

jQuery: extending HTML DOM elements with `$.extend()`

The jQuery's `$.extend()` utility method virtually works with any object. An interesting object we'd like to extend with new methods is the prototype object of the `HTMLElement` interface. Why this object is so important? Simply put, it's the base object of all the HTML elements contained in the DOM. If we extend it with new methods, we'll be able both to work within or outside the jQuery's wrapped set. The important thing to note here is that we can also use jQuery's methods in our custom methods. Let's see the details.

Here's how to extend the `HTMLElement`'s prototype object:

```
$.extend(HTMLElement.prototype, {
    destroy: function() {
        $(this).remove();
    },
    update: function(value) {
        this.innerHTML = value;
    }
});
```

Now every HTML element in the DOM has two new methods. Notice how the `this` keyword works: if we insert it within the jQuery's `$()` wrapper, we can use all the jQuery's methods. On the contrary, if we use it as is, we can use the core DOM methods. Here's an example:

```
var li1 = document.getElementsByTagName('li')[0];
var li2 = document.getElementsByTagName('li')[1];

$(li1).click(function() {
```

```
        $(this)[0].destroy();  
    });
```

```
$(li2).click(function() {  
    this.update('Test');  
});
```

You can see the demo below.

Demo

Live demo