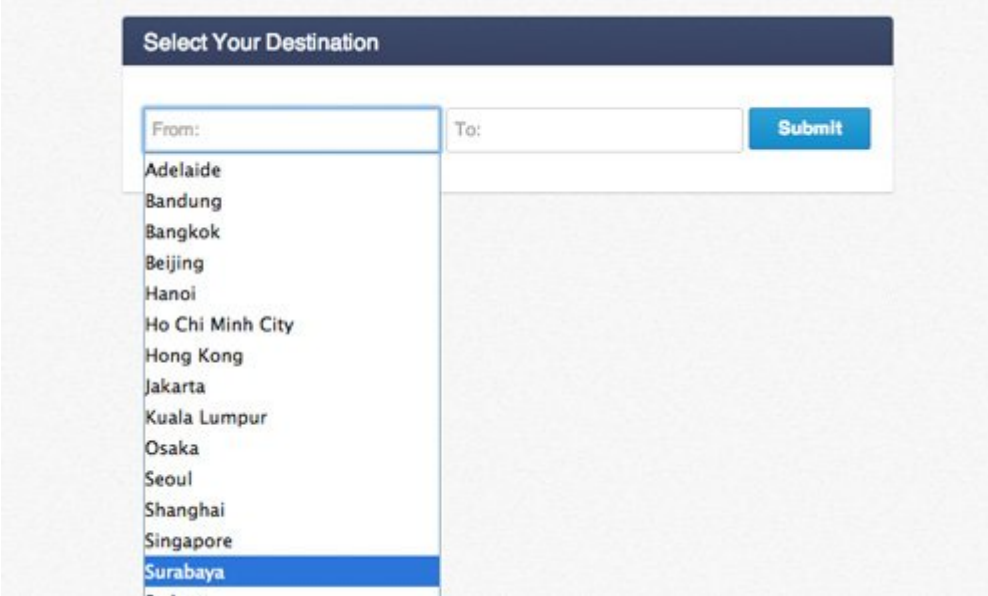


Una mirada a HTML5 DataList

Usted probablemente ha experimentado este tipo de respuesta en algún lugar de la web: al escribir en un formulario de entrada, una lista de sugerencias aparece en la parte inferior de la entrada.



The image shows a web form titled "Select Your Destination". It features two input fields: "From:" and "To:". The "From:" field is active, and a dropdown menu is displayed below it, listing various cities: Adelaide, Bandung, Bangkok, Beijing, Hanoi, Ho Chi Minh City, Hong Kong, Jakarta, Kuala Lumpur, Osaka, Seoul, Shanghai, Singapore, and Surabaya. A blue "Submit" button is located to the right of the "To:" field.

A través de los años, nos basamos en JavaScript para crear una función de este tipo. Hoy en día, estamos en condiciones de hacer lo mismo con el nuevo elemento HTML5 `<datalist>`. Este nuevo elemento HTML5 nos permite almacenar una lista de opciones que se mostrarán como el tipo de usuarios en la entrada.

HTML5 lista de atributos

El `<datalist>` puede vincularse a elementos `<input>` utilizando un nuevo atributo `lista`. HTML5, Podemos utilizar la lista en algunos tipos de entrada como texto y la búsqueda. Pero, de acuerdo con Mozilla Developer Network, el atributo de lista no es aplicable dentro de los siguientes tipos de entrada: oculto, casilla de verificación, radio, archivo y botón.

En el ejemplo siguiente, se añade una lista de las ciudades

con `<datalist>` y asignamos con el atributo ID, así.

```
[code lang=»html»]
<datalist id="city">
<option value="Adelaide">
<option value="Bandung">
<option value="Bangkok">
<option value="Beijing">
<option value="Hanoi">
<option value="Ho Chi Minh City">
<option value="Hong Kong">
<option value="Jakarta">
<option value="Kuala Lumpur">
<option value="Osaka">
<option value="Seoul">
<option value="Shanghai">
<option value="Singapore">
<option value="Surabaya">
<option value="Sydney">
<option value="Tokyo">
</datalist>
[/code]
```

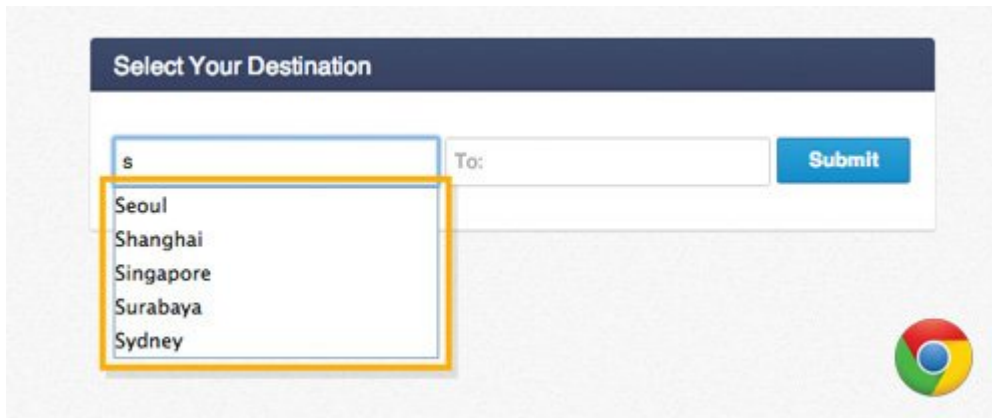
Para enganchar el `<datalist>` a un elemento `<input>`, sólo tiene que añadir atributo de lista y consulte el atributo id, como tal.

```
[code lang=»html»]
<input type="text" placeholder="From:" list="city">
<input type="text" placeholder="To:" list="city">
[/code]
```

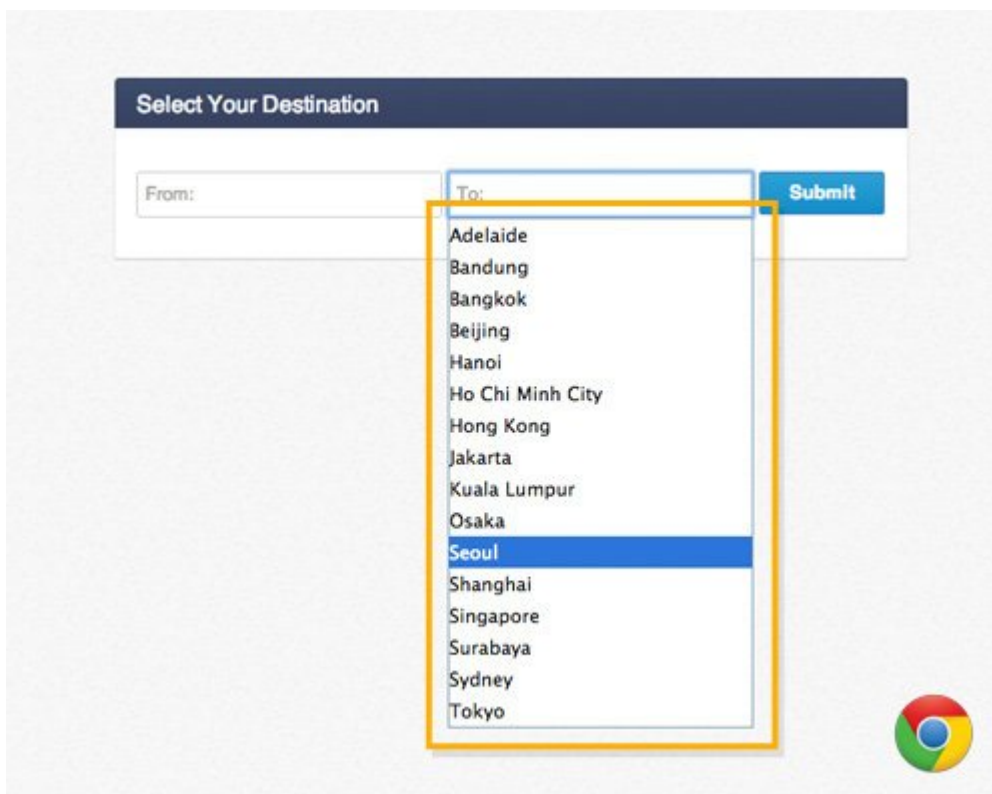
Comportamiento del navegador

Cada uno de los navegadores que soportan elemento `<datalist>`, Chrome, Opera y Firefox actúan ligeramente diferente. En Chrome, mostrará las opciones que empiezan con el valor que

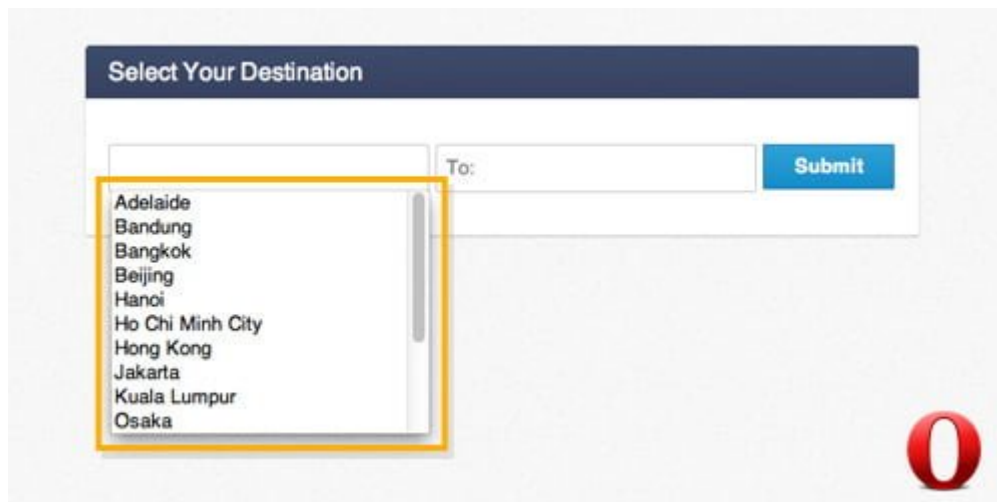
escribimos escribimos en el input. En el ejemplo siguiente, Chrome muestra los valores que comienzan con «s».



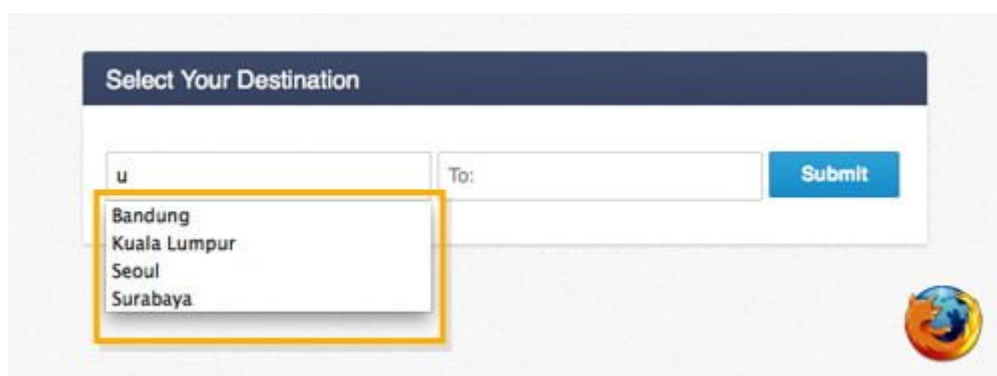
Por otro lado, como se hace clic en la entrada dos veces, mostrará todas las opciones disponibles, como se muestra a continuación.



En Opera, como nos estamos centrando en el formulario de entrada, se muestra inmediatamente todas las opciones y luego ordena las que se basan en las letras iniciales que digitamos



Firefox no mostrará nada al enfocarse en la entrada de texto. Se mostrarán las opciones que contienen los valores a medida que escribimos o digitamos. Por ejemplo, la siguiente imagen muestra que el navegador Firefox muestra las opciones que contienen la letra «u».



Safari por el momento, no soporta el elemento `<datalist>` e Internet Explorer se informa que soportara este elemento a partir de la versión 10.

El uso de Datalist HTML5 con polyfills

Hay algunas polyfills o trucos para reproducir una funcionalidad similar en los navegadores compatibles. En este post, vamos a poner en práctica las polyfills DataList de Michael Taylor. Para usarlo, necesitaremos jQuery y Modernizr para la detección de características del navegador.

En cuanto a Modernizr, tenemos que hacer algunas

personalizaciones. En la página de descarga Modernizr, compruebe las siguientes opciones.

- Input Attributes en la sección HTML5
- Modernizr.addTest en la sección de extensibilidad
- Modernizr.load en la sección adicional
- elem-datalist en detectar complementos

Ahora, vamos a abrir el documento HTML donde agregamos el elemento `<datalist>` y añadimos la biblioteca Modernizr que acabamos de descargar en la sección head.

```
[code lang=»html»]  
<script src="js/modernizr.js" type="text/javascript"></script>  
[/code]
```

Después de añadir Modernizr, podemos probar en el navegador si admite elemento lista de datos o datalist con el script de abajo.

```
[code lang=»js»]  
if (!Modernizr.datalistelem) {  
  alert('This browser does not support HTML5 datalist element,  
  so we will load the polyfills');  
}  
[/code]
```

La secuencia de comandos mostrará una alerta que dice «Este navegador no soporta elementos DataList HTML5, por lo que cargará los polyfills» cuando no se admite elementos DataList. En Safari, aparece esto.



A continuación, cree un nuevo archivo de JavaScript llamado `load.datalist.js` y añada esta línea. Esto tendrá como objetivo ejecutar la secuencia de comandos para la entrada que tiene el atributo de lista.

```
[code lang=»js»]
$('input[list]').datalist();
[/code]
```

Por último, vamos a cargar los jQuery, `jquery.datalist.js` y `load.datalist.js` utilizando `Modernizr.load`, de la siguiente manera.

```
[code lang=»js»]
Modernizr.load({
test: Modernizr.datalistelem,
nope:   ['js/jquery.js',    'js/jquery.datalist.js',
'js/load.datalist.js']
});
[/code]
```

Por lo tanto, se cargan sólo cuando el navegador no soporta elemento `<datalist>`.

Select Your Destination

To:

Submit

- Adelaide
- Bandung
- Bangkok
- Beijing
- Hanoi
- Ho Chi Minh City
- Hong Kong
- Jakarta
- Kuala Lumpur
- Osaka
- Seoul
- Shanghai
- Singapore
- Surabaya
- Sydney
- Tokyo

