jQuery Part 2

IMY 220 ● Lecture 11

Project

“Featuritis”

Never add stuff for the sake of adding them

Don’t add effects just “because it’s cool”

Always have a reason

If it looks cool, but makes the site harder/annoying/frustrating to use, change it or take it out

Project

Design

Look at what’s out there (https://www.awwwards.com/)

Doesn’t have to be flashy

Ask your brutally honest friends to use it

JavaScript

The Web Without JS

A small number (1-3% depending on which part of the world) browse the web without JS enabled

This has caused a debate around whether or not to cater for these small numbers

- Some government/public websites are required to work without JS
- Issues of accessibility come into play as well

https://www.wired.com/2015/11/i-turned-off-javascript-for-a-whole-week-and-it-was-glorious/

https://webmasters.stackexchange.com/questions/4733/should-i-worry-about-people-disabling-javascript

Progressive Enhancement

Depending on your site’s demographic, if up to 3% of your users might be browsing without JavaScript capabilities, and if you have a million customers, that’s 30,000 alienated customers!

“***Progressive enhancement*** *is a design philosophy that provides a baseline of essential content and functionality to as many users as possible, while delivering the best possible experience only to users of the most modern browsers that can run all the required code.*”

https://developer.mozilla.org/en-US/docs/Glossary/Progressive_enhancement

Progressive Enhancement

If a user came to our site using a browser lacking support for JavaScript, they'd see a button on the page that would do nothing when they clicked it.

- For the disclaimer functionality we can make the disclaimer to be visible to all users, so we place it in our HTML.
- Then, we add the ability to hide it for users with JavaScript.
- Don't display the show/hide button to users who'll be unable to make use of it.

You can add the button to the page via jQuery; that way, only users with JavaScript will see the button.

Graceful Degradation

“Graceful degradation is a design philosophy that centers around trying to build a modern web site/application that will work in the newest browsers, but falls back to an experience that while not as good still delivers essential content and functionality in older browsers.”

https://developer.mozilla.org/en-US/docs/Glossary/Graceful_degradation

GD and PE are closely related; both aim to create baseline functionality across all systems and enhanced user experience (UX) for everyone with the newest technology supported and enabled

Adding New Elements

So far we’ve seen the jQuery function used for selecting, but it does have another function of equal importance: creating new elements.

Any valid HTML string you put inside the jQuery function will be created and made ready for you to put on the page.

Here’s how we might create a simple paragraph element

Adding New Elements

```
$('<p>A new paragraph!</p>')
```

The above code does the following:

  It parses the HTML code into a DOM fragment.

  It selects the new element just like an ordinary jQuery selector.

You can directly work with the newly created element e.g. to add a class to our newly created element:

```
$('<p>A new paragraph!</p>').addClass('new');
```

Adding New Elements

```
$("div.container").append(
    $("<p></p>", {
        html: 'Some text',
        class: 'myClass'
    })
);
```

If you’re getting the values for the text, classes, etc. for an element from variables/functions/etc., this is the **recommended syntax**

This syntax allows you to create complicated elements in an orderly fashion

Adding New Elements

```
let element1 = $('<p></p>', {
      html: 'First element',
      class: 'myClass’
});

let element2 = $('<p></p>', {
      html: 'Second element',
      class: 'myClass’
});

let elementList = [element1, element2];

$('div.container').append(elementList);
```

You can also supply the append function with an array of newly created elements, in which case it appends each element individually

Adding New Elements

```
$('div.container').append(
      $('<p></p>', {
            html: 'Child of the div',
            class : 'myClass'
      })
      .append(
            $('<span></span>', {
                  html: 'Child of the paragraph'
            })
      )
);
```

You can also nest append statements to add child elements to new child elements

insertAfter or insertBefore

When the new element is created it needs to be inserted into the HTML page somewhere.

```
$('<button></button>'), {
   id: 'toggleButton',
   html: 'Toggle'
})
.insertAfter('#disclaimer');

$('#toggleButton').on('click', function(){
       $('#disclaimer').toggle();
});
```

The above example creates the new element (a button) and inserts it after the element with the id “disclaimer”.

insertAfter or insertBefore

The new element is added for every element that matches the selector

- **insertAfter** adds the new element as a sibling after the disclaimer element.

- **insertBefore** will also place the element as a sibling but before the disclaimer element.

Sibling = In terms of the DOM hierarchy, it is on the same level.

```
$('<b></b>', {html: 'Text'}).insertBefore('p');
```

The example will add “A PARAGRAPH” before every paragraph element of the page.

prependTo or appendTo

If you want to add the new element as a **child** of an element (instead of a sibling) you use the following:

```
$('<strong></strong>', {html: 'Begin'}).prependTo('#disclaimer');
$('<strong></strong>', {html: 'End'}).appendTo('#disclaimer');
```

This will place the new elements at the start and the end of the disclaimer element rather than before or after.

Removing existing elements

We can add a warning to the HTML to those with JavaScript disabled, then we take out the warning with JavaScript.

```html
<p id="no-script">We recommend that you have JavaScript
enabled!</p>
```

We then remove the element with jQuery

```javascript
$('#no-script').remove();
```

The remove action will remove:

- All selected elements from the DOM.
- All event handlers attached to the elements.
- All data attached to the elements.

Removing existing elements

remove does not take arguments but you can refine the selection:

```
$('#celebs tr').remove(':contains('Singer')');
```

The above code removes only the table rows from the celebrity table that contains the text “Singer”.

Traversing the DOM tree

jQuery provides a number of functions for traversing the DOM tree, such as

- children()
- parent()
- parents()
- find()

```
$('#hideButton').on('click', function() {
    $(this).parent().hide();
    // parent() goes up 1 level in the DOM tree
    // and selects the parent element
});
```

Traversing the DOM tree

.children() takes an optional selector and returns all child elements of an element that match the selector

(or all children if no selector is provided)

```
$("#users").children(".col").addClass("new");
```

.parents does the inverse, i.e. it searches up the DOM tree for any parent elements that match an optional selector

```
$(".col").parents("#users").addClass("new");
```

Traversing the DOM tree

.find() takes an optional selector and returns all descendants of an element that match the selector

In other words, not just the children of the element, but their children, children’s children, etc.

```
$("#users").find(".col").addClass("new");
```

The jQuery UI Library

The jQuery UI library is a collection of advanced jQuery widgets, effects, and interactions—such as date pickers, accordions, and drag-and-drop functionality

The jQuery UI Library

Downloading and including the jQuery UI library:

- More complicated to set up than the core library and plugins we've been using so far.

- This is because the full jQuery UI library is very large.

- Any given project will normally only require a small subset of the functionality contained in the jQuery UI, and the jQuery website provides us with a handy tool to create our own custom version of the jQuery UI containing only what we need and nothing more.

http://jqueryui.com/download

Scrolling

We will look at menus that stay where they are while the user scrolls.

Custom-themed scrollbars.

Use animation techniques and apply them to scrolling the document.

Scroll event:

> Every time a user interacts with a scroll bar (be it in the main window or a scrollable div) the scroll event fires.
>
> To catch the event we attach a handler to the element that has scroll bars (mostly the window element).
>
> To test this, we set the overflow CSS property of a div to scroll.

```
#news {
        height: 100px;
        width: 300px;
        overflow: scroll;
}
```

Scrolling

We now attach the scroll event handler to the element and output arbitrary text to the page every time someone scrolls the div.

```
$(window).on("scroll", function() {
      $('#header')
      .append(
            $("<span></span>", {
                  html: "You scrolled",
                  class: "scroll"
            }
      );
});
```

Scrolling

To get the distance of an element from the top of the document:

```
$(element).offset().top
```

To get the scroll distance (how far the user has scrolled down):

```
$(window).scrollTop()
```

Scrolling the document

When we have a lot of text on a HTML page we usually have links at the top that can jump to different parts on the same page.

We also have a link at the bottom to jump to the top.

We can add this functionality (and animate it) with jQuery.

Scrolling the document

To jump to the top of the page we use # in the href attribute in the link.

If we want to do this with jQuery, we use scrollTo, but we need to cancel the default action of the link otherwise the page will jump before the animation occurs.

```
$('a[href=#]').on("click", function() {
     $([document.documentElement, document.body])
     .animate({scrollTop: 0},'slow');

     return false;
     // Return false to cancel the default link action
});
```

Scrolling the document

We cancel the default action by using **return false;**

You can also see we use an attribute selector:

```
$('a[href=#]')
```

This will select the “a” elements with the matching attribute values.

Using data attributes

HTML5 is designed with extensibility in mind for data that should be associated with a particular element but need not have any defined meaning.

data-* attributes allow us to store extra information on standard, semantic HTML elements.

Using data attributes

HTML Syntax

Any attribute on any element whose attribute name starts with data- is a data attribute.

```
<article
id="electiccars"
data-columns="3"
data-index-number="12314"
data-parent="cars">
     …
</article>
```

Reading data attribute values

Reading values with JS or CSS:

https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Using_data_attributes

```
let article = document.getElementById('electriccars');

article.dataset.columns       // "3"
article.dataset.indexNumber   // "12314"
article.dataset.parent        // "cars"
```

Reading data attribute values

Reading values with jQuery:

Given the following HTML:

```
<div data-role="page"
data-last-value="43"
data-hidden="true"
data-options='{"name":"John"}'></div>
```

All of the following jQuery code will work:

```
$("div").data("role") === "page";
$("div").data("lastValue") === 43;
$("div").data("hidden") === true;
$("div").data("options").name === "John";
```

https://api.jquery.com/data/

Adding/changing data attribute values

Given two parameters, the **data** function sets the data-attribute to the value given

```
$("div").data("role", "page");

// This code sets/adds the
// value of "data-role" to "page"
```

Adding/changing data attribute values

Note that using the data() function to set data for elements does not change the HTML for that element, for example

```
<div data-role="panel"></div>
```

```
$("div").data("role", "page");
// The element still shows data-role="panel"

// But when we do the following
console.log($("div").data("role"));
// We get: page
```

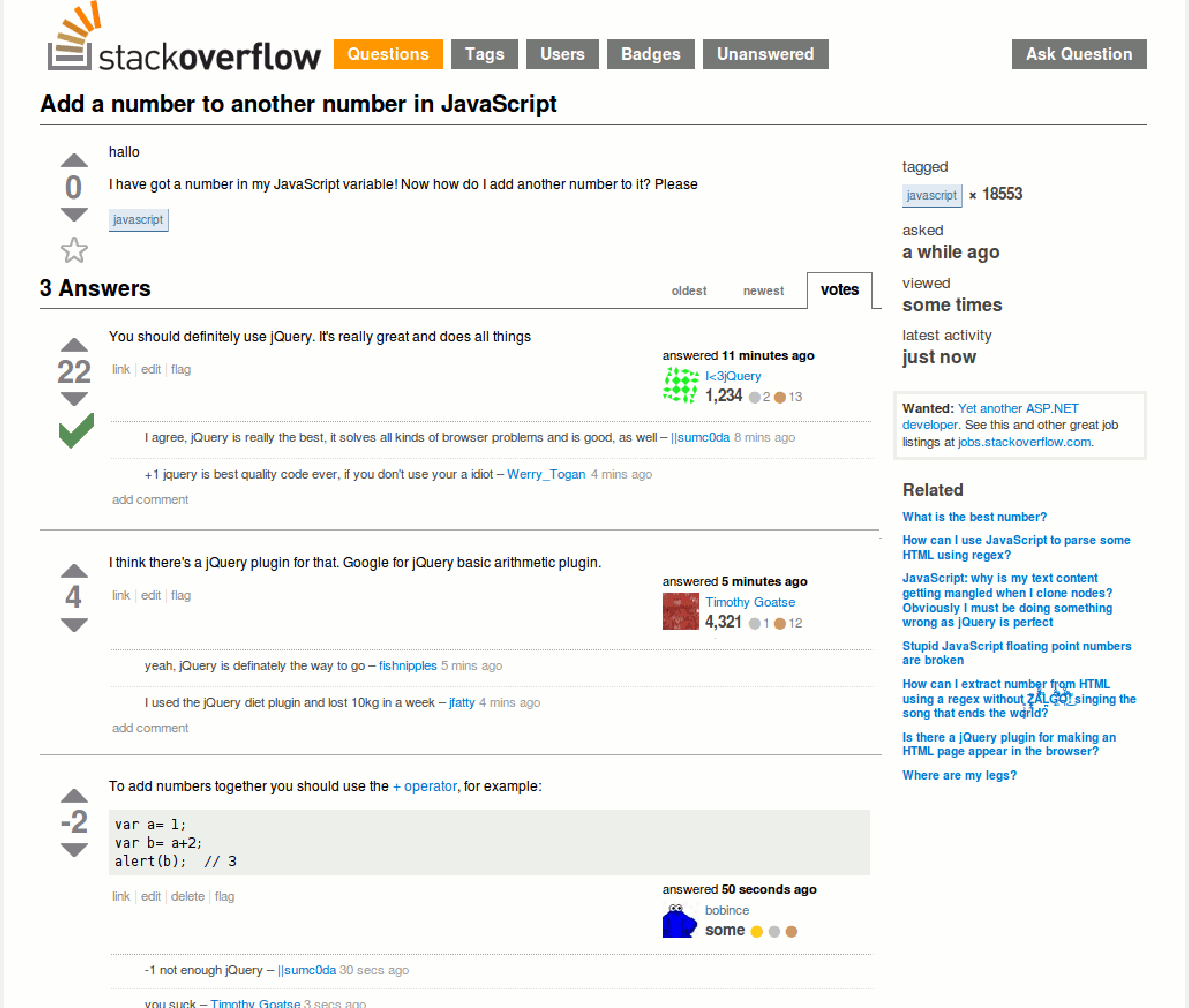
https://api.jquery.com/data/

Adding/changing data attribute values

If you really need to change the HTML, use the attr() function

```
$("div").attr("data-role", "page");
// The element now shows data-role="page"
```

https://api.jquery.com/data/



Add a number to another number in JavaScript

0

hallo

I have got a number in my JavaScript variable! Now how do I add another number to it? Please

javascript

3 Answers

oldest newest **votes**

22

You should definitely use jQuery. It's really great and does all things

link | edit | flag

answered **11 minutes ago**
l<3jQuery
1,234 ● 2 ● 13

I agree, jQuery is really the best, it solves all kinds of browser problems and is good, as well – ||sumc0da 8 mins ago

+1 jquery is best quality code ever, if you don't use your a idiot – Werry_Togan 4 mins ago

add comment

4

I think there's a jQuery plugin for that. Google for jQuery basic arithmetic plugin.

link | edit | flag

answered **5 minutes ago**
Timothy Goatse
4,321 ● 1 ● 12

yeah, jQuery is definately the way to go – fishnipples 5 mins ago

I used the jQuery diet plugin and lost 10kg in a week – jfatty 4 mins ago

add comment

-2

To add numbers together you should use the + operator, for example:

```
var a= 1;
var b= a+2;
alert(b);  // 3
```

link | edit | delete | flag

answered **50 seconds ago**
bobince
some ● ● ●

-1 not enough jQuery – ||sumc0da 30 secs ago

you suck – Timothy Goatse 3 secs ago

tagged
javascript × 18553

asked
a while ago

viewed
some times

latest activity
just now

Wanted: Yet another ASP.NET developer. See this and other great job listings at jobs.stackoverflow.com.

Related

What is the best number?

How can I use JavaScript to parse some HTML using regex?

JavaScript: why is my text content getting mangled when I clone nodes? Obviously I must be doing something wrong as jQuery is perfect

Stupid JavaScript floating point numbers are broken

How can I extract number from HTML using a regex without ZALGO singing the song that ends the world?

Is there a jQuery plugin for making an HTML page appear in the browser?

Where are my legs?

References

jQuery: Novice to Ninja by Earle Castledine and Craig Sharkie p29-49

http://api.jquery.com/

https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Using_data_attributes