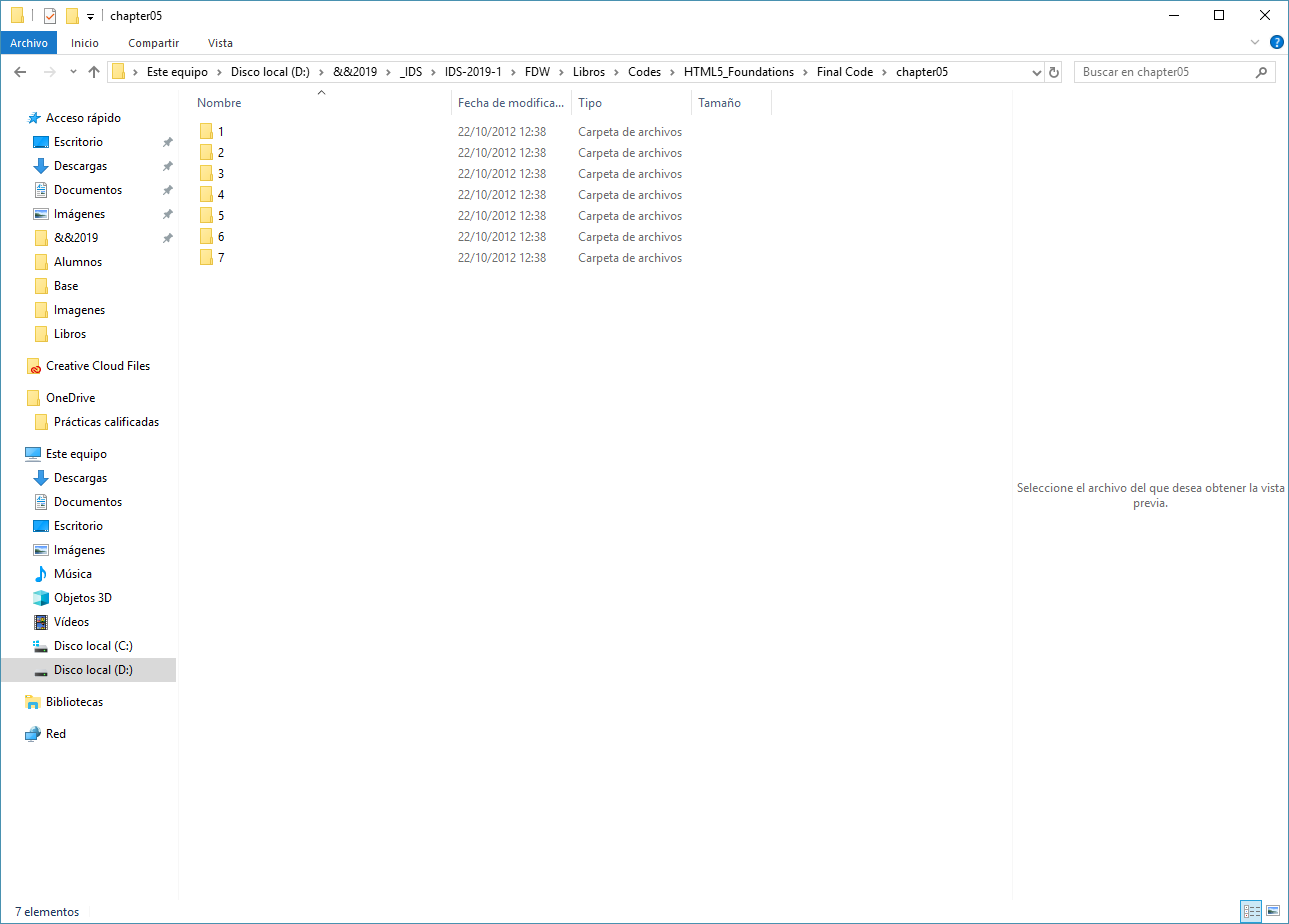
<input type=”submit” value=”Search”>

</div>

</form>

Por ahora, esto es todo lo que necesita saber sobre el elemento <form> para comenzar a crear su formulario de reservas. Más adelante en este tema, profundizará en los formularios y las entradas, pero primero configuremos la página de Reservas.

Como es habitual, este código se puede descargar del sitio web en <http://www.wiley.com/go/treehouse/html5foundations>. El código para el ejercicio-1 se puede encontrar en el código de la lección 5, carpeta 1.





## EJERCICIO 1

## Crear la nueva página de Reservas.

1. Cree un nuevo archivo llamado bookings.html en su editor de texto favorito y guárdelo en la raíz de la carpeta de su proyecto.
2. Abra el archivo about.html y copie su contenido en su nuevo archivo bookings.html.

No olvides actualizar los elementos <title> y <meta> y eliminar la clase activa del enlace de Inicio en este nuevo archivo.

1. Borre el contenido del elemento <section> que tiene el texto de la página de ID.
2. Dentro de este elemento <section>, cree un nuevo elemento <h1> con el texto Solicitar una reserva.
3. Debajo de este encabezado, agregue el siguiente párrafo de texto usando un elemento <p>.

<p>

Utilice el siguiente formulario para enviar una solicitud de reserva. Un miembro del personal lo contactará y confirmará su reserva.

</p>

1. Debajo de este párrafo, crea un nuevo elemento <form>.

<form> </form>

1. Establezca el atributo de acción en este elemento <form> en bookings.php.

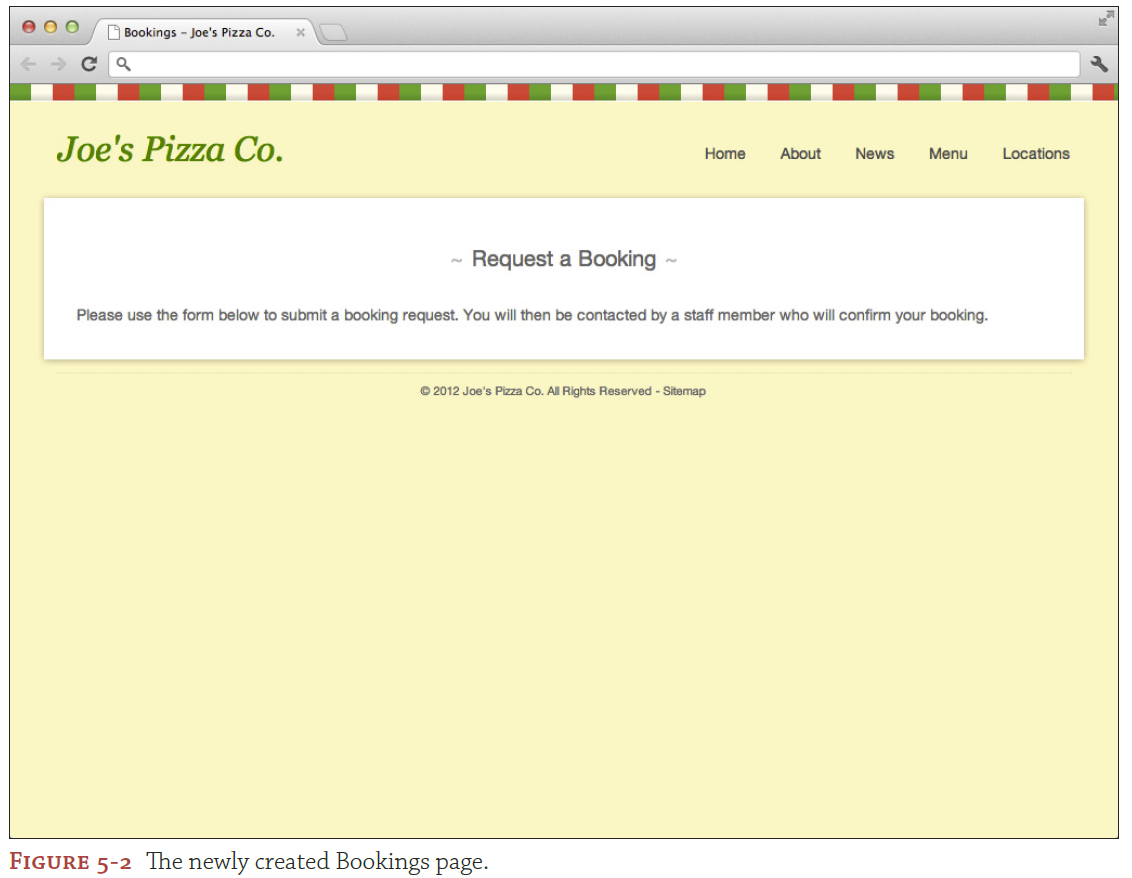
<form action = "bookings.php”> </form>

1. Establezca el atributo de método en este elemento <form> en POST.

<form action = "bookings.php” method = "POST”> </form>

1. Guarde su archivo bookings.html.

La Figura 5-2 muestra cómo debe verse su página de Reservas en esta etapa.



### El atributo action

***El atributo action se utiliza para especificar la ubicación del archivo al que se deben enviar los datos del formulario***. Esta debe ser una ruta absoluta o relativa. El archivo al que se hace referencia en el atributo de acción realizaría una acción, como guardar los datos en la base de datos o verificar los detalles de inicio de sesión de un usuario.

Para obtener más información sobre las rutas absolutas y relativas, consulte la sección "Rutas relativas y absolutas" en el Tema 2.

Si observa el formulario que acaba de agregar a la página de Reservas, puede ver que el atributo action hace referencia a un archivo bookings.php.

<form action = "bookings.php” method = "POST”>

El hecho de que haya utilizado una ruta relativa aquí significa que el archivo bookings.php debe estar en la misma carpeta que el archivo bookings.html.

No se preocupe por no tener un bookings.php en la carpeta de su proyecto. Solo vamos a fingir que está allí y que creará una nueva reserva cuando se envíe el formulario.

Para poder procesar los datos, necesitaría escribir un programa usando un lenguaje de programación como PHP, Ruby o Python, que está fuera del alcance de este primer tramo del curso. Si está interesado en obtener más información sobre la programación del lado del servidor, consulte este curso Treehouse de Jim Hoskins y Jason Seifer <http://teamtreehouse.com/library/programming-2/build-a-simple-version-of-facebook>.

### El Atributo método

El atributo método se utiliza para especificar cómo desea que se envíen los datos del formulario al servidor.

Hay dos formas de enviar datos de formulario, solicitudes GET y solicitudes POST. Si olvida especificar un atributo método en su formulario, el método GET se utiliza de forma predeterminada.

#### El método GET

El método GET envía los datos de su formulario dentro de una URL. Este método se usa comúnmente cuando se realizan tareas que solo necesitan recuperar datos existentes de un servidor web, como búsquedas o filtros de productos en sitios web de comercio electrónico. Cuando un usuario envía el formulario, el navegador comenzará a crear una nueva URL para enviar al usuario. Esto se verá como la siguiente URL.

<http://www.html5foundations.com/bookings.php?name=Joe%20Balochio&restaurant=310%20West%2038th%20Street,%20NY>

Esta URL tiene varias partes diferentes. Vamos a verlos con más detalle.

##### El nombre de dominio

La primera parte de la URL generada está formada por el protocolo (http: //) y el nombre de dominio completo (www.html5foundations.com).

Si está utilizando una URL absoluta, habría definido explícitamente este nombre de dominio en su atributo de acción.

Si está utilizando una URL relativa en su atributo de acción, el navegador tomará automáticamente el nombre de dominio de la página desde la que se envía el formulario. Por ejemplo, si su formulario se encuentra en la página [www.html5foundations.com/bookings.html](http://www.html5foundations.com/bookings.html), el navegador usará www.html5foundations.com como nombre de dominio.

##### El path

El segmento de ruta de la URL representa la ruta al archivo que procesará los datos del formulario. He resaltado el segmento de ruta de nuestra URL de ejemplo en negrita.

[http://www.html5foundations.com/**bookings.php**?name=Joe%20Balochio&restaurant=310%20West%2038th%20Street,%20NY](http://www.html5foundations.com/bookings.php?name=Joe%20Balochio&restaurant=310%20West%2038th%20Street,%20NY)

#### Los parámetros de URL

El segmento final de nuestra URL son los parámetros. Aquí es donde se transmiten los datos de su formulario.

El navegador primero agregará un carácter de signo de interrogación (?) Después de la ruta para indicar que todo después de este punto en la URL son datos.

<http://www.html5foundations.com/bookings.php?name=Joe%20Balochio&restaurant=310%20West%2038th%20Street,%20NY>

Cada uno de los campos del formulario se agrega a la URL como un par clave-valor. La clave se toma del atributo de nombre en el control de formulario (más sobre esto más adelante), y el valor es solo el valor del control de formulario en el momento en que se envió el formulario. En el siguiente ejemplo, la clave para el primer dato es name y el valor es Joe% 20Balochio. Cada par clave-valor se separa con un signo (&); Esto se resalta en negrita en el siguiente ejemplo.

<http://www.html5foundations.com/bookings.php?name=Joe%20Balochio&restaurant=310%20West%2038th%20Street,%20NY>

Una vez que el navegador termina de construir la URL, carga la nueva URL en la ventana actual del navegador.

***Los navegadores ponen un límite a la longitud de las URL y, por lo tanto, a la cantidad de datos que puede enviar utilizando el método GET. Este límite varía según el navegador, pero es típico un límite de 2,000 caracteres***. Si tiene una forma bastante grande o está utilizando elementos de <textarea>, debe usar el **método POST en su lugar**.

Una ventaja de usar el método GET es que la URL que se genera cuando se envía el formulario puede marcarse como favorita. Por ejemplo, si utiliza el método GET en un formulario de búsqueda, el usuario puede marcar la página de resultados como datos de formulario que contienen el término de búsqueda en la URL de la página. Si usas el método POST, no podrían hacer esto porque los datos se pasan "detrás de la escena".

Codificación de datos de formulario

Es posible que haya notado algunos caracteres extraños que aparecen en la URL de ejemplo (están en negrita a continuación):

name = Joe**%20**Balochio&restaurant=310**%20**West**%20**38th**%20**Street,**%20**NY

Seguramente los humanos no escribieron esto?

Correcto. Las URL no pueden contener espacios o caracteres especiales, por lo que el navegador codifica automáticamente cualquier espacio o caracteres especiales en sus datos al cambiarlos a un código ASCII hexadecimal. No tienes que preocuparte demasiado por esto; sólo tenga en cuenta que el navegador codifica los datos para usted.



## El método POST

**El método POST debe utilizarse al realizar cualquier acción que implique que los datos se guarden en una base de datos**. Si bien también es técnicamente posible guardar datos utilizando el método GET, se considera una práctica recomendada de seguridad usar POST para estas acciones.

La URL para los formularios que utilizan el método POST se construye en gran parte de la misma manera. La diferencia es que los datos del formulario no se agregan a la URL como parámetros (pero aún están codificados).

En su lugar, ***los datos se envían efectivamente en segundo plano y están completamente ocultos de la URL***.

Los datos todavía pueden ser recogidos y procesados ​​por el servidor.

El método POST tiene algunos beneficios de seguridad. Por ejemplo, no querría utilizar el método GET para un formulario de inicio de sesión, ya que los detalles de inicio de sesión de un usuario serían visibles en la URL (¡incluida su contraseña!). Además, a medida que se pasan los datos en la URL, no hay nada que impida que un usuario modifique fácilmente estos datos editando la URL en la barra de direcciones. A veces, esto puede interrumpir sus aplicaciones si no cuentan con las medidas adecuadas para tratar con datos incorrectos. Sin embargo, **es importante tener en cuenta que el uso del método POST no solucionará todos estos problemas y no es suficiente para que sus formularios estén completamente seguros** (pero ayuda).

## Agregar los campos del formulario a la página de reservas con los elementos <input> y <label>

El principal elemento de control de formulario que utilizará en sus formularios web es el elemento <input>.

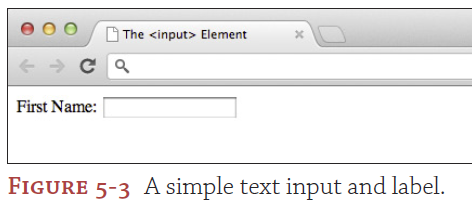
Este elemento es un tipo de persona, ya que puede usarse para recopilar todo tipo de datos de sus usuarios. Puede especificar qué tipo de datos desea recopilar utilizando el atributo type; en la mayoría de los navegadores, esto también alterará la forma en que se muestra el elemento. De forma predeterminada, todos los elementos <input> tienen el texto de tipo. Aprenderá sobre los diferentes tipos de entrada que puede usar más adelante en este tema y en el Lección 6.

Los campos de formulario son inútiles a menos que el usuario sepa lo que debe ingresar en cada casilla. El elemento <label> le permite asociar una etiqueta de texto con un campo de formulario. El siguiente ejemplo muestra un elemento <input> con un elemento <label> asociado.

<label for = "firstName”> First Name: </label>

<input name = "firstName" type = "text" id = "firstName">

La Figura 5-3 muestra cómo se mostraría este ejemplo en un navegador.



Cuando creas un elemento <input> también deberías crear un elemento <label> y asociarlo con la entrada. Para hacer esto, debe asignar al ID <input> un ID y luego especificar un atributo for en el elemento <label> con el valor establecido en el ID de la entrada, como en el ejemplo anterior.

La creación de etiquetas para los elementos <input> es importante por varias razones. Las tecnologías de asistencia, como los lectores de pantalla, se basarán en estas etiquetas para comprender sus formularios y ayudar a los usuarios a completarlos. La asociación adecuada de etiquetas con entradas también hará que la etiqueta se convierta en un destino de clic extendido para el campo de entrada. Esto significa que cuando el usuario hace clic en la etiqueta, el elemento <input> asociado se centrará en el navegador. Esto es especialmente útil para elementos como casillas de verificación o botones de radio (más sobre estos próximos) que son en sí mismos, objetivos bastante pequeños para que un usuario haga clic.

Para comenzar, el formulario de reservas incluirá siete campos:

* Nombre,
* Teléfono,
* Correo electrónico,
* Restaurante,
* Tiempo de reserva,
* Invitados
* y Marketing.

Ahora crearás elementos <input> y <label> para cada uno de estos.



## EJERCICIO-2

1. Abra su archivo bookings.html en su editor de texto favorito.
2. Cree un nuevo elemento <div> dentro del elemento <form> y establezca su clase en field.

* Esta clase se utilizará exclusivamente para fines de estilo. (Necesitará crear un nuevo elemento <div> como este para cada elemento <input> que agregue).

<div class = "field”> </div>

1. Dentro de este primer elemento <div>, cree un nuevo elemento <input> con el texto de tipo.

<div class=”field”>

<input type=”text”>

</div>

1. Establezca los atributos name y id de esta entrada como nombre. La razón para configurar ambos atributos es porque es posible que múltiples elementos tengan el mismo valor de nombre.

Esto ocurre cuando se trata de grupos de entradas, como botones de radio. El valor de id, sin embargo, siempre debe ser único. Esto significa que si desea apuntar a este elemento <input> individual con CSS o JavaScript, deberá usar la ID.

<div class=”field>

<input type=”text” name=”name” id=”name”>

</div>

1. Cree un elemento <label> y asígnele este elemento <input> utilizando el atributo for.

<div class=”field”>

<label for=”name”>Name:</label>

<input type=”text” name=”name” id=”name”>

</div>

1. Debajo de este elemento <div>, cree un nuevo <div> (nuevamente con el campo de clase) con un elemento <input> que tiene el texto de tipo y el nombre y el teléfono de identificación. A continuación, cree una etiqueta para esta entrada.

<div class=”field”>

<label for=”phone”>Phone:</label>

<input type=”text” name=”phone” id=”phone”>

</div>

1. Ahora cree un nuevo elemento <div> con un elemento <input> que tenga el texto de tipo y el nombre y el correo electrónico de ID. Crea una etiqueta para esta entrada.

<div class=”field”>

<label for=”email”>Email:</label>

<input type=”text” name=”email” id=”email”>

</div>

1. Cree un nuevo elemento <div> que tenga un elemento <input> con el texto de tipo y el nombre y el restaurante de ID. También debe crear una etiqueta para esta entrada.

<div class=”field”>

<label for=”restaurant”>Restaurant:</label>

<input type=”text” name=”restaurant” id=”restaurant”>

</div>

1. Cree otro nuevo elemento <div> con un <input> que tenga el texto de tipo y el nombre y el tiempo de reserva de ID. ¡No olvides la etiqueta!

<div class=”field”>

<label for=”bookingTime”>Booking Time:</label>

<input type=”text” name=”bookingTime” id=”bookingTime”>

</div>

1. Cree un elemento <div> con un <input> que tenga el texto de tipo y el nombre e invitados de ID. Añadir una etiqueta para esta entrada.

<div class=”field”>

<label for=”guests”>Number of Guests:</label>

<input type=”text” name=”guests” id=” guests”>

</div>

1. Crea un elemento final <div> con un elemento <input>. Esta vez, configure el tipo de casilla de verificación y el nombre y la ID de marketing. Establezca el atributo de valor en 1. Luego cree una etiqueta para este elemento con el texto que se muestra en el siguiente código.

<div class=”field”>

<label for=”marketing”>Please tick this box if you would like

to receive special offers from Joe’s Pizza Co.</label>

<input type=”checkbox” name=”marketing” id=”marketing”

value=”1”>

</div>

1. Guarde el archivo bookings.html.

### Código completo del elemento form

<form action=”bookings.php” method=”POST”>

<div class=”field”>

<label for=”name”>Name:</label>

<input type=”text” name=”name” id=”name”>

</div>

<div class=”field”>

<label for=”phone”>Phone:</label>

<input type=”text” name=”phone” id=”phone”>

</div>

<div class=”field”>

<label for=”email”>Email:</label>

<input type=”text” name=”email” id=”email”>

</div>

<div class=”field”>

<label for=”restaurant”>Restaurant:</label>

<input type=”text” name=”restaurant” id=”restaurant”>

</div>

<div class=”field”>

<label for=”bookingTime”>Booking Time:</label>

<input type=”text” name=”bookingTime” id=”bookingTime”>

</div>

<div class=”field”>

<label for=”guests”>Number of Guests:</label>

<input type=”text” name=”guests” id=”guests”>

</div>

<div class=”field”>

<label for=”marketing”>Please tick this box if you would like

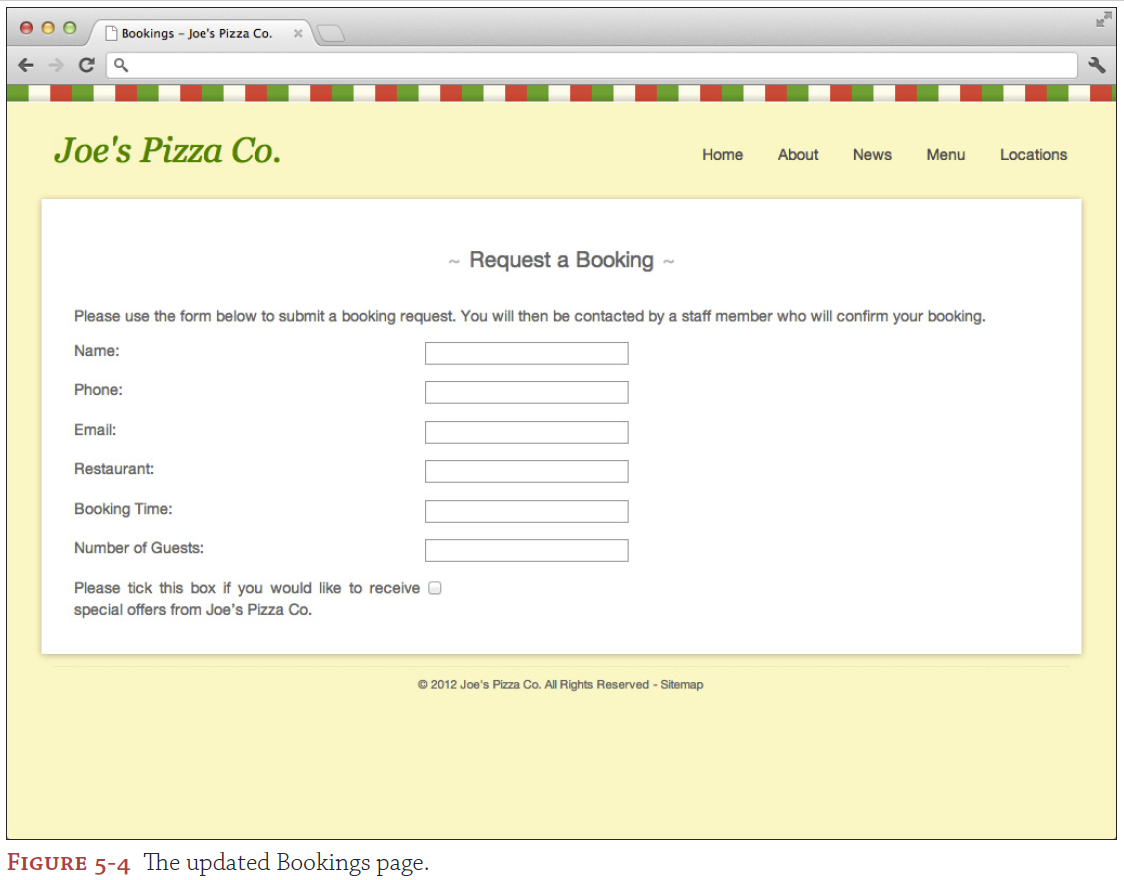
to receive special offers from Joe’s Pizza Co.</label>

<input type=”checkbox” name=”marketing” id=”marketing”

value=”1”>

</div>

</form>



En las siguientes secciones, aprenderá sobre los diferentes tipos de entradas que puede usar, así como cada uno de los diferentes atributos que puede aplicar al elemento <input>. Aquí solo va a cubrir los tipos y atributos ampliamente soportados de HTML4. En el tema 6 aprenderá sobre los nuevos tipos de entrada y atributos introducidos en HTML5.

## Atributos de input

Además de los atributos globales estándar que vio en el tema 3, el elemento <input> tiene una serie de atributos especializados, algunos de los cuales ya ha utilizado en la página de Reservas.

### Poner nombres a las entradas

Aunque no es estrictamente necesario, todos los elementos de entrada deben tener un atributo name. Si pretende enviar los datos a un servidor, deberá especificar un atributo name.

De lo contrario, el navegador no enviará sus datos.

<label for=”name”>Name:</label>

<input type=”text” name=”name” id=”name”>

Como mencioné anteriormente en este tema, el valor del atributo name formará la clave con la que se asociarán los datos de entrada cuando se envíen al servidor.

## Configuración de valores predeterminados

Puede establecer un valor predeterminado para un elemento de entrada utilizando el atributo value. Cuando el navegador carga la página, colocará automáticamente el valor del atributo de valor en el cuadro de entrada o establecerá un control especializado para este valor.

Este es un ejemplo de cómo podría establecer que el valor predeterminado de la entrada de invitados sea 2.

<input type=”text” name=”guests” id=”guests” value=”2”>

Campos de desactivación

Tiene dos formas de deshabilitar un campo de formulario para que un usuario no pueda modificar su valor. El primero es usar el atributo disabled; alternativamente, puede usar el atributo readonly.

<input type=”text” name=”name” disabled>

<input type=”text” name=”name” readonly>

La diferencia entre estos dos atributos es que el valor de los elementos <input> con el atributo de solo lectura aún se enviará al servidor, mientras que los elementos <input> que tienen el atributo deshabilitado no enviarán ningún dato. El atributo desactivado se ignora si el tipo de entrada está oculto.

### Size

*El atributo de tamaño se usa para especificar el número de caracteres que debe ser visible* para un usuario en cualquier momento, por lo tanto, altera el tamaño físico del elemento <input>. El tamaño por defecto es 20.

El atributo de tamaño no limita la cantidad de caracteres que un usuario puede ingresar en un campo; para eso, usas el atributo maxlength que verás en el tema 7.

<label for=”name”>Name:</label>

<input type=”text” name=”name” id=”name” size=”65”>

## Tipos de entrada

Hay una serie de diferentes tipos de entrada que puede utilizar para facilitar que los usuarios completen sus formularios web. En esta sección, aprenderá acerca de los tipos de entrada que son ampliamente compatibles con los navegadores web. En el tema 6 aprenderá sobre los nuevos tipos de entrada que se han introducido en HTML5.

### Texto

El tipo de entrada de texto ha sido tradicionalmente el tipo más comúnmente usado en los elementos <input>.

Como su nombre lo indica, el tipo de entrada de texto se usa para instancias en las que necesita recopilar algún texto del usuario. Este es también el tipo predeterminado para el elemento <input>, por lo que si olvida especificar un atributo de tipo o está utilizando un tipo que el navegador de un usuario no admite, el navegador simplemente mostrará un campo de texto.

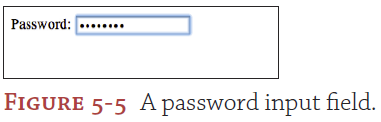
La entrada de nombre en su formulario utiliza el tipo de texto.

<label for = "name”> Name: </label>

<input type = "text" name = "name" id = "name" autofocus>

### Contraseñas

El tipo de contraseña se puede usar para indicar que una entrada está esperando una contraseña. El navegador no mostrará la entrada que se escribe en un campo de contraseña en texto sin formato; esto evita que esos demonios furtivos vean tu contraseña por encima del hombro. En su lugar, las entradas de contraseña muestran los caracteres como puntos negros o asteriscos, como se muestra en la Figura 5-5.



Aquí hay un ejemplo de un elemento <input> de tipo contraseña.

<label for=”pass”>Password:</label>

<input type=”password” name=”pass” id=”pass”>

### Checkboxes

El tipo de **casilla de verificación** es útil cuando necesita recopilar una respuesta verdadera o falsa del usuario.

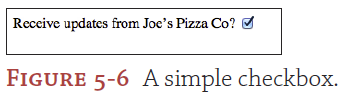
Por defecto, la casilla de verificación se mostrará cómo sin marcar; sin embargo, puede cambiar esto agregando el atributo checked al elemento <input>.

Las entradas con el tipo de casilla de verificación también deben especificar un atributo value; estos son los datos que se enviarán al servidor si se selecciona la casilla de verificación cuando se envía el formulario.

Este es un ejemplo de un elemento <input> con el cuadro de verificación de tipo configurado para mostrarse como marcado de forma predeterminada. El valor se ha establecido en 1 porque esta es la convención común para las respuestas verdaderas. ***Utilice 0 para falso***.

<label for=”contact”>Receive updates from Joe’s Pizza Co?</label>

<input type=”checkbox” name=”contact” id=”contact” value=”1” checked>



## Botones de radio

El tipo de radio se utiliza para mostrar un botón de radio. Esto es útil si tiene varias opciones diferentes para presentar a los usuarios, pero solo quiere que elijan una de estas opciones.

Cuando utilice varios botones de opción como opciones, cada elemento <input> debe tener el mismo atributo de nombre. Los datos asociados con una opción deben almacenarse en el atributo de valor.

Al igual que con las casillas de verificación, también puede establecer un valor predeterminado para las entradas de radio usando el atributo marcado.

Este es un ejemplo de cómo podría usar varios elementos <input> con el tipo de radio para recopilar el color favorito de un usuario.

<p>

What is your favorite color?

</p>

<div>

<label for=”green”>Green</label>

<input name=”favColor” id=”green” type=”radio” value=”green”>

<label for=”red”>Red</label>

<input name=”favColor” id=”red” type=”radio” value=”red” checked>

<label for=”orange”>Orange</label>

<input name=”favColor” id=”orange” type=”radio” value=”orange”>

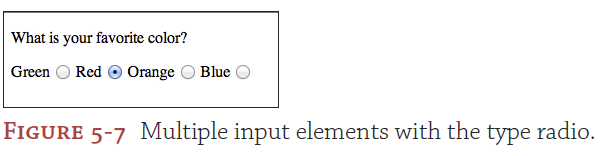
<label for=”blue”>Blue</label>

<input name=”favColor” id=”blue” type=”radio” value=”blue”>

</div>

Observe cómo, en este ejemplo, todos los elementos <input> tienen el mismo atributo de nombre. También hemos indicado que el rojo debe ser la opción predeterminada utilizando el atributo marcado.

La Figura 5-7 muestra cómo debería verse este ejemplo en su navegador web.



## Botones de envío

El tipo submit se puede utilizar para definir un botón de envío para su formulario. El texto para el botón se debe establecer en el atributo de valor.

Aquí hay un ejemplo de un botón de envío.

<input type=”submit” value=”Send”>

También puede crear botones Enviar utilizando el elemento <button> que conocerá más adelante en este tema.

## Campos ocultos

Puede haber ocasiones en las que desee enviar datos al servidor pero ocultar esos datos a sus usuarios. El tipo hidden está diseñado para permitirte hacer esto.

Cuando utilice el tipo hidden, debe especificar un nombre para su elemento <input> y establecer el valor del atributo value en los datos que desea enviar con el formulario.

<input type=”hidden” name=”id” value=”123”>



Has cubierto mucho en esta sección hasta ahora. Para un resumen rápido, vea el video de Nick Pettit sobre las entradas en Treehouse: <http://teamtreehouse.com/library/websites/html/forms/inputs>.

### Agregar un cuadro de mensaje con el elemento <textarea>

Además de los campos que ya ha agregado a su formulario, sería bueno tener un cuadro de texto donde los usuarios puedan dejar comentarios sobre sus reservas. Los elementos de entrada son excelentes, pero no son muy fáciles de usar cuando se trata de ingresar grandes cantidades de texto, especialmente si ese texto necesita abarcar varias líneas. En su lugar, puede agregar el elemento <textarea> a su formulario para campos de entrada con gran cantidad de texto.

El elemento <textarea> comparte muchos de los mismos atributos que el elemento <input>, como el enfoque automático, el nombre y el marcador de posición; Sin embargo, también tiene los atributos cols y rows. El atributo cols se puede usar para especificar cuántos caracteres deben caber en una sola línea en el área de texto, y el atributo row especifica cuántas filas deben ser visibles para el usuario. Usando estos dos atributos, teóricamente puedes establecer la altura y el ancho de <textarea> (aunque es posible que desees hacerlo usando CSS).

Aquí hay un ejemplo de un elemento <textarea> que usa los atributos de columnas y filas para especificar su tamaño.

<textarea name = "message" cols = "50" rows = "10”> </textarea>

Observe cómo el elemento <textarea> tiene una etiqueta final. A diferencia del elemento <input>, <textarea> no es un elemento vacío. Cualquier contenido de texto que coloque entre las etiquetas de inicio y finalización se mostrará en el área de texto.





## EJERCICIO-3

1. Abra el archivo bookings.html.
2. Debajo de los campos de formulario existentes, cree un nuevo elemento <div> con el campo de clase.
3. Dentro de este elemento <div>, cree un nuevo elemento <textarea>.
4. Establezca los atributos id y name en este <textarea> para que sean mensajes.
5. Establezca el atributo cols en este <textarea> para que sea 50.
6. Establezca el atributo de las filas en 10.
7. Sobre este elemento <textarea>, cree un nuevo elemento <label>.
8. Establezca el atributo for en este <label> para que sea un mensaje.
9. Agregue el siguiente texto dentro del elemento <label>: Solicitudes especiales :.
10. Guarde el archivo bookings.html

Ahora ha agregado un <textarea> a su formulario que los visitantes pueden usar para proporcionar algunos comentarios adicionales sobre sus solicitudes de reserva. Así es como debería verse su nuevo código.

<div class=”field”>

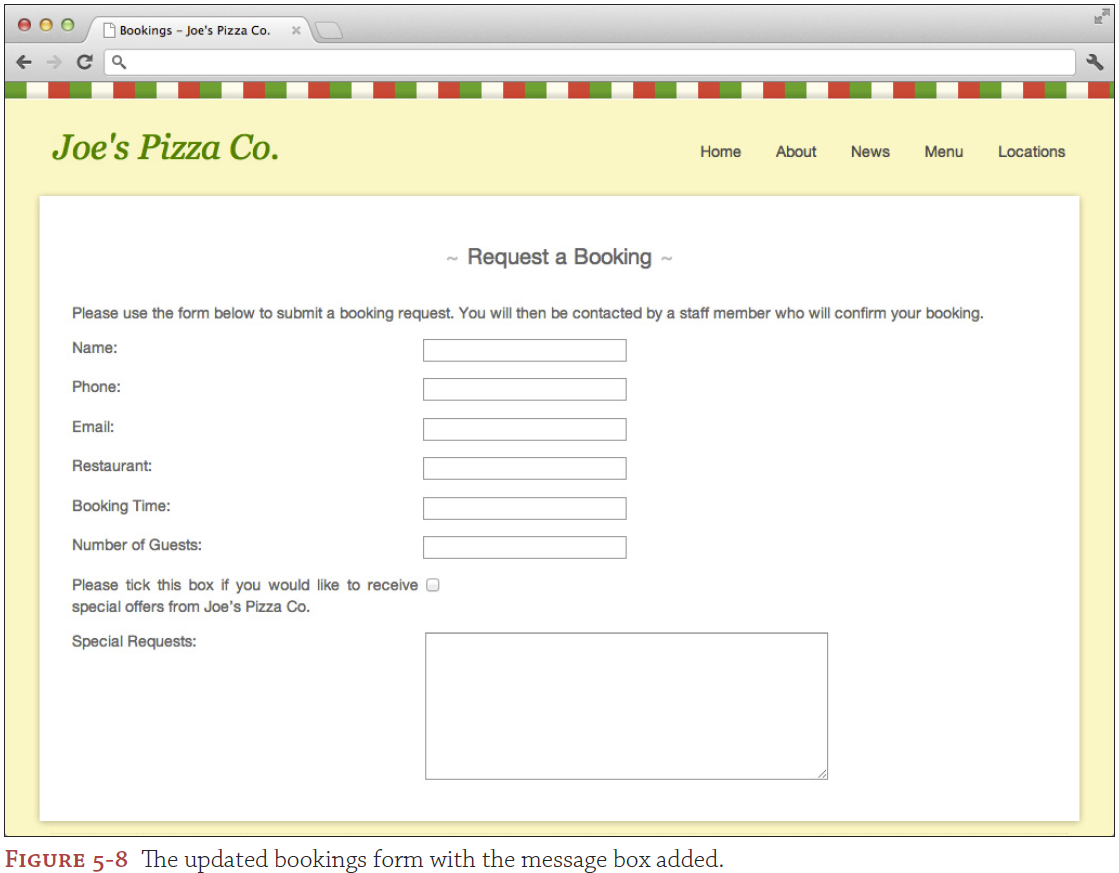
<label for=”message”>Special Requests:</label>

<textarea name=”message” id=”message” cols=”50” rows=”10”>

</textarea>

</div>

La Figura 5-8 muestra cómo debe verse su formulario de reservas en su navegador.



## Agregar un botón de envío a su página de reservas

Su formulario de reserva está casi completo, pero le falta un componente importante: un botón Enviar que los usuarios pueden hacer clic para enviar sus solicitudes de reserva.

El elemento <button> tiene una serie de funciones diferentes que se pueden definir utilizando su atributo de tipo. Para su página de Reservas, utilizará el tipo de envío, pero también puede usar los tipos de reinicio o botón para otras funciones, como se describe brevemente. Si olvida especificar un tipo, enviar se utilizará de forma predeterminada.

Para crear un botón que envíe un formulario, use el tipo de envío, como en el siguiente ejemplo:

<button type=”submit”>Submit Form</button>

El texto dentro del elemento del botón aparecerá en el botón.





## EJERCICIO-4

## Añadir un botón a su forma

1. Abra el archivo bookings.html.
2. Cree un nuevo elemento <div> justo antes de la etiqueta final del elemento <form> y asígnele el campo de clase.
3. Dentro de este elemento <div>, cree un nuevo elemento <button>.
4. Establezca el atributo de tipo en este <botón> para enviar.
5. Agregue el siguiente texto entre las etiquetas del elemento <button>: Solicitar reserva.
6. Guarde el archivo bookings.html.

Aquí es cómo debe verse esta parte de su código:

<div clase = "filed">

<button type = "submit"> Solicitar reserva </button>

</div>

Ahora debe tener un botón Enviar al final de su formulario para que los visitantes puedan solicitar reservas.

La Figura 5-9 muestra cómo debe verse su formulario de reservas actualizado en su navegador.

Con el elemento <button>, puede crear un botón que sea similar a uno creado usando <input type = "submit">. La diferencia es que puede colocar otros elementos HTML dentro de un elemento <button>; por ejemplo, podría tener un elemento <img> dentro de un botón.

### Botones de reinicio

Al configurar el atributo de tipo de un elemento <button> para restablecer, puede crear un botón que restablecerá todos los campos de un formulario. Este es un ejemplo de cómo puede crear un botón Restablecer para sus formularios.

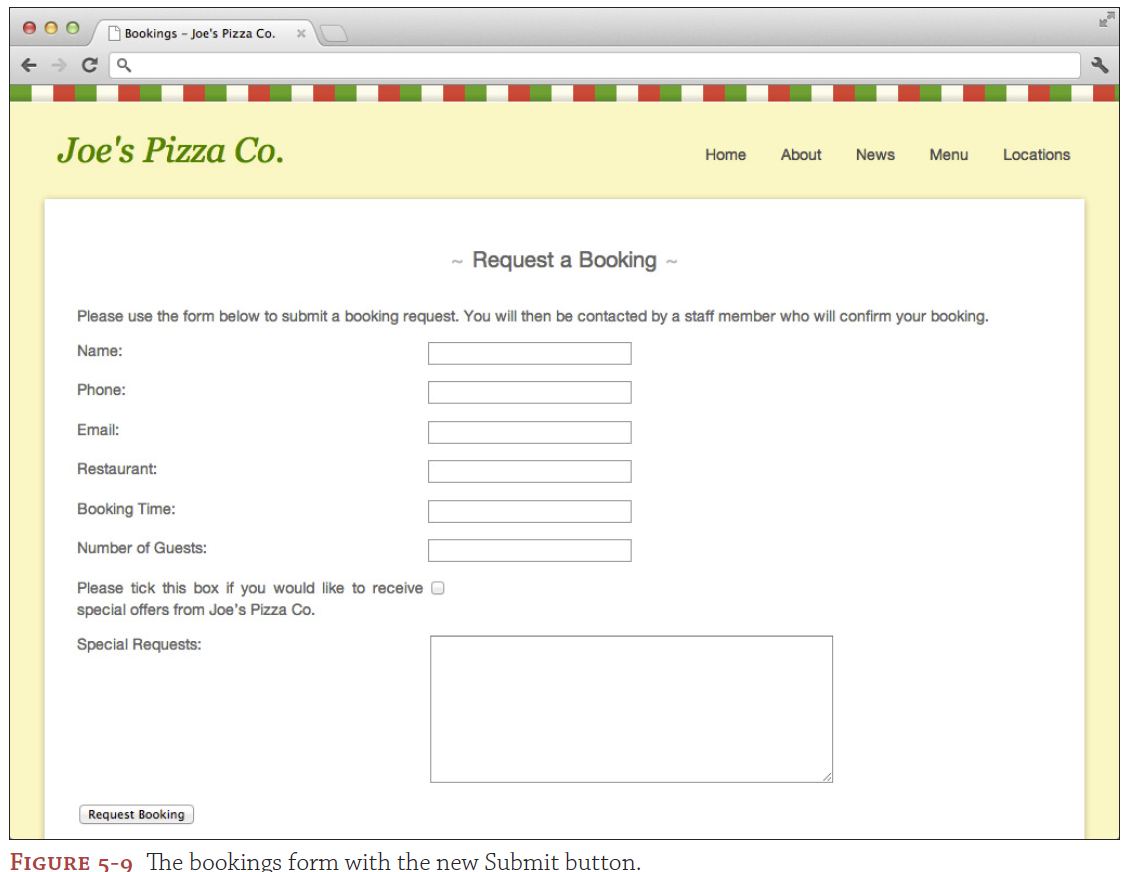
<button type = "reset”> Restablecer formulario </button>

### Botones tontos

Hay otro tipo que se puede usar en el elemento <botón>. Me gusta referirme a estos botones como botones tontos porque en realidad no realizan ninguna acción. ***Para que estos botones sean "inteligentes", debe usar JavaScript para indicar al navegador que ejecute algún código JavaScript cuando se presiona el botón***. Este código de JavaScript podría realizar una acción, como iniciar o detener la reproducción de un video. Aprenderás cómo hacer esto en el Lección 11.

Aquí hay un ejemplo de un botón tonto.

<button type=”button”>Do Something</button>



## Agregar un menú desplegable a su página de reservas

Ya has aprendido cómo puedes usar los botones de radio para crear múltiples opciones para que los usuarios puedan elegir. En esta sección, aprenderá a usar los elementos <seleccionar> y <opción> para crear un menú desplegable. Luego, utilizará estos elementos para crear un menú desplegable para el campo de restaurante en su página de Reservas, eliminando la necesidad de que el usuario escriba la ubicación del restaurante.

### Los elementos <select> y <option>

Para crear un menú desplegable, utiliza un elemento <select>. Este elemento tiene un propósito similar al elemento <input> en el sentido de que se usa para establecer el nombre del control de formulario, así como otros atributos, como su clase e id.

El elemento <select> también es responsable de agrupar las opciones y debe contener uno o más elementos <option>, como se muestra en el siguiente ejemplo.

<select name=”color” id=”color”>

<option>Green</option>

<option>Red</option>

<option>Blue</option>

</select>

Cada uno de estos elementos <option> representa un valor potencial para el control de formulario (el texto contenido dentro del elemento se usa como valor aquí). El concepto es similar a cómo funcionan los múltiples botones de radio. Aquí es cómo se vería el ejemplo anterior si se implementara usando botones de radio.

<label for=”green”>Green</label>

<input type=”radio” name=”color” id=”green” value=”Green”>

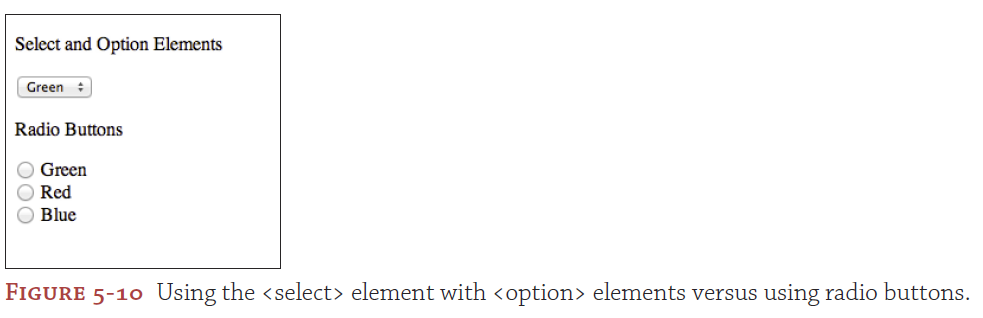
<label for=”red”>Red</label>

<input type=”radio” name=”color” id=”red” value=”Red”>

<label for=”blue”>Blue</label>

<input type=”radio” name=”color” id=”blue” value=”Blue”>

La principal diferencia es que todas las opciones se muestran a la vez cuando se usan los botones de radio, mientras que el elemento <select> agrupará las opciones y las mostrará en un menú desplegable. Esto es particularmente útil si tiene una gran cantidad de opciones para elegir, porque el espacio de pantalla que ocupa el menú desplegable no aumenta cuando agrega más opciones. La figura 5-10 muestra cómo estos dos ejemplos difieren.





## Agregar el menú desplegable del restaurante a su página

El campo de restaurante en su formulario de reservas se utiliza para indicar al coordinador de reservas en qué restaurante al cliente le gustaría hacer una reserva. Actualmente, este campo solo usa una entrada de texto normal; sin embargo, esta es una mala práctica porque confía en que el cliente escriba una dirección precisa en el campo de ubicación. Una mejor solución sería crear un menú desplegable que ya contenga las tres ubicaciones de restaurantes para que los clientes puedan seleccionar el restaurante que desean en lugar de tener que escribir la ubicación.



## EJERCICIO 5

1. Abra el archivo bookings.html.
2. Localice el elemento <input> que se usa actualmente para el campo de restaurante y elimínelo.

<div class=”field”>

<label for=”restaurant”>Restaurant:</label>

<input type=”text” id=”restaurant” name=”restaurant”>

</div>

1. Crea un nuevo elemento <select> en su lugar.

<div class=”field”>

<label for=”restaurant”>Restaurant:</label>

<select></select>

</div>

1. Establezca los atributos de nombre e identificación en este elemento <select> en restaurante.

<select name = "restaurant" id = "restaurant"> </select>

1. Dentro de este elemento <select>, cree un nuevo elemento <option> con el texto 310 West 38th Street, NY.

<option>310 West 38th Street, NY</option>

1. Cree otro elemento <option> debajo de ese con el texto 2450 Broadway, NY.

<option>2450 Broadway, NY</option>

7. Finalmente, cree un elemento más <option> con el texto 200 West 44th Street, NY.

<option> 200 West 44th Street, NY </option>

8. Guarde el archivo bookings.html.

El campo de su restaurante ahora debería verse como el siguiente.

<div class=”field”>

<label for=”restaurant”>Restaurant:</label>

<select id=”restaurant” name=”restaurant”>

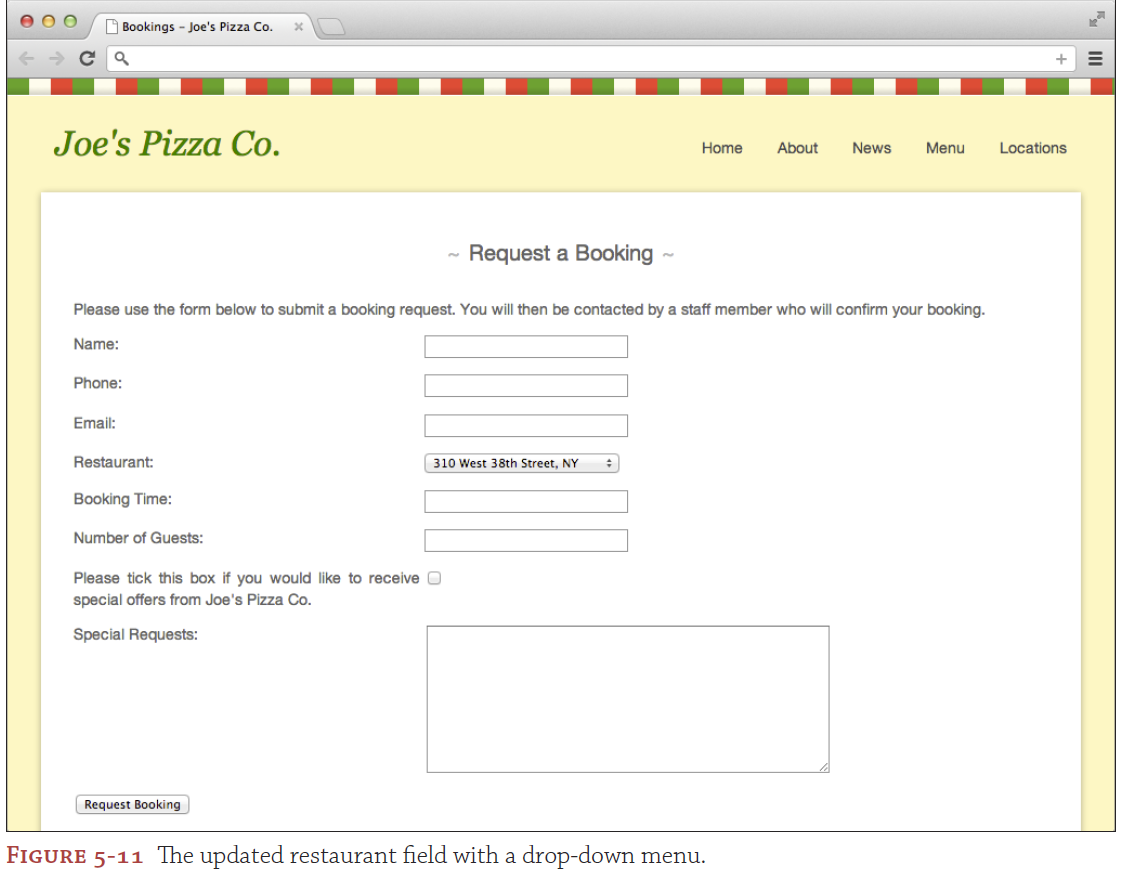
<option>310 West 38th Street, NY</option>

<option>2450 Broadway, NY</option>

<option>200 West 44th Street, NY</option>

</select>

</div>



## Más sobre el uso de opciones

Hasta ahora, en esta sección, ha aprendido cómo crear menús desplegables básicos utilizando los elementos <select> y <option>. Ahora aprenderá algunas técnicas ligeramente más avanzadas que le permitirán hacer cosas como configurar la opción predeterminada, agregar un valor que sea diferente del texto de la opción y cómo agrupar las opciones en subgrupos en el menú desplegable.

### Configuración de una opción predeterminada

Para configurar la opción predeterminada en un menú desplegable, simplemente debe aplicar el atributo seleccionado al elemento <option> deseado.

<select name=”color” id=”color”>

<option>Green</option>

**<option selected>Red</option>**

<option>Blue</option>

</select>

En este ejemplo, la opción Red se seleccionaría de forma predeterminada.

## Usando el atributo value para el elemento <option>

Habrá ocasiones en que el texto dentro de un elemento <option> debe ser diferente del valor que se envía cuando se envía el formulario.

Un ejemplo sería si necesitara recopilar una ID de usuario. Es poco probable que un usuario administrador pueda recordar el ID de usuario numérico de cada usuario en el sistema, por lo que desearía que los nombres de los usuarios se muestren en el menú desplegable pero que se envíen sus ID en el formulario. Puede hacer esto agregando un atributo de valor al elemento <option>.

<option value = ”23”> Joe Balochio </option>

Cuando se envía el formulario, se utilizará el valor en lugar del texto contenido entre las etiquetas <option>.

### Permitir selecciones múltiples

Puede permitir que sus usuarios seleccionen más de una opción. Especificar el atributo múltiple en el elemento <select> le permitirá hacer esto. Los usuarios luego tendrán que presionar y mantener presionada la tecla CTRL (o la tecla CMD en una Mac) para seleccionar múltiples opciones de la lista.

### Opciones de agrupación

Cuando tenga un elemento <select> con varias opciones, es posible que desee utilizar el elemento <optgroup> para facilitar a los usuarios encontrar el que desean. El elemento <optgroup> debe colocarse dentro del elemento <select> y debe contener todos los elementos <option> que están relacionados con el grupo. Puede especificar un nombre para un grupo de opciones usando su atributo de título.

Este código de ejemplo muestra cómo podría crear un menú desplegable que tenga varios grupos de opciones.

<select name=”food” id=”food”>

<optgroup title=”Dairy”>

<option>Milk</option>

<option>Egg</option>

<option>Cheese</option>

</optgroup>

<optgroup title=”Bakery”>

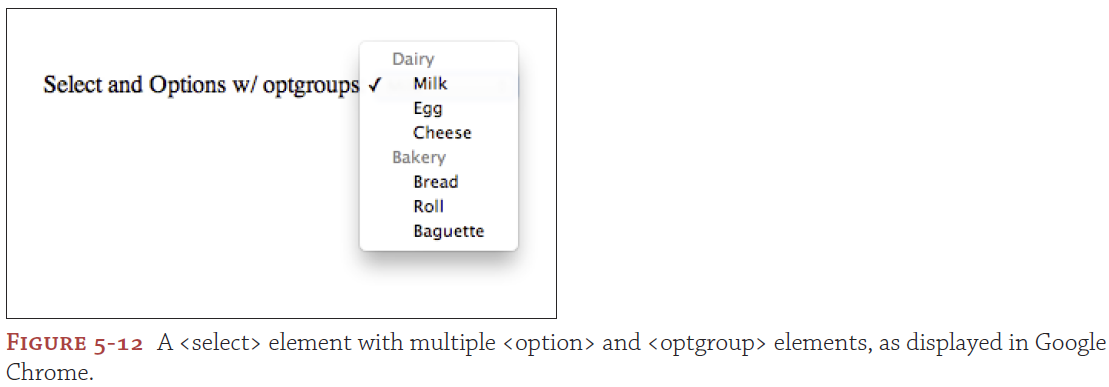
<option>Bread</option>

<option>Rolls</option>

<option>Baguette</option>

</optgroup>

</select>



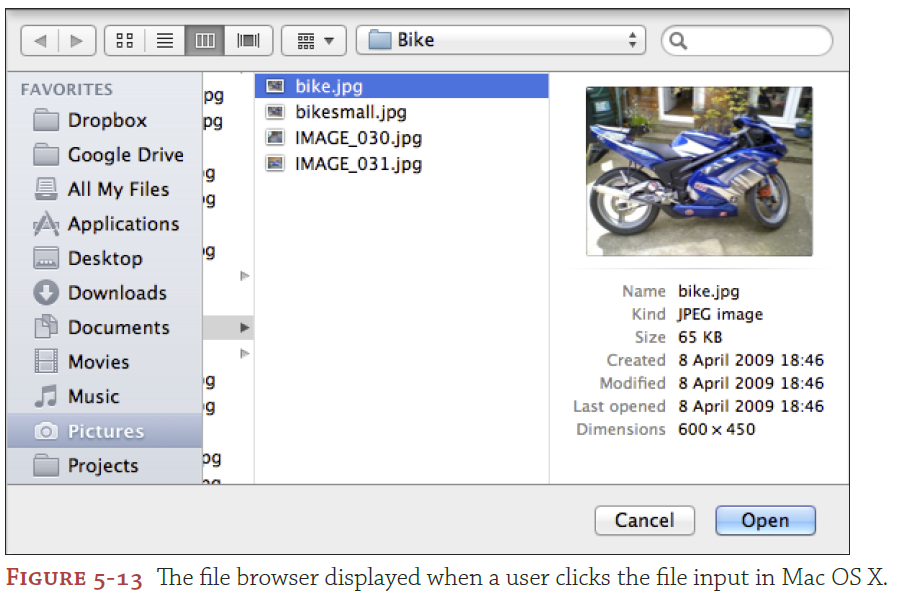
## Manejo de archivos

Hasta ahora, ha aprendido cómo puede recopilar texto de un usuario y cómo puede crear menús desplegables y botones de radio para restringir los valores de entrada. Ahora es el momento de ver cómo puede permitir a los usuarios cargar archivos. Para esta tarea, utiliza otro tipo para el elemento <input>:file.

## El tipo de entrada file

El tipo de entrada file muestra un botón que inicia un explorador de archivos para que los usuarios puedan seleccionar un archivo de su disco duro para cargarlo en su sitio web. En el siguiente ejemplo de código se muestra un ejemplo de una entrada de archivo en su forma más simple.

<input name=”avatar” type=”file”>



### Seleccionando múltiples archivos

Antes de que apareciera HTML5, la única manera de poder manejar con confianza múltiples cargas de archivos era usar la tecnología Flash de Adobe. Esto causó algunos dolores de cabeza a los desarrolladores que solo querían una solución simple para múltiples cargas de archivos. Una vez más, HTML5 ha llegado al rescate al permitir múltiples cargas de archivos con el simple uso del atributo múltiple.

El siguiente ejemplo muestra cómo puede permitir que un usuario seleccione varios archivos utilizando un solo elemento <input>.

<label id=”pictures”>Upload Pictures:</label>

<input name=”pictures” id=”pictures” type=”file” multiple>

Esta es una nueva adición en HTML5. Los navegadores web más antiguos no admiten el atributo múltiple.

### Especificando tipos de archivos aceptados

Si está esperando que un usuario suba una imagen de sí mismo para usarla como avatar en su sitio de redes sociales, debe asegurarse de que no intente cargar una hoja de cálculo. El atributo accept le permite especificar qué tipo de archivo está esperando para que el navegador pueda iniciar un navegador de archivos que solo permita al usuario seleccionar esos tipos de archivos.

Algunos valores posibles para el atributo de aceptación son: imagen / \*, audio / \* y video / \*. Puede encontrar una lista completa de estos valores (también conocidos como tipos MIME) aquí: <http://www.webmastertoolkit.com/mime-types.shtml>.

Aquí hay un ejemplo de un archivo <entrada> que solo acepta archivos de imagen.

<label id=”photos”>Upload Photos:</label>

<input name=”photos” id=”photos” type=”file” accept=”image/\*”

multiple>

El atributo accept es una nueva adición a HTML5, por lo que no funcionará en navegadores web más antiguos.

## Agrupando los campos de entrada en su página de reservas

## usando <fieldset> y <legend>

Al crear formularios web más grandes, a menudo le resultará útil agrupar campos de formulario relevantes. Puede hacerlo utilizando los elementos <fieldset> y <legend>.

***El elemento <fieldset> se utiliza para definir un grupo de entradas y debe contener todos los elementos de formulario relacionados con ese grupo***. ***El elemento <legend> se usa para especificar un nombre para un grupo de fieldset y se debe colocar después de la etiqueta de inicio del elemento <fieldset>,*** como se muestra en el siguiente ejemplo.

<fieldset>

<legend>Contact Details</legend>

<div class=”field”>

<label for=”name”>Name:</label>

<input type=”text” name=”name” id=”name”>

</div>

<div class=”field”>

<label for=”email”>Email:</label>

<input type=”text” name=”email” id=”email”>

</div>

</fieldset>

También puede establecer el ***atributo disabled*** en un elemento <fieldset>. Esto deshabilitará todos los campos dentro del conjunto de campos para que el usuario no pueda editarlos.

Echa un vistazo a este video de Treehouse donde Nick Pettit cubre campos, leyendas y etiquetas:

<http://teamtreehouse.com/library/websites/html/forms/fieldsets-andlabels>.

Ahora va a agregar los toques finales a su formulario de reservas, creando dos grupos de campos de entrada: uno para los detalles de contacto del cliente y otro para los detalles de la reserva.





## EJERCICIO-6

1. Abra su archivo bookings.html.
2. Cree un nuevo elemento <fieldset> en la parte superior de su formulario, justo debajo de la etiqueta de inicio para el elemento <form>.
3. Dentro de este <fieldset>, cree un elemento <legend> con el texto Detalles de contacto.
4. Ahora mueva las entradas de nombre, teléfono y correo electrónico a este elemento <fieldset>, directamente debajo del elemento <legend>.
5. Debajo de este <fieldset>, crea otro elemento <fieldset>.
6. Cree un nuevo <legend> en este <fieldset> con el texto Información de la reserva.
7. Mueva los campos de restaurante, tiempo de reserva, invitados, marketing y formulario de mensaje a este elemento <fieldset>, dejando el botón fuera de él.
8. Guarde el archivo bookings.html.

Ahora debería tener dos grupos de campos de formulario, como se muestra en el siguiente código.

<form action = "bookings.php” method = "POST”>

<fieldset>

<legend>Contact Details</legend>

<div class=”field”>

<label for=”name”>Name:</label>

<input type=”text” name=”name” id=”name”>

</div>

<div class=”field”>

<label for=”phone”>Phone:</label>

<input type=”text” name=”phone” id=”phone”>

</div>

<div class=”field”>

<label for=”email”>Email:</label>

<input type=”text” name=”email” id=”email”>

</div>

</fieldset>

<fieldset>

<legend>Booking Information</legend>

<div class=”field”>

<label for=”restaurant”>Restaurant:</label>

<select name=”restaurant” id=”restaurant”>

<option>310 West 38th Street, NY</option>

<option>2450 Broadway, NY</option>

<option>200 West 44th Street, NY</option>

</select>

</div>

<div class=”field”>

<label for=”bookingTime”>Booking Time:</label>

<input type=”text” name=”bookingTime” id=”bookingTime”>

</div>

<div class=”field”>

<label for=”guests”>Number of Guests:</label>

<input type=”number” name=”guests” id=”guests”>

</div>

<div class=”field”>

<label for=”marketing”>Please tick this box if you would like

to receive special offers from Joe’s Pizza Co.</label>

<input type=”checkbox” name=”marketing” id=”marketing”

value=”1”>

</div>

<div class=”field”>

<label for=”message”>Special Requests:</label>

<textarea id=”message” name=”message” cols=”50” rows=”10”>

</textarea>

</div>

</fieldset>

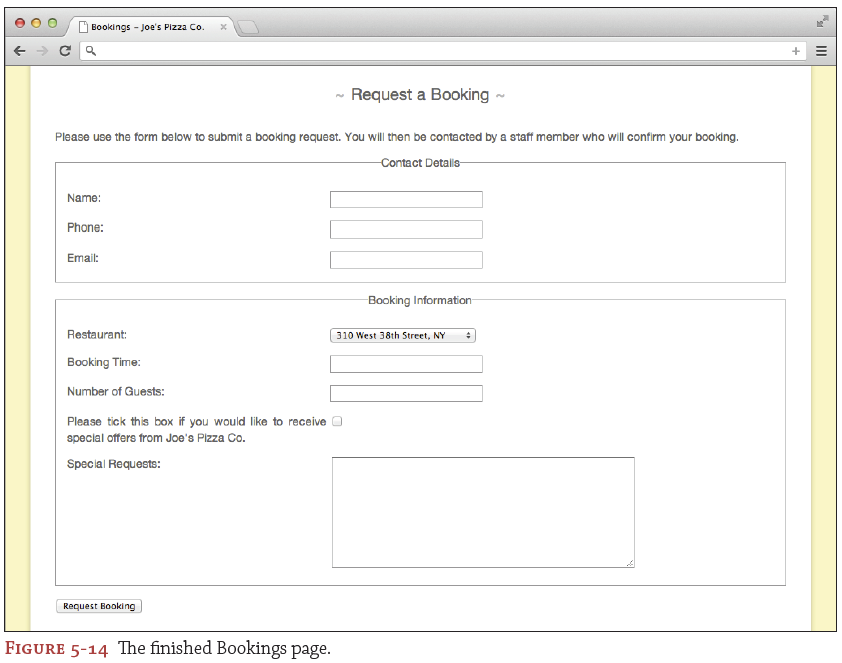
<div class=”field”>

<button type=”submit”>Request Booking</button>

</div>

</form>

¡Felicidades! Ya ha terminado el formulario de reservas (por ahora). El envío de este formulario no hará nada porque no tiene un archivo bookings.php para procesar los datos. La Figura 5-14 muestra cómo debe verse su página de Reservas. En los siguientes dos temas, aprenderá a usar HTML5 para hacer que este formulario web sea un poco más emocionante.





## Actualización de la navegación del sitio y el sitemap

A medida que sus sitios web maduren, se encontrará agregando nuevas páginas. Cada vez que haga esto, deberá actualizar la navegación del sitio en cada una de sus páginas, así como actualizar la lista de páginas en la página Mapa del sitio.

Ahora que ha creado su página de reservas, debe actualizar la navegación del sitio y el mapa del sitio para incluir enlaces a esta nueva página.



## EJERCICIO 7

1. Abra el archivo sitemap.html.
2. Localice el elemento <nav> en el encabezado de la página.
3. Al final de la lista de navegación, cree un nuevo elemento <li>.
4. Dentro de este elemento <li>, cree un nuevo enlace usando el elemento <a>:

<li>

<a href=”bookings.html” title=”Make a booking”>Bookings</a>

</li>

1. Guarde este archivo y repita los pasos 1 a 4 para cada página de su sitio web. Es posible que le resulte más fácil copiar y pegar el nuevo elemento <li> en la sección de navegación en cada uno de los archivos. No olvide agregar la clase activa al enlace cuando la agregue a la página de Reservaciones.
2. Vuelva al sitemap.html y localice la lista de enlaces que conforman el sitemap.

Este elemento <ul> tiene el ID del mapa del sitio.

1. Usando el código del Paso 4, agregue un nuevo enlace al mapa del sitio.
2. Guarde el archivo sitemap.html.

La Figura 5-15 muestra cómo debe verse la página de Sitemap actualizada en su navegador web.

