Learn

Creating Autocomplete Dropdowns with the Datalist Element

Matt West

writes on March 11, 2014

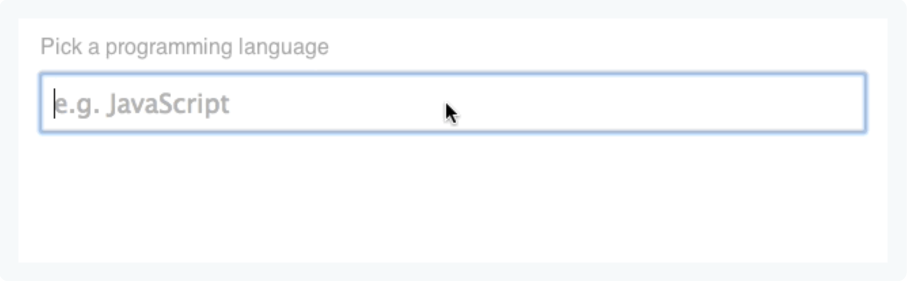
f 34 0 g+ 0 in 8 ✉

The `<datalist>` element, a new addition in the HTML5 specification, allows developers to create native autocomplete dropdowns for their web applications. This type of user interface component is commonly used on form fields and search boxes as it helps the user input data faster. In the past, autocomplete dropdowns could only be achieved using JavaScript.

In this blog post, you're going to learn how to use the `<datalist>` element to create native autocomplete dropdowns for your applications.

Let's get started.

The `<datalist>` Element



A datalist autocomplete dropdown in Google Chrome.

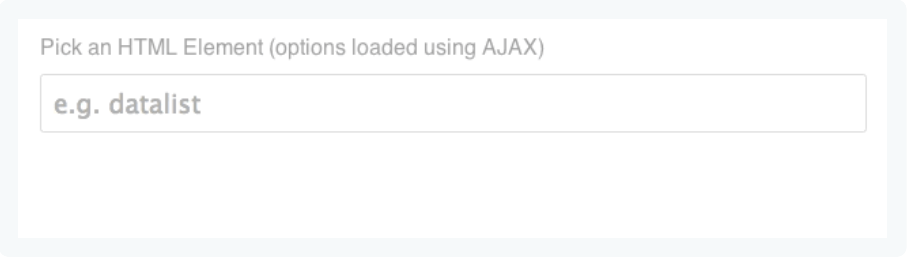
The `<datalist>` element is used to specify all of the possible values for the autocomplete list. Each of these values is defined using an `<option>` element, as shown in the example below.

```
<datalist id="languages">
  <option value="HTML">
  <option value="CSS">
  <option value="JavaScript">
  <option value="Java">
  <option value="Ruby">
  <option value="PHP">
  <option value="Go">
  <option value="Erlang">
  <option value="Python">
  <option value="C">
  <option value="C#">
  <option value="C++">
</datalist>
```

The `<datalist>` element should have an `id` attribute. To link your datalist to an `<input>` element, you need to specify a `list` attribute on the input and set its value to the `id` of your datalist.

```
<input type="text" list="languages">
```

Loading Options via AJAX



Pick an HTML Element (options loaded using AJAX)

e.g. datalist

A datalist autocomplete dropdown with options loaded via AJAX.

If you want to add a large number of options to your `<datalist>`, you may want to store those values in an external JSON file. You can then fetch this file via AJAX and populate the `<datalist>` options once the page has loaded. Let's look at an example of how to do this.

Like before, we start with `<input>` and `<datalist>` elements. Ensure that you link the two using the `id` and `list` attributes.

```
<input type="text" id="ajax" list="json-datalist" placehol
<datalist id="json-datalist"></datalist>
```

We then need to get references to these two elements in the JavaScript code.

```
// Get the <datalist> and <input> elements.
var dataList = document.getElementById('json-datalist');
var input = document.getElementById('ajax');
```

Next, we need to load the JSON file (`html-elements.json`) and populate the options for the `<datalist>` element. For this example, we're just using an array of strings, but it's also possible to use a more complex data structure.

```
// Create a new XMLHttpRequest.
var request = new XMLHttpRequest();

// Handle state changes for the request.
request.onreadystatechange = function(response) {
  if (request.readyState === 4) {
    if (request.status === 200) {
      // Parse the JSON
      var jsonOptions = JSON.parse(request.responseText);

      // Loop over the JSON array.
      jsonOptions.forEach(function(item) {
        // Create a new <option> element.
        var option = document.createElement('option');
        // Set the value using the item in the JSON array.
        option.value = item;
        // Add the <option> element to the <datalist>.
        dataList.appendChild(option);
      });

      // Update the placeholder text.
      input.placeholder = "e.g. datalist";
    } else {
      // An error occured :(
      input.placeholder = "Couldn't load datalist options
    }
  }
};

// Update the placeholder text.
input.placeholder = "Loading options...";

// Set up and make the request.
request.open('GET', 'html-elements.json', true);
request.send();
```

This code will load the `html-elements.json` file using an `XMLHttpRequest`. Once the file has been fetched, we parse the raw JSON data so that we have a JavaScript array to work with. We then loop over each of the items in this array. Each time we create a new `<option>` element, set the `value` attribute to the current item in the array, and then add this `<option>` element to the `<datalist>`.

In this example we're also using the `<input>` element's `placeholder` property to display the loading status for the datalist options.

See the Demo

View on CodePen

Browser Support

Support for the `<datalist>` element among browsers is pretty good. Safari is the only modern web browser not to include support for `<datalist>`. It's worth noting that the implementation of this element in IE10 is known to be buggy.

IE	FIREFOX	CHROME	SAFARI	OPERA
10.0+	4.0+	20.0+	–	9.0+

Source: http://caniuse.com/#feat=datalist

Final Thoughts

In this post, you've learned how to use the `<datalist>` element to create native autocomplete dropdowns for your web applications. You've also learned how to use AJAX to populate datalist options from an external JSON file.

The `<datalist>` element is just one of a number of new elements that provide native implementations of commonly used UI components. By making these elements really simple to use, browser vendors are removing the need for developers to rely on JavaScript libraries in these areas. Standardizing the behavior of UI components also has the added benefit of creating a more consistent experience for users across the web.

Useful Links

- Can I use... datalist
- WHATWG Spec

autocomplete code datalist

dropdown html5 learn programming

make a website | UI | web app

19 Responses to “Creating Autocomplete Dropdowns with the Datalist Element”

Chirag Bansal on July 18, 2017 at 7:05 am said:

Actually it is showing error “Cannot set property ‘placeholder’ of null” but when I printed the jsonOptions in console, it is printing the array that it fetching from the json file.

So can you suggest some solution for this error?
Thanks in Advance

Reply

ram on July 10, 2017 at 11:05 am said:

Is there a way to limit the size of dropdown list window?

Reply

> **Chirag on July 18, 2017 at 11:46 pm said:**
>
> Great Tutorial but this is the error which i am facing and i have already gone through different solutions available but couldn’t solve this issue:
>
> ajaxautocomplete.html:26 Uncaught TypeError: Cannot set property ‘appendChild’ of null
> at ajaxautocomplete.html:26
> at Array.forEach ()
> at XMLHttpRequest.request.onreadystatechange (ajaxautocomplete.html:20)
>
> **Reply**
>
>> **Chirag on July 18, 2017 at 11:51 pm said:**
>>
>> Thanks guys, problem is resolved!

Reply

Django on June 29, 2017 at 12:09 am said:

Hello,
Do we option to select more then one input from the drop down menu?
There is a way to do it by "select multiple" attribute in HTML5 and we can do it through js too.
But I want to select the same option more then one time.
Please let me know if there is any alternative.

Thanks.

Reply

BNP on June 5, 2017 at 8:58 am said:

Hey,
Thank for the tutorial.
I can't seem to create a legit request variable.
I'm using backbone and receiving back, as an AJAX result, a collection.

Shouldn't request = JSON.stringify(collection) do the trick [as it generates a valid JSON out of the collection]?

The error I'm receiving in the chrome's console is "Cannot create property 'onreadystatechange' on string"

Thanks

Reply

> Matt West on June 22, 2017 at 4:29 am said:
>
> Hey!
>
> Can you share your code in a gist (https://gist.github.com) and I'll take a look to see what's wrong.
>
> Reply

Joe on October 18, 2016 at 2:32 am said:

Pretty Cool, thanks for sharing 🙂

Reply

Ubaid on June 15, 2016 at 10:02 pm said:

Great Tutorial Matt.

Is there a way to limit the size of dropdown list window. I have created a similar autocomplete dropdown with a lot options.

Reply

Christine on December 15, 2015 at 12:45 am said:

Awesome feature, and very to easy follow tutorial of how it is to be used.
Thanks Matt!

Reply

hiims on December 5, 2015 at 5:27 am said:

awesome!
thanks a lot

Reply

人気スーパーコピーブランド時計激安通販専門店私達は長年の実体商店の販売経験を持って、先進とプロの技術を持って、高品質のスーパーコピー時計づくりに 取り組んでいます。最高品 on November 14, 2015 at 1:55 pm said:

スーパーコピーブランドショップ有名ブランド品の輸入業、卸売り業、インターネット小売販売N級,,S級 ヴィトン、シャネル、グッチ、財布、バッグ。 高級腕時計ROLEX 、OMEGA 、CARTIER 、 BVLGARI、是非ご覧下さい。2015超人気腕時計、LOUIS　VUITTON新品の大量入荷!商品の数量は多い、品質はよい、価格は低い、現物写真！本当の意味でのハイレプがほしい方にお勧めです。経営方針: 品質を重視、納期も厳守、信用第一は当社の方針です。
人気スーパーコピーブランド時計激安通販専門店私達は長年の実体商店の販売経験を持って、先進とプロの技術を持って、高品質のスーパーコピー時計づくりに 取り組んでいます。最高品質のロレックス時計コピー、カルティエ時計コピー、IWC時計コピー、ブライトリング時計コピー、パネライ時計コピー激安販売中商品の数量は多い、品質はよい。海外直営店直接買い付け！★ 2015年注文割引開催中,　全部の商品割引10% ★ 在庫情報随時更新! ★ 実物写真、付属品を完備する。 ★ 100%を厳守する。 ★ 送料は無料です(日本全国)!★ お客さんたちも大好評です★ 経営方針: 品質を重視、納期も厳守、信用第一！税関の没収する商品は再度無料にして発送します http://www.gginza.com/watch/iwc/index.html

Reply

imran on October 5, 2015 at 1:31 pm said:

what this json code get a data from database with query?

Reply

jewellery sets for rent in chennai|artificial bridal jewellery for rent in chennai|bridal jewellery sets in chennai for rent|bridal jewellery for rent in chennai|kundan bridal jewellery sets for rent in chennai|bridal jewellery set for rent in chennai|bri on September 17, 2015 at 4:29 pm said:

Thank you, I've recently been searching for info approximately this topic for a long time and yours is the best I have found out till now. However, what about the bottom line? Are you positive concerning the source?|

Reply

Matt Hinchliffe on March 12, 2014 at 7:12 am said:

It would be interesting if you could gather screenshots across a range of mobile devices to see how their 'adaptive' interfaces handle this (E.G. when a user interacts with a number field they're displayed a keyboard with only numbers).

Reply

> Matt West on March 12, 2014 at 7:18 am said:
>
> Hi Matt,
>
> Unfortunately I don't think this is supported on mobile yet (except in the Blackberry browser). Source: http://caniuse.com/#feat=datalist
>
> Reply
>
> > Matt Hinchliffe on March 12, 2014 at 7:50 am said:
> >
> > It looks like iOS is certainly behind =(
> >
> > Both Chrome and Firefox on Android (4.4) support it, https://dl.dropboxusercontent.com/u/2664340/Screenshot_2014-03-12-11-15-25.png
> >
> > Reply

Brandon Brown on March 11, 2014 at 1:13 pm said:

Great post Matt 🙂 I had no idea that the datalist element existed and that it could do native browser autocompletion

Reply

> **Matt West** on **March 11, 2014 at 3:35 pm** said:
>
> Thanks for reading Brandon 🙂
>
> **Reply**

Leave a Reply

Name *

Email Address (will not be published) *

Website

Submit Comment

Stay current

Sign up for our newsletter, and we'll send you news and tutorials on web design, coding, business, and more!

Email Address

Subscribe

About • Careers • Blog • Affiliate Program • Terms • Privacy • Press Kit • Contact