Front-end Essentials

jQuery & AJAX

Table of Contents

Lesson Objectives

- Understand the core concepts of jQuery
- Able to use jQuery to manipulate HTML
- Understand the core concepts of AJAX
- Able to use AJAX (thanks to jQuery) to create dynamic web page

Section 1

jQuery Overview

jQuery Overview – What is jQuery?

- jQuery is a library of JavaScript Functions.
- jQuery greatly simplifies JavaScript programming.
- jQuery is easy to learn.

jQuery Overview – What is jQuery?

- jQuery is a lightweight "write less, do more" JavaScript library.
- The purpose of jQuery is to make it much easier to use JavaScript on your website.

jQuery Overview – Why use jQuery?

- **The jQuery library contains the following features:**
  - HTML element selections
  - HTML element manipulation
  - CSS manipulation
  - HTML event functions
  - JavaScript Effects and animations
  - HTML DOM traversal and modification
  - AJAX
  - Utilities

jQuery Overview – How to add jQuery?

- The jQuery library is stored as a single JavaScript file, containing all the jQuery methods.
- It can be added to a web page with the following mark-up:

```
<head>
<script type="text/javascript" src = "jquery.js"></script>
</head>
```

- Or you can use the hosted jQuery library from Google or Microsoft.

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery
      /1.11.3/jquery.min.js"></script>
</head>
```

jQuery Overview – Summary

- A library of JavaScript Functions.
- A lightweight "write less, do more" JavaScript library.
- Contains many features that help us to developer web easier
- To Add jQuery: download its source code and add it to our web page (via `<script>` tag)

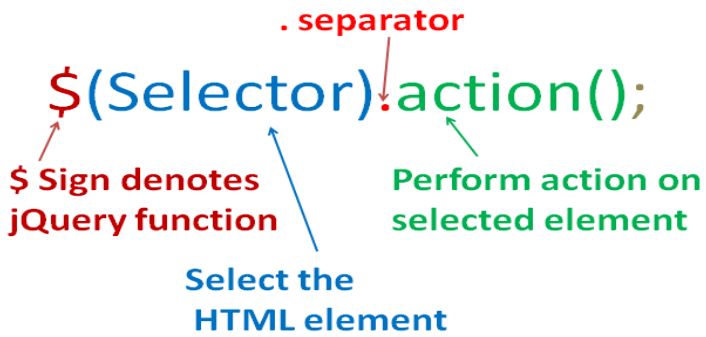
Section 2

jQuery Syntax

jQuery Syntax – Basic Syntax (1)

- The jQuery syntax is tailor made for **selecting** HTML elements and perform some **action** on the element(s).
- Basic syntax is:

  $(selector).action()



jQuery Syntax – Basic Syntax (2)

- **Examples:**
  - ✓ $(this).hide() - hides the current element.
  - ✓ $("p").hide() - hides all <p> elements.
  - ✓ $(".test").hide() - hides all elements with class="test".
  - ✓ $("#test").hide() - hides the element with id="test".

jQuery Syntax – Document Ready (1)

- You might have noticed that all jQuery methods are inside a document ready event:

```
$(document).ready(function(){
   // jQuery methods go here...
});
```

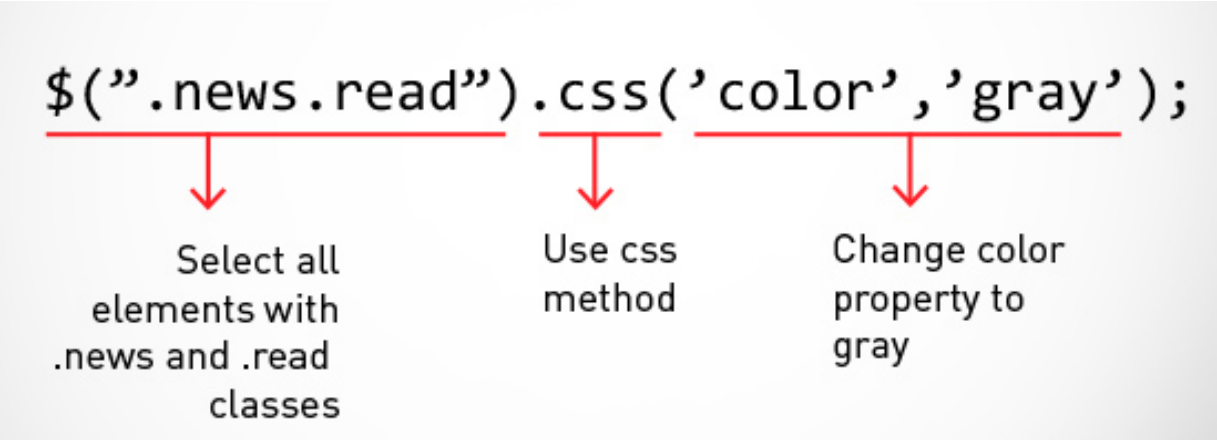
- This is to prevent any jQuery code from running before the document is finished loading (is ready).
  - ✓ It is good practice to wait for the document to be fully loaded and ready before working with it.

jQuery Syntax – Document Ready (1)

- Here are some examples of actions that can fail if methods are run before the document is fully loaded:
  - ✓ Trying to hide an element that is not created yet
  - ✓ Trying to get the size of an image that is not loaded yet

jQuery Syntax – Selectors (1)

- jQuery selectors are one of the most important parts of the jQuery library.
- jQuery selectors allow you to select and then manipulate HTML element(s).



jQuery Syntax – Selectors (2)

- jQuery selectors are used to "find" (or select) HTML elements based on their id, classes, types, attributes, values of attributes and much more
- All selectors in jQuery start with the dollar sign and parentheses: $().

jQuery Syntax – Selectors Examples (1)

Syntax	Description
$("*")	Selects all elements
$(this)	Selects the current HTML element
$("p.intro")	Selects all <p> elements with class="intro"
$("p:first")	Selects the first <p> element
$("ul li:first")	Selects the first element of the first
$("ul li:first-child")	Selects the first element of every

jQuery Syntax – Selectors Examples (2)

Syntax	Description
$("[href]")	Selects all elements with an href attribute
$("a[target='_blank']")	Selects all <a> elements with a target attribute value equal to "_blank"
$("a[target!='_blank']")	Selects all <a> elements with a target attribute value NOT equal to "_blank"
$(":button")	Selects all <button> elements and <input> elements of type="button"
$("tr:even")	Selects all even <tr> elements
$("tr:odd")	Selects all odd <tr> elements

jQuery Syntax – Element Selector

- The jQuery element selector selects elements based on the element name.
- Example 1:

```
$("p")
```

- Example 2:

```
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide();
    });
});
```

- Refs:

jQuery Syntax – Question?

- Question 1: Use the correct selector to hide all elements with an href attribute.

```
<script>
$(document).ready(function(){
    $("selector").hide();
});
</script>
```

jQuery Syntax - Summary

- The jQuery syntax is tailor made for **selecting** HTML elements and perform some **action** on the element(s).
- Basic syntax is:

  $(selector).action()

- Document Ready is to prevent any JavaScript/jQuery code from running before the document is finished loading (is ready).

Section 3

jQuery Events

jQuery Events – What is Event?

- All the different visitor's actions that a web page can respond to are called events.
- **Examples**:
  - ✓ moving a mouse over an element
  - ✓ selecting a radio button
  - ✓ clicking on an element

jQuery Events – Common Events

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

jQuery Events – Example

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
     $(document).ready(function() {
            $("button").click(function() {
                 $("p").hide();
     });
     });
</script>
</head>
<body>
     <h2>This is a heading</h2>
     <p>This is a paragraph.</p>
     <p>This is another paragraph.</p>
     <button>Click me</button>
</body>
</html>
```

jQuery Events – Summary

- **Events**: all the different visitor's actions that a web page can respond to
- **Common Events:** mouse click, mouse hover, keyup, keypress, document load, document unload

Section 4

jQuery Callback Functions

jQuery Callback Functions – Overview

- A callback function is executed after the current effect is 100% finished.
- JavaScript statements are executed line by line. However, with effect, the next line of code can be run even though the animation is not finished. This can create errors.
  - ✓ To prevent this, you can create a callback function.
- A callback function is executed after the current effect is finished.

jQuery Callback Functions – Example

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function() {
      $("button").click(function() {
            $("p").hide(1000, function() {
                  alert("The paragraph is now hidden");
            });
      });
});
</script>
</head>
<body>
<button>Hide</button>
<p>This is a paragraph with little content.</p>
</body>
</html>
```

jQuery Callback Functions – Practical Time (1)

- Demo with:
  - ✓ **draggable** action;
  - ✓ **addClass** action;
  - ✓ **removeClass** action;

jQuery Callback Functions – Practical Time (2)

Code snippet

```
$(function() {
    $("#button").on("click", function() {
        $("#effect").addClass("newClass");
    });

    function callback() {
        setTimeout(function() {
        $("#effect").removeClass("newClass");
        }, 1500);
    }
});
```

jQuery Callback Functions – Summary

- A callback function is executed after the current effect is 100% finished.
- JavaScript statements are executed line by line.
- However, with effect, the next line of code can be run even though the animation is not finished.
- A callback function is executed after the current effect is finished.

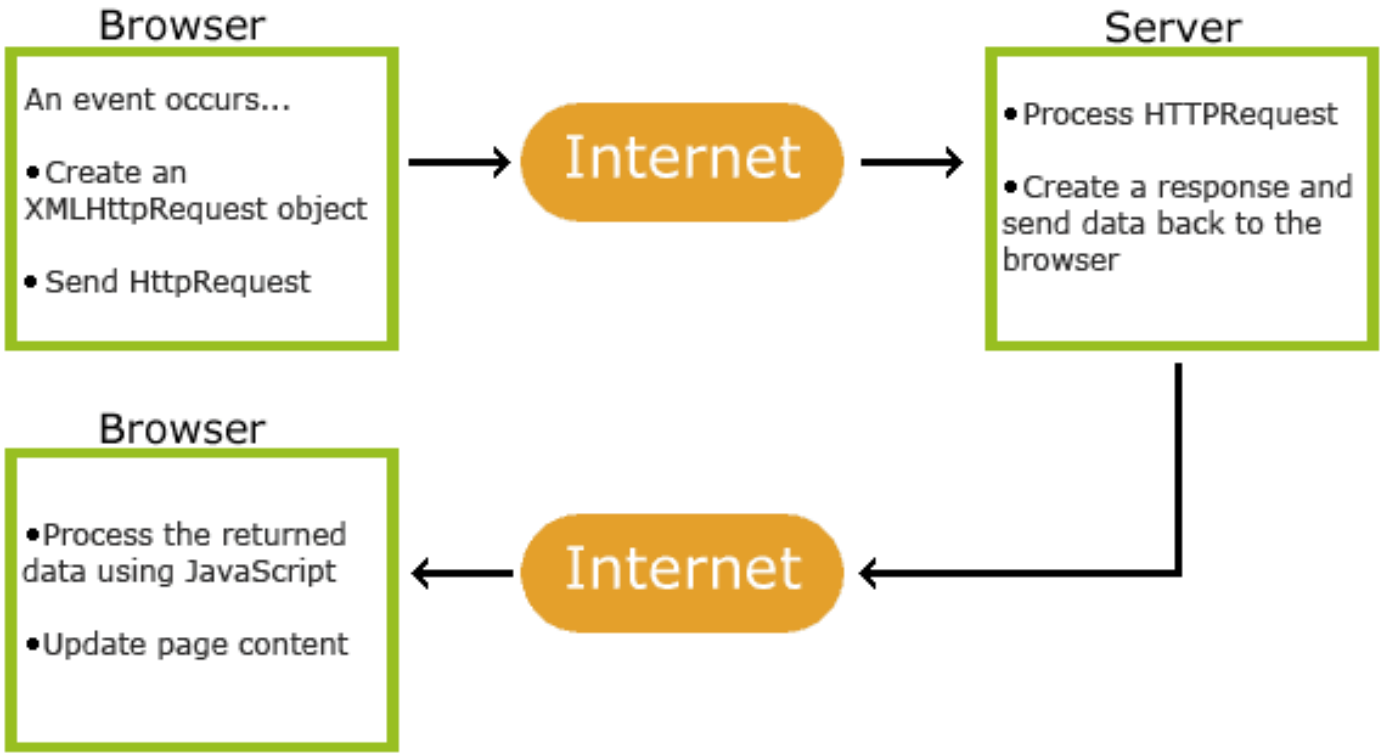
Section 5

AJAX

AJAX – Overview (1)

- Asynchronous JavaScript and XML
- AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page.
- AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

AJAX – How it works?



AJAX – Overview (2)

- AJAX is based on internet standards, and uses a combination of:
  - ✓ XMLHttpRequest object (to exchange data asynchronously with a server)
  - ✓ JavaScript/DOM (to display/interact with the information)
  - ✓ CSS (to style the data)
  - ✓ XML (often used as the format for transferring data)
- AJAX applications are browser- and platform-independent!

AJAX – Example

Code snippet

```
function loadXMLDoc() {
     var xmlhttp;
     if (window.XMLHttpRequest) {// code for IE7+, Firefox, Chrome, Opera, Safari
          xmlhttp = new XMLHttpRequest();
     } else {// code for IE6, IE5
          xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
     }
     xmlhttp.onreadystatechange = function() {
          if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
               document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
          }
     }
     xmlhttp.open("GET", "ajax_info.txt", true);
     xmlhttp.send();
}
```

AJAX – Working with jQuery (1)

- jQuery provides a rich set of methods for AJAX web development.
- With jQuery AJAX, you can request TXT, HTML, XML or JSON data from a remote server using both HTTP Get and HTTP Post
- **And you can load remote data directly into selected HTML elements of your web page!**

AJAX – Working with jQuery (2)

- The jQuery load() method is a simple (but very powerful) AJAX function. It has the following syntax:

$(selector).load(url,*data,callback*)

- Use the **selector** to define the HTML element(s) to change, and the **url parameter** to specify a web address for your data.

AJAX – Working with jQuery (3)

- **Syntax:**

```
jQuery.ajax({
    type:"POST",//Phương thức gửi request là POST hoặc GET
    data:"id=12&name=abc", //tham số gửi kèm
    dataType:"xml",//kiểu dữ liệu trả về, mặc định là text
    url:"/login/servletLogin",//Đường dẫn tới nơi xử lý request ajax
    success: function (){//hàm gọi về khi thực hiện thành công
          // mã lệnh
    }
});
```

AJAX – Example (1)

```
<script type="text/javascript">
    jQuery(document).ready(function() {
        jQuery("#aj    axButton").click(function() {
            jQuery.ajax({
                type : "POST",
                url : "ajax.html",
                success : function(html) {
                    jQuery("#responseDiv").html(html);
                }
            });
        });
    });
</script>
```

AJAX – Example (2)

```
<style>
#sampleTable {
 border-collapse: collapse;
}
#sampleTable td {
 border: 1px solid black;
 width: 100px;
}
</style>
<table id="sampleTable">
      <tr>
            <td>Name</td>
            <td>Year</td>
      </tr>
      <tr>
            <td>Van A</td>
            <td>1982</td>
      </tr>
</table>
```

AJAX – Summary

- AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page.
- AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes
- jQuery provides a rich set of methods for AJAX web development: load() method and ajax() method

Thank you