JQuery

Events, animations, AJAX

Shadi Lahham - Programmazione web - Frontend - Javascript

Events

Review: JS Functions

```
// define a function
let sayHello = function (name) {
  console.log("Hello, " + name + "!");
}

// call a function
sayHello("Dolly");
```

Review: Callback Functions

```
let sayHello = function (name) {
  console.log("Hello, " + name + "!");
}

let textAFriend = function (name, callback) {
  callback(name);
}

// call the function with a named callback function
textAFriend("Dolly", sayHello);
```

Review: Callback Functions

```
let sayAwesome = function (name, callback) {
  callback(name);
}

//call the function with an anonymous callback function
sayAwesome("Dolly", function() {
  console.log("You are awesome " + name + "!");
});
```

jQuery: Events

```
// First Example, with named callback & .on
let onButtonClick = function() {
  console.log('clicked!');
};

$('button').on('click', onButtonClick);


// Second Example, with anonymous callback & .on
$('button').on('click', function () {
  console.log('clicked!');
});


// Third Example, with .click (& a named callback)
$('button').click(onButtonClick)
```

jQuery: Events

- Keyboard Events
  - 'keydown'
  - 'keypress'
  - 'keyup'
- Mouse Events
  - 'click'
  - 'mousedown'
  - 'mouseup'
  - 'mousemove'
- Form Events
  - 'change'
  - 'focus'
  - 'blur'

jQuery: Preventing default event

```
// default event for clicking on link is to go to new page
$('a').on('click', function (event) {
  event.preventDefault();
  console.log('Not going there!');
});

// default event is to submit form and reload page
$('form').on('submit', function (event) {
  event.preventDefault();
  console.log('Not submitting, time to validate!');
});
```

Animations

jQuery: Effects and animations

```
// on page load
$('.kitty-image').show(3000);

$('.kitty-image').fadeIn(3000);

// with an event handler, as a callback
$('button').click(function() {
  $('.kitty-image').show();
});

$('button').mouseover(function(){
  $(this).css('color', 'red');
});
```

jQuery: Animate

```
$(selector).animate({params},speed,callback);
```

```
$("button").click(function(){
    $("div").animate({
        left: '250px',
        opacity: '0.5',
        height: '150px',
        width: '150px'
    });
});
```

jQuery: Animate queue

You can define animations that run in sequence

```
$("button").click(function(){
    let div = $("div");
    div.animate({height: '300px', opacity: '0.4'}, "slow");
    div.animate({width: '300px', opacity: '0.8'}, "slow");
    div.animate({height: '100px', opacity: '0.4'}, "slow");
    div.animate({width: '100px', opacity: '0.8'}, "slow");
});
```

More here:
[jQuery Effects - Animation]

Plugins

jQuery: Plugins

Copy the files (not recommended):

1. Download the plugin and associated files from the site or github repo.

2. Copy them into your project folder.

3. In the HTML, reference any associated CSS files.
```
<link rel="type/stylesheet" type="text/css" href="tablesorter.css">
```

4. In the HTML, add a <script> tag for the jQuery plugin itself.
```
<script src="lib/tablesorter.js"><script>
```

5. In the JavaScript, call the jQuery plugin on your DOM.
```
$('table').tableSorter();
```

jQuery Plugin Registry

jQuery: Plugins

Hotlink the source files

1. In the HTML, add a <script> tag that hotlinks to the CDN or source of file.

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.17.0/jquery.validate.min.js"><scrip
t>
```

2. In the JavaScript, call the jQuery plugin on your DOM.

```
$("#commentForm").validate();
```

Note: always link to the minified js files

Example from: jQuery Form Validation Plugin

More jQuery

Patterns and Anti-patterns

Pattern: name variables with $

```
let $node = $('#myNode');
```

Pattern: store references to callback functions

```
let myCallback = function(argument) {
  // do something cool
};
$(document).on('click', 'p', myCallback);
```

Anti-pattern: anonymous functions

```
$(document).on('click', 'p', function(argument) {
  // do something anonymous
});
```

Chaining

```
banner.css('color', 'red');
banner.html('Welcome!');
banner.show();

// Is the same as:
banner.css('color', 'red').html('Welcome!').show();

// Is the same as:
banner.css('color', 'red')
      .html('Welcome!')
      .show();
```

DOM Readiness

DOM manipulation and event binding doesn’t work if the <script> is in the <head>

```
<head>
  <script>
    $('body').append( myNode ); // breaks - the DOM is not fully loaded yet
  </script>
</head>
```

DOM Readiness - jQuery

```
$(document).ready(function() {
  // the DOM is fully loaded
});
```

```
$(window).on('load', function() {
  // the DOM and all assets (including images) are loaded
});
```

The ready event:
fired when the DOM is fully loaded and accesses to elements are safe

The load event:
fired after the DOM and all assets have loaded (much slower)

DOM Readiness - Plain JS

```
function ready(fn) {
  if (document.attachEvent ? document.readyState === 'complete' : document.readyState !== 'loading')
{
    fn();
  } else {
    document.addEventListener('DOMContentLoaded', fn);
  }
}

ready(function() {
  // the DOM is fully loaded
});
```

DOM Readiness - Plain JS

A simpler solution?
Using only 'DOMContentLoaded' is not secure.
The callback will not execute if the event has already fired!

```
document.addEventListener('DOMContentLoaded', function() {
  // might never be called
});
```

The load event:

```
window.onload = function() {
  // the DOM and all assets (including images) are loaded
};
```

More info:
Quick Tip: Replace jQuery's Ready() with Plain JavaScript
Document: DOMContentLoaded event

AJAX

AJAX with jQuery

```
$.ajax({
  type: 'GET',
  url: 'filename.json',
  dataType: 'json',
  success: function(data) {
  },
  error: function(xhr, status, e) {
  }
});
```

AJAX with jQuery - more complete example

```
$.ajax({
  type: 'GET',
  url: 'https://api.myjson.com/bins/wgonb',
  dataType: 'json',
  success: function(response) {
    let books = response.books;
    for (let i = 0; i < books.length; i++) {
      let book = books[i];
      let p = $('<p>');
      p.html(book.title + ' by ' + book.author);
      $('body').append(p);
    }
  },
  error: function(xhr, status, e) {
    console.log(status, e);
  }
});
```

AJAX with jQuery 1.5+

```
$.ajax({
  method: 'POST',
  url: 'some.php',
  data: { name: 'John', location: 'Boston' }
})
  .done(function(msg) {
    console.log('Data Saved: ' + msg);
  })
  .fail(function(jqXHR, textStatus) {
    console.log('Request failed: ' + textStatus);
  });
```

You can chain the success and error handlers
The syntax is simpler and cleaner

Details: jQuery.ajax()

Plain JS GET request

```
// instantiate a new request
let request = new XMLHttpRequest();

// add event listeners
request.addEventListener('load', function() {
  // transform a string into a usable object
  console.log(JSON.parse(request.responseText));
});

request.open('get', '/path/to/api', true); // third parameter async
request.setRequestHeader('Content-type', 'application/json');
request.send();
```

Plain JS POST request

```
let request = new XMLHttpRequest();
request.open('POST', 'myservice/user');
request.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');
request.onload = function() {
  if (request.status === 200) {
    let userInfo = JSON.parse(request.responseText);
  }
};
request.send(
  JSON.stringify({
    name: 'John Smith',
    age: 34
  })
);
```

Your turn

1.Fun

- Use a jquery plugin to make your page more colorful
  - Create an HTML page with various text elements
    - <h1> - <h6>, <p>, <a>, , , etc
    - Use the funText jQuery plugin on your page
      - https://github.com/briznad/funText/
    - Hotlink directly to the github file
  - Here is a demo of what it can do
    - funText demo