Object Selection and Object Creation with JQuery $("…")

Source: https://api.jquery.com/jquery/#jQuery-html-ownerDocument

Creating New Elements

If a string is passed as the parameter to $(), if the string appears to be an HTML snippet (i.e., it starts with <tag ... >), jQuery attempts to create new DOM elements according to the HTML, and a jQuery object referring to these elements is returned.

If not HTML code, the string is interpreted as a selector expression (e.g., $("dvi")).

When the string has a single tag, closing tag is optional, i.e., $("<li>") is the same as $("<li></li>"), and quick-closing is allowed $("<img/>"). A good practice is to always use well formed HTML (paired all tags).

Example 1:

```
$( "<p id='test'>My <em>new</em> text</p>" ).appendTo( "body" )
```

Example 2:

```
$( "<div/>", {
 "class": "test",
 text: "Click me!",
 click: function() {
   $( this ).toggleClass( "test" );
 }
}).appendTo( "body" );
```

JQuery append() vs appendTo()

Reference: https://www.tutorialspoint.com/What-is-the-difference-between-append-and-appendTo-in-jQuery#:~:text=What%20is%20the%20difference%20between%20append()%20and%20appendTo()%20in%20jQuery%3F,-jQueryWeb%20Development&text=The%20append%20(content)%20method%20appends,%2C%20specified%2C%20set%20of%20elements.

- append (content) appends content to the inside of every matched element
- appendTo (selector) appends all of the matched elements to a set of elements

Example

$("div").click(function () { $(this).append('<div></div>'); });	$("div").click(function () { $(this).appendTo("#result"); }); <p id = "result"> THIS IS TEST </p>

Styling the (Drop-Down) Menu Items

Reference: https://www.htmlgoodies.com/beyond/javascript/modify-jquery-autocomplete-widget-response-data.html

Methods for customizing auto-complete drop-down content begin with an underscore and are exposed via the.data() method with the "ui-autocomplete" key. It allows us to attach data of any type to DOM elements. It provides two parameters as input to the customized function: the **<ul>** element that newly created <li> elements are appended to, and the current menu **item**, i.e., the response object that holds the label and value attributes.

The code below adds two elements to the menu item: an <img> element (or <span> if none exists) and a <span> for match highlighting.

```
<li class="ui-menu-item">
 <div tabindex="-1" class="ui-menu-item-wrapper" id="ui-id-2">
  <img class="m-icon" src="data:image/jpeg;base64,/9j.../Z">
  <span class="m-name">
   <span class="highlight">Jam</span>ie's Italian
  </span>
 </div>
</li>
```

The _renderItem() method sets the logo source based on the item.icon variable that has to be set somewhere before:

```
$('#autocomplete').autocomplete({
//autocomplete options...
}).data('ui-autocomplete')._renderItem = function( ul, item ) {
 var $div = $("<div></div>");
 if (item.icon) {
  $("<img class='m-icon'>").attr("src", item.icon).appendTo($div);
 } else {
  $("<span class='x-icon'>no img</span>").appendTo($div);
 }
 //highlighting code...
};
```